ANL-7659
RETURN TO ANL (IDAHO) LIBRARY.

Argonne National Laboratory

THE INVERSION OF DE HAAS-VAN ALPHEN DATA:

I. Closed Surfaces with Inversion Symmetry

by

R. L. Aurbach, J. B. Ketterson,
F. M. Mueller, and L. R. Windmiller

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

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

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

NOTICE

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

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.65

ANL-7659
Mathematics and Computers

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

THE INVERSION OF DE HAAS-VAN ALPHEN DATA:

I. Closed Surfaces with Inversion Symmetry

by

R. L. Aurbach, J. B. Ketterson,
F. M. Mueller, and L. R. Windmiller

Solid State Science Division

July 1970

TABLE OF CONTENTS

		Page
I.	MAIN PROGRAMS	
	A. Notes on the Program Conventions	1
	B. Inversion Programs J111-J1110 and J121-J1210	2
	C. Fermi-Velocity Programs J131-J1310	8
	D. Density-of-States Programs J141-J1410	10
	E. Fermi-Volume Programs J151-J1510	11
	F. Kubic Harmonic Generator HAGEN	12
II.	SPHERICAL HARMONIC GENERATOR	
	A. Notes on the Spherical Harmonic Generator Package	14
	B. Subroutine IYLM	14
	C. Subroutine YLM	15
	D. Subroutine BLKDAT	15
	E. Subroutine COMSET	15
	F. Subroutine SPHERE	15
	G. Subroutine GRAD	17
	H. Limitations of the Spherical Harmonic Generator Package	19
III.	POLYNOMIAL GENERATOR PACKAGE	
	A. Harmonic Function Generators xxPOLY	21
	B. Harmonic Function Theta-Derivative Generators xxPOLT	23
	C. Harmonic Function Phi-Derivative Generators xxPOLP	23
	D. Legendre Polynomial Generator COET	24
IV.	MULTIPURPOSE SUBROUTINES	
	A. Least Squares Fitting Routine KLSQ	25
	B. Lab to Crystal Coordinate Transformation LAB	26

C. Matrix Inverter MATINV 28

D. Harmonic Number-Number of Terms Transfer LTOM and MTOL 28

E. Equivalent Pocket Transformation TRANS 29

F. Spherical Mapping Transformation ELLIPS 30

V. VECTOR SUBROUTINES

A. Spherical to Cartesian Coordinate Transformation GOTTA, DGOTTA 31

B. Vector Product CRISS, DCRISS 32

C. Unit Vector Generator WOW, DWOW 33

D. Cubic Transformation ORD, DORD 33

E. Arc Tangent Function FLOP, DFLOP 34

F. Cartesian to Spherical Coordinate Transformation GETTA, DGETTA 34

G. Vector Rotator OILER 35

H. Vector Rotator EULER 36

VI. SPECIAL-PURPOSE SUBROUTINES

A. Wigner Matrix Elements WIG 36

B. Eigenvalue and Eigenvector Calculation EIGEN 38

APPENDIXES

A. Inversion Programs Input Data Format 40

B. Kubic Harmonic Expansion Coefficients 47

C. Programs 56

ACKNOWLEDGMENTS 311

LIST OF FIGURES

No.	Title	Page
1.	The Values of $k(\partial k/\partial E)$ in the (100) Plane for the Electron Surface in Palladium as Calculated by Mueller, Freeman, Dimmock and Furdyna. Shown also are the fits for ℓ = 36 and 60.	6
2.	The Values of $k(\partial k/\partial E)$ in the (110) Plane for the Electron Surface in Palladium. Shown also are the fits for ℓ = 36 and 60.	7
3.	The Total Time and the Time per Harmonic to Compute SPHERE.	18
4.	The Total Time and the Time per Harmonic to Compute GRAD.	20

2. Input-data Format

The format for the input data of these programs is listed in Appendix A.

B. Inversion Programs J111-J1110 and J121-J1210

1. Purpose

These programs accomplish inversions of de Haas-van Alphen data for closed Fermi surfaces with inversion symmetry and with a single-valued radius vector measured from the inversion center. The J11x series invert extremal cross-sectional areas to give the square of the Fermi radius; the J12x series invert cyclotron effective masses, $(1/\pi)(\partial A/\partial E)$, to give $(2/\pi)\ k(\partial k/\partial E)$.

For the representation of the surface, we use symmetry-adapted spherical harmonics. Since the radius vector is a real quantity, we deal with the real spherical harmonics:

$$C^g_{\ell,m} = \frac{1}{\sqrt{2}}\ [Y_{\ell,m}(\theta,\varphi) + Y_{\ell,-m}(\theta,\varphi)];$$

$$C^u_{\ell,m} = \frac{1}{i\sqrt{2}}\ [Y_{\ell,m}(\theta,\varphi) - Y_{\ell,-m}(\theta,\varphi)];$$

$$C_{\ell,o} = Y_{\ell,o}(\theta,\varphi).$$

We expand the area $A(\theta,\varphi)$ and the square of the radius vector $k^2(\theta,\varphi)$ as follows:

$$k^2(\theta,\varphi) = \sum_{\ell,m} \left[\gamma^g_{\ell,m} C^g_{\ell,m}(\theta,\varphi) + \gamma^u_{\ell,m} C^u_{\ell,m}(\theta,\varphi)\right];$$

$$A(\theta,\varphi) = \sum_{\ell,m} \left[\beta^g_{\ell,m} C^g_{\ell,m}(\theta,\varphi) + \beta^u_{\ell,m} C^u_{\ell,m}(\theta,\varphi)\right].$$

The inversion theorem states that

$$\beta_{\ell,m} = \pi\ P_\ell(0)\ \gamma_{\ell,m},$$

where $P_\ell(0)$ is the Legendre function of order ℓ. A unique connection between radii and areas requires even values of ℓ, since $P_\ell(0) = 0$ for odd ℓ. The requirement even ℓ is equivalent to requiring inversion symmetry, since the parity of spherical harmonics is $(-1)^\ell$. Similarly,

$$m^* = \frac{1}{\pi}\frac{\partial A(\theta,\varphi)}{\partial E} = \sum_{\ell,m}\left[\beta'^{g}_{\ell,m}C^{g}_{\ell,m}(\theta,\varphi) + \beta'^{u}_{\ell,m}C^{u}_{\ell,m}(\theta,\varphi)\right];$$

$$\frac{2}{\pi}k\frac{\partial k}{\partial E} = \frac{\partial k^2(\theta,\varphi)}{\pi\,\partial E} = \sum_{\ell,m}\left[\gamma'^{g}_{\ell,m}C^{g}_{\ell,m}(\theta,\varphi) + \gamma'^{u}_{\ell,m}C^{u}_{\ell,m}(\theta,\varphi)\right],$$

where $\beta'_{\ell,m} = \pi P_\ell(0)\gamma'_{\ell,m}$.

Often one encounters surfaces that are nearly ellipsodial in shape. This is particularly true in semimetals. The expansions above, however, do not terminate for ellipsoids and converge slowly if the ellipsoid is quite elongated. For such surfaces, it is convenient to perform a "spherical mapping" (ellipsoidal transformation) on the coordinates in order to map the "ellipsoid" into a "sphere." We do this by the following transformation:

$$k'_x = \alpha k_x; \quad k'_y = k_y; \quad k'_z = \gamma k_z.$$

Let T represent this transformation, i.e., $T(k_x, k_y, k_z) = (k'_x, k'_y, k'_z)$, or, in polar coordinates, $T(k,\theta,\varphi) = (k',\theta',\varphi')$. We will also need the inverse transformation $T^{-1}(k'_x,k'_y,k'_z) = (k_x,k_y,k_z)$ or equivalently, $T^{-1}(k',\theta',\varphi') = (k,\theta,\varphi)$. It is easy to show the following:

$$k'^2(T\theta,T\varphi) = k^2(\theta,\varphi)[\sin^2\theta(\sin^2\varphi + \alpha^2\cos^2\varphi) + \gamma^2\cos^2\theta];$$

$$\cos\varphi' = \frac{\alpha\cos\varphi}{(\sin^2\varphi + \alpha^2\cos^2\varphi)^{1/2}};$$

$$\cos\theta' = \frac{\gamma\cos\theta}{[\sin^2\theta(\sin^2\varphi + \alpha^2\cos^2\varphi) + \gamma^2\cos^2\theta]^{1/2}}.$$

The question arises as to the relation between the area in the transformed system and the untransformed system. It can be shown that

$$A'(T^{-1}\theta,T^{-1}\varphi) = \alpha\gamma[\sin^2\theta(\sin^2\varphi + \alpha^{-2}\cos^2\varphi) + \gamma^{-2}\cos^2\theta]^{1/2}A(\theta,\varphi).$$

2. Subroutines Required

xxPOLY	DCRISS	
BLKDAT	DGOTTA	
COMSET	DWOW	
IYLM	TRANS	
YLM	ELLIPS	
SPHERE	COET	
LTOM	MTOL	
KLSQ	EULER	All except J111 and J121
LAB	DFLOP	All except J111 and J121
MATINV		

ABSTRACT

In this report we have collected a series of mathematical techniques and detailed computer programs which allow the inversion of de Haas-van Alphen data on closed Fermi surfaces that possess inversion symmetry and have a single-valued radius vector measured from that center. De Haas-van Alphen data include extremal cross-sectional areas and cyclotron effective masses, whereas the techniques discussed here allow the determination from such data of the Fermi radius and Fermi velocity. The programs have been applied to the platinum group near-noble metals and the arsenic group semimetals. Programs are presented, however, which are applicable to all ten of the point groups containing the inversion element. Also included are programs for the generation of Kubic harmonics as linear combinations of spherical harmonics. The coefficients for this expansion are tabulated to order $\ell = 60$.

The programs employing Kubic harmonics have been used extensively and are therefore reasonably well "debugged." The programs built around the C_{2h} and D_{4h} point groups have been employed somewhat less, while the remaining point groups have not been tested at all.

I. MAIN PROGRAMS

A. Notes on the Program Conventions

1. Naming System for Programs

The inversion-scheme programs are named by a convention of the form J1YX, where

a. The J1 is arbitrary.
b. The number Y indicates the purpose of the program, using the following code:

(1) Inversion of $A(\theta,\varphi)$ to give $k^2(\theta,\varphi)$;
(2) Inversion of $m^*(\theta,\varphi)$ to give $\frac{1}{\pi} 2k \frac{dk}{dE} (\theta,\varphi)$;
(3) Fermi velocities;
(4) Density of states;
(5) Number of carriers (volume).

c. The number x indicates the crystal point group for which the program is written using the following code:

X	Group	Mnemonic
1	O_h	KU
2	D_{6h}	HX
3	C_{6h}	C6
4	D_{3d}	D3
5	S_6 (C_{3i})	S6
6	D_{4h}	SQ
7	C_{4h}	C4
8	V_h (D_{2h})	VH
9	C_{2h}	C2
10	S_2 (C_i)	S2

The mnenomic is the first two letters of the name used for the harmonic function and harmonic derivative programs associated with that point group. The numbers X are also the numbers that index the proper groups in subroutines LTOM and MTOL.

3. Output

The output of this routine consists of a listing of the input data, the expansion coefficients obtained, the root-mean-square error, and the values of the areas (or masses) and radii ($k\ \partial k/\partial E$) in the selected planes of the crystal.

4. Algorithm

We distinguish between two cases:

a. Cubic Symmetry. The oriented input data are read and the angular variables of each point determined in the crystal coordinate system. A standard least-mean-squares routine is used to determine the expansion coefficients for the input data (areas or masses). Provision is made for varying selected groups of the expansion coefficients while maintaining the rest constant in order to achieve an iterative fitting technique.

The inversion theorem is applied to determine the expansion coefficients of the radii, or $2k\ (\partial k/\partial E)$, as the case may be. These coefficients are then used to calculate the areas (masses) and radii ($k\ \partial k/\partial E$) in the selected planes.

b. Other Symmetries. The same procedure as that for cubic symmetry is followed, except that a spherical mapping procedure may be used throughout the calculations.

References: I. M. Lifshitz and A. V. Pogorelov, Dokl. Akad. Nauk. SSR 96, 1143 (1954).

F. M. Mueller, Phys. Rev. 148, 636 (1966).

F. M. Mueller and M. G. Priestley, Phys. Rev. 148, 638 (1966).

L. L. Foldy, Phys. Rev. 170, 670 (1968).

J. B. Ketterson, L. R. Windmiller, S. Hörnfeldt, and F. M. Mueller, Solid State Commun. 6, 851 (1968).

J. B. Ketterson and L. R. Windmiller, Phys. Rev. 1, 463 (1970).

R. Aurbach, J. B. Ketterson, and L. R. Windmiller, Proc. of the Conf. on Physics of Semimetals and Narrow Gap Semiconductors, Dallas, Texas, 1970 (to be published).

5. Restrictions

The program is limited through dimension statements to 10 planes of data with 600 points per plane and fits involving no more than 91 expansion coefficients.

6. Remarks

It might seem that a 91-term fit is far more detailed than is actually necessary. However, Figs. 1 and 2 illustrate a case in which such a high-order fit is required. The data points are values of $k(\partial k/\partial E)$ for the Γ-centered electron surface in palladium in the (100) and (110) planes as determined from a band-structure calculation using the interpolation scheme. Shown also are fits for $\ell = 36$ (37 terms), and $\ell = 60$ (91 terms). The improvement of the fit with increasing ℓ demonstrates the necessity of using a high-order expansion in this particular case.

Reference: F. M. Mueller, A. J. Freeman, J. O. Dimmock, and A. M. Furdyna, Phys. Rev. B1, 4617-4635 (15 June 1970).

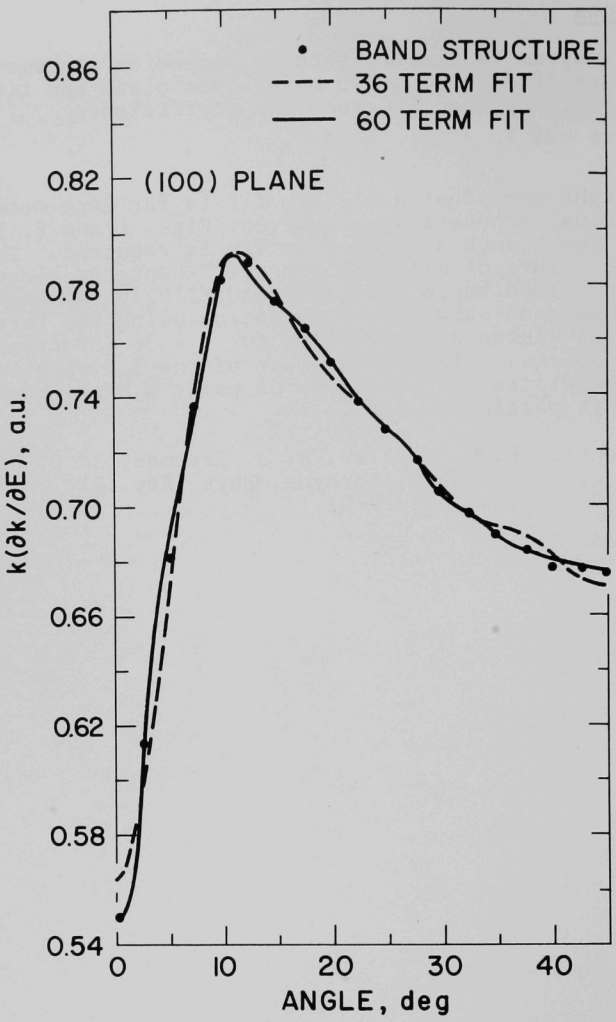

Fig. 1. The Values of $k(\partial k/\partial E)$ in the (100) Plane for the Electron Surface in Palladium as Calculated by Mueller, Freeman, Dimmock and Furdyna. Shown also are the fits for ℓ = 36 and 60.

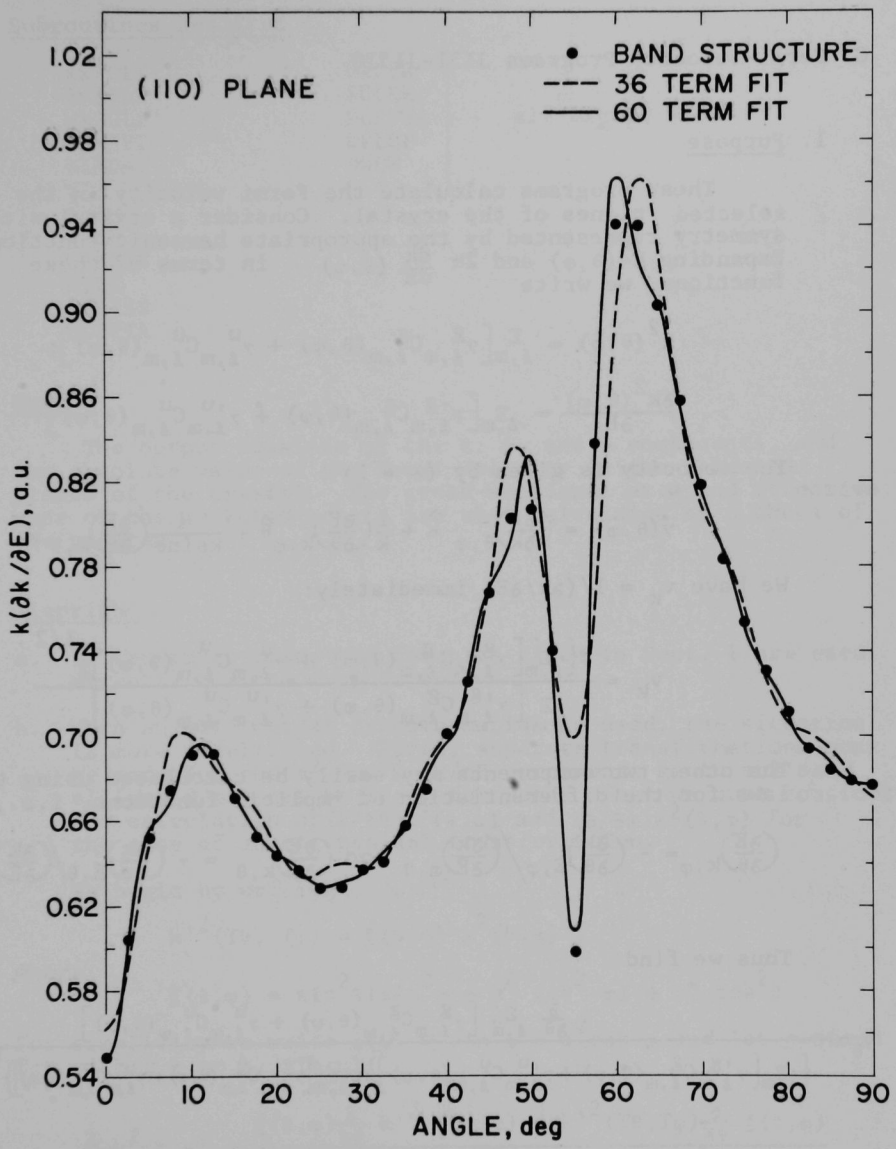

Fig. 2. The values of $k(\partial k/\partial E)$ in the (110) Plane for the Electron Surface in Palladium. Shown also are the fits for ℓ = 36 and 60.

C. Fermi-Velocity Programs J131-J1310

1. Purpose

These programs calculate the Fermi velocity in the selected planes of the crystal. Consider a crystal with symmetry represented by the appropriate harmonic function. Expanding $k^2(\theta,\varphi)$ and $2k\,\frac{dk}{dE}(\theta,\varphi)$ in terms of these functions, we write

$$k^2(\theta,\varphi) = \sum_{\ell,m}\left[\gamma^g_{\ell,m}C^g_{\ell,m}(\theta,\varphi) + \gamma^u_{\ell,m}C^u_{\ell,m}(\theta,\varphi)\right];$$

$$\frac{\partial k^2(\theta,\varphi)}{\partial E} = \sum_{\ell,m}\left[\gamma'^g_{\ell,m}C^g_{\ell,m}(\theta,\varphi) + \gamma'^u_{\ell,m}C^u_{\ell,m}(\theta,\varphi)\right].$$

The velocity is given by ($\hbar = 1$)

$$\vec{v}(\theta,\varphi) = \left(\frac{\partial E}{\partial k}\right)_{\theta,\varphi}\hat{k} + \frac{1}{k}\left(\frac{\partial E}{\partial \theta}\right)_{k,\varphi}\hat{\theta} + \frac{1}{k\sin\theta}\left(\frac{\partial E}{\partial \varphi}\right)_{k,\theta}\hat{\varphi}\,.$$

We have $v_k = 1/(\partial k/\partial E)$ immediately:

$$v_k = \frac{2\left\{\sum_{\ell,m}\left[\gamma^g_{\ell,m}C^g_{\ell,m}(\theta,\varphi) + \gamma^u_{\ell,m}C^u_{\ell,m}(\theta,\varphi)\right]\right\}^{1/2}}{\sum_{\ell,m}\left[\gamma'^g_{\ell,m}C^g_{\ell,m}(\theta,\varphi) + \gamma'^u_{\ell,m}C^u_{\ell,m}(\theta,\varphi)\right]}\,.$$

The other two components may easily be calculated using the laws for the differentiation of implicit functions, i.e.,

$$\left(\frac{\partial E}{\partial \theta}\right)_{k,\varphi} = -\left(\frac{\partial k}{\partial \theta}\right)_{E,\varphi}\Big/\left(\frac{\partial k}{\partial E}\right)_{\varphi,\theta} \quad\text{and}\quad \left(\frac{\partial E}{\partial \varphi}\right)_{k,\theta} = -\left(\frac{\partial k}{\partial \varphi}\right)_{E,\theta}\Big/\left(\frac{\partial k}{\partial E}\right)_{\varphi,\theta}.$$

Thus we find

$$v_\theta = -\frac{\frac{\partial}{\partial\theta}\sum_{\ell,m}\left[\gamma^g_{\ell,m}C^g_{\ell,m}(\theta,\varphi) + \gamma^u_{\ell,m}C^u_{\ell,m}(\theta,\varphi)\right]}{\left\{\sum_{\ell,m}\left[\gamma'^g_{\ell,m}C^g_{\ell,m}(\theta,\varphi) + \gamma'^u_{\ell,m}C^u_{\ell,m}(\theta,\varphi)\right]\right\}\left\{\sum_{\ell,m}\left[\gamma^g_{\ell,m}C^g_{\ell,m}(\theta,\varphi) + \gamma^u_{\ell,m}C^u_{\ell,m}(\theta,\varphi)\right]\right\}^{\frac{1}{2}}};$$

$$v_\varphi = -\frac{(\sin\theta)^{-1}\frac{\partial}{\partial\varphi}\sum_{\ell,m}\left[\gamma^g_{\ell,m}C^g_{\ell,m}(\theta,\varphi) + \gamma^u_{\ell,m}C^u_{\ell,m}(\theta,\varphi)\right]}{\left\{\sum_{\ell,m}\left[\gamma'^g_{\ell,m}C^g_{\ell,m}(\theta,\varphi) + \gamma'^u_{\ell,m}C^u_{\ell,m}(\theta,\varphi)\right]\right\}\left\{\sum_{\ell,m}\left[\gamma^g_{\ell,m}C^g_{\ell,m}(\theta,\varphi) + \gamma^u_{\ell,m}C^u_{\ell,m}(\theta,\varphi)\right]\right\}^{\frac{1}{2}}}.$$

2. Subroutines Required

xxPOLY	OILER	all except J131
xxPOLT	EULER	
xxPOLP	DGETTA	
COMSET	DFLOP	
BLKDAT	DWOW	
YLM		
IYLM		
GRAD		
SPHERE		
MTOL		
DCRISS		
DGOTTA		

3. Output

The output consists of the k, θ, and φ components, and the absolute value of the Fermi velocity in the selected planes of the crystal. The cross-sectional area and effective mass of the selected orbits are also calculated as a check of the program.

4. Algorithm

a. In the cubic case, the formulas given in Sect. 1 are used directly for the computations.

b. When an ellipsoidal transformation is used, the situation is more complicated. First, separate transformations must be made for the k^2 and k dk/dE components since the same ellipsoid may not be ideal for both. Then we must consider the calculation of $(\partial/\partial\theta)k^2(\theta,\varphi)$ and $(\partial/\partial\varphi)k^2(\theta,\varphi)$ for the case of an ellipsoidal transformation.

We begin by writing

$$k'^2(T\theta,\ T\varphi) = \xi(\theta,\varphi)\ k^2(\theta,\varphi)\ ,$$

where

$$\xi(\theta,\varphi) = \sin^2\theta(\sin^2\varphi + \alpha^2\cos^2\varphi) + \gamma^2\cos^2\theta\ .$$

For simplicity we write $\varepsilon(\varphi) = \sin^2\varphi + \alpha^2\cos^2\varphi$ and let ψ stand for either θ or φ. Then

$$\frac{\partial}{\partial\psi}k^2(\theta,\varphi) = \frac{\xi(\theta,\varphi)\frac{\partial}{\partial\psi}k'^2(T\theta,T\varphi) - k'^2(T\theta,T\varphi)\frac{\partial}{\partial\psi}\xi(\theta,\varphi)}{\xi^2(\theta,\varphi)}$$

and

$$\frac{\partial}{\partial\psi}k'^2(T\theta,T\varphi) = \left[\frac{\partial(T\theta)}{\partial\psi}\frac{\partial}{\partial(T\theta)} + \frac{\partial(T\varphi)}{\partial\psi}\frac{\partial}{\partial(T\varphi)}\right]k'^2(T\theta,T\varphi).$$

Direct computation leads to the following values of the derivatives:

$$\frac{\partial}{\partial\theta}\,\xi(\theta,\varphi) = 2\sin\theta\cos\theta\,[\varepsilon(\varphi) - \gamma^2]\ ;$$

$$\frac{\partial}{\partial\varphi}\,\xi(\theta,\varphi) = 2(1-\alpha^2)\sin^2\theta\sin\varphi\cos\varphi;$$

$$\frac{\partial(T\varphi)}{\partial\theta} = 0;$$

$$\frac{\partial(T\varphi)}{\partial\varphi} = \frac{\alpha}{\varepsilon(\varphi)}\ ;$$

$$\frac{\partial(T\theta)}{\partial\theta} = \frac{\gamma\varepsilon^{1/2}(\varphi)}{\xi(\theta,\varphi)}\ ;$$

$$\frac{\partial(T\theta)}{\partial\varphi} = \frac{\gamma(1-\alpha^2)\sin\theta\cos\theta\sin\varphi\cos\varphi}{\varepsilon^{1/2}(\varphi)\,\xi(\theta,\varphi)}\ .$$

These formulas enable one to apply correctly the ellipsoidal mapping to the computation of Fermi velocities. The area and effective mass in the selected planes are calculated by Simpson's rule integrations as an overall check on program consistency.

References: J. B. Ketterson, L. R. Windmiller, S. Hörnfeldt, and F. M. Mueller, Solid State Commun. 6, 851 (1966)

J. B. Ketterson and L. R. Windmiller Phys. Rev. 1, 463 (1970)

R. Aurbach, J. B. Ketterson, and L. R. Windmiller, Proc. of the Conf. on Semimetals and Narrow Gap Semiconductors, Dallas, Texas (1970), to be published.

D. Density-of-States Programs J141-J1410

1. Purpose

These programs calculate the density of states N(E) at $E = E_f$. We use

$$N(E_f) = \frac{2}{(2\pi)^3}\int_{E=E_f}\frac{dS}{|\nabla|}$$

$$= \frac{2}{(2\pi)^3}\int\frac{k^2(E_f,\theta,\varphi)\sin\theta\,d\theta\,d\varphi}{v_k(E_f,\theta,\varphi)}\ .$$

2. Subroutines Required

xxPOLY	OILER	all except J141
COMSET	EULER	
BLKDAT	DGOTTA	
IYLM	DGETTA	
YLM	DFLOP	
SPHERE	DWOW	
MTOL		

3. Output

The output consists of the values of v_k and k at each mesh point of the integration, followed by the value of the density of states in several different units.

4. Algorithm

Integration was by two-dimensional Simpson's Rule and proceeded over the smallest unique section of the Fermi surface. The ellipsoidal mapping was used where appropriate. Separate mappings were used for the k^2 expansion and the 2k (dk/dE) expansion.

References: J. B. Ketterson, L. R. Windmiller, S. Hörnfeldt, and F. M. Mueller, Solid State Commun. 6, 851 (1968)

E. Fermi-Volume Programs J151-J1510

1. Purpose

These programs calculate the volume Ω of the Fermi surface and the total number of states N.

$$\Omega = \int d^3k = \frac{1}{3} \int k^3(\theta,\varphi) \sin\theta \, d\theta \, d\varphi;$$

$$N = \frac{2}{(2\pi)^3} \Omega .$$

2. Subroutines Required

xxPOLY	OILER	all except J151
BLKDAT	EULER	
COMSET	DGOTTA	
IYLM	DGETTA	
YLM	DFLOP	
SPHERE	DWOW	
MTOL		

3. Output

The output consists of a listing of the value of k at each mesh point of the integration. The values of Ω and N are printed in a variety of units.

4. Algorithm

Integration was by two-dimensional Simpson's Rule. The integration is extended only over the smallest unique sections of the surface. Where appropriate, the ellipsoidal mapping was used, in which case the integrations are performed in the untransformed system. An alternative method of performing the volume integrations is to work in the transformed system and make use of the relation $\int d^3r = 1/\alpha\beta \int d^3r'$.

F. Kubic Harmonic Generator--Hagen

1. Purpose

The Kubic harmonics ${}_iK_\ell(\theta,\varphi)$, which have the transformation properties of the full cubic group, can be written as a linear combination of the real even spherical harmonics $C^g_{\ell,m}$, where $C^g_{\ell,m} = (Y_{\ell,m} + Y_{\ell,-m})/\sqrt{2}$. This program computes the expansion coefficients ${}_ia_{\ell m}$

$$ {}_iK_\ell(\theta,\varphi) = \sum_m {}_ia_{\ell m} C^g_{\ell,m}(\theta,\varphi). $$

2. Subroutines required

WIG
EIGEN

3. Output

Printed Output: For every (even) order of $\ell \leq$ LMAX, the matrix D (see below), the residual vectors $\vec{R} = \overleftrightarrow{D}\,\vec{a} - \vec{a}$, the root-mean-square deviation

$$ \sqrt{\frac{1}{L}\sum_{m=1}^{L} R_i^2}\,, $$

and the values of the coefficients ${}_ia_{\ell m}$ are printed.

Punched Output: The number of the coefficient vector ${}_i\vec{a}_\ell$, its ℓ and i parameters, and its values are punched. In other programs using these coefficients, this information is stored as the arrays NAM and SYM.

4. Algorithm

The restrictions imposed by cubic symmetry on the linear combination of the $C^g_{\ell m}$ are:

a. Inversion. This implies that ℓ is even.
b. A fourfold rotation about the z axis. This implies that m = 0 mod (4).
c. Invariance under a rotation by $\pi/2$ about the y axis. This implies that the ${}_ia_{\ell m}$ satisfy

$$\sum_m {}_ia_{\ell m} d^\ell_{m'm}(\pi/2) = {}_ia_{\ell m'}$$

where

$$d^\ell_{m'm}(\beta)$$

is the Wigner matrix element for a rotation by β about the y axis. Contracting the above sum to nonnegative orders of m, we compute the matrix D_{jk}, j = 1/4 m' + 1, k = 1/2 m + 1, defined by

$$D_{11} = d^\ell_{oo}(\pi/2)$$

$$D_{1k} = 1/\sqrt{2}\ (d^\ell_{om}\ (\pi/2) + d^\ell_{o-m}\ (\pi/2))$$

$$D_{j1} = 1/\sqrt{2}\ (d^\ell_{m'o}\ (\pi/2) + d^\ell_{-m'o}\ (\pi/2))$$

$$D_{jk} = 1/2\ (d^\ell_{m'm}\ (\pi/2) + d^\ell_{-m'm}(\pi/2) + d^\ell_{m'-m}(\pi/2) + d^\ell_{-m'-m}(\pi/2)).$$

The matrix equation is now

$$\sum_k D_{jk}\ {}_ia_{\ell k} = {}_ia_{\ell j}\ .$$

HAGEN calculates the D matrix and solves for the orthonomal eigenvectors ${}_i\vec{a}_\ell$, including the proper degeneracy. The coefficients ${}_ia_{\ell m}$ are collected in Appendix B to order ℓ = 60.

References: F. M. Mueller and M. G. Preistley, Phys. Rev. 148, 638 (1966).

5. Program Restrictions

The program is restricted to values of ℓ not greater than 60. In practice, due to the propagation of rounding errors in subroutine WIG, use of this routine for values of ℓ greater than 40 should not be attempted except on machines having wide double-precision arithmetic (as CDC 3600).

6. Acknowledgment

We are grateful for valuable discussions with Professor M. G. Priestley.

II. SPHERICAL HARMONIC GENERATOR

A. Notes on the Spherical Harmonic Generator Package

1. Purpose

This package of six subroutines is used in the calculation of the spherical harmonics and their derivatives. The members of the package are IYLM, YLM, SPHERE, GRAD, COMSET, and BLKDAT subroutines. The subroutines SPHERE and GRAD calculate the spherical harmonics and their theta derivatives respectively for all even orders of ℓ, in the range $0 \leq \ell \leq 60$, and non negative indices m, in the range $0 \leq m \leq \ell$. These values are stored in two one-dimensional arrays XPLM and DYLM, respectively, which are contained in common block *YMATRX*. The function IYLM computes the location of a particular value with order ℓ and index m. YLM returns the value with order ℓ and index m for either the spherical harmonics or their derivatives, depending on the value of an additional input parameter. COMSET, which is called only once in the execution of any given procedure, and the BLKDAT program called by COMSET provide necessary initialization for the proper working of subroutines SPHERE and GRAD.

2. Restrictions

A. Other subroutines required: the package requires no additional subroutines.

B. Common storage: *YMATRX*
FACTOR

B. IYLM--Usage

1. Entry point: IYLM

2. Calling sequence: I = IYLM (L, M), where L and M are the indices of the harmonic or harmonic derivative and I is the index of XPLM or DYLM that references that value.

3. Common block necessary: None

C. YLM--Usage

1. Entry point: YLM

2. Calling sequence: X = YLM (L, M, I) where

   a. L and M are the indices of the harmonic or harmonic derivative desired.

   b. I: if I = +1, a spherical harmonic is referenced;
      if I = -1, a theta derivative is referenced.

   c. X is the value of the harmonic or harmonic derivative desire

3. Common Block necessary: COMMON/YMATRIX/XPLM (961), DYLM (961)

D. BLKDAT

1. Usage

   a. Entry point: BLKDAT

   b. Calling sequence: Call BLKDAT

   c. Common block necessary: COMMON/FACTOR/FL(121), OMEGA (961), T(961), COEF (61)

2. Algorithm

   This routine initializes the array FL of common block FACTOR. FL contains the natural logarithm of n! in element n + 1, for all values of n, $0 \leq n \leq 120$. These numbers were computed on a CDC 3600 in double precision as an accumulated sum and are believed accurate to 15 significant figures.

E. COMSET

1. Usage

   a. Entry point: COMSET

   b. Calling sequence: CALL COMSET

   c. Common blocks necessary: *FACTOR*

2. Algorithm

   This subroutine, which should be only called once per job, initializes the OMEGA and T arrays of common block FACTOR.

F. SPHERE

1. Usage

   a. Entry point: SPHERE

   b. Calling sequence: CALL SPHERE (T, P, LMAX) where

(1) T and P are the theta and phi values for the point of interest,

(2) LMAX is the largest ℓ value for which harmonics are to be calculated.

c. Common blocks necessary: YMATRX, which contains the output of the routine.
FACTOR, which contains storage for coefficients used by the routine.

2. Algorithm

a. Method. Each harmonic is expanded in a cosine series and evaluated directly. Four cases can be distinguished:

(1) $|x| = |\cos\theta| \le 0.77$;

$$Y_{\ell m} = \left[\frac{2\ell+1}{4\pi}\,\frac{(\ell-m)!}{(\ell+m)!}\right]^{1/2}(1-x^2)^{m/2}\sum_{\nu=0}^{[(\ell-m)/2]}\omega_\nu^{\ell m}x^{\ell-m-2\nu};$$

where

$$\omega_\nu^{\ell m} = -\,\omega_{\nu-1}^{\ell m}\,\frac{(\ell-m-2\nu+2)(\ell-m-2\nu+1)}{2\nu(2\ell-2\nu+1)}$$

and

$$\omega_o^{\ell m} = \frac{(2\ell!)}{\ell!(\ell-m)!2^\ell}$$

and $[(\ell-m)/2]$ is the largest integer contained in $(\ell-m)/2$.

(2) $|x| = |\cos\theta| > 0.77$;

$$Y_{\ell m} = \left[\frac{2\ell+1}{4\pi}\,\frac{(\ell-m)!}{(\ell+m)!}\right]^{1/2}(1-x^2)^{m/2}\sum_{\nu=0}^{\ell-m}T_\nu^{\ell m}(1-x)^\nu;$$

where

$$T_\nu^{\ell m} = -T_{\nu-1}^{\ell m}\,\frac{(\ell-m-\nu+1)(\ell+m+\nu)}{2\nu(m+\nu)}$$

and

$$T_o^{\ell m} = \frac{(\ell+m)!}{(\ell-m)!m!2^m}$$

(3) $x = \cos\theta = 0$;

$Y_{\ell m} = 0$, if $\ell - m = 0 \bmod (2)$, otherwise

$$Y_{\ell m} = \left[\frac{2\ell+1}{4\pi}\,\frac{(\ell-m)!}{(\ell+m)!}\right]^{1/2}(1-x^2)^{m/2}\,\omega_{[(\ell-m)/2]}^{\ell m}.$$

(4) $x = \cos\theta = 1$;

$$Y_{\ell m} = 0,\ m \neq 0;$$
$$Y_{\ell 0} = \left[\frac{2\ell + 1}{4\pi}\right]^{1/2} .$$

To prevent problems of overflow and underflow, the normalization factor

$$N_{\ell m} = \left[\frac{2\ell + 1}{4\pi}\ \frac{(\ell - m)!}{(\ell + m)!}\right]^{1/2}$$

was included in the $\omega_0^{\ell m}$ and $T_0^{\ell m}$ coefficients. These coefficients are computed in COMSET. The factorials enter as logarithms to prevent overflow. The sums were evaluated using nested multiplication to increase stability.

b. Acknowledgments. We wish to thank Dr. Arnold C. Wahl who provided the algorithm and Dr. William Cody for valuable programming suggestions.

3. Remarks

Figure 3 shows the total time and the time per harmonic to compute SPHERE.

G. GRAD

1. Usage

a. Entering point: GRAD

b. Calling sequence: CALL GRAD (T, P, LMAX), where

(1) T and P are the theta and phi values of the point of interest;

(2) LMAX is the largest ℓ-value for which harmonic derivatives are to be calculated.

c. Common blocks necessary: YMATRX
FACTOR

2. Algorithm

This routine uses a direct term-by-term differentiation of the expansion used in SPHERE. Again we consider four cases:

a. $|x| = |\cos\theta| \leq 0.77$;

$$\frac{\partial Y_{\ell m}}{\partial\theta} = N_{\ell m}(1 - x^2)^{(m-1)/2}\left[-\frac{1 - x^2}{x}\sum_{\nu=0}^{[(\ell - m)/2]} \omega_\nu^{\ell m}(\ell-m-2\nu)x^{\ell-m-2\nu}\right.$$
$$\left. + mx\sum_{\nu=0}^{[(\ell - m)/2]} \omega_\nu^{\ell m} x^{\ell-m-2\nu}\right] .$$

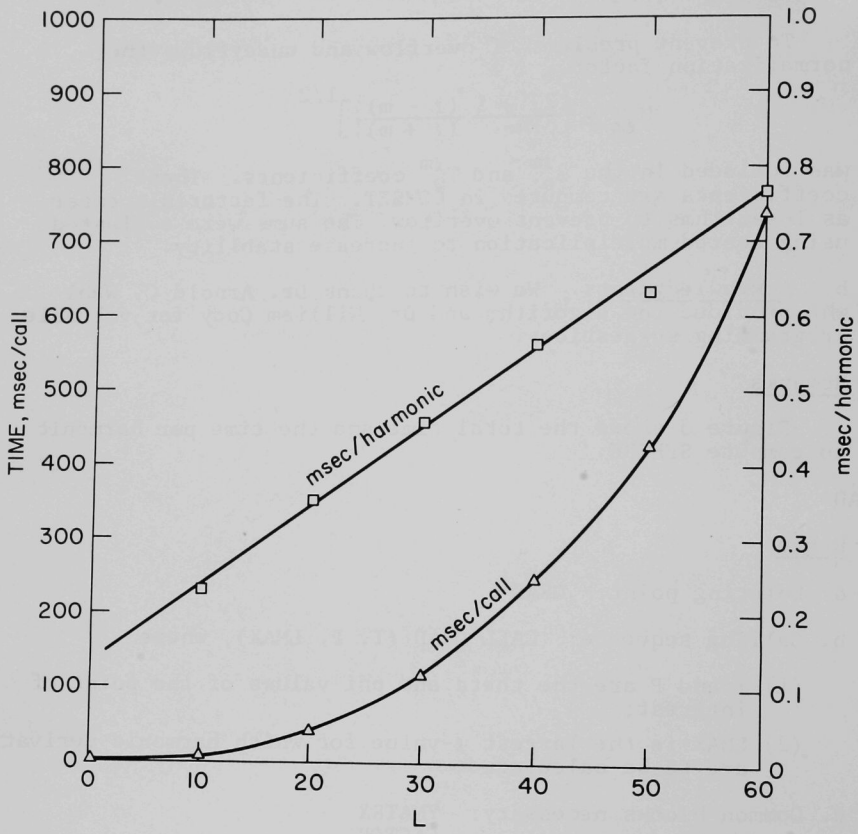

Fig. 3. The Total Time and the Time per Harmonic to Compute SPHERE

b. $|x| = |\cos\theta| > 0.77;$

$$\frac{\partial Y_{\ell m}}{\partial\theta} = N_{\ell m}\,(1 - x^2)^{(m-1)/2}\left[(1 + x)\sum_{\nu=0}^{\ell-m} T_\nu^{\ell m}(1 - x)^\nu + mx\sum_{\nu=0}^{\ell-m} T_\nu^{\ell m}(1 - x)^\nu\right].$$

c. $x = \cos\theta = 0;$

$$\frac{\partial Y_{\ell m}}{\partial\theta} = 0,\ \ell - m = 0 \bmod (2);$$

$$\frac{\partial Y_{\ell m}}{\partial\theta} = -\,N_{\ell m}(1 - x^2)^{(m-1)/2}\,\omega^{\ell m}_{[(\ell - m)/2]}\,.$$

d. $x = \cos\theta = 1,$

$$\frac{\partial Y_{\ell m}}{\partial\theta} = 0,\ m \neq 1;$$

$$\frac{\partial Y_{\ell m}}{\partial\theta} = -N_{\ell m}\,T_o^{\ell m}\,.$$

3. Remarks

Figure 4 shows the total time and time per harmonic to compute GRAD.

H. Limitations of the Spherical Harmonic Generator Package

Due to rounding error, the Spherical Harmonic Generator Package produces accurate results at all angles only for $\ell \leq 50$. The package works to $\ell = 60$ only for $0 \leq \theta \leq 30^o$ and $55^o \leq \theta \leq 90^o$. In the range $30^o < \theta < 55^o$ the precision available using the IBM System 360 computer is not great enough to calculate the terms for $\ell > 50$ with $m \leq 20$. This is not a serious limitation for the cubic system where the symmetry operation can be used (through the subroutine DORD) to keep the unit vector in the basic 48th of the unit sphere in which $54.7^o < \theta < 90^o$.

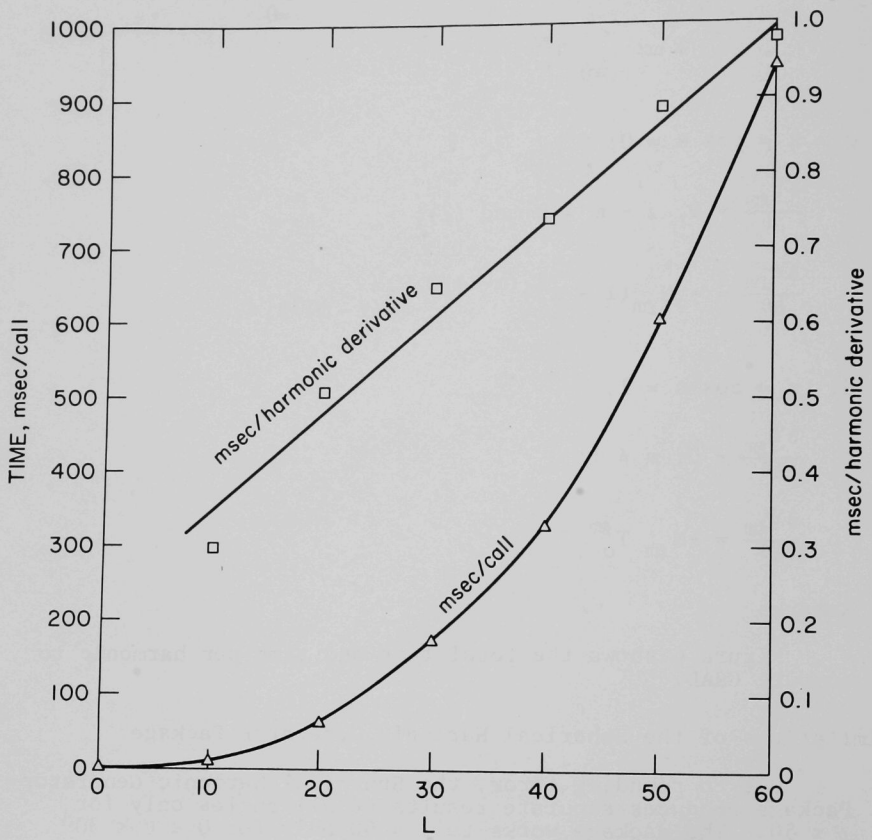

Fig. 4. The Total Time and the Time per Harmonic to Compute GRAD

III. POLYNOMIAL GENERATOR PACKAGE

A. Harmonic-function Generators, xxPOLY

1. Purpose

The 10 harmonic-function generator routines produce polynomials that are invariant under the group operations of the ten crystal groups that have full inversion symmetry. These subroutines are:

Subroutine name	Crystal group
KUPOLY	O_h
HXPOLY	D_{6h}
C6POLY	C_{6h}
D3POLY	D_{3d}
S6POLY	S_6
SQPOLY	D_{4h}
C4POLY	C_{4h}
VHPOLY	V_h
C2POLY	C_{2h}
S2POLY	S_2

2. Restrictions

a. Other subroutines required: SPHERE
YLM
IYLM
BLKDAT
COMSET

b. Common blocks necessary: *KUBE*
CARRY

3. Usage

a. Entry point: xxPOLY

b. Calling sequence: CALL xxPOLY (THETA, PHI, B) where

(1) THETA and PHI are the spherical polar coordinates of the point at which the harmonics are to be evaluated;

(2) B is a vector containing the values of the harmonics evaluated at that point. B is 91 elements long. Only the first KE elements are evaluated (see below). The elements of B have their ℓ value specified by the corresponding element of NAM (in common block KUBE).

c. Common blocks necessary:

(1) KUBE contains: NAM (91), SYM (91, 16), KE, KLN. In subroutine KUPOLY, NAM and SYM are specified as input data. For the other subroutines, NAM is evaluated in the subroutine and SYM is unused (but dimensioned). NAM contains the ℓ numbers of the elements of B. KE and KLN are parameters giving information on how many harmonics are to be calculated. KLN is the maximum ℓ number, and KE gives the total number of harmonics up to and including the harmonics of order ℓ.

(2) CARRY contains a group of parameters including the variable NUT. NUT = 69 if the current values of the element of NAM are the correct ℓ values. Otherwise, these values are computed in the subroutine. (Note: In subroutine KUPOLY, NUT is not specified.)

4. Algorithm

A. The method used is to utilize proper linear combinations of the spherical harmonics which have the required symmetry. Inversion symmetry requires that ℓ be even. For all subroutines except KUPOLY, the rotational symmetry (about the axis of highest symmetry) determines the allowed m values. The existence of an additional mirror symmetry element determines if only symmetric ($C^g_{\ell m}$) or if both symmetric and antisymmetric ($C^u_{\ell m}$) harmonics are to be allowed:

$$C^g_{\ell,m} = (2)^{-1/2}\,[Y_{\ell,m} + Y_{\ell,-m}];$$

$$C^u_{\ell,m} = -i(2)^{-1/2}[Y_{\ell,m} - Y_{\ell,-m}].$$

For the O_h group (subroutine KUPOLY), additional symmetry considerations lead to the use of linear combinations of the $C^g_{\ell,m}$:

$$_iK_\ell(\theta,\varphi) = \sum_m {}_ia_{\ell m}\, C^g_{\ell,m}(\theta,\varphi)\ .$$

The $_ia_{\ell m}$ are computed in the program HAGEN and are listed in Appendix B up to order $\ell = 60$.

5. Remarks

If we write the spherical harmonic $Y_{\ell m}$ as $Y_{\ell m} = \overline{Y}_{\ell m}\, e^{im\varphi}$, where $\overline{Y}_{\ell m}$ is the quantity computed in subroutine SPHERE, then

$$\left.\begin{aligned} C^g_{\ell,m} &= \sqrt{2}\,\overline{Y}_{\ell m}\cos m\varphi \\ C^u_{\ell,m} &= \sqrt{2}\,\overline{Y}_{\ell m}\sin m\varphi \end{aligned}\right\}\ m \neq 0$$

and

$$C^g_{\ell,0} = \overline{Y}_{\ell,0}$$

$$C^u_{\ell,0} = 0.$$

This accounts for the factors of $\sqrt{2}$ in the harmonic function generator subroutines.

B. Harmonic-function Theta-derivative Generators, xxPOLT

1. Purpose

These 10 subroutines compute the partial derivatives of the functions calculated in xxPOLY with respect to theta. The naming convention is the same as for the harmonic function generator.

2. Restrictions

a. Other subroutine required: GRAD
YLM
IYLM
BLKDAT
COMSET

b. Common blocks necessary: *KUBE*
CARRY

3. Usage

a. Entry point: xxPOLT

b. Calling sequence: CALL xxPOLT (THETA, PHI, B) where

(1) THETA and PHI are the spherical polar coordinates of the point at which the harmonics are to be generated;

(2) B is the vector containing the harmonic derivatives. (See Sect. A.3.b, of the polynomial generator package write-up.),

c. Common blocks necessary: KUBE, CARRY (See Sect. A.3.c above).

4. Algorithm

The method is the same as for the xxPOLY routines.

C. Harmonic-function Phi-derivative Generators, xxPOLP

1. Purpose

These 10 subroutines compute the partial derivatives of the functions calculated in xxPOLY with respect to phi. The naming convention is the same as for the xxPOLY routines.

2. Restrictions

a. Other subroutines required: SPHERE
YLM
IYLM
BLKDAT
COMSET

b. Common blocks necessary: *KUBE*
CARRY

3. Usage

a. Entry point: xxPOLP

b. Calling sequence: CALL xxPOLP (THETA, PHI, B) where

(1) THETA and PHI are the spherical polar coordinates of the point at which the harmonic derivatives are to be calculated;

(2) B is the vector containing the harmonic derivative. (See Sect. A.3.b above.)

c. Common blocks necessary: KUBE, CARRY (See Sect. A.3.c above.)

4. Algorithm

The method is the same as for the xxPOLY routines.

D. Legendre Polynomial Generator COET

1. Purpose

This subroutine evaluates the Legendre polynomial $P_\ell(x)$ at the point $x = 0$.

2. Restrictions

a. Other subroutines required: None

b. Common storage: *KUBE*

3. Usage

a. Entry point: COET

b. Calling sequence: CALL COET (B), where B is a 91-element vector containing the values of the $P_\ell(0)$ functions. The ℓ values of the elements are given by the corresponding elements of the vector NAM in common block KUBE.

c. Common blocks required: KUBE containing NAM (91), SYM (91, 16), KE, KLN. This block is described in Section A above on polynomial generator package routines.

4. Algorithm

The function is calculated directly as an accumulated product.

IV. MULTIPURPOSE SUBROUTINES

A. KLSQ

1. Purpose

This routine is a least-squares-fitting routine.

2. Restrictions

a. Other subroutines required:

xxPOLY	ELLIPS
LAB	MATINV
TRANS	EULER
DGOTTA	DFLOP
DCRISS	OILER
DWOW	DGETTA

b. Common storage: *WORK*
OVER
INTO
TEEPEE
ORIENT
LIMIT

3. Usage

a. Entry point: KLSQ

b. Calling sequence: CALL KLSQ (NUM, KE, CAL, DELTA, FUNC, IDELTA), where

(1) NUM is the number of data planes;

(2) KE is the number of harmonics included in the fit;

(3) CAL is the calibration parameter;

(4) DELTA is the standard deviation;

(5) FUNC is the name of the particular harmonic generator xxPOLY to be used;

(6) IDELTA is set to zero if standard deviations are not required.

c. Common blocks necessary:

WORK contains working arrays to be shared by several sources;
OVER is the necessary common block for MATINV;
INTO contains the input data IN(J,I), R(J,I);
TEEPEE contains the matrices of (θ,φ) values for the points of each data plane;
ORIENT contains parameters necessary for the orientation of the data planes;
LIMIT contains information concerning the blocks of parameters to be varied and the order these blocks will be varied.

4. Algorithm

The lab angles are converted into (θ,φ) values (crystal angles) by subroutine LAB (with auxiliary routine TRANS). When surfaces having several symmetry-equivalent pockets are being fit, TRANS rotates the data points so that all the data refer to a single pocket. Otherwise, TRANS is a dummy subroutine. If an ellipsoidal transformation is required, ELLIPS is used to perform it on the input data. Otherwise, it too is a dummy routine. Then a standard least-squares procedure is performed that determines the best set of coefficients to fit the data to a set of orthogonal polynomials (the appropriate harmonics). Finally if IDELTA $\neq$ 0, the standard deviation is calculated.

B. LAB

1. Purpose

This subroutine transforms the input angular data to spherical polar angles relative to some crystal axes.

2. Restrictions

a. Other subroutines required: DGOTTA
DCRISS
DWOW
TRANS

b. Common storage: *INTO*
TEEPEE
ANGLE
ORIENT

3. Usage

a. Entry point: LAB

b. Calling sequence: CALL LAB (I, CAL), where

(1) I is the number of the plane being considered;

(2) CAL is the calibration parameter;

(3) TEEPEE contains the (θ,φ) values of the NT(I) points in plane I;

c. Common blocks necessary: INTO contains input data, TEEPEE contains output angles, ANGLE and ORIENT contain information necessary for data plane orientation.

4. Algorithm

a. Method. The input angular data are in the form of integers which may be read directly from the sample rotator. Such a number is converted into degrees by multiplying by 0.1/CAL, where CAL = 0.965 for the current sample rotator. The angle is, of course, corrected to a real zero (parameter NZ). The directions (T1, P1) and (T2, P2) specify two unit vectors X and Y, which define the data plane in question. Since, in general, these are not perpendicular,

$$\vec{Z} = \frac{\vec{X} \times \vec{Y}}{|\vec{X} \times \vec{Y}|} \quad \text{and} \quad \vec{Y}' = \frac{\vec{Z} \times \vec{X}}{|\vec{Z} \times \vec{X}|}$$

are formed. Under this construction, $\vec{X} \cdot \vec{Y}' = 0$. The data-point direction is therefore expressed in component form in the directions of $\vec{X}$ and $\vec{Y}'$. Then the effect of a cone angle is included. Inaccuracies in the sample-mounting procedure result in the magnetic field rotating on a cone relative to the sample axis for which the apex angle differs from 90° by a small quantity (usually of the order of tenths of a degree). This rotation must be taken into account for accurate determination of crystal directions. Next TRANS is called. TRANS is used for symmetry-equivalent Fermi-surface sheets consisting of pockets centered at points of the Fermi surface other than Γ. TRANS is used to perform coordinate rotations which superimpose the various pockets so that areas or radii can be calculated for them individually. When there is only one surface, TRANS is a dummy subroutine. Finally, the unit vector used throughout the calculations described above is converted to the polar coordinates θ and φ and is stored in common block TEEPEE. This routine calculates θ and φ values for all NT(I) points of plane I.

References: L. R. Windmiller and J. B. Ketterson, Rev. Sci. Inst. 39, 1672 (1968)

F. M. Mueller, L. R. Windmiller, and J. B. Ketterson, J. Appl. Phys. 41, 2312 (1970).

C. MATINV

1. Purpose

This subroutine calculates the inverse $\underline{A}^{-1}$ of a square matrix $\underline{A}$.

2. Restrictions

a. Other subroutines required: None

b. Common storage: *OVER*

3. Usage

a. Entry point: MATINV

b. Calling sequence: CALL MATINV

c. Common block necessary: The common block OVER must be supplied in the calling program

COMMON/OVER/A,N

where

(1) A is the matrix to be inverted;

(2) N is the order of A; $N \geq 1$.

4. Algorithm

a. Method. Gauss-Jordon elimination with full pivoting.

b. References: This routine is a version of the subroutine ANL F402S of the ANL subroutine library. The write up of this routine should be consulted for additional information.

5. Remarks. The A matrix is arbitrarily given the dimensions 91 x 91, so that any nonsingular matrix of order $\leq$ 91 may be inverted. The revision of this routine to other maximum dimensions is straightforward.

D. LTOM AND MTOL

1. Purpose

In working with the linear combinations of the spherical harmonics that are summed to form the representation of the data, we desire to label a given series in either of two ways:

a. By the total number of terms (the parameter KE), or

b. By the largest ℓ value to appear in a series (the parameter KLN).

These routines transfer between these representations.

2. Restrictions

a. Other subroutines required: None.

b. Common storage: None.

3. Usage: LTOM

a. Entry point: LTOM

b. Calling sequence: KE = LTOM (KLN, I), where I indicates the crystal group.

4. Usage: MTOL

a. Entry point: MTOL

b. Calling sequence: KLN = MTOL (KE, I)

5. Algorithm

The number of terms for a given ℓ value is determined by the properties of the group being considered.

Group	I
O_h	1
D_{6h}	2
C_{6h}	3
D_{3d}	4
$S_6(C_{3i})$	5
D_{4h}	6
C_{4h}	7
$V_h(D_{2h})$	8
C_{2h}	9
S_2	10

E. TRANS

1. Purpose

This subroutine is used when the Fermi surface consists of several symmetry-equivalent pockets. TRANS performs the necessary rotations on the data that map the pockets into a single pocket.

2. Restrictions

a. Other subroutines required: In general, EULER

b. Common storage: In general, block *ROTATE*

3. Usage

a. Entry point: TRANS

b. Calling sequence: CALL TRANS (W, I, J) where W is the unit position vector of the Jth point of the Ith plane of data.

c. Common blocks necessary: ROTATE--contains an integer array which tells which pocket a given point is from.

4. Algorithm

The method is a straightforward procedure of rotation. Three subroutines are illustrated. The first is a dummy routine performing no rotations, which is provided merely to resolve the external reference in subroutine LAB when symmetry-equivalent pockets do not exist. The second is routine for handling D_{4h} pockets in a cubic crystal (as in the d-like pockets of the Fermi surface of platinum). The third is a routine for handling C_{2h} pockets in a D_{3d} crystal (as in the electron surface in antimony or arsenic).

F. ELLIPS

1. Purpose

This subroutine performs the ellipsoid transformation. Both the regular and a dummy version (for use with O_h groups) are provided.

2. Restrictions

a. Other subroutines required: OILER
EULER
DFLOP
DGOTTA
DGETTA

b. Common storage: *TEEPEE*
INTO
CARRY

3. Usage

a. Entry point: ELLIPS

b. Calling sequence: CALL ELLIPS (I, J, T, P, RS) where

(1) the Jth point of the Ith plane is considered;

(2) the (transformed) angular coordinates are (T, P);

(3) the value of the quantity to be fit (transformed) is RS.

c. Common blocks necessary

(1) TEEPEE contains the angular coordinates of the points in the Ith plane. It is used to determine (T,P).

(2) INTO contains the input data. It is used to obtain RS.

(3) CARRY contains information on the transformation to be performed.

4. Algorithm

a. Method: The version of ELLIPS for the O_h group is essentially a dummy subroutine. The general version of ELLIPS, however, is designed to allow three options: (1) If $\gamma = 1$, no transformation is performed; (2) if LUMP = 0, the parameter to be fit is taken to be $1/(\text{AREA})^2$ rather than AREA; (3) if $\gamma \neq 1$, the ellipsoidal transformation is performed.

References: J. B. Ketterson and L. R. Windmiller, Phys. Rev. 1, 463 (1970)

V. VECTOR SUBROUTINES

A. GOTTA and DGOTTA

1. Purpose

GOTTA produces a unit vector in the direction (θ, φ), where θ and φ are spherical polar coordinates. DGOTTA is the double-precision version of GOTTA.

2. Restrictions

a. Other subroutines required: None

b. Common storage: None

3. Usage

a. Entry point: GOTTA (DGOTTA)

b. Calling sequence: CALL GOTTA (X, T, P)
CALL DGOTTA (X, T, P),

where

(1) X is the three-component Cartesian resultant vector;

(2) T and P are the angles φ and θ, respectively.

4. Algorithm

Method: $x_1 = \cos\theta \sin\varphi;$

$x_2 = \sin\theta \sin\varphi;$

$x_3 = \cos\varphi.$

B. CRISS and DCRISS

1. Purpose

CRISS computes the components of a unit vector in the direction of the cross product of two vectors. DCRISS is the double-precision version of CRISS.

2. Restrictions

a. Other subroutines required: WOW (DWOW)

b. Common storage: None

3. Usage

a. Entry point: CRISS (DCRISS)

b. Calling sequence: CALL CRISS (X, Y, Z)
CALL DCRISS (X, Y, Z),

where

(1) X and Y are three-component Cartesian vectors;

(2) Z is a unit Cartesian vector in the direction $\vec{X} \times \vec{Y}$.

4. Algorithm

Method: $Z_j = (\vec{X} \times \vec{Y})_j / \left[\sum_{i=1}^{3} Z_i^2 \right]^{1/2}.$

C. WOW and DWOW

1. Purpose

WOW takes a vector $\vec{X}$ and computes a unit vector in the same direction as $\vec{X}$. DWOW is the double-precision version of WOW.

2. Restrictions

a. Other subroutines required: None.

b. Common storage: None.

3. Usage

a. Entry point: WOW (DWOW)

b. Calling sequence: CALL WOW (X), CALL DWOW (X), where X is a three (Cartesian) component vector. WOW replaces X with the unit vector $\vec{X}/|\vec{X}|$.

4. Algorithm

Method: $X_j = X_j / \left(\sum_{i=1}^{3} X_i^2 \right)^{1/2}$.

D. ORD and DORD

1. Purpose

These subroutines order the components of a vector X in decreasing order of magnitude:

$$|x| \geq |y| \geq |z| ,$$

DORD is the double-precision version of ORD.

2. Restrictions

a. Other subroutines required: None.

b. Common storage: None.

3. Usage

a. Entry point: ORD, DORD

b. Calling sequence: CALL ORD (X)
CALL DORD (X),

where X is the vector whose components are to be ordered.

E. FLOP and DFLOP

1. Purpose

These subroutines calculate an angle θ in the range $0 \leq \theta \leq 2\pi$ given $\sin\theta$ and $\cos\theta$. DFLOP is the double-precision version of FLOP. Some machines include the subroutines ATAN2(Y, X) and DATAN2 (Y, X), which perform the same functions as FLOP and DFLOP, in which case this subroutine is not required.

2. Restrictions

a. Other subroutines needed: None.

b. Common storage: None.

3. Usage

a. Entry point: FLOP, DFLOP

b. Calling sequence:

```
CALL FLOP (S, C, T)
CALL DFLOP (S, C, T),
```

where

(1) $S = \sin\theta$;

(2) $C = \cos\theta$;

(3) $T = \theta$.

4. Algorithm

$\theta = \arctan(\sin\theta/\cos\theta)$, where the special case of $\cos\theta = 0$ is handled explicitly to prevent a divide check exception.

F. GETTA and DGETTA

1. Purpose

These subroutines determine the spherical polar coordinates (θ and φ) of a vector $\bar{W}$. DGETTA is a double-precision version of GETTA.

2. Restrictions

a. Other subroutines needed: FLOP, WOW {(in GETTA); DFLOP, DWOW {(in DGETTA)

b. Common storage: None.

3. Usage

a. Entry point: GETTA, DGETTA

b. Calling sequence: CALL GETTA (W, THETA, PHI)
CALL DGETTA (W, THETA, PHI),

where

(1) W is the input vector;

(2) THETA and PHI are the output spherical polar coordinates.

4. Algorithm

$$\vec{W} = (X, Y, Z);$$

$$\vec{W} \rightarrow \frac{\vec{W}}{(X^2 + Y^2 + Z^2)^{1/2}};$$

$$\theta = \cos^{-1}(Z);$$

$$\varphi = \tan^{-1}(Y/X).$$

G. OILER

1. Purpose

This subroutine rotates a vector represented by the spherical polar coordinates θ and φ by the rotation operator specified by the Euler angles α, β and γ.

2. Restrictions

a. Other subroutines required: DGOTTA
DGETTA
DFLOP
EULER

b. Common storage: None.

3. Usage

a. Entry point: OILER

b. Calling sequence: CALL OILER (ALPHA, BETA, GAMMA, THETA, PHI, I), where

(1) ALPHA, BETA, and GAMMA are the three Euler angles;

(2) (THETA, PHI) are the spherical polar coordinates of the vector to be rotated;

(3) If $I > 0$, the Euler angle rotation R is performed; if $I < 0$, the Euler angle rotation R^{-1} is performed.

4. Method

The angles θ and φ are converted to a unit vector in the same direction. This vector is rotated by R or R^{-1} in subroutine EULER. The resultant vector is converted back to spherical polar coordinates.

H. EULER

1. Purpose

This subroutine rotates a vector $\vec{X}$ by the Euler angle-rotation operator R.

2. Restrictions

a. Other subroutines required: None.

b. Common storage: None

3. Usage

a. Entry point: EULER

b. Calling sequence: Call EULER (ALPHA, BETA, GAMMA, X, I), where

(1) ALPHA, BETA and GAMMA are the three Euler Angles;

(2) X is the vector to be rotated;

(3) If $I > 0$, the Euler angle rotation R is performed; if $I < 0$, the Euler angle rotation R^{-1} is performed.

4. Algorithm

Method. The transformation matrix R of the operator R is computed directly. If R^{-1} is to be used, R^{T} (R transpose) $= R^{-1}$ is used.

Reference: H. Goldstein; Classical Mechanics, Addison-Wesley Publishing Co., Reading, Mass. (1950), pp. 107-109.

VI. SPECIAL-PURPOSE SUBROUTINES

A. WIG

1. Purpose

This subroutine calculates the Wigner matrix elements $d^{j}_{m'm}(\beta)$ for a rotation through the Euler angle $\beta = \pi/2$. If

R is a rotation operator and we consider the Euler angles α, β, and γ, then

$$R\ \psi_{jm} = \sum_{m'} e^{-im'\alpha}\ d^{j}_{m'm}(\beta)\ e^{-im\gamma}\psi_{jm'}\ .$$

2. Restrictions

a. Other subroutines required: None.

b. Common storage: None.

3. Usage

a. Entry point: WIG

b. Calling sequence: CALL WIG (J, M1, M, AN), where

(1) J, M1, and M are the obvious quantum numbers;

(2) AN is the value of $d^{j}_{m'm}\ (\pi/2)$;

c. Common blocks necessary: None.

4. Algorithm

a. Method. The value of $d^{j}_{m'm}(\pi/2)$ is calculated from the equations

$$d^{j}_{m'm}(\beta) = \left[\frac{(j-m)!(j+m')!}{(j+m)!(j-m')!}\right]^{1/2} \frac{(\cos\beta/2)^{2j+m-m'}(-\sin\beta/2)^{m'-m}}{(m'-m)!}$$

$$\times\ {}_2F_1\ (m'-j, -m-j; m'-m+1;\ -\tan^2\frac{\beta}{2});\ m' \geq m;$$

$$d^{j}_{m'm}(\beta) = (-1)^{m'-m} d^{j}_{mm'}(\beta);\ d^{j}_{m'm}(\beta) = (-1)^{m'-m}\ d^{j}_{-m'-m}(\beta)$$

where ${}_2F_1(m'-j,\ -m-j;\ m'-m+1;\ -\tan^2\frac{\beta}{2})$ is the confluent hypergeometric function.

Reference: M. E. Rose, Elementary Theory of Angular Momentum, John Wiley and Sons, Inc., New York (1963), p. 53.

5. Remarks

This subroutine is designed for even values of m and m', and for $\beta = \pi/2$. The modification to greater generality should be straightforward.

The algorithm used is subject to rounding error at high orders of j. For $j > 40$, use of a machine with very wide double-precision arithmetic (such as the CDC 3600) is recommended. Use of a nesting algorithm in the calculation of the hypergeometric function would probably also increase stability in the calculation, although it was not attempted in this program.

B. EIGEN

1. Purpose

This subroutine finds all eigenvalues λ_i (including proper multiplicity) and all eigenvectors $\vec{x}_i$ of the real symmetric matrix A, satisfying $A\vec{x} = \lambda\vec{x}$. Options include limitation of the computation to the n largest eigenvalues and avoidance of the calculation of eigenvectors.

2. Restrictions

a. Other subroutines required: None.

b. Common storage: One block of labeled common, called F202.

3. Usage

a. Entry point: EIGEN

b. Calling sequence: CALL EIGEN (A, B, N, VALU, M, ANORM, NMAX) where

(1) A is the input matrix of order N, which is destroyed in the calculation. If eigenvectors are calculated, they are stored columnwise at A.

(2) B is an N x N array used as work space if eigenvectors are desired. If eigenvectors are not desired, B will not be used, but it still must be included in the calling sequence.

(3) VALU is an array for the storage of eigenvalues in descending order by magnitude.

(4) M is the number of eigenvalues to be found. If M is negative, eigenvectors are not calculated.

(5) ANORM returns the matrix norm, defined as $\sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij}^2}$.

(6) NMAX is the row DIMENSION of A and B.

c. Common blocks necessary: None

4. Algorithm

a. Method. The matrix A is reduced to tridiagonal form by the Householder reduction. The eigenvalues of the tridiagonal matrix are found by Given's method, after which the eigenvectors are computed by the Wilkinson inverse iteration. Gram-Schmidt orthonormalization ensures that eigenvectors of multiple roots are orthogonal. Finally, the eigenvectors are transformed back to the eigenvectors of A.

b. Reference: Subroutine EIGEN is subroutine ANL F202S of the ANL subroutine library. The write up of this routine in the Applied Mathematics Division of Argonne National Laboratory should be consulted for additional information and references.

5. Remarks

The eigenvalues are accurate to 2^{-51} x ANORM. The eigenvectors should be accurate to 11 or 12 decimal places.

APPENDIX A

Inversion Programs--Input-data Format

For programs J111-J1110 and J121-J1210, Data Block will refer to the following:

Name	No. of Cards	Format	Remarks
ZZ	1	F10.4	Correction parameter
T1	1	F10.4	The spherical polar unit vectors (1,T1,P1) and (1,T2,P2) define the data plane.
P1	1	F10.4	
T2	1	F10.4	
P2	1	F10.4	
TL	1	F10.4	Cone apex angle
NT	1	I5	Number of points in plane
NL	1	I5	Lowest value of angle variable
NZ	1	I5	Value of angle variable corresponding to the field parallel to the magnet axis
IN,R	NT	15X, I5, F10.6	IN is the angle variable, R the experimental value at that angle.

Alternative form of data cards:

IN,R,NCODE	NT	15X,I5,F10.6,14X,I1	NCODE is used to specify which pocket of the Fermi surface (for a surface composed of a set of pockets) is being considered.

For programs J111-J1110 and J121-J1210, Traceback Block will refer to the following:

Name	No. of Cards	Format	Remarks
T11	1	F10.4	Polar coordinates defining the traceback plane.
P11	1	F10.4	
T21	1	F10.4	
P21	1	F10.4	
TL2	1	F10.4	Cone apex angle of traceback plane.
NTOT	1	I5	Number of angular steps to be made in traceback plane.
NSPACE	1	I5	Size of angular step.
NSTART	1	I5	Start position of angular coordinate (not used in J13x programs)
NAXIS	1	I5	Value of angular coordinate at (T11, P11) (not used in J13x programs).

1. INPUT DATA: J111, J121

Name	No. of Cards	Format	Remarks
TITLE	1	20A4	Label for the data being considered.
LMIN	1	I5	Starting ℓ value.
LMAX	1	I5	Ending ℓ value.
NAM, SYM	455	15X,I5/4F20.15/4F20.15/4F20.15/4F20.15	Coefficients for the Kubic harmonics.
LMID	1	I5	Separating point between ℓ values to be held constant and ℓ values to be varied.
LNIT	1	I5	Starting iteration number to determine if lower or upper upper set of ℓ values is to be varied first.
NIT	1	I5	Number of iterations.
NINIT	1	I5	Number of expansion coefficients to be read.

II,B	NINIT	10X,I4,E17.6	Coefficient number and initial values of expansion coefficients.
CAL	1	F10.4	Calibration parameter for angle variable.
NUM	1	I5	Number of data planes.
Data blocks			Data
NUMB	1	I5	Number of traceback planes
Traceback blocks			Data

In the following, only those parameters not previously used will be defined.

2. INPUT DATA: J112-J1110 and J122-J1210

Name	No. of Cards	Format	Remarks
TITLE	1	20A4	
LMIN	1	I5	
LMAX	1	I5	
LUMP	1	I5	Lump = 0 implies the fit is to $1/(\text{AREA})^2$ (or to $1/(m^*)^2$).
GAMMA	1	F10.4	Spherical mapping parameter.
ALPHA	1	F10.4	(J118, J119, J1110, J128, J129, J1210 only). Spherical mapping parameter.
E1	1	F10.4	(J1110, J1210 only) Ellipsoidal mapping parameter.
E2	1	F10.4	(J119, J1110, J129, J1210 only) Ellipsoidal mapping parameter.
E3	1	F10.4	(J1110, J1210 only) Ellipsoidal mapping parameter.

LMID	1	I5
LNIT	1	I5
NIT	1	I5
NINIT	1	I5
II,B	NINIT	10X,I4,E17.6
CAL	1	F10.4
NUM	1	I5
DATA BLOCKS		
NUMB	1	I5
TRACEBACK BLOCKS		

3. INPUT DATA: J131

Name	No. of Cards	Format	Remarks
NAM,SYM	455	15X,I5/4F20.15/4F20.15/4F20.15/4F20.15	
LC	1	I4	Number of expansion coefficients for k^2.
IN,C	LC	10X,I4,E17.6	k^2 expansion coefficients.
LD	1	I4	Number of expansion coefficients for $(1/\pi)2k\ dk/dE$.
IN,D	LD	10X,I4,E17.6	$(1/\pi)2kdk/dE$ expansion coefficients.
NUMB	1	I5	Number of traceback planes.
TRACEBACK BLOCKS			

INPUT DATA: J132-J1310

GAMMA	1	F10.4		k^2 parameters
ALPHA	1	F10.4	(J138, J139, J1310 only)	
EA1	1	F10.4	(J1310 only)	
EA2	1	F10.4	(J139, J1310 only)	
EA3	1	F10.4	(J1310 only)	

LC	1	I4		
IN,C	LC	10X,I4,E17.6		
GAMMB	1	F10.4		$\frac{1}{\pi}$2k dk/dE parameters
ALPHB	1	F10.4	(J138,J139,J1310 only)	
EB1	1	F10.4	(J1310 only)	
EB2	1	F10.4	(J139,J1310 only)	
EB3	1	F10.4	(J1310 only)	
LD	1	I4		
IN,D	LD	10X,I4,E17.6		
NUMB	1	I5		

TRACEBACK BLOCKS

4. INPUT DATA: J141

Name	No. of Cards	Format	Remarks	
NAM,SYM	455	15X,I5/4F20.15/4F20.15/4F20.15/4F20.15		
LC	1	I4		
IN,C	LC	10X,I4,E17.6		
LD	1	I4		
IN,D	LD	10X,I4,E17.6		
N	1	I4	Numbers of integration steps +1 for θ integration	Must be odd no.
M	1	I4	Numbers of integration steps +1 for φ	
A	1	F10.6	Lattice constant in Å.	
NCELL	1	I4	Number of atoms per unit cell.	

INPUT DATA: J142-J1410

GAMMA	1	F10.4	
ALPHA	1	F10.4	(J148,J149,J1410 only)
EA1	1	F10.4	(J1410 only)
EA2	1	F10.4	(J149, J1410 only)
EA3	1	F10.4	(J1410 only)
LC	1	I4	
IN, C	LC	10X,I4,E17.6	
GAMMB	1	F10.4	
ALPHB	1	F10.4	(J148,J149,J1410 only)
EB1	1	F10.4	(J1410 only)
EB2	1	F10.4	(J149,J1410 only)
EB3	1	F10.4	(J1410 only)
LD	1	I4	
IN,D	LD	10X,I4,E17.6	
N	1	I4	
M	1	I4	
VOL	1	F10.4	Volume of unit cell in $(\text{Å})^3$.

5. INPUT DATA: J151

Name	No. of Cards	Format	Remarks
NAM,SYM	455	15X,I5/4F20.15/4F20.15/4F20.15/4F20.15	
L	1	I4	Number of expansion coefficients for k^2.
IN,C	L	10X,I4,E17.6	k^2 expansion coefficients.
N	1	I4	

M	1	I4	
A	1	F10.6	
NCELL	1	I4	

INPUT DATA: J152-J1510

GAMMA	1	F10.4	
ALPHA	1	F10.4	(J158,J159,J1510 only)
E1	1	F10.4	(J1510 only)
E2	1	F10.4	(J159, J1510 only)
E3	1	F10.4	(J1510 only)
L	1	I4	
IN, C	L	10X,I4,E17.6	
N	1	I4	
M	1	I4	
VOL	1	F10.4	

APPENDIX B

Kubic Harmonic Expansion Coefficients

These coefficients are presented here in the same format as read by the main calling programs. The first line of each group gives the number of the kubic harmonic, followed by the values of ℓ and i for that harmonic (as defined in the Hagen program). This is then followed by the values of ${}_i a_{\ell m}$ as defined in the Hagen program.

KUBIC HARMONIC EXPANSION COEFFICIENTS

1 0	1		
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
2 4	1		
.763762615824817	0.645497224366409	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
3 6	1		
.353553390588786	-0.935414346691687	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
4 8	1		
.718070330811315	0.381881307912408	0.581843335152371	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
5 10	1		
.411425367870834	-0.586301969975466	-0.697838926018449	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
6 12	1		
.695502665941603	0.314125566801522	0.348449537588749	0.544227975842659
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
7 12	2		
.000000000000000	0.558979374196497	-0.806267508174642	0.193583998479880
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
8 14	1		
.440096461963549	-0.457681828615023	-0.491132301416656	-0.596348480685265
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
9 16	1		
.681361683309660	0.275868022756185	0.290489869839803	0.327569746979862
.517645425847149	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
10 16	2		
.000000000000000	0.637048214630340	-0.329990348807769	-0.647980748501141
.255728159332648	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
11 18	1		
.457915144208528	-0.386455976316938	-0.402094634002424	-0.437465928524034
.536571491727955	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000

12 18	2		
.000000000000000	0.148727527117444	-0.637746007749229	0.723341669436195
.218945156411792	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
13 20	1		
.671414949683822	0.249826188952284	0.257828465102648	0.274693332583411
.312489189767803	0.497199560064473	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
14 20	2		
.000000000000000	0.662995391001459	-0.112952591471185	-0.427384410235391
.528104333527153	0.293474364065332	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
15 22	1		
.470327478040417	-0.339860080595827	-0.348712525919836	-0.366503004333936
.401831606184714	-0.495876669498102	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
16 22	2		
.000000000000000	0.214974719277961	-0.670455529878382	0.151269299742125
.627459855240886	-0.296119876926241	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
17 24	1		
.663917794998270	0.230436286688928	0.235420938588504	0.245104945162893
.262927777766890	0.300744878324622	0.480660304798221	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
18 24	2		
.000000000000000	0.672701793955639	-0.003610688737058	-0.244969194096484
.446693402598612	-0.431713899975875	0.318649440465379	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
19 24	3		
.000000000000000	0.000000000000000	0.265830384390533	-0.680361133956467
.634242147571058	-0.252040683662926	0.027597432526363	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
20 26	1		
.479596346165636	-0.306426266863127	-0.312024897975789	-0.322631850453035
.341164199926425	-0.375900931176147	-0.465773462579818	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
21 26	2		
.000000000000000	0.260499090909434	-0.639562455326086	-0.116813297747285
.367984850876383	0.507573744238471	-0.341190249702777	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
22 28	1		
.657999990260578	0.215198937989044	0.218566603205545	0.224825290708395
.235356956749456	0.253603776734963	0.291164731039316	0.466822220165341
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
23 28	2		
.000000000000000	0.676034145653830	0.056323410874938	-0.125152165637701
.313765075195988	-0.433188029586745	-0.353503996855579	0.336269200990500
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000

24 28	3		
.000000000000000	0.000000000000000	0.377786082914099	-0.637888735858724
.013429084884820	0.568618020872236	-0.352895826981694	0.048305046561836
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
25 30	1		
.486850799803506	-0.280977125308709	-0.284787663767929	-0.291763294546399
.303174059219600	-0.321899282629602	-0.355892797349952	-0.442272516374942
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
26 30	2		
.000000000000000	0.294081043255574	-0.597019237131462	-0.243755925668665
.139179571389832	0.423777995485580	0.402713797971956	-0.369500031840289
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
27 30	3		
.000000000000000	0.000000000000000	0.069936050926117	-0.360823909737519
.675716768018901	-0.590607398960856	0.241463897385984	-0.034640884590772
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
28 32	1		
.653171715079225	0.202780001112842	0.205187214560283	0.209540928753995
.216508277597313	0.227498041705985	0.245911633392097	0.283099747932283
.454959993774537	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
29 32	2		
.000000000000000	0.676543561494327	0.091164015511822	-0.047140589808805
.205531019742921	-0.344896162103396	-0.406194937539112	-0.289440043408831
.348983477721049	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
30 32	3		
.000000000000000	0.000000000000000	0.448271891313198	-0.524469263007632
.295055591064738	0.258148473934853	0.442951643541164	-0.411733455875947
.067492740185116	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
31 34	1		
.492724561619980	-0.260792870423757	-0.263526362519769	-0.268421729655529
.276120663991605	-0.287905216500803	-0.306582187251479	-0.339813861530274
.423219899814285	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
32 34	2		
.000000000000000	0.319990832773328	-0.555391453497577	-0.304627662029816
.011032718928618	0.262127304282330	0.421049824661168	0.315863758201886
.387895443003799	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
33 34	3		
.000000000000000	0.000000000000000	0.117231336493205	-0.485848475807870
.545358227464021	0.103487418826262	-0.563814418885158	0.347045529990282
.061282390066481	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
34 36	1		
.649133062659530	0.192386397306109	0.194180361526378	0.197367100001429
.202308723510214	0.209697370766662	0.220939567938331	0.239384422788135
.276153237238759	0.444603580959665	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
35 36	2		
.000000000000000	0.675720997634926	0.112211474095602	0.004709685597732
.125234332484979	-0.255425557901617	-0.354081184617826	-0.374421531138069
.236396953681833	0.358347869565478	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000

36 36	3		
.000000000000000	0.000000000000000	0.495146942885185	-0.414223374158610
.381590917328140	-0.036946379773326	0.363426413627167	0.314637268907973
.444932655584125	0.085037644363183	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
37 36	4		
.000000000000000	0.000000000000000	0.000000000000000	0.122160274464477
.440035824394727	0.663330070674419	-0.537239431650960	0.244048461729108
.056888546729169	0.004048773426916	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
38 38	1		
.497604045638582	-0.244291853032337	-0.246332006950979	-0.249931476784695
.255448760050058	-0.263544800043746	-0.275486333332083	-0.294002943912346
.326501264993567	-0.407336526703148	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
39 38	2		
.000000000000000	0.340650333404483	-0.517792631086195	-0.332685553919873
.105947206036944	0.127596253030788	0.318503413582221	0.395504310305114
.244510104625078	-0.400011967671162	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
40 38	3		
.000000000000000	0.000000000000000	0.157527060036955	-0.543404339114204
.352627451036824	0.369702007577871	-0.207411059349397	-0.448509519024810
.409011484291113	-0.086027469165856	0.000000000000000	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
41 40	1		
.645688618242275	0.183510287963145	0.184891005726968	0.187313038815773
.190989810991596	0.196303212331259	0.203951430707093	0.215328301765112
.233728995437559	0.270065180826350	0.435431229976530	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
42 40	2		
.000000000000000	0.674258599567111	0.125202437182452	0.039993815749767
.066550467388879	-0.180906686924573	-0.284264051304490	-0.350584604253527
.341894150747976	-0.192001426301431	0.365338497300399	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
43 40	3		
.000000000000000	0.000000000000000	0.527550594139029	-0.321749269714928
.382857275559218	-0.206261886705761	0.135347854546126	0.384625860220694
.199258437784010	-0.461405841175292	0.100939334844952	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
44 40	4		
.000000000000000	0.000000000000000	0.000000000000000	0.203110292659403
.560711638478097	0.437863360486517	0.186540520946437	-0.529476556141162
.356047857101657	-0.103147799996805	0.008485891821465	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
45 42	1		
.501739858722431	-0.230484300860553	-0.232054963398696	-0.234796686563641
.238924941408186	-0.244815727310197	-0.253130340162897	-0.265111658598471
.283415726509702	-0.315224932790443	-0.393808138891472	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
46 42	2		
.000000000000000	0.357549544605718	-0.484646436576440	-0.343636214071012
.165783348184050	0.028220973983935	0.209734616448259	0.339500820206013
.361609193467302	0.185642597429251	-0.407965643265925	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
47 42	3		
.000000000000000	0.000000000000000	0.191840381306974	-0.565411722738645
.184707470987632	0.411570985444996	0.110226802155012	-0.342747285641963
.319737570862344	0.443121713062283	-0.108557401559665	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000

48	42	4	
.000000000000000	0.000000000000000	0.000000000000000	0.031779060589542
.193841176893329	0.482401214452693	-0.645452880868106	0.505328911734978
.231339931335242	0.056498273583202	-0.005273377262711	0.000000000000000
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
49	44	1	
.642704794183373	0.175808161013265	0.176898485049605	0.178793701936229
.181627581026987	0.185628631366853	0.191185967913043	0.198996049497509
.210436254859815	0.228749526177126	0.264656025457953	0.427213216382370
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
50	44	2	
.000000000000000	0.672500347776804	0.133260479862656	0.064509105304751
.023449771890682	-0.121922348018416	-0.219033048248093	-0.298887589175138
.339870296716981	-0.310461815002782	-0.154460471534549	0.370598016306758
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
51	44	3	
.000000000000000	0.000000000000000	0.550641769659705	-0.247370270542888
.350273700445541	-0.284101990459021	-0.049268462145847	0.239233911474003
.362352669311804	0.100572575129263	-0.466722451150417	0.115289169223615
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
52	44	4	
.000000000000000	0.000000000000000	0.000000000000000	0.269684289953148
.585879148609820	0.168883696325793	0.408463762789324	-0.131291793597484
.417496652946284	0.418712541817513	-0.146978762932122	0.013701052546594
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
53	46	1	
.505302491656039	-0.218715444625559	-0.219955249802297	-0.222102403731697
.225293853291078	-0.229759302044840	-0.235879362920969	-0.244308362533047
.256263281655265	-0.274332423956366	-0.305500200789538	-0.382088116042723
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
54	46	2	
.000000000000000	0.371658698997635	-0.455596085106663	-0.345376933946682
.203516838690120	-0.043038790158789	0.118320309762566	0.256950323535420
.341097662072571	0.325677328830352	0.136695083463565	-0.413065737127909
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
55	46	3	
.000000000000000	0.000000000000000	0.221142731119471	-0.568118707029498
.054245552471002	0.362761535907339	0.273609149830008	-0.086861525278437
.379414907263708	-0.200754614044854	0.458844710017729	-0.128821849863016
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
56	46	4	
.000000000000000	0.000000000000000	0.000000000000000	0.059834424120709
.308507444977295	0.564184173228568	-0.345272770078736	-0.241082834490953
.524372886720812	-0.346020200253406	0.103848263346663	-0.011119028016992
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
57	48	1	
.640086714280187	0.169037625862984	0.169916908496816	0.171434820855211
.173679322058888	0.176795636278257	0.181018847768428	0.186739674649289
.194649458742788	0.206108019570820	0.224308867924265	0.259796664613532
.419779930052755	0.000000000000000	0.000000000000000	0.000000000000000
58	48	2	
.000000000000000	0.670625091195689	0.138184581253881	0.081828902390043
.008540098193407	-0.075851470515772	-0.163478269641928	-0.244027959193772
.304001053889806	-0.325193931101239	-0.280922823156288	-0.122413766806858
.374565580961644	0.000000000000000	0.000000000000000	0.000000000000000
59	48	3	
.000000000000000	0.000000000000000	0.567508046588046	-0.188203528847225
.306723347930529	-0.308145781142230	-0.166937885082007	0.073914950704420
.293379193346482	0.318600614067691	0.018245030473281	-0.464541567031119
.128218117817596	0.000000000000000	0.000000000000000	0.000000000000000

60 48	4		
.000000000000000	0.000000000000000	0.000000000000000	0.323961269168649
.565138651116285	-0.036874847232866	0.365119847891037	0.204460862398264
.294701769904350	-0.278997995999816	0.446314337954391	-0.186887316383945
.019455767310774	0.000000000000000	0.000000000000000	0.000000000000000
61 48	5		
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.054986221069157	-0.254197639420454	0.521361014645663	-0.620653109974228
.467839603385073	-0.227226905277348	0.068541935950634	-0.011321242151553
.000597151559987	0.000000000000000	0.000000000000000	0.000000000000000
62 50	1		
.508412448922172	-0.208532820484834	-0.209531802473066	-0.211251504861139
.213782904957043	-0.217274157861539	-0.221960315593606	-0.228220999444602
.236700747122086	-0.248591012259567	-0.266418319013610	-0.296990893191833
.371793796570273	0.000000000000000	0.000000000000000	0.000000000000000
63 50	2		
.000000000000000	0.383637807171908	-0.430102309801441	-0.342088024946861
.227131336803723	-0.093810063797719	0.046322954937750	0.178174775250227
.282107413077028	0.331965994584607	0.290528115045163	0.095641473444630
.416159691834764	0.000000000000000	0.000000000000000	0.000000000000000
64 50	3		
.000000000000000	0.000000000000000	0.246310210226511	-0.560334169698763
.043182450647691	0.285006686281122	0.329069748186157	0.109596877213335
.211867417518079	-0.363460556945938	-0.098188198455318	0.462323955180182
.146932800096693	0.000000000000000	0.000000000000000	0.000000000000000
65 50	4		
.000000000000000	0.000000000000000	0.000000000000000	0.087942891232160
.391057023633039	0.528806160669774	-0.034404201187499	-0.421255219873274
.085823416968196	0.425682476161455	-0.412225001084153	0.149129597957653
.018015807313077	0.000000000000000	0.000000000000000	0.000000000000000
66 52	1		
.637764865081408	0.163021785298042	0.163743400506064	0.164982542857615
.166799320471910	0.169290444508079	0.172606688949600	0.176985643141961
.182816933782306	0.190785312322987	0.202233068695932	0.220307321953442
.255391215061536	0.413002674635209	0.000000000000000	0.000000000000000
67 52	2		
.000000000000000	0.668728259275667	0.141059206995124	0.094217111431135
.032556480912717	-0.039875560776636	-0.117665063309687	-0.193814616148302
.259377626956848	-0.302792970025621	-0.308514980526525	-0.253558190044714
.094820200662070	0.377550444907683	0.000000000000000	0.000000000000000
68 52	3		
.000000000000000	0.000000000000000	0.580080475207069	-0.141115597078169
.262111705538700	-0.303127364677493	-0.231673074289574	-0.056071903276461
.162513263614528	0.313136963755824	0.265637064352632	-0.049449214374363
.457330618402921	0.139867527432216	0.000000000000000	0.000000000000000
69 52	4		
.000000000000000	0.000000000000000	0.000000000000000	0.368129362432228
.523891126416856	-0.173001782019128	0.234767943038605	0.329233561162255
.004433562763325	-0.351719525395310	-0.145032220825669	0.449210960396158
.222203930359683	0.025550322637628	0.000000000000000	0.000000000000000
70 52	5		
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.103026306929678	-0.391539595912036	0.556937014349387	-0.243252729920641
.286098019656492	0.499806837346114	-0.343788275298721	0.127625684108352
.024329199053227	0.001428778088462	0.000000000000000	0.000000000000000
71 54	1		
.511157608489157	-0.199612564465497	-0.200431501314597	-0.201834659877932
.203884695307352	-0.206681348921848	-0.210377754341607	-0.215210131249478
.221554576426570	-0.230044923777314	-0.241848355188267	-0.259435125386517
.289455424441257	-0.362648047586845	0.000000000000000	0.000000000000000

72 54 2			
.000000000000000	0.393951963902509	-0.407634839612001	-0.336073133010359
.241567880264483	-0.129981489542843	-0.009146059756858	0.110837160684241
.217100140831462	0.293125411655637	0.317004745753366	0.257379776703601
.060914211152522	-0.417815634464205	0.000000000000000	0.000000000000000
73 54 3			
.000000000000000	0.000000000000000	0.268077722859744	-0.546976987097878
.114784521387264	0.204334551192005	0.324390680674696	0.224017131797154
.029694962039230	-0.280021799371752	-0.321125507194665	-0.012672904994815
.457647347546299	-0.163070961087215	0.000000000000000	0.000000000000000
74 54 4			
.000000000000000	0.000000000000000	0.000000000000000	0.114977819932392
.448037127171119	0.445566338494245	0.176375693983573	-0.321584040975722
.250578778883209	0.270660439186031	0.290672520255612	-0.442186554661021
.190434961128631	-0.025633091820055	0.000000000000000	0.000000000000000
75 54 5			
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.014192605694006	-0.100274297050419	0.306630991006386	-0.537943542571156
.600011327944230	-0.442203144426458	0.215097625408816	-0.066190152934723
.011482485559100	-0.000792939553116	0.000000000000000	0.000000000000000
76 56 1			
.635686980240280	0.157627934138873	0.158229002736334	0.159256840270245
.160753846503212	0.162786883065564	0.165457225637510	0.168918219456828
.173407878748549	0.179313403001288	0.187312377158378	0.198730079166126
.216670195390179	0.251366558652080	0.406781553050678	0.000000000000000
77 56 2			
.000000000000000	0.666861088451697	0.142563123856235	0.103148313177371
.050780323618710	-0.011649695987899	-0.080299034822019	-0.150250394435716
.215315273817396	-0.267703772755340	-0.297421868905076	-0.291019581367436
.228394248730183	-0.070874503782761	0.379775890658493	0.000000000000000
78 56 3			
.000000000000000	0.000000000000000	0.589611342322314	-0.103447458637675
.220480068674078	-0.283533193243784	-0.260685189990909	-0.144433682464296
.038001094935680	0.221119670302869	0.310004726095940	0.210318489502242
.104574666458575	-0.446781251426728	0.150374740202096	0.000000000000000
79 56 4			
.000000000000000	0.000000000000000	0.000000000000000	0.404179030636442
.475200920416682	-0.254740500720800	0.097745662587840	0.316351979796309
.206923919573455	-0.144590113010054	-0.342797549077659	-0.027917491016069
.435151173842314	-0.252813851504470	0.031829878782446	0.000000000000000
80 56 5			
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.150277546075813	-0.477077763542184	0.454791934826062	0.099526197482191
.404260664021422	0.030595829151935	0.407632896058203	-0.408666440569505
.184210043222265	-0.039958243189176	0.002582757224332	0.000000000000000
81 58 1			
.513603748258902	-0.191716038167215	-0.192397299088043	-0.193560214036552
.195249243221042	-0.197533912163635	-0.200518277368246	-0.204357337730471
.209286414235976	-0.215676953153888	-0.224152014081483	-0.235856037263147
.253207916182873	-0.282715041576012	-0.354444450160372	0.000000000000000
82 58 2			
.000000000000000	0.402938637060288	-0.387727907342196	-0.328640592233569
.249940425630484	-0.155758781740587	-0.051581761345005	0.055580507852028
.156926260591717	0.241420205849863	0.295008274319116	0.299058021817473
.226687468210002	0.031302578459872	-0.418425371230114	0.000000000000000
83 58 3			
.000000000000000	0.000000000000000	0.287042314899736	-0.530848299851641
.166994631854323	0.130733190864703	0.290194909575803	0.275817400346568
.107001556661406	-0.133244278571510	-0.307979861805507	-0.267197630259034
.057282278990897	0.447577310304041	-0.177436637321080	0.000000000000000

84 58 4			
.000000000000000	0.000000000000000	0.000000000000000	0.140368614127510
.485606984831975	0.348487346993352	0.292313166508393	-0.148180943768239
.345222130061302	-0.047942447595233	0.344064986318699	0.155535148125637
.446059920934204	0.226894389597874	-0.033694011705848	0.000000000000000
85 58 5			
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.029382619735316	-0.181076451754052	0.444430842180736	-0.517273781631957
.165981686284795	0.320995296417095	-0.493444455089048	0.333063810401654
.125009602870705	0.024883352907636	-0.001904464682366	0.000000000000000
86 60 1			
.633812892032438	0.152754209015256	0.153261279785511	0.154125480778021
.155377525676158	0.157065080064058	0.159258732466697	0.162062113697175
.165629541832459	0.170198346819234	0.176152641455701	0.184162708210351
.195537529685680	0.213340119706118	0.247665693790623	0.401037519091915
87 60 2			
.000000000000000	0.665050560084637	0.143134658192139	0.109608040476814
.064737540007627	0.010639446267987	-0.049887268085513	-0.113287318126822
.175136185811425	-0.229961201301194	-0.270941065842635	-0.289354478081805
.273424799670465	-0.205335575748904	-0.049946891324907	0.381406570733816
88 60 3			
.000000000000000	0.000000000000000	0.596938136746758	-0.073104001961838
.183222296996973	-0.257491103024222	-0.267245352537429	-0.198497540906828
.059915486162936	0.112324846448246	0.255651865591062	0.292219360300805
.156402474443894	-0.149122516217176	-0.434066635076306	0.159866710895585
89 60 4			
.000000000000000	0.000000000000000	0.000000000000000	0.433762486187334
.425645239978621	-0.298371839220636	-0.017408555254406	0.240331471977697
.287243529506668	0.069175161861494	-0.237597224822821	-0.296506656253769
.068564540903026	0.409849630937970	-0.278908890984894	0.038177665627700
90 60 5			
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.194670738143031	-0.523048802351695	0.309733873538789	0.289298537252762
.226463096754742	-0.294669497365248	0.230473603136488	0.269684022212459
.431799985424732	0.234886886923050	-0.057375068217880	0.004044998677955
91 60 6			
.000000000000000	0.000000000000000	0.000000000000000	0.000000000000000
.000000000000000	0.024426224201306	-0.137655850940064	0.355095966246154
.552463418556727	0.573335286026122	-0.413620078274107	0.209227338080382
.072794191848516	0.016456542011383	-0.002105677907298	0.000088240581408

APPENDIX C

Programs

The programs in this appendix are listed in the same order in which they appear in the body of this report.

AREA INVERSION PROGRAMS

```
C ***********************************************************************
C *                                                                     *
C *                          J111                                       *
C *                                                                     *
C *                                                                     *
C *    NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                    *
C *                                                                     *
C *    INVERSION OF AREA DATA TO GIVE RADIUS**2                         *
C *                                                                     *
C *    SYMMETRY GROUP:  OH.                                             *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      EXTERNAL KUPOLY
 1    FORMAT (15X,I5/4F20.15/4F20.15/4F20.15/4F20.15)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT (15X,I5,F10.6)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $     7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $     4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $     I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (1H1,9X,'#',5X,'J',8X,'AREA',9X,'RADIUS**2'/21X,
     $     ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (///10X,' RADIUS**2 IS NEGATIVE.'/)
 13   FORMAT (1H1)
 14   FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,2D17.8)
 19   FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'AREA',10X,'RADIUS')
 22   FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
```

```
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
 30   FORMAT (20A4)
 31   FORMAT (1H1,////40X,' SYMMETRY GROUP OH'//
     $      ' ',10X,20A4)
42    FORMAT(I5,2F10.5,2D17.8,21X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
C
C     READ DATA
C
      CALL COMSET
      READ (5,30)  (TITLE(I),I=1,20)
      WRITE (6,31) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,1) (NAM(I),(SYM(I,J),J=1,16),I=1,91)
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,1)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,1)
      KE=NINIT
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10)  II, BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (5,3) NUM
      WRITE (6,5)
      DO 16 I = 1,NUM
      READ (5,2) ZZ(I)
      READ (5,2) T1(I),P1(I)
      READ (5,2) T2(I),P2(I),TL(I)
      READ (5,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $      NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 16 J = 1,NTI
 16   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE(6,22)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
```

```
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
C
C     HARMONIC DO LOOP
C
      IF (LMIN .LT. 4) LMIN = 4
      IF (LMIN .GT. 60) LMIN = 60
      IF (LMAX .LT. LMIN) LMAX = LMIN
      IF (LMAX .GT. 60) LMAX = 60
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,1)
      KE = NC
      N = KE
      MID=LTOM(LMID,1)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 17 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 17 J = 1,91
 17   A(I,J) = 0.
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,KUPOLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,8)
      DO 18 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 18   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
```

```
      NTTT=NTOT(I) +1
      WRITE (6,19)
      DO 200   J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      T=AXT
      P=AXP
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      CALL KUPOLY(T,P,BE)
      AREA = 0.
      ARAD2 = 0.
      DO 20 K = 1,KE
      AREA = AREA + BE(K)*C(K)
 20   ARAD2 = ARAD2 + BE(K)*D(K)
      IF (ARAD2 .GE. 0.) GO TO 21
      WRITE (6,11)
      ARAD2 = DABS(ARAD2)
 21   ARAD = DSQRT(ARAD2)
      JO=(J-1)*NSPACE(I)+NSTART(I)
      WRITE(6,14) JO,AXT,AXP,AREA,ARAD
      WRITE(10,42) JO,AXT,AXP,AREA,ARAD
200   CONTINUE
291   CONTINUE
139   CONTINUE
 500  CONTINUE
      STOP
      END


C ***********************************************************************
C *                                                                     *
C *                          J112                                       *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *                                                                     *
C *   INVERSION OF AREA DATA TO GIVE RADIUS**2                          *
C *                                                                     *
C *   SYMMETRY GROUP: D6H                                               *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
```

```
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL HXPOLY
1     FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
2     FORMAT (F10.4)
3     FORMAT (I5)
4     FORMAT(15X,I5,F10.6,14X,I1)
5     FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $      7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $      4X,'NLOW',5X,'NZERO')
6     FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
7     FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $      I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
8     FORMAT (///9X,' #',5X,'J',8X,'AREA',11X,'RADIUS**2'/21X,
     $      ' COEFFICIENTS',5X,'COEFFICIENTS'/)
9     FORMAT (5X,2I6,2E17.6)
10    FORMAT (10X,I4,E17.6)
11    FORMAT (///10X,' RADIUS**2 IS NEGATIVE.'/)
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
13    FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
16    FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
17    FORMAT (1H1,////40X,' SYMMETRY GROUP D6H'//
     $      ' ',10X,20A4)
18    FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'AREA',12X,'RADIUS',
     &12X,'AREA',12X,'RADIUS')
23    FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
```

```
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      ALPHA=1.0D0
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,2)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,2)
      KE=NINIT
      T=0.0
      P=0.0
      CALL HXPOLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10) II,BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      NUT=0
      E1 = 0.0
      E2=0.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $      NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
 20   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
```

```
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,2)
      KE = NC
      N = KE
      MID=LTOM(LMID,2)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,HXPOLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
```

```
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (    6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(ALPHA2*SP2+CP2)
      AB = ALPHA*CT
      CALL DFLOP(AA,AB,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL HXPOLY(TT,PP,BE)
      CALL HXPOLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
```

```
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      AR1 = DSQRT(AR1)
      AR2 = DSQRT(AR2)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL HXPOLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J113                                       *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *                                                                     *
C *   INVERSION OF AREA DATA TO GIVE RADIUS**2                          *
C *                                                                     *
C *   SYMMETRY GROUP:  C6H                                              *
```

```
C *                                                                    *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL C6POLY
 1    FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT(15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $     7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $     4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $     I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'AREA',11X,'RADIUS**2'/21X,
     $     ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (///10X,' RADIUS**2 IS NEGATIVE.'/)
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP C6H'//
     $     ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'AREA',12X,'RADIUS',
     &12X,'AREA',12X,'RADIUS')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
```

```
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      ALPHA=1.0D0
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,3)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,3)
      KF=NINIT
      T=0.0
      P=0.0
      CALL C6POLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10) II,BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
25    CONTINUE
      NUT=0
      E1 = 0.0
      E2=0.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $      NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
20    R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
```

```
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,3)
      KE = NC
      N = KE
      MID=LTOM(LMID,3)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,C6POLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
```

```
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (   6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(ALPHA2*SP2+CP2)
      AB = ALPHA*CT
      CALL DFLOP(AA,AB,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL C6POLY(TT,PP,BE)
      CALL C6POLY(TTT,PPP,B)
      A1 = 0.0
```

```
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      AR1 = DSQRT(AR1)
      AR2 = DSQRT(AR2)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL C6POLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END
```

```
C ***********************************************************************
C *                                                                     *
C *                          J114                                       *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *                                                                     *
C *   INVERSION OF AREA DATA TO GIVE RADIUS**2                          *
C *                                                                     *
C *   SYMMETRY GROUP:  D3D                                              *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3),  AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL D3POLY
 1    FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT(15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $      7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $      4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $      I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'AREA',11X,'RADIUS**2'/21X,
     $      ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2F17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (///10X,' RADIUS**2 IS NEGATIVE.'/)
 12   FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
```

```
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP D3D'//
     $      ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'AREA',12X,'RADIUS',
     &12X,'AREA',12X,'RADIUS')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      ALPHA=1.0D0
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,4)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,4)
      KE=NINIT
      T=0.0
      P=0.0
      CALL D3POLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10) II,BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      NUT=0
      E1 = 0.0
      E2=0.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
```

```
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $      NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
20    R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,4)
      KE = NC
      N = KE
      MID=LTOM(LMID,4)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,D3POLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
```

```
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (    6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(ALPHA2*SP2+CP2)
      AB = ALPHA*CT
      CALL DFLOP(AA,AB,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
```

```
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL D3POLY(TT,PP,BE)
      CALL D3POLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      AR1 = DSQRT(AR1)
      AR2 = DSQRT(AR2)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL D3POLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
```

```
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = C
 500  CONTINUE
      STOP
      END


C ***********************************************************************
C *                                                                     *
C *                          J115                                       *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *                                                                     *
C *   INVERSION OF AREA DATA TO GIVE RADIUS**2                          *
C *                                                                     *
C *   SYMMETRY GROUP: S6                                                *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL S6POLY
1     FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT(15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $      7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
```

```
     $      4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $      I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'AREA',11X,'RADIUS**2'/21X,
     $      ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (///10X,' RADIUS**2 IS NEGATIVE.'/)
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP S6(C3I)'//
     $      ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'AREA',12X,'RADIUS',
     &12X,'AREA',12X,'RADIUS')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      ALPHA=1.0D0
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,5)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,5)
      KE=NINIT
      T=0.0
      P=0.0
      CALL S6POLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10) II,BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      NUT=0
      E1 = 0.0
```

```
      E2=0.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $     NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
 20   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,5)
      KE = NC
      N = KE
      MID=LTOM(LMID,5)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
```

```
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,S6POLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (    6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
 201  AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILFR(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
```

```
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(ALPHA2*SP2+CP2)
      AB = ALPHA*CT
      CALL DFLOP(AA,AB,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL S6POLY(TT,PP,BE)
      CALL S6POLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      AR1 = DSQRT(AR1)
      AR2 = DSQRT(AR2)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL S6POLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
```

```
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J116                                       *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *                                                                     *
C *   INVERSION OF AREA DATA TO GIVE RADIUS**2                          *
C *                                                                     *
C *   SYMMETRY GROUP: D4H                                               *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),TZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
```

```
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL SQPOLY
1     FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT(15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $     7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $     4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $     I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'AREA',11X,'RADIUS**2'/21X,
     $     ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (///10X,' RADIUS**2 IS NEGATIVE.'/)
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP D4H'//
     $     ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'AREA',12X,'RADIUS',
     &12X,'AREA',12X,'RADIUS')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      ALPHA=1.0D0
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,6)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,6)
      KE=NINIT
```

```
      T=0.0
      P=0.0
      CALL SQPOLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10) II,BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
25    CONTINUE
      NUT=0
      E1 = 0.0
      E2=0.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $      NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (9,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
20    R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
```

```
      NC = LTOM(LNC,6)
      KE = NC
      N = KE
      MID=LTOM(LMID,6)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,SQPOLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I)+1
      WRITE (   6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
```

```
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(ALPHA2*SP2+CP2)
      AB = ALPHA*CT
      CALL DFLOP(AA,AB,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL SQPOLY(TT,PP,BE)
      CALL SQPOLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
93    AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
94    AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
95    CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      AR1 = DSQRT(AR1)
      AR2 = DSQRT(AR2)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
```

```
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL SQPOLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J117                                       *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *                                                                     *
C *   INVERSION OF AREA DATA TO GIVE RADIUS**2                          *
C *                                                                     *
C *   SYMMETRY GROUP: C4H                                               *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
```

```
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC   ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL C4POLY
1     FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT(15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $      7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $      4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $      I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'AREA',11X,'RADIUS**2'/21X,
     $      ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (///10X,' RADIUS**2 IS NEGATIVE.'/)
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP C4H'//
     $      ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'AREA',12X,'RADIUS',
     &12X,'AREA',12X,'RADIUS')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
```

```
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      ALPHA=1.0D0
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,7)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,7)
      KE=NINIT
      T=0.0
      P=0.0
      CALL C4POLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10) II,BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      NUT=0
      E1 = 0.0
      E2=0.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $     NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
 20   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
```

```
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,7)
      KE = NC
      N = KE
      MID=LTOM(LMID,7)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,C4POLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (    6,27)
```

```
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(ALPHA2*SP2+CP2)
      AB = ALPHA*CT
      CALL DFLOP(AA,AB,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL C4POLY(TT,PP,BE)
      CALL C4POLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
```

```
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      AR1 = DSQRT(AR1)
      AR2 = DSQRT(AR2)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL C4POLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END


C ***********************************************************************
C *                                                                     *
C *                          J118                                       *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *                                                                     *
C *   INVERSION OF AREA DATA TO GIVE RADIUS**2                          *
C *                                                                     *
C *   SYMMETRY GROUP: VH                                                *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
```

```
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC   ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL VHPOLY
1     FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT(15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $     7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $     4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $     I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'AREA',11X,'RADIUS**2'/21X,
     $     ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (///10X,' RADIUS**2 IS NEGATIVE.'/)
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP VH'//
     $     ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'AREA',12X,'RADIUS',
     &12X,'AREA',12X,'RADIUS')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
```

```
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      READ (5,2) ALPHA
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,8)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,8)
      KE=NINIT
      T=0.0
      P=0.0
      CALL VHPOLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10) II,BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      NUT=0
      E1 = 0.0
      E2=0.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $      NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
 20   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
```

```
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,8)
      KE = NC
      N = KE
      MID=LTOM(LMID,8)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,VHPOLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
```

```
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (   6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL DILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(ALPHA2*SP2+CP2)
      AB = ALPHA*CT
      CALL DFLOP(AA,AB,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL VHPOLY(TT,PP,BE)
      CALL VHPOLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KF
      A1 = A1 + C(K)*BE(K)
```

```
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. C) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      AR1 = DSQRT(AR1)
      AR2 = DSQRT(AR2)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL VHPOLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J119                                       *
C *                                                                     *
```

```
C *                                                                    *
C *    NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                   *
C *                                                                    *
C *    INVERSION OF AREA DATA TO GIVE RADIUS**2                        *
C *                                                                    *
C *    SYMMETRY GROUP: C2H                                             *
C *                                                                    *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST-CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL C2POLY
 1    FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT (15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $      7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $      4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $      I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'AREA',11X,'RADIUS**2'/21X,
     $      ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (///10X,' RADIUS**2 IS NEGATIVE.'/)
 12   FORMAT(////4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
 14   FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP C2H'//
```

```
     $      ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'AREA',12X,'RADIUS',
     &12X,'AREA',12X,'RADIUS')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      READ (5,2) ALPHA,E2
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,9)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,9)
      KE=NINIT
      T=0.0
      P=0.0
      CALL C2POLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10) II,BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      NUT=0
      E1 = 0.0
      E3 = 0.0
      E2=E2*PI/180.0D0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $      NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.CDO
```

```
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
 20   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,9)
      KE = NC
      N = KE
      MID=LTOM(LMID,9)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,C2POLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
```

```
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (   6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(ALPHA2*SP2+CP2)
      AB = ALPHA*CT
      CALL DFLOP(AA,AB,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
```

```
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL C2POLY(TT,PP,BE)
      CALL C2POLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      AR1 = DSQRT(AR1)
      AR2 = DSQRT(AR2)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL C2POLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
```

```
291   CONTINUE
 139  NUT = C
 500  CONTINUE
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J1110                                      *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *                                                                     *
C *   INVERSION OF AREA DATA TO GIVE RADIUS**2                          *
C *                                                                     *
C *   SYMMETRY GROUP: S2                                                *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3),  AZ(3),  AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL S2POLY
 1    FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT(15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $     7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $     4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $     I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'AREA',11X,'RADIUS**2'/21X,
```

```
     $      ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (///10X,' RADIUS**2 IS NEGATIVE.'/)
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONF ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP S2'//
     $      ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'AREA',12X,'RADIUS',
     &12X,'AREA',12X,'RADIUS')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA,ALPHA,E1,E2,E3
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,10)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,10)
      KE=NINIT
      T=0.0
      P=0.0
      CALL S2POLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10) II,BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      NUT=0
      E1=E1*PI/180.0D0
      E2=E2*PI/180.0D0
      E3=E3*PI/180.0D0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
```

```
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $     NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
 20   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,10)
      KE = NC
      N = KE
      MID=LTOM(LMID,10)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,S2POLY,1)
C
```

```
C       INVERSION OF FITTING COEFFICIENTS.
C
        CALL COET(BE)
        WRITE (6,13)
        IF (LUMP .EQ. 0) WRITE (6,18)
        IF (GAMMA .NE. 1.0) WRITE (6,16)
        WRITE (6,8)
        DO 22 I = 1,KE
        D(I) = C(I)/(PI * BE(I))
        WRITE(6,9) I,NAM(I),C(I),D(I)
 22     WRITE (7,10) I,D(I)
C
C       EVALUATION OF FITTING RESULTS FOR
C           PRESELECTED ANGLES.
C
        IF (LUMP .EQ. 0) WRITE (6,18)
        WRITE (6,13)
        IF (NUMB.EQ.0)  GO TO 139
        DO 291  I=1,NUMB
        AT11=T11(I)
        AP11=P11(I)
        AT21=T21(I)
        AP21=P21(I)
        ATL=TL2(I)
        CALL DGOTTA(AX,AT11,AP11)
        CALL DGOTTA(AY,AT21,AP21)
        CALL DCRISS(AX,AY,AZ)
        CALL DCRISS(AZ,AX,AY)
        AYC=DCOS(ATL)
        AYS=DSIN(ATL)
        NTTT=NTOT(I) +1
        WRITE (    6,27)
        WRITE (6,19)
        DO 205 J=1,NTTT
        J11=(J-1)*NSPACE(I)
        J11=J11+NSTART(I)-NAXIS(I)
        APR=J11*0.1/CAL
        APR=APR*PI/180.0D0
        AZC=DCOS(APR)
        AZS=DSIN(APR)
        DO 201  K=1,3
        AS(K)=AX(K)*AZC+AY(K)*AZS
 201    AW(K)=AS(K)*AYS+AZ(K)*AYC
        AXT=DARCOS(AW(3))
        AXP=DATAN2(AW(2),AW(1))
        XT=AXT
        XP=AXP
        IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
        T=XT
        P=XP
        IF (LUMP .EQ. 0) GO TO 91
        IF (GAMMA .EQ. 1.0) GO TO 91
        ST = DSIN(XT)
        CT = DCOS(XT)
        SP = DSIN(XP)
        CP = DCOS(XP)
        ST2 = ST * ST
        CT2 = CT * CT
        SP2 = SP * SP
        CP2 = CP * CP
```

```
      AA = GAMMA*ST*DSQRT(ALPHA2*SP2+CP2)
      AB = ALPHA*CT
      CALL DFLOP(AA,AB,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL S2POLY(TT,PP,BE)
      CALL S2POLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      AR1 = DSQRT(AR1)
      AR2 = DSQRT(AR2)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL S2POLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
```

```
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END
```

MASS INVERSION PROGRAMS

```
C ***********************************************************************
C *                                                                     *
C *                          J121                                       *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *                                                                     *
C *   INVERSION OF MASS DATA TO GIVE (1/PI)2K(DK/DE).                   *
C *                                                                     *
C *   SYMMETRY GROUP:  OH.                                              *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      EXTERNAL KUPOLY
 1    FORMAT (15X,I5/4F20.15/4F20.15/4F20.15/4F20.15)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT (15X,I5,F10.6)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $     7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $     4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $     I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (1H1,9X,'#',5X,'J',8X,'MASS',9X,'(1/PI)2K(DK/DE)'/21X,
     $     ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (///10X,' K(DK/DE) IS NEGATIVE.'/)
 13   FORMAT (1H1)
 14   FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,2D17.8)
 19   FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'MASS',10X,'2K(DK/DE)'
     &)
```

```
22    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
 30   FORMAT (20A4)
 31   FORMAT (1H1,////40X,' SYMMETRY GROUP OH'//
     $      ' ',10X,20A4)
42    FORMAT(I5,2F10.5,2D17.8,21X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
C
C     READ DATA
C
      CALL COMSET
      READ (5,30)  (TITLE(I),I=1,20)
      WRITE (6,31) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,1) (NAM(I),(SYM(I,J),J=1,16),I=1,91)
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,1)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,1)
      KE=NINIT
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10)  II, BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (5,3) NUM
      WRITE (6,5)
      DO 16 I = 1,NUM
      READ (5,2) ZZ(I)
      READ (5,2) T1(I),P1(I)
      READ (5,2) T2(I),P2(I),TL(I)
      READ (5,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $      NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 16 J = 1,NTI
 16   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE(6,22)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
```

```
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
C
C     HARMONIC DO LOOP
C
      IF (LMIN .LT. 4) LMIN = 4
      IF (LMIN .GT. 60) LMIN = 60
      IF (LMAX .LT. LMIN) LMAX = LMIN
      IF (LMAX .GT. 60) LMAX = 60
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,1)
      KE = NC
      N = KE
      MID=LTOM(LMID,1)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 17 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 17 J = 1,91
 17   A(I,J) = 0.
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,KUPOLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,8)
      DO 18 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 18   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
```

```
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (6,19)
      DO 200   J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      T=AXT
      P=AXP
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      CALL KUPOLY(T,P,BE)
      AREA = 0.
      ARAD2 = 0.
      DO 20 K = 1,KE
      AREA = AREA + BE(K)*C(K)
 20   ARAD2 = ARAD2 + BE(K)*D(K)
      IF (ARAD2 .GE. 0.) GO TO 21
      WRITE (6,11)
      ARAD2 = DABS(ARAD2)
 21   ARAD = DSQRT(ARAD2)
      JO=(J-1)*NSPACE(I)+NSTART(I)
      WRITE(6,14)   JO,AXT,AXP,AREA,ARAD
      WRITE(10,42)  JO,AXT,AXP,AREA,ARAD
200   CONTINUE
291   CONTINUE
139   CONTINUE
 500  CONTINUE
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J122                                       *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *
C *   INVERSION OF MASS DATA TO GIVE (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  D6H                                              *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
```

```
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC   ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL HXPOLY
1     FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT (15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $     7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $     4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $     I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'MASS',11X,'K(DK/DE)'/21X,
     $     ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (10X,' K(DK/DE) IS NEGATIVE.')
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     & 'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP D6H'//
     $     ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'MASS',10X,'2K(DK/DE)'
     &,11X,'MASS',10X,'2K(DK/DE)')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
```

```
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      ALPHA = 1.0
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,2)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,2)
      KE=NINIT
      T=0.0
      P=0.0
      CALL HXPOLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10)  II, BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      NUT=0
      E1 = 0.0
      E2=0.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $      NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
 20   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     & , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
```

```
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,2)
      KE = NC
      N = KE
      MID=LTOM(LMID,2)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,HXPOLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
```

```
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (   6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(SP2 + CP2/ALPHA2)
      CALL DFLOP(AA,CT,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL HXPOLY(TT,PP,BE)
      CALL HXPOLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
```

```
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL HXPOLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J123                                       *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *
C *   INVERSION OF MASS DATA TO GIVE (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  C6H                                              *
C *                                                                     *
C ***********************************************************************
```

```
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL C6POLY
 1    FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT (15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $      7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $      4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $      I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'MASS',11X,'K(DK/DE)'/21X,
     $      ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (10X,' K(DK/DE) IS NEGATIVE.')
 12   FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
 14   FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP C6H'//
     $      ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
 19   FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'MASS',10X,'2K(DK/DE)'
     &,11X,'MASS',10X,'2K(DK/DE)')
 23   FORMAT (20A4)
 24   FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
 27   FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
```

```
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      ALPHA = 1.0
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,3)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,3)
      KE=NINIT
      T=0.0
      P=0.0
      CALL C6POLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10)  II, BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      NUT=0
      E1 = 0.0
      E2=C.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $     NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
 20   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
```

```
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS1I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,3)
      KE = NC
      N = KE
      MID=LTOM(LMID,3)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,C6POLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
```

```
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (   6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(SP2 + CP2/ALPHA2)
      CALL DFLOP(AA,CT,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL C6POLY(TT,PP,BE)
      CALL C6POLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
```

```
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL C6POLY(T,P,B)
      AREA = C.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J124                                       *
C *                                                                     *
```

```
C *                                                                    *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                    *
C *
C *   INVERSION OF MASS DATA TO GIVE (1/PI)2K(DK/DE)                   *
C *                                                                    *
C *   SYMMETRY GROUP:  D3D                                             *
C *                                                                    *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC   ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL D3POLY
 1    FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT (15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $     7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $     4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $     I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'MASS',11X,'K(DK/DE)'/21X,
     $     ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (10X,' K(DK/DE) IS NEGATIVE.')
 12   FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
 14   FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP D3D'//
```

```
     $      ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'MASS',10X,'2K(DK/DE)'
     &,11X,'MASS',10X,'2K(DK/DE)')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      ALPHA=1.0D0
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,4)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,4)
      KE=NINIT
      T=0.0
      P=0.0
      CALL D3POLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10)  II, BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      NUT=0
      E1 = 0.0
      E2=0.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $      NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
```

```
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
 20   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,4)
      KE = NC
      N = KE
      MID=LTOM(LMID,4)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,D3POLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
```

```
 22    WRITE (7,10) I,D(I)
C
C      EVALUATION OF FITTING RESULTS FOR
C           PRESELECTED ANGLES.
C
       IF (LUMP .EQ. 0) WRITE (6,18)
       WRITE (6,13)
       IF (NUMB.EQ.0)  GO TO 139
       DO 291  I=1,NUMB
       AT11=T11(I)
       AP11=P11(I)
       AT21=T21(I)
       AP21=P21(I)
       ATL=TL2(I)
       CALL DGOTTA(AX,AT11,AP11)
       CALL DGOTTA(AY,AT21,AP21)
       CALL DCRISS(AX,AY,AZ)
       CALL DCRISS(AZ,AX,AY)
       AYC=DCOS(ATL)
       AYS=DSIN(ATL)
       NTTT=NTOT(I) +1
       WRITE (   6,27)
       WRITE (6,19)
       DO 205 J=1,NTTT
       J11=(J-1)*NSPACE(I)
       J11=J11+NSTART(I)-NAXIS(I)
       APR=J11*0.1/CAL
       APR=APR*PI/180.0D0
       AZC=DCOS(APR)
       AZS=DSIN(APR)
       DO 201  K=1,3
       AS(K)=AX(K)*AZC+AY(K)*AZS
 201   AW(K)=AS(K)*AYS+AZ(K)*AYC
       AXT=DARCOS(AW(3))
       AXP=DATAN2(AW(2),AW(1))
       XT=AXT
       XP=AXP
       IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
       T=XT
       P=XP
       IF (LUMP .EQ. 0) GO TO 91
       IF (GAMMA .EQ. 1.0) GO TO 91
       ST = DSIN(XT)
       CT = DCOS(XT)
       SP = DSIN(XP)
       CP = DCOS(XP)
       ST2 = ST * ST
       CT2 = CT * CT
       SP2 = SP * SP
       CP2 = CP * CP
       AA = GAMMA*ST*DSQRT(SP2 + CP2/ALPHA2)
       CALL DFLOP(AA,CT,XTT)
       AA = ALPHA*SP
       CALL DFLOP(AA,CP,XPP)
       TT = XTT
       PP = XPP
       AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
       CALL DFLOP(AA,CT,XTTT)
       AA=SP/ALPHA
       CALL DFLOP(AA,CP,XPPP)
```

```
      TTT = XTTT
      PPP = XPPP
      CALL D3POLY(TT,PP,BE)
      CALL D3POLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL D3POLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = 0
 500  CONTINUE
```

```
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J125                                       *
C *                                                                     *
C *                                                                     *
C *    NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                    *
C *
C *    INVERSION OF MASS DATA TO GIVE (1/PI)2K(DK/DE)                   *
C *                                                                     *
C *    SYMMETRY GROUP:  S6                                              *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL S6POLY
 1    FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT (15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $      7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $      4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $      I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'MASS',11X,'K(DK/DE)'/21X,
     $      ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
```

```
 11   FORMAT (10X,' K(DK/DE) IS NEGATIVE.')
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP S6'//
     $      ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'MASS',10X,'2K(DK/DE)'
     &,11X,'MASS',10X,'2K(DK/DE)')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      ALPHA=1.0D0
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,5)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,5)
      KE=NINIT
      T=0.0
      P=0.0
      CALL S6POLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10)  II, BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      NUT=0
      E1 = 0.0
      E2=0.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
```

```
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $      NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
 20   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,5)
      KE = NC
      N = KE
      MID=LTOM(LMID,5)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,S6POLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
```

```
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (    6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(SP2 + CP2/ALPHA2)
      CALL DFLOP(AA,CT,XTT)
```

```
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL S6POLY(TT,PP,BE)
      CALL S6POLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL S6POLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
```

```
 103  CONTINUE
      ARAC = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J126                                       *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *
C *   INVERSION OF MASS DATA TO GIVE (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  D4H                                              *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL SQPOLY
 1    FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT(15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $     7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
```

```
     $      4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $      I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'MASS',11X,'K(DK/DE)'/21X,
     $      ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (10X,' K(DK/DE) IS NEGATIVE.')
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     & 'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP D4H'//
     $      ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'MASS',10X,'2K(DK/DE)'
     &,11X,'MASS',10X,'2K(DK/DE)')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      ALPHA=1.0D0
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,6)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,6)
      KE=NINIT
      T=0.0
      P=0.0
      CALL SQPOLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10)  II, BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
  25  CONTINUE
      NUT=0
      E1 = 0.0
```

```
      E2=C.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $     NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
 20   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,6)
      KE = NC
      N = KE
      MID=LTOM(LMID,6)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
```

```
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,SQPOLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (   6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
```

```
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(SP2 + CP2/ALPHA2)
      CALL DFLOP(AA,CT,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL SQPOLY(TT,PP,BE)
      CALL SQPOLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL SQPOLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
```

```
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J127                                       *
C *                                                                     *
C *                                                                     *
C *    NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                    *
C *                                                                     *
C *    INVERSION OF MASS DATA TO GIVE (1/PI)2K(DK/DE)                   *
C *                                                                     *
C *    SYMMETRY GROUP:  C4H                                             *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
```

```
      COMMON/ROTATE/NCODE
      EXTERNAL C4POLY
1     FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT(15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $     7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $     4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $     I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'MASS',11X,'K(DK/DE)'/21X,
     $     ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (10X,' K(DK/DE) IS NEGATIVE.')
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP C4H'//
     $     ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'MASS',10X,'2K(DK/DE)'
     &,11X,'MASS',10X,'2K(DK/DE)')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA
      ALPHA=1.0D0
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,7)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,7)
      KE=NINIT
      T=0.0
      P=0.0
      CALL C4POLY(T,P,B)
```

```
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10)  II, BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
25    CONTINUE
      NUT=0
      E1 = 0.0
      E2=0.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $     NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
20    R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,7)
      KE = NC
      N = KE
```

```
      MID=LTOM(LMID,7)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,C4POLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTCT(I) +1
      WRITE (   6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCCS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
```

```
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(SP2 + CP2/ALPHA2)
      CALL DFLOP(AA,CT,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL C4POLY(TT,PP,BE)
      CALL C4POLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*18C.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL C4POLY(T,P,B)
      AREA = 0.0
```

```
      ARAC = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
 102  WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
 291  CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END


C ***********************************************************************
C *                                                                     *
C *                          J128                                       *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *
C *   INVERSION OF MASS DATA TO GIVE (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  VH                                               *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
```

```
      COMMON/LIMIT/MID,NIT,BC   ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL VHPOLY
1     FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT (15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $     7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $     4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $     I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'MASS',11X,'K(DK/DE)'/21X,
     $     ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (10X,' K(DK/DE) IS NEGATIVE.')
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP VH (C3I)'//
     $     ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'MASS',10X,'2K(DK/DE)'
     &,11X,'MASS',10X,'2K(DK/DE)')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA,ALPHA
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,8)
      WRITE (6,43) LMID,MID
```

```
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLA=MTOL(NINIT,8)
      KE=NINIT
      T=0.0
      P=0.0
      CALL VHPOLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10)  II, BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      NUT=0
      E1 = 0.0
      E2=0.0D0
      E3 = 0.0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $     NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
 20   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
```

```
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,8)
      KE = NC
      N = KE
      MID=LTOM(LMID,8)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,VHPOLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (   6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
```

```
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(SP2 + CP2/ALPHA2)
      CALL DFLOP(AA,CT,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL VHPOLY(TT,PP,BE)
      CALL VHPOLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
```

```
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL VHPOLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
 102  WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
 291  CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J129                                       *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *                                                                     *
C *   INVERSION OF MASS DATA TO GIVE (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  C2H                                              *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
```

```
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL C2POLY
1     FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT (15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $     7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $     4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $     I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'MASS',11X,'K(DK/DE)'/21X,
     $     ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (10X,' K(DK/DE) IS NEGATIVE.')
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP C2H'//
     $     ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'MASS',10X,'2K(DK/DE)'
     &,11X,'MASS',10X,'2K(DK/DE)')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
```

```
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA,ALPHA,E2
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,9)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,9)
      KE=NINIT
      T=0.0
      P=0.0
      CALL C2POLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10)  II, BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
25    CONTINUE
      NUT=0
      E1 = 0.0
      E3 = 0.0
      E2=E2*PI/180.0D0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $     NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
20    R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
```

```
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,9)
      KE = NC
      N = KE
      MID=LTOM(LMID,9)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ (NUM,KE,CAL,DELTA,C2POLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (   -6,27)
```

```
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(SP2 + CP2/ALPHA2)
      CALL DFLOP(AA,CT,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL C2POLY(TT,PP,BE)
      CALL C2POLY(TTT,PPP,B)
      A1 = 0.0
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
```

```
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL C2POLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
102   WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
291   CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J1210                                      *
C *                                                                     *
C *                                                                     *
C *   NON-SYMMETRY PLANE ELECTRON SURFACE INVERSION                     *
C *
C *   INVERSION OF MASS DATA TO GIVE (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP: S2                                                *
C *                                                                     *
C ***********************************************************************
C
C     INITIALIZATION
C
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI
      REAL*8 SP,CP,ST,CT,SP2,CP2,ST2,CT2
```

```
      REAL*8 XT,XP,XTT,XPP,XTTT,XPPP,TEMP
      REAL*4 ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16)
      DIMENSION A(91,91),B(91),BE(91),C(91),D(91)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),ZZ(10),TL(10)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION NCODE(600,10)
      DIMENSION TITLE(20)
      DIMENSION BC(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION NSTART(10),NAXIS(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/LIMIT/MID,NIT,BC  ,LNIT
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      COMMON/INTO/IN,R
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,ALPHA2,GAMMA,G2,E1,E2,E3,JJJJ
      COMMON/ROTATE/NCODE
      EXTERNAL S2POLY
1     FORMAT(1X,I4,7X,F10.5,4X,F10.5,38X,2D17.8)
 2    FORMAT (F10.4)
 3    FORMAT (I5)
 4    FORMAT (15X,I5,F10.6,14X,I1)
 5    FORMAT (1H1,'PLANE',7X,'ZZ',10X,'THETA(1)',7X,'PHI(1)',
     $      7X,'THETA(2)',7X,'PHI(2)',6X,'CONE ANGLE',6X,'# POINTS',
     $      4X,'NLOW',5X,'NZERO')
 6    FORMAT (/2X,I2,2X,6(4X,F10.6),7X,I5,6X,I5,5X,I5)
 7    FORMAT (1H1,5X,'THE FIT INCLUDES TERMS TO ORDER L =',
     $      I3//' THE TOTAL NUMBER OF HARMONICS IS',I3////)
 8    FORMAT (///9X,' #',5X,'J',8X,'MASS',11X,'K(DK/DE)'/21X,
     $      ' COEFFICIENTS',5X,'COEFFICIENTS'/)
 9    FORMAT (5X,2I6,2E17.6)
 10   FORMAT (10X,I4,E17.6)
 11   FORMAT (10X,' K(DK/DE) IS NEGATIVE.')
12    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING',5X,'START'
     &,6X,'AXIS')
 13   FORMAT (1H1)
14    FORMAT (1X,I4,7X,F10.5,4X,F10.5,4X,4D17.8)
 16   FORMAT (///15X,' ELLIPSOIDAL TRANSFORMATION USED.'/)
 17   FORMAT (1H1,////40X,' SYMMETRY GROUP S2'//
     $      ' ',10X,20A4)
 18   FORMAT(///15X,' 1/AREA**2 FIT.'/)
19    FORMAT(/1X,'ANGLE',8X,'THETA',10X,'PHI',15X,'MASS',10X,'2K(DK/DE)'
     &,11X,'MASS',10X,'2K(DK/DE)')
 23   FORMAT (20A4)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5,6X,I5,5X,I5)
27    FORMAT(//1X,'ROTOR',13X,'LAB SYSTEM',19X,'TRANSFORMED SYSTEM',20X,
     & 'LAB SYSTEM')
41    FORMAT(I5,2F10.5,2D17.8,21X)
42    FORMAT(2F10.5,D17.8,2F10.5,D17.8,6X)
43    FORMAT(//5X,'THE FIT INVOLVES TERMS DIVIDED AT L=',I3,
     &'  CORRESPONDING TO',I4,' HARMONICS')
44    FORMAT(//5X,'THE NUMBER OF ITERATIONS IS',I2,
     &' STARTING FROM THE ITERATION NUMBER',I2)
```

```
      PI = 3.141592653589793
      NUT = 0
C
C     READ DATA
C
      CALL COMSET
      READ (5,23) (TITLE(I),I=1,20)
      WRITE (6,17) (TITLE(I),I=1,20)
      READ (5,3) LMIN,LMAX
      READ (5,3) LUMP
      READ (5,2) GAMMA,ALPHA,E1,E2,E3
      READ (5,3) LMID,LNIT,NIT
      MID=LTOM(LMID,10)
      WRITE (6,43) LMID,MID
      WRITE(6,44) NIT,LNIT
      READ(5,3) NINIT
      KLN=MTOL(NINIT,10)
      KE=NINIT
      T=0.0
      P=0.0
      CALL S2POLY(T,P,B)
      CALL COET(BE)
      DO 25 I = 1,NINIT
      READ (5,10)  II, BC(I)
      BC(I) = BC(I)*BE(I)*PI
      WRITE (6,10)  I,BC(I)
 25   CONTINUE
      NUT=0
      E1=E1*PI/180.0D0
      E2=E2*PI/180.0D0
      E3=E3*PI/180.0D0
      READ (5,2) CAL
      CAL = 1.0/CAL
      READ (9,3) NUM
      WRITE (6,5)
      DO 20 I = 1,NUM
      READ (9,2) ZZ(I)
      READ (9,2) T1(I),P1(I)
      READ (9,2) T2(I),P2(I),TL(I)
      READ (9,3) NT(I),NL(I),NZ(I)
      WRITE (6,6) I,ZZ(I),T1(I),P1(I),T2(I),P2(I),TL(I),
     $     NT(I),NL(I),NZ(I)
      NTI = NT(I)
      READ (5,4) (IN(J,I),R(J,I),NCODE(J,I),J=1,NTI)
      T1(I) = T1(I) * PI/180.0D0
      T2(I) = T2(I) * PI/180.0D0
      P1(I) = P1(I) * PI/180.0D0
      P2(I) = P2(I) * PI/180.0D0
      DO 20 J = 1,NTI
 20   R(J,I) = R(J,I) - ZZ(I)
      READ (5,3) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE (6,12)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I),NSTART(I),NAXIS(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I),NSTART(I),NAXIS(I)
      T11(I)=T11(I)*PI/180.0D0
```

```
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
      G2 = GAMMA * GAMMA
      ALPHA2 = ALPHA * ALPHA
      JJJJ = 0
      IF (E1 .NE. 0.0) JJJJ = 1
      IF (E2 .NE. 0.0) JJJJ = 1
      IF (E3 .NE. 0.0) JJJJ = 1
C
C     HARMONIC DO LOOP
C
      IF (LMAX .LT. LMIN) LMAX = LMIN
      DO 500 LNC = LMIN,LMAX,2
      DELTA=LNC-LMIN
      KLN = LNC
      NC = LTOM(LNC,10)
      KE = NC
      N = KE
      MID=LTOM(LMID,10)
      IF (MID.GE.KE)  MID=KE
      WRITE (6,7) KLN,KE
      DO 21 I = 1,91
      B(I) = 0.
      C(I) = 0.
      D(I) = 0.
      DO 21 J = 1,91
 21   A(I,J) = 0.0
C
C     LEAST SQUARE FIT OF DATA.
C
      CALL KLSQ(NUM,KE,CAL,DELTA,S2POLY,1)
C
C     INVERSION OF FITTING COEFFICIENTS.
C
      CALL COET(BE)
      WRITE (6,13)
      IF (LUMP .EQ. 0) WRITE (6,18)
      IF (GAMMA .NE. 1.0) WRITE (6,16)
      WRITE (6,8)
      DO 22 I = 1,KE
      D(I) = C(I)/(PI * BE(I))
      WRITE(6,9) I,NAM(I),C(I),D(I)
 22   WRITE (7,10) I,D(I)
C
C     EVALUATION OF FITTING RESULTS FOR
C          PRESELECTED ANGLES.
C
      IF (LUMP .EQ. 0) WRITE (6,18)
      WRITE (6,13)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  I=1,NUMB
      AT11=T11(I)
      AP11=P11(I)
      AT21=T21(I)
      AP21=P21(I)
      ATL=TL2(I)
      CALL DGOTTA(AX,AT11,AP11)
```

```
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(I) +1
      WRITE (   6,27)
      WRITE (6,19)
      DO 205 J=1,NTTT
      J11=(J-1)*NSPACE(I)
      J11=J11+NSTART(I)-NAXIS(I)
      APR=J11*0.1/CAL
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      XT=AXT
      XP=AXP
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,XT,XP,1)
      T=XT
      P=XP
      IF (LUMP .EQ. 0) GO TO 91
      IF (GAMMA .EQ. 1.0) GO TO 91
      ST = DSIN(XT)
      CT = DCOS(XT)
      SP = DSIN(XP)
      CP = DCOS(XP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      AA = GAMMA*ST*DSQRT(SP2 + CP2/ALPHA2)
      CALL DFLOP(AA,CT,XTT)
      AA = ALPHA*SP
      CALL DFLOP(AA,CP,XPP)
      TT = XTT
      PP = XPP
      AA = ST*DSQRT(SP2 + CP2*ALPHA2)/GAMMA
      CALL DFLOP(AA,CT,XTTT)
      AA=SP/ALPHA
      CALL DFLOP(AA,CP,XPPP)
      TTT = XTTT
      PPP = XPPP
      CALL S2POLY(TT,PP,BE)
      CALL S2PCLY(TTT,PPP,B)
      A1 = 0.C
      A2 = 0.0
      AR1  = 0.0
      AR2 = 0.0
      DO 93 K = 1,KE
      A1 = A1 + C(K)*BE(K)
 93   AR1 = AR1 + D(K)*B(K)
      TEMP = G2*ST2*(ALPHA2*SP2+CP2) + ALPHA2*CT2
      A2 = A1/DSQRT(TEMP)
      TEMP = ST2*(ALPHA2*CP2+SP2) + G2*CT2
      AR2 = AR1/TEMP
```

```
      IF (AR1 .LT. 0) GO TO 94
      IF (AR2 .LT. 0) GO TO 94
      GO TO 95
 94   AR1 = DABS(AR1)
      AR2 = DABS(AR2)
      WRITE (6,11)
 95   CONTINUE
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      WRITE (6,14) JO,AXT,AXP,A1,AR1,A2,AR2
      WRITE(10,41) JO,AXT,AXP,A2,AR2
      TT=TT*180.0D0/PI
      PP=PP*180.0D0/PI
      TTT=TTT*180.0D0/PI
      PPP=PPP*180.0D0/PI
200   WRITE(10,42)TT,PP,A1,TTT,PPP,AR1
205   CONTINUE
      GO TO 291
 91   CALL S2POLY(T,P,B)
      AREA = 0.0
      ARAD = 0.0
      DO 100 K = 1,KE
      AREA = AREA + C(K)*B(K)
 100  ARAD = ARAD + D(K)*B(K)
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      JO=(J-1)*NSPACE(I)+NSTART(I)
      IF (LUMP .GT. 0) GO TO 101
      AREA = 1.0/AREA
      IF (AREA .LT. 0.0) AREA = 0.0
      AREA = DSQRT(AREA)
      ARAD = 0.0
      GO TO 102
 101  IF (ARAD .GE. 0.0) GO TO 103
      WRITE (6,11)
      ARAD = DABS(ARAD)
 103  CONTINUE
      ARAD = DSQRT(ARAD)
 102  WRITE(6,1) JO,AXT,AXP,AREA,ARAD
      WRITE(10,41) JO,AXT,AXP,AREA,ARAD
      GO TO 205
 291  CONTINUE
 139  NUT = 0
 500  CONTINUE
      STOP
      END
```

FERMI VELOCITY PROGRAMS

```
C ***********************************************************************
C *                                                                     *
C *                          J131                                       *
C *                                                                     *
C *   FERMI VELOCITY PROGRAM                                            *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  THE C(I) ARE RADIUS**2 COEFFICIENTS.                     *
C *        2)  THE D(I) ARE (1/PI)2K(DK/DE) COEFFICIENTS.               *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 T11,T21,P11,P21,TL2
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),B(91),IN(91)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/KUBE/NAM,SYM,KE,KLN
 1    FORMAT(15X,I5/4F20.15/4F20.15/4F20.15/4F20.15)
 2    FORMAT (I4)
 3    FORMAT (10X,I4,E17.6)
 4    FORMAT (1H1,4X,'#',7X,'RADIUS**2 COEFFICIENTS'/)
 5    FORMAT (1H1,4X,'#',7X,'(1/PI)2K(DK/DE) COEFFICIENTS'/)
 6    FORMAT (I5,12X,E13.6)
 7    FORMAT (I5,15X,E13.6)
8     FORMAT(1H1,2X,'ROTOR',3X,'THETA',5X,'PHI',4X,'FERMI VELOCITY',
     $5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',/27X,
     $'(K COMPONENT)',4X,'(THETA COMPONENT)',3X,'(PHI COMPONENT)',5X,
     $'(ABS. VALUE)'/)
9     FORMAT(3X,I3,3X,F6.2,3X,F6.2,4E19.6)
 10   FORMAT (//10X,' AREA = ',F10.6//10X,' EFFECTIVE MASS = ',F10.6)
12    FORMAT(I3,2F6.2,4E16.6,1X)
14    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING')
15    FORMAT(I5)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5)
      READ (5,1) (NAM(I),(SYM(I,J),J=1,16),I=1,91)
      DO 11 I = 1,91
      C(I) = 0.0
 11   D(I) = 0.0
      READ (5,2) LC
      READ (5,3) (IN(I),C(I),I=1,LC)
      WRITE (6,4)
      WRITE (6,6) (IN(I),C(I),I=1,LC)
      READ (5,2) LD
      READ (5,3) (IN(I),D(I),I=1,LD)
      WRITE (6,5)
      WRITE (6,7) (IN(I),D(I),I=1,LD)
      CALL COMSET
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,1)
      PI = 3.141592653589793
      READ(5,15) NUMB
      IF (NUMB.EQ.0) GO TO 289
```

```
      WRITE(6,14)
      DO 290  I=1,NUMB
      READ (5,2) T11(I), P11(I)
      READ (5,2) T21(I), P21(I), TL2(I)
      READ (5,3) NTOT(I),NSPACE(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
C
C     CALCULATION OF FERMI VELOCITY
C
      WRITE (6,8)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  II=1,NUMB
      AT11=T11(II)
      AP11=P11(II)
      AT21=T21(II)
      AP21=P21(II)
      ATL=TL2(II)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(II) + 1
      AREA = 0.0
      DADE = 0.0
      DO 200   J=1,NTTT
      APR=J-1
      APR=APR*NSPACE(II)
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      T=AXT
      P=AXP
      TT=AXT
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      VK = 0.0
      VT = 0.0
      VP = 0.0
      VD = 0.0
      VC = 0.0
      CALL KUPOLY(T,P,B)
      DO 20 I = 1,KE
      VC = VC + C(I)*B(I)
 20   VD = VD + D(I)*B(I)
      VC = DSQRT(VC)
      VD = PI * VD
```

```
      CALL KUPOLT(T,P,B)
      DO 21 I = 1,KE
 21   VT = VT + C(I)*B(I)
      CALL KUPOLP(T,P,B)
      DO 22 I = 1,KE
 22   VP = VP + C(I)*B(I)
      VK = (VC + VC)/VD
      VT = -VT/(VD*VC)
      GP = 0.0
      IF (DABS(TT) .GT. 1.0D-06) GP = 1.0D0/(DSIN(TT)*VD*VC)
      VP = -VP * GP
      VABS = VK*VK + VT*VT + VP*VP
      VABS = DSQRT(VABS)
      JJ=(J-1)*NSPACE(II)
      WRITE(6,9) JJ,AXT,AXP,VK,VT,VP,VABS
      WRITE(10,12)  JJ,AXT,AXP,VK,VT,VP,VABS
C
C     SIMPSON'S RULE INTEGRATION FOR THE FERMI AREA AND EFFECTIVE MASS
C
      RSQ = VC * VC
      V = VC/VK
      K=J-1
      IF ( MOD(K,2) .EQ. 0) GO TO 30
      AV=V+V+V+V
      AD = RSQ + RSQ + RSQ + RSQ
      GO TO 33
 30   AV=V+V
      AD = RSQ + RSQ
      IF(J.EQ.1) GO TO 32
      IF(J.EQ.NTTT) GO TO 32
      GO TO 33
 32   AV = V
      AD = RSQ
 33   AREA = AREA  + AD
      DADE = DADE + AV
 200  CONTINUE
      DT=NSPACE(II)*PI/180.0D0
      AMULT=NSPACE(II)*NTOT(II)
      AMULT=360.0D0/AMULT
      AREA=AREA*AMULT*DT/6.0D0
      DADE=2.0D0*DADE*AMULT*DT/6.0D0
      EM = DADE/PI
      WRITE (6,10) AREA,EM
 291  CONTINUE
 139  CONTINUE
      STOP
      END


C ***********************************************************************
C *                                                                     *
C *                          J132                                       *
C *                                                                     *
C *   FERMI VELOCITY PROGRAM                                            *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:   D6H                                             *
```

```
C *                                                                    *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      REAL*4 T11,T21,P11,P21,TL2
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),B(91),IN(91)
      DIMENSION A(91),F(91),VV(3)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3,JJJJ
 1    FORMAT (F10.4)
 2    FORMAT (I4)
 3    FORMAT (10X,I4,E17.6)
 4    FORMAT (1H1,4X,'#',7X,'RADIUS**2 COEFFICIENTS'/)
 5    FORMAT (1H1,4X,'#',7X,'(1/PI)2K(DK/DE) COEFFICIENTS'/)
 6    FORMAT (I5,12X,E13.6)
 7    FORMAT (I5,15X,E13.6)
8     FORMAT(1H1,2X,'ROTOR',3X,'THETA',5X,'PHI',4X,'FERMI VELOCITY',
     $5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',/27X,
     $'(K COMPONENT)',4X,'(THETA COMPONENT)',3X,'(PHI COMPONENT)',5X,
     $'(ABS. VALUE)'/)
9     FORMAT(3X,I3,3X,F6.2,3X,F6.2,4E19.6)
 10   FORMAT (//10X,' AREA = ',F10.6//10X,' EFFECTIVE MASS = ',F10.6)
12    FORMAT(I3,2F6.2,4E16.6,1X)
14    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING')
15    FORMAT(I5)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5)
      LUMP = 1
      NUT = 0
      CALL COMSET
      DO 11 I = 1,91
      C(I) = 0.0
 11   D(I) = 0.0
      READ (5,1) GAMMA
      ALPHA = 1.0
      EA1 = 0.0
      EA2 = 0.0
      EA3 = 0.0
      READ (5,2) LC
      READ (5,3) (IN(I),C(I),I=1,LC)
      WRITE (6,4)
      WRITE (6,6) (IN(I),C(I),I=1,LC)
      READ (5,1) GAMMB
      ALPHB = 1.0
      EB1 = 0.0
      EB2=0.0
      EB3 = 0.0
      READ (5,2) LD
      READ (5,3) (IN(I),D(I),I=1,LD)
      WRITE (6,5)
      WRITE (6,7) (IN(I),D(I),I=1,LD)
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,2)
      PI = 3.141592653589793
      JJJJ = 0
```

```
      IIII = 0
      IF (EA1 .NE. 0.0) JJJJ = 1
      IF (EA2 .NE. 0.0) JJJJ = 1
      IF (EA3 .NE. 0.0) JJJJ = 1
      IF (EB1 .NE. 0.0) IIII = 1
      IF (EB2 .NE. 0.0) IIII = 1
      IF (EB3 .NE. 0.0) IIII = 1
      AA2 = ALPHA * ALPHA
      AB2 = ALPHB * ALPHB
      GA2 = GAMMA * GAMMA
      GB2 = GAMMB * GAMMB
      READ(5,15) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE(6,14)
      DO 290  I=1,NUMB
      READ (5,1) T11(I), P11(I)
      READ (5,1) T21(I), P21(I), TL2(I)
      READ (5,15)NTOT(I),NSPACE(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
C
C     CALCULATION OF FERMI VELOCITY
C
      WRITE (6,8)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  II=1,NUMB
      AT11=T11(II)
      AP11=P11(II)
      AT21=T21(II)
      AP21=P21(II)
      ATL=TL2(II)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(II) + 1
      AREA = 0.0
      DADE = 0.0
      DO 200   J=1,NTTT
      APR=J-1
      APR=APR*NSPACE(II)
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      TA=AXT
      TB=AXT
      PA=AXP
```

```
      PB=AXP
      TT=AXT
      PP=AXP
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      VK = 0.0
      VT = 0.0
      VP = 0.0
      VD = 0.0
      VC = 0.0
C
C     DO THE ELLIPSOIDAL TRANSFORMATION FOR THE K(DK/DE) COEFFICIENTS
C
      IF (GAMMB .EQ. 1.0) GO TO 40
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP = DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMB*CT/DSQRT(ST2*(SP2+AB2*CP2)+GB2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP = SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2)+GB2*CT2
      GO TO 41
 40   COEFB = 1.0D0
      T = TT
      P = PP
 41   CALL HXPOLY(T,P,B)
      DO 42 I = 1,KE
 42   VD = VD + D(I)*B(I)
      VD = VD * PI/COEFB
C
C     ELLIPSOIDAL TRANSFORMATION FOR THE K**2 COEFFICIENTS
C
      IF (GAMMA .EQ. 1.0) GO TO 45
      IF (JJJJ .NE. 0) CALL OILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMA*CT/DSQRT(ST2*(SP2+AA2*CP2)+GA2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      EPSILN = SP2 + AA2*CP2
```

```
      XI = ST2 * EPSILN + GA2 * CT2
      DXIDT = 2.0D0*ST*CT*(EPSILN-GA2)
      DXIDP = 2.0D0*(1.0-AA2)*ST2*SP*CP
      DPDP = ALPHA/EPSILN
      DTDP = GAMMA*(1.0-AA2)*ST*CT*SP*CP
      DTDP = DTDP/(DSQRT(EPSILN) * XI)
      DTDT = GAMMA*DSQRT(EPSILN)/XI
      COEFA = XI
      GO TO 46
 45   COEFA = 1.0D0
      EPSILN = 1.0D0
      XI = 1.0D0
      DXIDT = 0.0
      DXIDP = 0.0
      DPDP = 1.0D0
      DTDP = 0.0
      DTDT = 1.0D0
      T = TT
      P = PP
 46   CALL HXPOLY(T,P,A)
      CALL HXPOLT(T,P,B)
      CALL HXPOLP(T,P,F)
      DO 52 I = 1,KE
      VC = VC + C(I)*A(I)
      VT = VT + C(I)*B(I)
 52   VP = VP + C(I)*F(I)
      VC = VC/COEFA
      RSQ = VC
      VC = DSQRT(VC)
      VK = (VC+VC)/VD
      VTTEMP = (DTDT*VT - RSQ*DXIDT)/XI
      VPTEMP = (DTDP*VT + DPDP*VP - RSQ*DXIDP)/XI
      VT = VTTEMP
      VP = VPTEMP
      VT = -VT/(VD*VC)
      GP = 0.0
      IF (DABS(TT) .GT. 1.0D-06) GP = 1.0D0/(DSIN(TT)*VD*VC)
      VP = -VP * GP
      VABS = VK*VK + VT*VT + VP*VP
      VABS = DSQRT(VABS)
      JJ=(J-1)*NSPACE(II)
      WRITE(6,9) JJ,AXT,AXP,VK,VT,VP,VABS
      WRITE(10,12)  JJ,AXT,AXP,VK,VT,VP,VABS
C
C     SIMPSON'S RULE INTEGRATION FOR THE FERMI AREA AND EFFECTIVE MASS.
C
      V = VC/VK
      K=J-1
      IF ( MOD(K,2) .EQ. 0) GO TO 30
      AV=V+V+V+V
      AD = RSQ + RSQ + RSQ + RSQ
      GO TO 33
30    AV=V+V
      AD = RSQ + RSQ
      IF(J.EQ.1) GO TO 32
      IF(J.EQ.NTTT) GO TO 32
      GO TO 33
 32   AV = V
      AD = RSQ
33    AREA = AREA  + AD
```

```
      DADE = DADE + AV
200   CONTINUE
      DT=NSPACE(II)*PI/180.0D0
      AMULT=NSPACE(II)*NTOT(II)
      AMULT=360.0D0/AMULT
      AREA=AREA*AMULT*DT/6.0D0
      DADE=2.0D0*DADE*AMULT*DT/6.0D0
 182  EM = DADE/PI
      WRITE (6,10) AREA,EM
291   CONTINUE
 139  STOP
      END


C ***********************************************************************
C *                                                                     *
C *                          J133                                       *
C *                                                                     *
C *   FERMI VELOCITY PROGRAM                                            *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  C6H                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      REAL*4 T11,T21,P11,P21,TL2
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),B(91),IN(91)
      DIMENSION A(91),F(91),VV(3)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3,JJJJ
 1    FORMAT (F10.4)
 2    FORMAT (I4)
 3    FORMAT (10X,I4,E17.6)
 4    FORMAT (1H1,4X,'#',7X,'RADIUS**2 COEFFICIENTS'/)
 5    FORMAT (1H1,4X,'#',7X,'(1/PI)2K(DK/DE) COEFFICIENTS'/)
 6    FORMAT (I5,12X,E13.6)
 7    FORMAT (I5,15X,E13.6)
 8    FORMAT(1H1,2X,'ROTOR',3X,'THETA',5X,'PHI',4X,'FERMI VELOCITY',
     $5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',/27X,
     $'(K COMPONENT)',4X,'(THETA COMPONENT)',3X,'(PHI COMPONENT)',5X,
     $'(ABS. VALUE)'/)
 9    FORMAT(3X,I3,3X,F6.2,3X,F6.2,4E19.6)
 10   FORMAT (//10X,' AREA = ',F10.6//10X,' EFFECTIVE MASS = ',F10.6)
 12   FORMAT(I3,2F6.2,4E16.6,1X)
 14   FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING')
 15   FORMAT(I5)
 24   FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5)
      LUMP = 1
      NUT = 0
      CALL COMSET
```

```
      DO 11 I = 1,91
      C(I) = 0.0
11    D(I) = 0.0
      READ (5,1) GAMMA
      ALPHA = 1.0
      EA1 = 0.0
      EA2 = 0.0
      EA3 = 0.0
      READ (5,2) LC
      READ (5,3) (IN(I),C(I),I=1,LC)
      WRITE (6,4)
      WRITE (6,6) (IN(I),C(I),I=1,LC)
      READ (5,1) GAMMB
      ALPHB = 1.0
      EB1 = 0.0
      EB2 = 0.0
      EB3 = 0.0
      READ (5,2) LD
      READ (5,3) (IN(I),D(I),I=1,LD)
      WRITE (6,5)
      WRITE (6,7) (IN(I),D(I),I=1,LD)
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,3)
      PI = 3.141592653589793
      JJJJ = 0
      IIII = 0
      IF (EA1 .NE. 0.0) JJJJ = 1
      IF (EA2 .NE. 0.0) JJJJ = 1
      IF (EA3 .NE. 0.0) JJJJ = 1
      IF (EB1 .NE. 0.0) IIII = 1
      IF (EB2 .NE. 0.0) IIII = 1
      IF (EB3 .NE. 0.0) IIII = 1
      AA2 = ALPHA * ALPHA
      AB2 = ALPHB * ALPHB
      GA2 = GAMMA * GAMMA
      GB2 = GAMMB * GAMMB
      READ(5,15) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE(6,14)
      DO 290  I=1,NUMB
      READ (5,1) T11(I), P11(I)
      READ (5,1) T21(I), P21(I), TL2(I)
      READ (5,15)NTOT(I),NSPACE(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
C
C     CALCULATION OF FERMI VELOCITY
C
      WRITE (6,8)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  II=1,NUMB
      AT11=T11(II)
      AP11=P11(II)
      AT21=T21(II)
```

```
      AP21=P21(II)
      ATL=TL2(II)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(II) + 1
      AREA = 0.0
      DADE = 0.0
      DO 200   J=1,NTTT
      APR=J-1
      APR=APR*NSPACE(II)
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      TA=AXT
      TB=AXT
      PA=AXP
      PB=AXP
      TT=AXT
      PP=AXP
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      VK = 0.0
      VT = 0.0
      VP = 0.0
      VD = 0.0
      VC = 0.0
C
C     DO THE ELLIPSOIDAL TRANSFORMATION FOR THE K(DK/DE) COEFFICIENTS
C
      IF (GAMMB .EQ. 1.0) GO TO 40
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP = DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMB*CT/DSQRT(ST2*(SP2+AB2*CP2)+GB2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP = SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2)+GB2*CT2
      GO TO 41
 40   COEFB = 1.0D0
      T = TT
      P = PP
```

```
 41   CALL C6POLY(T,P,B)
      DO 42 I = 1,KE
 42   VD = VD + D(I)*B(I)
      VD = VD * PI/COEFB
C
C     ELLIPSOIDAL TRANSFORMATION FOR THE K**2 COEFFICIENTS
C
      IF (GAMMA .EQ. 1.0) GO TO 45
      IF (JJJJ .NE. 0) CALL CILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMA*CT/DSQRT(ST2*(SP2+AA2*CP2)+GA2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      EPSILN = SP2 + AA2*CP2
      XI = ST2 * EPSILN + GA2 * CT2
      DXIDT = 2.0D0*ST*CT*(EPSILN-GA2)
      DXIDP = 2.0D0*(1.0-AA2)*ST2*SP*CP
      DPDP = ALPHA/EPSILN
      DTDP = GAMMA*(1.0-AA2)*ST*CT*SP*CP
      DTDP = DTDP/(DSQRT(EPSILN) * XI)
      DTDT = GAMMA*DSQRT(EPSILN)/XI
      COEFA = XI
      GO TO 46
 45   COEFA = 1.0D0
      EPSILN = 1.0D0
      XI = 1.0D0
      DXIDT = 0.0
      DXIDP = 0.0
      DPDP = 1.0D0
      DTDP = 0.0
      DTDT = 1.0D0
      T = TT
      P = PP
 46   CALL C6POLY(T,P,A)
      CALL C6POLT(T,P,B)
      CALL C6POLP(T,P,F)
      DO 52 I = 1,KE
      VC = VC + C(I)*A(I)
      VT = VT + C(I)*B(I)
 52   VP = VP + C(I)*F(I)
      VC = VC/COEFA
      RSQ = VC
      VC = DSQRT(VC)
      VK = (VC+VC)/VD
      VTTEMP = (DTDT*VT - RSQ*DXIDT)/XI
      VPTEMP = (DTDP*VT + DPDP*VP - RSQ*DXIDP)/X
      VT = VTTEMP
      VP = VPTEMP
      VT = -VT/(VD*VC)
```

```
      GP = 0.0
      IF (DABS(TT) .GT. 1.0D-06) GP = 1.0D0/(DSIN(TT)*VD*VC)
      VP = -VP * GP
      VABS = VK*VK + VT*VT + VP*VP
      VABS = DSQRT(VABS)
      JJ=(J-1)*NSPACE(II)
      WRITE(6,9) JJ,AXT,AXP,VK,VT,VP,VABS
      WRITE(10,12)  JJ,AXT,AXP,VK,VT,VP,VABS
C
C     SIMPSON'S RULE INTEGRATION FOR THE FERMI AREA AND EFFECTIVE MASS.
C
      V = VC/VK
      K=J-1
      IF ( MOD(K,2) .EQ. 0) GO TO 30
      AV=V+V+V+V
      AD = RSQ + RSQ + RSQ + RSQ
      GO TO 33
30    AV=V+V
      AD = RSQ + RSQ
      IF(J.EQ.1) GO TO 32
      IF(J.EQ.NTTT) GO TO 32
      GO TO 33
 32   AV = V
      AD = RSQ
33    AREA = AREA  + AD
      DADE = DADE + AV
200   CONTINUE
      DT=NSPACE(II)*PI/180.0D0
      AMULT=NSPACE(II)*NTOT(II)
      AMULT=360.0D0/AMULT
      AREA=AREA*AMULT*DT/6.0D0
      DADE=2.0D0*DADE*AMULT*DT/6.0D0
 182  EM = DADE/PI
      WRITE (6,10) AREA,EM
291   CONTINUE
 139  STOP
      END


C ***********************************************************************
C *                                                                     *
C *                          J134                                       *
C *                                                                     *
C *    FERMI VELOCITY PROGRAM                                           *
C *                                                                     *
C *    INPUT DATA:                                                      *
C *         1)  COEFFICIENTS C(I) FOR K**2                              *
C *         2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                   *
C *                                                                     *
C *    SYMMETRY GROUP:  D3D                                             *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      REAL*4 T11,T21,P11,P21,TL2
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),B(91),IN(91)
      DIMENSION A(91),F(91),VV(3)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
```

```
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,AA2,GAMMA.GA2.EA1,EA2,EA3,JJJJ
 1    FORMAT (F10.4)
 2    FORMAT (I4)
 3    FORMAT (10X,I4,E17.6)
 4    FORMAT (1H1,4X,'#',7X,'RADIUS**2 COEFFICIENTS'/)
 5    FORMAT (1H1,4X,'#',7X,'(1/PI)2K(DK/DE) COEFFICIENTS'/)
 6    FORMAT (I5,12X,E13.6)
 7    FORMAT (I5,15X,E13.6)
8     FORMAT(1H1,2X,'ROTOR',3X,'THETA',5X,'PHI',4X,'FERMI VELOCITY',
     $5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',/27X,
     $'(K COMPONENT)',4X,'(THETA COMPONENT)',3X,'(PHI COMPONENT)',5X,
     $'(ABS. VALUE)'/)
9     FORMAT(3X,I3,3X,F6.2,3X,F6.2,4E19.6)
 10   FORMAT (//10X,' AREA = ',F10.6//10X,' EFFECTIVE MASS = ',F10.6)
12    FORMAT(I3,2F6.2,4E16.6,1X)
14    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING')
15    FORMAT(I5)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5)
      LUMP = 1
      NUT = 0
      CALL COMSET
      DO 11 I = 1,91
      C(I) = 0.0
 11   D(I) = 0.0
      READ (5,1) GAMMA
      ALPHA = 1.0
      EA1 = 0.0
      EA2 = 0.0
      EA3 = 0.0
      READ (5,2) LC
      READ (5,3) (IN(I),C(I),I=1,LC)
      WRITE (6,4)
      WRITE (6,6) (IN(I),C(I),I=1,LC)
      READ (5,1) GAMMB
      ALPHB = 1.0
      EB1 = 0.0
      EB2 = 0.0
      EB3 = 0.0
      READ (5,2) LD
      READ (5,3) (IN(I),D(I),I=1,LD)
      WRITE (6,5)
      WRITE (6,7) (IN(I),D(I),I=1,LD)
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,4)
      PI = 3.141592653589793
      JJJJ = 0
      IIII = 0
      IF (EA1 .NE. 0.0) JJJJ = 1
      IF (EA2 .NE. 0.0) JJJJ = 1
      IF (EA3 .NE. 0.0) JJJJ = 1
      IF (EB1 .NE. 0.0) IIII = 1
      IF (EB2 .NE. 0.0) IIII = 1
      IF (EB3 .NE. 0.0) IIII = 1
      AA2 = ALPHA * ALPHA
      AB2 = ALPHB * ALPHB
      GA2 = GAMMA * GAMMA
```

```
      GB2 = GAMMB * GAMMB
      READ(5,15) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE(6,14)
      DO 290  I=1,NUMB
      READ (5,1) T11(I), P11(I)
      READ (5,1) T21(I), P21(I), TL2(I)
      READ (5,15)NTOT(I),NSPACE(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
C
C     CALCULATION OF FERMI VELOCITY
C
      WRITE (6,8)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  II=1,NUMB
      AT11=T11(II)
      AP11=P11(II)
      AT21=T21(II)
      AP21=P21(II)
      ATL=TL2(II)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(II) + 1
      AREA = 0.0
      DADE = 0.0
      DO 200   J=1,NTTT
      APR=J-1
      APR=APR*NSPACE(II)
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      TA=AXT
      TB=AXT
      PA=AXP
      PB=AXP
      TT=AXT
      PP=AXP
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      VK = 0.0
      VT = 0.0
      VP = 0.0
      VD = 0.0
      VC = 0.0
```

```
C
C     DO THE ELLIPSOIDAL TRANSFORMATION FOR THE K(DK/DE) COEFFICIENTS
C
      IF (GAMMB .EQ. 1.0) GO TO 40
      IF (IIII .NE. 0) CALL CILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP = DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMB*CT/DSQRT(ST2*(SP2+AB2*CP2)+GB2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP = SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2)+GB2*CT2
      GO TO 41
 40   COEFB = 1.0D0
      T = TT
      R = PP
 41   CALL D3POLY(T,P,B)
      DO 42 I = 1,KE
 42   VD = VD + D(I)*B(I)
      VD = VD * PI/COEFB
C
C     ELLIPSOIDAL TRANSFORMATION FOR THE K**2 COEFFICIENTS
C
      IF (GAMMA .EQ. 1.0) GO TO 45
      IF (JJJJ .NE. 0) CALL CILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMA*CT/DSQRT(ST2*(SP2+AA2*CP2)+GA2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      EPSILN = SP2 + AA2*CP2
      XI = ST2 * EPSILN + GA2 * CT2
      DXIDT = 2.0D0*ST*CT*(EPSILN-GA2)
      DXIDP = 2.0D0*(1.0-AA2)*ST2*SP*CP
      DPDP = ALPHA/EPSILN
      DTDP = GAMMA*(1.0-AA2)*ST*CT*SP*CP
      DTDP = DTDP/(DSQRT(EPSILN) * XI)
      DTDT = GAMMA*DSQRT(EPSILN)/XI
      COEFA = XI
      GO TO 46
 45   COEFA = 1.0D0
```

```
      EPSILN = 1.0D0
      XI = 1.0D0
      DXIDT = 0.0
      DXIDP = 0.0
      DPDP = 1.0D0
      DTDP = 0.0
      DTDT = 1.0D0
      T = TT
      P = PP
 46   CALL D3POLY(T,P,A)
      CALL D3POLT(T,P,B)
      CALL D3POLP(T,P,F)
      DO 52 I = 1,KE
      VC = VC + C(I)*A(I)
      VT = VT + C(I)*B(I)
 52   VP = VP + C(I)*F(I)
      VC = VC/COFFA
      RSQ = VC
      VC = DSQRT(VC)
      VK = (VC+VC)/VD
      VTTEMP = (DTDT*VT - RSQ*DXIDT)/XI
      VPTEMP = (DTDP*VT + DPDP*VP - RSQ*DXIDP)/XI
      VT = VTTEMP
      VP = VPTEMP
      VT = -VT/(VD*VC)
      GP = 0.0
      IF (DABS(TT) .GT. 1.0D-06) GP = 1.0D0/(DSIN(TT)*VD*VC)
      VP = -VP * GP
      VABS = VK*VK + VT*VT + VP*VP
      VABS = DSQRT(VABS)
      JJ=(J-1)*NSPACE(II)
      WRITE(6,9) JJ,AXT,AXP,VK,VT,VP,VABS
      WRITE(10,12)  JJ,AXT,AXP,VK,VT,VP,VABS
C
C     SIMPSON'S RULE INTEGRATION FOR THE FERMI AREA AND EFFECTIVE MASS.
C
      V = VC/VK
      K=J-1
      IF ( MOD(K,2) .EQ. 0) GO TO 30
      AV=V+V+V+V
      AD = RSQ + RSQ + RSQ + RSQ
      GO TO 33
 30   AV=V+V
      AD = RSQ + RSQ
      IF(J.EQ.1) GO TO 32
      IF(J.EQ.NTTT) GO TO 32
      GO TO 33
 32   AV = V
      AD = RSQ
 33   AREA = AREA  + AD
      DADE = DADE + AV
 200  CONTINUE
      DT=NSPACE(II)*PI/180.0D0
      AMULT=NSPACE(II)*NTOT(II)
      AMULT=360.0D0/AMULT
      AREA=AREA*AMULT*DT/6.0D0
      DADE=2.0D0*DADE*AMULT*DT/6.0D0
 182  EM = DADE/PI
      WRITE (6,10) AREA,EM
 291  CONTINUE
```

```
 139  STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J135                                       *
C *                                                                     *
C *    FERMI VELOCITY PROGRAM                                           *
C *                                                                     *
C *    INPUT DATA:                                                      *
C *         1)  COEFFICIENTS C(I) FOR K**2                              *
C *         2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                   *
C *                                                                     *
C *    SYMMETRY GROUP:  S6  (C3I)                                       *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      REAL*4 T11,T21,P11,P21,TL2
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),B(91),IN(91)
      DIMENSION A(91),F(91),VV(3)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3,JJJJ
 1    FORMAT (F10.4)
 2    FORMAT (I4)
 3    FORMAT (10X,I4,E17.6)
 4    FORMAT (1H1,4X,'#',7X,'RADIUS**2 COEFFICIENTS'/)
 5    FORMAT (1H1,4X,'#',7X,'(1/PI)2K(DK/DE) COEFFICIENTS'/)
 6    FORMAT (I5,12X,E13.6)
 7    FORMAT (I5,15X,E13.6)
8     FORMAT(1H1,2X,'ROTOR',3X,'THETA',5X,'PHI',4X,'FERMI VELOCITY',
     $5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',/27X,
     $'(K COMPONENT)',4X,'(THETA COMPONENT)',3X,'(PHI COMPONENT)',5X,
     $'(ABS. VALUE)'/)
9     FORMAT(3X,I3,3X,F6.2,3X,F6.2,4E19.6)
 10   FORMAT (//10X,' AREA = ',F10.6//10X,' EFFECTIVE MASS = ',F10.6)
12    FORMAT(I3,2F6.2,4E16.6,1X)
14    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING')
15    FORMAT(I5)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5)
      LUMP = 1
      NUT = 0
      CALL COMSET
      DO 11 I = 1,91
      C(I) = 0.0
 11   D(I) = 0.0
      READ (5,1) GAMMA
      ALPHA = 1.0
      EA1 = 0.0
      EA2 = 0.0
      EA3 = 0.0
      READ (5,2) LC
      READ (5,3) (IN(I),C(I),I=1,LC)
```

```
      WRITE (6,4)
      WRITE (6,6) (IN(I),C(I),I=1,LC)
      READ (5,1) GAMMB
      ALPHB = 1.0
      EB1 = 0.0
      EB2 = 0.0
      EB3 = 0.0
      READ (5,2) LD
      READ (5,3) (IN(I),D(I),I=1,LD)
      WRITE (6,5)
      WRITE (6,7) (IN(I),D(I),I=1,LD)
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,5)
      PI = 3.141592653589793
      JJJJ = 0
      IIII = 0
      IF (EA1 .NE. 0.0) JJJJ = 1
      IF (EA2 .NE. 0.0) JJJJ = 1
      IF (EA3 .NE. 0.0) JJJJ = 1
      IF (EB1 .NE. 0.0) IIII = 1
      IF (EB2 .NE. 0.0) IIII = 1
      IF (EB3 .NE. 0.0) IIII = 1
      AA2 = ALPHA * ALPHA
      AB2 = ALPHB * ALPHB
      GA2 = GAMMA * GAMMA
      GB2 = GAMMB * GAMMB
      READ(5,15) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE(6,14)
      DO 290  I=1,NUMB
      READ (5,1) T11(I), P11(I)
      READ (5,1) T21(I), P21(I), TL2(I)
      READ (5,15)NTOT(I),NSPACE(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
C
C     CALCULATION OF FERMI VELOCITY
C
      WRITE (6,8)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  II=1,NUMB
      AT11=T11(II)
      AP11=P11(II)
      AT21=T21(II)
      AP21=P21(II)
      ATL=TL2(II)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(II) + 1
      APEA = 0.0
```

```
      DADE = 0.0
      DO 200   J=1,NTTT
      APR=J-1
      APR=APR*NSPACF(II)
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      TA=AXT
      TB=AXT
      PA=AXP
      PB=AXP
      TT=AXT
      PP=AXP
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      VK = 0.0
      VT = 0.0
      VP = 0.0
      VD = 0.0
      VC = 0.0
C
C     DO THE ELLIPSOIDAL TRANSFORMATION FOR THE K(DK/DE) COEFFICIENTS
C
      IF (GAMMB .EQ. 1.0) GO TO 40
      IF (IIII .NE. 0) CALL CILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP = DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMB*CT/DSQRT(ST2*(SP2+AB2*CP2)+GB2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP = SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2)+GB2*CT2
      GO TO 41
 40   COEFB = 1.0D0
      T = TT
      P = PP
 41   CALL S6POLY(T,P,B)
      DO 42 I = 1,KE
 42   VD = VD + D(I)*B(I)
      VD = VD * PI/COEFB
C
C     ELLIPSOIDAL TRANSFORMATION FOR THE K**2 COEFFICIENTS
C
      IF (GAMMA .EQ. 1.0) GO TO 45
      IF (JJJJ .NE. 0) CALL CILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
```

```
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMA*CT/DSQRT(ST2*(SP2+AA2*CP2)+GA2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      EPSILN = SP2 + AA2*CP2
      XI = ST2 * EPSILN + GA2 * CT2
      DXIDT = 2.0D0*ST*CT*(EPSILN-GA2)
      DXIDP = 2.0D0*(1.0-AA2)*ST2*SP*CP
      DPDP = ALPHA/EPSILN
      DTDP = GAMMA*(1.0-AA2)*ST*CT*SP*CP
      DTDP = DTDP/(DSQRT(EPSILN) * XI)
      DTDT = GAMMA*DSQRT(EPSILN)/XI
      COEFA = XI
      GO TO 46
45    COEFA = 1.0D0
      EPSILN = 1.0D0
      XI = 1.0D0
      DXIDT = 0.0
      DXIDP = 0.0
      DPDP = 1.0D0
      DTDP = 0.0
      DTDT = 1.0D0
      T = TT
      P = PP
46    CALL S6POLY(T,P,A)
      CALL S6POLT(T,P,B)
      CALL S6POLP(T,P,F)
      DO 52 I = 1,KF
      VC = VC + C(I)*A(I)
      VT = VT + C(I)*B(I)
52    VP = VP + C(I)*F(I)
      VC = VC/COEFA
      RSQ = VC
      VC = DSQRT(VC)
      VK = (VC+VC)/VD
      VTTEMP = (DTDT*VT - RSQ*DXIDT)/XI
      VPTEMP = (DTDP*VT + DPDP*VP - RSQ*DXIDP)/XI
      VT = VTTEMP
      VP = VPTEMP
      VT = -VT/(VD*VC)
      GP = 0.0
      IF (DABS(TT) .GT. 1.CD-06) GP = 1.0D0/(DSIN(TT)*VD*VC)
      VP = -VP * GP
      VABS = VK*VK + VT*VT + VP*VP
      VABS = DSQRT(VABS)
      JJ=(J-1)*NSPACE(II)
      WRITE(6,9) JJ,AXT,AXP,VK,VT,VP,VABS
      WRITE(10,12)  JJ,AXT,AXP,VK,VT,VP,VABS
C
C     SIMPSON'S RULE INTEGRATION FOR THE FERMI AREA AND EFFECTIVE MASS.
```

```
C
      V = VC/VK
      K=J-1
      IF ( MOD(K,2) .EQ. 0) GO TO 30
      AV=V+V+V+V
      AD = RSQ + RSQ + RSQ + RSQ
      GO TO 33
30    AV=V+V
      AD = RSQ + RSQ
      IF(J.EQ.1) GO TO 32
      IF(J.EQ.NTTT) GO TO 32
      GO TO 33
 32   AV = V
      AD = RSQ
33    AREA = AREA  + AD
      DADE = DADE + AV
200   CONTINUE
      DT=NSPACE(II)*PI/180.0D0
      AMULT=NSPACE(II)*NTOT(II)
      AMULT=360.0D0/AMULT
      AREA=AREA*AMULT*DT/6.0D0
      DADE=2.0D0*DADE*AMULT*DT/6.0D0
 182  EM = DADE/PI
      WRITE (6,10) AREA,EM
291   CONTINUE
 139  STOP
      END


C ***********************************************************************
C *                                                                     *
C *                          J136                                       *
C *                                                                     *
C *   FERMI VELOCITY PROGRAM                                            *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  D4H                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      REAL*4 T11,T21,P11,P21,TL2
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),B(91),IN(91)
      DIMENSION A(91),F(91),VV(3)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3,JJJJ
 1    FORMAT (F10.4)
 2    FORMAT (I4)
 3    FORMAT (10X,I4,E17.6)
 4    FORMAT (1H1,4X,'#',7X,'RADIUS**2 COEFFICIENTS'/)
 5    FORMAT (1H1,4X,'#',7X,'(1/PI)2K(DK/DE) COEFFICIENTS'/)
 6    FORMAT (I5,12X,E13.6)
```

```
 7     FORMAT (I5,15X,E13.6)
8      FORMAT(1H1,2X,'ROTOR',3X,'THETA',5X,'PHI',4X,'FERMI VELOCITY',
      $5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',/27X,
      $'(K COMPONENT)',4X,'(THETA COMPONENT)',3X,'(PHI COMPONENT)',5X,
      $'(ABS. VALUE)'/)
9      FORMAT(3X,I3,3X,F6.2,3X,F6.2,4E19.6)
 10    FORMAT (//10X,' AREA = ',F10.6//10X,' EFFECTIVE MASS = ',F10.6)
12     FORMAT(I3,2F6.2,4E16.6,1X)
14     FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
      &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING')
15     FORMAT(I5)
24     FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5)
       LUMP = 1
       NUT = 0
       CALL COMSET
       DO 11 I = 1,91
       C(I) = 0.0
 11    D(I) = 0.0
       READ (5,1) GAMMA
       ALPHA = 1.0
       EA1 = 0.0
       EA2 = 0.0
       EA3 = 0.0
       READ (5,2) LC
       READ (5,3) (IN(I),C(I),I=1,LC)
       WRITE (6,4)
       WRITE (6,6) (IN(I),C(I),I=1,LC)
       READ (5,1) GAMMB
       ALPHB = 1.0
       EB1 = 0.0
       EB2 = 0.0
       EB3 = 0.0
       READ (5,2) LD
       READ (5,3) (IN(I),D(I),I=1,LD)
       WRITE (6,5)
       WRITE (6,7) (IN(I),D(I),I=1,LD)
       KE = MAX0(LC,LD)
       KLN = MTOL(KE,6)
       PI = 3.141592653589793
       JJJJ = 0
       IIII = 0
       IF (EA1 .NE. 0.0) JJJJ = 1
       IF (EA2 .NE. 0.0) JJJJ = 1
       IF (EA3 .NE. 0.0) JJJJ = 1
       IF (EB1 .NE. 0.0) IIII = 1
       IF (EB2 .NE. 0.0) IIII = 1
       IF (EB3 .NE. 0.0) IIII = 1
       AA2 = ALPHA * ALPHA
       AB2 = ALPHB * ALPHB
       GA2 = GAMMA * GAMMA
       GB2 = GAMMB * GAMMB
       READ(5,15) NUMB
       IF (NUMB.EQ.0) GO TO 289
       WRITE(6,14)
       DO 290  I=1,NUMB
       READ (5,1) T11(I), P11(I)
       READ (5,1) T21(I), P21(I), TL2(I)
       READ (5,15)NTOT(I),NSPACE(I)
       WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
      &  , NSPACE(I)
```

```
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
C
C     CALCULATION OF FERMI VELOCITY
C
      WRITE (6,8)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  II=1,NUMB
      AT11=T11(II)
      AP11=P11(II)
      AT21=T21(II)
      AP21=P21(II)
      ATL=TL2(II)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(II) + 1
      AREA = 0.0
      DADE = 0.0
      DO 200   J=1,NTTT
      APR=J-1
      APR=APR*NSPACE(II)
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      TA=AXT
      TB=AXT
      PA=AXP
      PB=AXP
      TT=AXT
      PP=AXP
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      VK = 0.0
      VT = 0.0
      VP = 0.0
      VD = 0.0
      VC = 0.0
C
C     DO THE ELLIPSOIDAL TRANSFORMATION FOR THE K(DK/DE) COEFFICIENTS
C
      IF (GAMMB .EQ. 1.0) GO TO 40
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP = DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
```

```
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMB*CT/DSQRT(ST2*(SP2+AB2*CP2)+GB2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP = SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2)+GB2*CT2
      GO TO 41
 40   COEFB = 1.0D0
      T = TT
      P = PP
 41   CALL SQPOLY(T,P,B)
      DO 42 I = 1,KE
 42   VD = VD + D(I)*B(I)
      VD = VD * PI/COEFB
C
C     ELLIPSOIDAL TRANSFORMATION FOR THE K**2 COEFFICIENTS
C
      IF (GAMMA .EQ. 1.0) GO TO 45
      IF (JJJJ .NE. 0) CALL OILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMA*CT/DSQRT(ST2*(SP2+AA2*CP2)+GA2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      EPSILN = SP2 + AA2*CP2
      XI = ST2 * EPSILN + GA2 * CT2
      DXIDT = 2.0D0*ST*CT*(EPSILN-GA2)
      DXIDP = 2.0D0*(1.0-AA2)*ST2*SP*CP
      DPDP = ALPHA/EPSILN
      DTDP = GAMMA*(1.0-AA2)*ST*CT*SP*CP
      DTDP = DTDP/(DSQRT(EPSILN) * XI)
      DTDT = GAMMA*DSQRT(EPSILN)/XI
      COEFA = XI
      GO TO 46
 45   COEFA = 1.0D0
      EPSILN = 1.0D0
      XI = 1.0D0
      DXIDT = 0.0
      DXIDP = 0.0
      DPDP = 1.0D0
      DTDP = 0.0
      DTDT = 1.0D0
      T = TT
      P = PP
 46   CALL SQPOLY(T,P,A)
```

```
      CALL SQPOLT(T,P,B)
      CALL SQPOLP(T,P,F)
      DO 52 I = 1,KE
      VC = VC + C(I)*A(I)
      VT = VT + C(I)*B(I)
 52   VP = VP + C(I)*F(I)
      VC = VC/COEFA
      RSQ = VC
      VC = DSQRT(VC)
      VK = (VC+VC)/VD
      VTTEMP = (DTDT*VT - RSQ*DXIDT)/XI
      VPTEMP = (DTDP*VT + DPDP*VP - RSQ*DXIDP)/XI
      VT = VTTEMP
      VP = VPTEMP
      VT = -VT/(VD*VC)
      GP = 0.0
      IF (DABS(TT) .GT. 1.0D-06) GP = 1.0D0/(DSIN(TT)*VD*VC)
      VP = -VP * GP
      VABS = VK*VK + VT*VT + VP*VP
      VABS = DSQRT(VABS)
      JJ=(J-1)*NSPACE(II)
      WRITE(6,9) JJ,AXT,AXP,VK,VT,VP,VABS
      WRITE(10,12)  JJ,AXT,AXP,VK,VT,VP,VABS
C
C     SIMPSON'S RULE INTEGRATION FOR THE FERMI AREA AND EFFECTIVE MASS.
C
      V = VC/VK
      K=J-1
      IF ( MOD(K,2) .EQ. 0) GO TO 30
      AV=V+V+V+V
      AD = RSQ + RSQ + RSQ + RSQ
      GO TO 33
30    AV=V+V
      AD = RSQ + RSQ
      IF(J.EQ.1) GO TO 32
      IF(J.EQ.NTTT) GO TO 32
      GO TO 33
 32   AV = V
      AD = RSQ
33    AREA = AREA  + AD
      DADE = DADE + AV
200   CONTINUE
      DT=NSPACE(II)*PI/180.0D0
      AMULT=NSPACE(II)*NTOT(II)
      AMULT=360.0D0/AMULT
      AREA=AREA*AMULT*DT/6.0D0
      DADE=2.0D0*DADE*AMULT*DT/6.0D0
 182  EM = DADE/PI
      WRITE (6,10) AREA,EM
291   CONTINUE
 139  STOP
      END


C ***********************************************************************
C *                                                                     *
C *                          J137                                       *
C *                                                                     *
C *   FERMI VELOCITY PROGRAM                                            *
C *                                                                     *
```

```
C *    INPUT DATA:                                                        *
C *         1)  COEFFICIENTS C(I) FOR K**2                                *
C *         2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                     *
C *                                                                       *
C *    SYMMETRY GROUP:  C4H                                               *
C *                                                                       *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      REAL*4 T11,T21,P11,P21,TL2
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),B(91),IN(91)
      DIMENSION A(91),F(91),VV(3)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3,JJJJ
 1    FORMAT (F10.4)
 2    FORMAT (I4)
 3    FORMAT (10X,I4,E17.6)
 4    FORMAT (1H1,4X,'#',7X,'RADIUS**2 COEFFICIENTS'/)
 5    FORMAT (1H1,4X,'#',7X,'(1/PI)2K(DK/DE) COEFFICIENTS'/)
 6    FORMAT (I5,12X,E13.6)
 7    FORMAT (I5,15X,E13.6)
 8    FORMAT(1H1,2X,'ROTOR',3X,'THETA',5X,'PHI',4X,'FERMI VELOCITY',
     $5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',/27X,
     $'(K COMPONENT)',4X,'(THETA COMPONENT)',3X,'(PHI COMPONENT)',5X,
     $'(ABS. VALUE)'/)
 9    FORMAT(3X,I3,3X,F6.2,3X,F6.2,4E19.6)
 10   FORMAT (//10X,' AREA = ',F10.6//10X,' EFFECTIVE MASS = ',F10.6)
 12   FORMAT(I3,2F6.2,4E16.6,1X)
 14   FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING')
 15   FORMAT(I5)
 24   FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5)
      LUMP = 1
      NUT = 0
      CALL COMSET
      DO 11 I = 1,91
      C(I) = 0.0
 11   D(I) = 0.0
      READ (5,1) GAMMA
      ALPHA = 1.0
      EA1 = 0.0
      EA2 = 0.0
      EA3 = 0.0
      READ (5,2) LC
      READ (5,3) (IN(I),C(I),I=1,LC)
      WRITE (6,4)
      WRITE (6,6) (IN(I),C(I),I=1,LC)
      READ (5,1) GAMMB
      ALPHB = 1.0
      EB1 = 0.0
      EB2 = 0.0
      EB3 = 0.0
      READ (5,2) LD
      READ (5,3) (IN(I),D(I),I=1,LD)
      WRITE (6,5)
```

```
      WRITE (6,7) (IN(I),D(I),I=1,LD)
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,7)
      PI = 3.141592653589793
      JJJJ = 0
      IIII = 0
      IF (EA1 .NE. 0.0) JJJJ = 1
      IF (EA2 .NE. 0.0) JJJJ = 1
      IF (EA3 .NE. 0.0) JJJJ = 1
      IF (EB1 .NE. 0.0) IIII = 1
      IF (EB2 .NE. 0.0) IIII = 1
      IF (EB3 .NE. 0.0) IIII = 1
      AA2 = ALPHA * ALPHA
      AB2 = ALPHB * ALPHB
      GA2 = GAMMA * GAMMA
      GB2 = GAMMB * GAMMB
      READ(5,15) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE(6,14)
      DO 290  I=1,NUMB
      READ (5,1) T11(I), P11(I)
      READ (5,1) T21(I), P21(I), TL2(I)
      READ (5,15)NTOT(I),NSPACE(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
C
C     CALCULATION OF FERMI VELOCITY
C
      WRITE (6,8)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  II=1,NUMB
      AT11=T11(II)
      AP11=P11(II)
      AT21=T21(II)
      AP21=P21(II)
      ATL=TL2(II)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(II) + 1
      AREA = 0.0
      DADE = 0.0
      DO 200   J=1,NTTT
      APR=J-1
      APR=APR*NSPACE(II)
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
```

```
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      TA=AXT
      TB=AXT
      PA=AXP
      PB=AXP
      TT=AXT
      PP=AXP
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      VK = 0.0
      VT = 0.0
      VP = 0.0
      VD = 0.0
      VC = 0.0
C
C     DO THE ELLIPSOIDAL TRANSFORMATION FOR THE K(DK/DE) COEFFICIENTS
C
      IF (GAMMB .EQ. 1.0) GO TO 40
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP = DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMB*CT/DSQRT(ST2*(SP2+AB2*CP2)+GB2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0DC
      TB = DARCOS(TEMP)
      SPP = SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2)+GB2*CT2
      GO TO 41
 40   COEFB = 1.0D0
      T = TT
      P = PP
 41   CALL C4POLY(T,P,B)
      DO 42 I = 1,KE
 42   VD = VD + D(I)*B(I)
      VD = VD * PI/COEFB
C
C     ELLIPSOIDAL TRANSFORMATION FOR THE K**2 COEFFICIENTS
C
      IF (GAMMA .EQ. 1.0) GO TO 45
      IF (JJJJ .NE. 0) CALL OILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMA*CT/DSQRT(ST2*(SP2+AA2*CP2)+GA2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
```

```
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      EPSILN = SP2 + AA2*CP2
      XI = ST2 * EPSILN + GA2 * CT2
      DXIDT = 2.0D0*ST*CT*(EPSILN-GA2)
      DXIDP = 2.0D0*(1.0-AA2)*ST2*SP*CP
      DPDP = ALPHA/EPSILN
      DTDP = GAMMA*(1.0-AA2)*ST*CT*SP*CP
      DTDP = DTDP/(DSQRT(EPSILN) * XI)
      DTDT = GAMMA*DSQRT(EPSILN)/XI
      COEFA = XI
      GO TO 46
 45   COEFA = 1.0D0
      EPSILN = 1.0D0
      XI = 1.0D0
      DXIDT = 0.0
      DXIDP = 0.0
      DPDP = 1.0D0
      DTDP = 0.0
      DTDT = 1.0D0
      T = TT
      P = PP
 46   CALL C4POLY(T,P,A)
      CALL C4POLT(T,P,B)
      CALL C4POLP(T,P,F)
      DO 52 I = 1,KE
      VC = VC + C(I)*A(I)
      VT = VT + C(I)*B(I)
 52   VP = VP + C(I)*F(I)
      VC = VC/COEFA
      RSQ = VC
      VC = DSQRT(VC)
      VK = (VC+VC)/VD
      VTTEMP = (DTDT*VT - RSQ*DXIDT)/XI
      VPTEMP = (DTDP*VT + DPDP*VP - RSQ*DXIDP)/XI
      VT = VTTEMP
      VP = VPTEMP
      VT = -VT/(VD*VC)
      GP = 0.0
      IF (DABS(TT) .GT. 1.0D-06) GP = 1.0D0/(DSIN(TT)*VD*VC)
      VP = -VP * GP
      VABS = VK*VK + VT*VT + VP*VP
      VABS = DSQRT(VABS)
      JJ=(J-1)*NSPACE(II)
      WRITE(6,9) JJ,AXT,AXP,VK,VT,VP,VABS
      WRITE(10,12)  JJ,AXT,AXP,VK,VT,VP,VABS
C
C     SIMPSON'S RULE INTEGRATION FOR THE FERMI AREA AND EFFECTIVE MASS.
C
      V = VC/VK
      K=J-1
      IF ( MOD(K,2) .EQ. 0) GO TO 30
      AV=V+V+V+V
      AD = RSQ + RSQ + RSQ + RSQ
      GO TO 33
30    AV=V+V
      AD = RSQ + RSQ
      IF(J.EQ.1) GO TO 32
```

```
      IF(J.EQ.NTTT) GO TO 32
      GO TO 33
 32   AV = V
      AD = RSQ
33    AREA = AREA  + AD
      DADE = DADE + AV
200   CONTINUE
      DT=NSPACE(II)*PI/180.0D0
      AMULT=NSPACE(II)*NTOT(II)
      AMULT=360.0D0/AMULT
      AREA=AREA*AMULT*DT/6.0D0
      DADE=2.0D0*DADE*AMULT*DT/6.0D0
 182  EM = DADE/PI
      WRITE (6,10) AREA,EM
291   CONTINUE
 139  STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J138                                       *
C *                                                                     *
C *   FERMI VELOCITY PROGRAM                                            *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  VH                                               *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      REAL*4 T11,T21,P11,P21,TL2
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),B(91),IN(91)
      DIMENSION A(91),F(91),VV(3)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3,JJJJ
  1   FORMAT (F10.4)
  2   FORMAT (I4)
  3   FORMAT (10X,I4,E17.6)
  4   FORMAT (1H1,4X,'#',7X,'RADIUS**2 COEFFICIENTS'/)
  5   FORMAT (1H1,4X,'#',7X,'(1/PI)2K(DK/DE) COEFFICIENTS'/)
  6   FORMAT (I5,12X,E13.6)
  7   FORMAT (I5,15X,E13.6)
 8    FORMAT(1H1,2X,'ROTOR',3X,'THETA',5X,'PHI',4X,'FERMI VELOCITY',
     $5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',/27X,
     $'(K COMPONENT)',4X,'(THETA COMPONENT)',3X,'(PHI COMPONENT)',5X,
     $'(ABS. VALUE)'/)
 9    FORMAT(3X,I3,3X,F6.2,3X,F6.2,4E19.6)
 10   FORMAT (//10X,' AREA = ',F10.6//10X,' EFFECTIVE MASS = ',F10.6)
 12   FORMAT(I3,2F6.2,4E16.6,1X)
 14   FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING')
```

```
15    FORMAT(I5)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5)
      LUMP = 1
      NUT = 0
      CALL COMSET
      DO 11 I = 1,91
      C(I) = 0.0
 11   D(I) = 0.0
      READ (5,1) GAMMA,ALPHA
      EA1 = 0.0
      EA2 = 0.0
      EA3 = 0.0
      READ (5,2) LC
      READ (5,3) (IN(I),C(I),I=1,LC)
      WRITE (6,4)
      WRITE (6,6) (IN(I),C(I),I=1,LC)
      READ (5,1) GAMMB.ALPHB
      EB1 = 0.0
      EB2 = 0.0
      EB3 = 0.0
      READ (5,2) LD
      READ (5,3) (IN(I),D(I),I=1,LD)
      WRITE (6,5)
      WRITE (6,7) (IN(I),D(I),I=1,LD)
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,8)
      PI = 3.141592653589793
      JJJJ = 0
      IIII = 0
      IF (EA1 .NE. 0.0) JJJJ = 1
      IF (EA2 .NE. 0.0) JJJJ = 1
      IF (EA3 .NE. 0.0) JJJJ = 1
      IF (EB1 .NE. 0.0) IIII = 1
      IF (EB2 .NE. 0.0) IIII = 1
      IF (EB3 .NE. 0.0) IIII = 1
      AA2 = ALPHA * ALPHA
      AB2 = ALPHB * ALPHB
      GA2 = GAMMA * GAMMA
      GB2 = GAMMB * GAMMB
      READ(5,15) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE(6,14)
      DO 290  I=1,NUMB
      READ (5,1) T11(I), P11(I)
      READ (5,1) T21(I), P21(I), TL2(I)
      READ (5,15)NTOT(I),NSPACE(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
C
C     CALCULATION OF FERMI VELOCITY
C
      WRITE (6,8)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  II=1,NUMB
```

```
      AT11=T11(II)
      AP11=P11(II)
      AT21=T21(II)
      AP21=P21(II)
      ATL=TL2(II)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(II) + 1
      APEA = 0.0
      DADE = 0.0
      DO 200   J=1,NTTT
      APR=J-1
      APR=APR*NSPACE(II)
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      TA=AXT
      TB=AXT
      PA=AXP
      PB=AXP
      TT=AXT
      PP=AXP
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      VK = 0.0
      VT = 0.0
      VP = 0.0
      VD = 0.0
      VC = 0.0
C
C     DO THE ELLIPSOIDAL TRANSFORMATION FOR THE K(DK/DE) COEFFICIENTS
C
      IF (GAMMB .EQ. 1.0) GO TO 40
      IF (IIII .NE. 0) CALL CILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP = DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMB*CT/DSQRT(ST2*(SP2+AB2*CP2)+GB2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP = SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2)+GB2*CT2
      GO TO 41
```

```
 40   COEFB = 1.0D0
      T = TT
      P = PP
 41   CALL VHPOLY(T,P,B)
      DO 42 I = 1,KE
 42   VD = VD + D(I)*B(I)
      VD = VD * PI/COEFB
C
C     ELLIPSOIDAL TRANSFORMATION FOR THE K**2 COEFFICIENTS
C
      IF (GAMMA .EQ. 1.0) GO TO 45
      IF (JJJJ .NE. 0) CALL CILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMA*CT/DSQRT(ST2*(SP2+AA2*CP2)+GA2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      EPSILN = SP2 + AA2*CP2
      XI = ST2 * EPSILN + GA2 * CT2
      DXIDT = 2.0D0*ST*CT*(EPSILN-GA2)
      DXIDP = 2.0D0*(1.0-AA2)*ST2*SP*CP
      DPDP = ALPHA/EPSILN
      DTDP = GAMMA*(1.0-AA2)*ST*CT*SP*CP
      DTDP = DTDP/(DSQRT(EPSILN) * XI)
      DTDT = GAMMA*DSQRT(EPSILN)/XI
      COEFA = XI
      GO TO 46
 45   COEFA = 1.0D0
      EPSILN = 1.0D0
      XI = 1.0D0
      DXIDT = 0.0
      DXIDP = 0.0
      DPDP = 1.0D0
      DTDP = 0.0
      DTDT = 1.0D0
      T = TT
      P = PP
 46   CALL VHPOLY(T,P,A)
      CALL VHPOLT(T,P,B)
      CALL VHPOLP(T,P,F)
      DO 52 I = 1,KE
      VC = VC + C(I)*A(I)
      VT = VT + C(I)*B(I)
 52   VP = VP + C(I)*F(I)
      VC = VC/COEFA
      RSQ = VC
      VC = DSQRT(VC)
      VK = (VC+VC)/VD
      VTTEMP = (DTDT*VT - RSQ*DXIDT)/XI
      VPTEMP = (DTDP*VT + DPDP*VP - RSQ*DXIDP)/XI
```

```
      VT = VTTEMP
      VP = VPTEMP
      VT = -VT/(VD*VC)
      GP = 0.0
      IF (DABS(TT) .GT. 1.0D-06) GP = 1.0D0/(DSIN(TT)*VD*VC)
      VP = -VP * GP
      VABS = VK*VK + VT*VT + VP*VP
      VABS = DSQRT(VABS)
      JJ=(J-1)*NSPACE(II)
      WRITE(6,9) JJ,AXT,AXP,VK,VT,VP,VABS
      WRITE(10,12)  JJ,AXT,AXP,VK,VT,VP,VABS
C
C     SIMPSON'S RULE INTEGRATION FOR THE FERMI AREA AND EFFECTIVE MASS.
C
      V = VC/VK
      K=J-1
      IF ( MOD(K,2) .EQ. 0) GO TO 30
      AV=V+V+V+V
      AD = RSQ + RSQ + RSQ + RSQ
      GO TO 33
30    AV=V+V
      AD = RSQ + RSQ
      IF(J.EQ.1) GO TO 32
      IF(J.EQ.NTTT) GO TO 32
      GO TO 33
 32   AV = V
      AD = RSQ
33    AREA = AREA  + AD
      DADE = DADE + AV
200   CONTINUE
      DT=NSPACE(II)*PI/180.0D0
      AMULT=NSPACE(II)*NTOT(II)
      AMULT=360.0D0/AMULT
      AREA=AREA*AMULT*DT/6.0D0
      DADE=2.0D0*DADE*AMULT*DT/6.0D0
 182  EM = DADE/PI
      WRITE (6,10) AREA,EM
291   CONTINUE
 139  STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J139                                       *
C *                                                                     *
C *   FERMI VELOCITY PROGRAM                                            *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  C2H                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      REAL*4 T11,T21,P11,P21,TL2
```

```
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),B(91),IN(91)
      DIMENSION A(91),F(91),VV(3)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3,JJJJ
 1    FORMAT (F10.4)
 2    FORMAT (I4)
 3    FORMAT (10X,I4,E17.6)
 4    FORMAT (1H1,4X,'#',7X,'RADIUS**2 COEFFICIENTS'/)
 5    FORMAT (1H1,4X,'#',7X,'(1/PI)2K(DK/DE) COEFFICIENTS'/)
 6    FORMAT (I5,12X,E13.6)
 7    FORMAT (I5,15X,E13.6)
8     FORMAT(1H1,2X,'ROTOR',3X,'THETA',5X,'PHI',4X,'FERMI VELOCITY',
     $5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',/27X,
     $'(K COMPONENT)',4X,'(THETA COMPONENT)',3X,'(PHI COMPONENT)',5X,
     $'(ABS. VALUE)'/)
9     FORMAT(3X,I3,3X,F6.2,3X,F6.2,4E19.6)
 10   FORMAT (//10X,' AREA = ',F10.6//10X,' EFFECTIVE MASS = ',F10.6)
12    FORMAT(I3,2F6.2,4E16.6,1X)
14    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING')
15    FORMAT(I5)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5)
      LUMP = 1
      NUT = 0
      CALL COMSET
      DO 11 I = 1,91
      C(I) = 0.0
 11   D(I) = 0.0
      READ (5,1) GAMMA,ALPHA,EA2
      EA1 = 0.0
      EA3 = 0.0
      READ (5,2) LC
      READ (5,3) (IN(I),C(I),I=1,LC)
      WRITE (6,4)
      WRITE (6,6) (IN(I),C(I),I=1,LC)
      READ (5,1) GAMMB,ALPHB,EB2
      EB1 = 0.0
      EB3 = 0.0
      READ (5,2) LD
      READ (5,3) (IN(I),D(I),I=1,LD)
      WRITE (6,5)
      WRITE (6,7) (IN(I),D(I),I=1,LD)
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,09)
      PI = 3.141592653589793
      JJJJ = 0
      IIII = 0
      IF (EA1 .NE. 0.0) JJJJ = 1
      IF (EA2 .NE. 0.0) JJJJ = 1
      IF (EA3 .NE. 0.0) JJJJ = 1
      IF (EB1 .NE. 0.0) IIII = 1
      IF (EB2 .NE. 0.0) IIII = 1
      IF (EB3 .NE. 0.0) IIII = 1
      EA2=EA2*PI/180.0D0
      EB2=EB2*PI/180.0D0
      AA2 = ALPHA * ALPHA
      AB2 = ALPHB * ALPHB
```

```
      GA2 = GAMMA * GAMMA
      GB2 = GAMMB * GAMMB
      READ(5,15) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE(6,14)
      DO 290  I=1,NUMB
      READ (5,1) T11(I), P11(I)
      READ (5,1) T21(I), P21(I), TL2(I)
      READ (5,15)NTOT(I),NSPACE(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
C
C     CALCULATION OF FERMI VELOCITY
C
      WRITE (6,8)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  II=1,NUMB
      AT11=T11(II)
      AP11=P11(II)
      AT21=T21(II)
      AP21=P21(II)
      ATL=TL2(II)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(II) + 1
      AREA = 0.0
      DADE = 0.0
      DO 200   J=1,NTTT
      APR=J-1
      APR=APR*NSPACE(II)
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      TA=AXT
      TB=AXT
      PA=AXP
      PB=AXP
      TT=AXT
      PP=AXP
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      VK = 0.0
      VT = 0.0
      VP = 0.0
      VD = 0.0
```

```
      VC = 0.0
C
C     DO THE ELLIPSOIDAL TRANSFORMATION FOR THE K(DK/DE) COEFFICIENTS
C
      IF (GAMMB .EQ. 1.0) GO TO 40
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP = DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMB*CT/DSQRT(ST2*(SP2+AB2*CP2)+GB2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP = SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2)+GB2*CT2
      GO TO 41
 40   COEFB = 1.0D0
      T = TT
      P = PP
 41   CALL C2POLY(T,P,B)
      DO 42 I = 1,KE
 42   VD = VD + D(I)*B(I)
      VD = VD * PI/COEFB
C
C     ELLIPSOIDAL TRANSFORMATION FOR THE K**2 COEFFICIENTS
C
      IF (GAMMA .EQ. 1.0) GO TO 45
      IF (JJJJ .NE. 0) CALL OILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMA*CT/DSQRT(ST2*(SP2+AA2*CP2)+GA2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      EPSILN = SP2 + AA2*CP2
      XI = ST2 * EPSILN + GA2 * CT2
      DXIDT = 2.0D0*ST*CT*(EPSILN-GA2)
      DXIDP = 2.0D0*(1.0-AA2)*ST2*SP*CP
      DPDP = ALPHA/EPSILN
      DTDP = GAMMA*(1.0-AA2)*ST*CT*SP*CP
      DTDP = DTDP/(DSQRT(EPSILN) * XI)
      DTDT = GAMMA*DSQRT(EPSILN)/XI
      COEFA = XI
      GO TO 46
```

```
45    COEFA = 1.0D0
      EPSILN = 1.0D0
      XI = 1.0D0
      DXIDT = 0.0
      DXIDP = 0.0
      DPDP = 1.0D0
      DTDP = 0.0
      DTDT = 1.0D0
      T = TT
      P = PP
46    CALL C2POLY(T,P,A)
      CALL C2POLT(T,P,B)
      CALL C2POLP(T,P,F)
      DO 52 I = 1,KE
      VC = VC + C(I)*A(I)
      VT = VT + C(I)*B(I)
52    VP = VP + C(I)*F(I)
      VC = VC/COEFA
      RSQ = VC
      VC = DSQRT(VC)
      VK = (VC+VC)/VD
      VTTEMP = (DTDT*VT - RSQ*DXIDT)/XI
      VPTEMP = (DTDP*VT + DPDP*VP - RSQ*DXIDP)/XI
      VT = VTTEMP
      VP = VPTEMP
      VT = -VT/(VD*VC)
      GP = 0.0
      IF (DABS(TT) .GT. 1.0D-06) GP = 1.0D0/(DSIN(TT)*VD*VC)
      VP = -VP * GP
      VABS = VK*VK + VT*VT + VP*VP
      VABS = DSQRT(VABS)
      JJ=(J-1)*NSPACE(II)
      WRITE(6,9) JJ,AXT,AXP,VK,VT,VP,VABS
      WRITE(10,12)  JJ,AXT,AXP,VK,VT,VP,VABS
C
C     SIMPSON'S RULE INTEGRATION FOR THE FERMI AREA AND EFFECTIVE MASS.
C
      V = VC/VK
      K=J-1
      IF ( MOD(K,2) .EQ. 0) GO TO 30
      AV=V+V+V+V
      AD = RSQ + RSQ + RSQ + RSQ
      GO TO 33
30    AV=V+V
      AD = RSQ + RSQ
      IF(J.EQ.1) GO TO 32
      IF(J.EQ.NTTT) GO TO 32
      GO TO 33
 32   AV = V
      AD = RSQ
33    AREA = AREA  + AD
      DADE = DADE + AV
200   CONTINUE
      DT=NSPACE(II)*PI/180.0D0
      AMULT=NSPACE(II)*NTOT(II)
      AMULT=360.0D0/AMULT
      AREA=AREA*AMULT*DT/6.0D0
      DADE=2.0D0*DADE*AMULT*DT/6.0D0
 182  EM = DADE/PI
      WRITE (6,10) AREA,EM
```

```
291   CONTINUE
139   STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J1310                                      *
C *                                                                     *
C *   FERMI VELOCITY PROGRAM                                            *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:   S2                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T11,T21,P11,P21,TL2
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),B(91),IN(91)
      DIMENSION A(91),F(91),VV(3)
      DIMENSION T11(10),T21(10),P11(10),P21(10),TL2(10)
      DIMENSION  NTOT(10),NSPACE(10)
      DIMENSION  AX(3), AY(3), AZ(3), AS(3),AW(3)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3,JJJJ
 1    FORMAT (F10.4)
 2    FORMAT (I4)
 3    FORMAT (10X,I4,E17.6)
 4    FORMAT (1H1,4X,'#',7X,'RADIUS**2 COEFFICIENTS'/)
 5    FORMAT (1H1,4X,'#',7X,'(1/PI)2K(DK/DE) COEFFICIENTS'/)
 6    FORMAT (I5,12X,E13.6)
 7    FORMAT (I5,15X,E13.6)
8     FORMAT(1H1,2X,'ROTOR',3X,'THETA',5X,'PHI',4X,'FERMI VELOCITY',
     $5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',5X,'FERMI VELOCITY',/27X,
     $'(K COMPONENT)',4X,'(THETA COMPONENT)',3X,'(PHI COMPONENT)',5X,
     $'(ABS. VALUE)'/)
9     FORMAT(3X,I3,3X,F6.2,3X,F6.2,4E19.6)
 10   FORMAT (//10X,' AREA = ',F10.6//10X,' EFFECTIVE MASS = ',F10.6)
12    FORMAT(I3,2F6.2,4E16.6,1X)
14    FORMAT(///4X,'PLANE',7X,'THETA(1)',5X,'PHI(1)',7X,'THETA(2)',7X,
     &  'PHI(2)',6X,'CONE ANGLE',5X,'# POINTS',5X,'SPACING')
15    FORMAT(I5)
24    FORMAT (/5X,I3,3X,5(4X,F10.6),5X,I5,7X,I5)
      LUMP = 1
      NUT = 0
      CALL COMSET
      DO 11 I = 1,91
      C(I) = 0.0
 11   D(I) = 0.0
      READ (5,1) GAMMA,ALPHA,EA1,EA2,EA3
      READ (5,2) LC
      READ (5,3) (IN(I),C(I),I=1,LC)
      WRITE (6,4)
      WRITE (6,6) (IN(I),C(I),I=1,LC)
      READ (5,1) GAMMB,ALPHB,EB1,EB2,EB3
```

```
      READ (5,2) LD
      READ (5,3) (IN(I),D(I),I=1,LD)
      WRITE (6,5)
      WRITE (6,7) (IN(I),D(I),I=1,LD)
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,10)
      PI = 3.141592653589793
      JJJJ = 0
      IIII = 0
      IF (EA1 .NE. 0.0) JJJJ = 1
      IF (EA2 .NE. 0.0) JJJJ = 1
      IF (EA3 .NE. 0.0) JJJJ = 1
      IF (EB1 .NE. 0.0) IIII = 1
      IF (EB2 .NE. 0.0) IIII = 1
      IF (EB3 .NE. 0.0) IIII = 1
      EA1=EA1*PI/180.0D0
      EA2=EA2*PI/180.0D0
      EA3=EA3*PI/180.0D0
      EB1=EB1*PI/180.0D0
      EB2=EB2*PI/180.0D0
      EB3=EB3*PI/180.0D0
      AA2 = ALPHA * ALPHA
      AB2 = ALPHB * ALPHB
      GA2 = GAMMA * GAMMA
      GB2 = GAMMB * GAMMB
      READ(5,15) NUMB
      IF (NUMB.EQ.0) GO TO 289
      WRITE(6,14)
      DO 290  I=1,NUMB
      READ (5,1) T11(I), P11(I)
      READ (5,1) T21(I), P21(I), TL2(I)
      READ (5,15)NTOT(I),NSPACE(I)
      WRITE (6,24) I,T11(I),  P11(I), T21(I),P21(I),TL2(I),NTOT(I)
     &  , NSPACE(I)
      T11(I)=T11(I)*PI/180.0D0
      T21(I)=T21(I)*PI/180.0D0
      P11(I)=P11(I)*PI/180.0D0
      P21(I)=P21(I)*PI/180.0D0
290   TL2(I)=TL2(I)*PI/180.0D0
289   CONTINUE
C
C     CALCULATION OF FERMI VELOCITY
C
      WRITE (6,8)
      IF (NUMB.EQ.0)  GO TO 139
      DO 291  II=1,NUMB
      AT11=T11(II)
      AP11=P11(II)
      AT21=T21(II)
      AP21=P21(II)
      ATL=TL2(II)
      CALL DGOTTA(AX,AT11,AP11)
      CALL DGOTTA(AY,AT21,AP21)
      CALL DCRISS(AX,AY,AZ)
      CALL DCRISS(AZ,AX,AY)
      AYC=DCOS(ATL)
      AYS=DSIN(ATL)
      NTTT=NTOT(II) + 1
      AREA = 0.0
      DADE = 0.0
```

```
      DO 200   J=1,NTTT
      APR=J-1
      APR=APR*NSPACE(II)
      APR=APR*PI/180.0D0
      AZC=DCOS(APR)
      AZS=DSIN(APR)
      DO 201  K=1,3
      AS(K)=AX(K)*AZC+AY(K)*AZS
201   AW(K)=AS(K)*AYS+AZ(K)*AYC
      AXT=DARCOS(AW(3))
      AXP=DATAN2(AW(2),AW(1))
      TA=AXT
      TB=AXT
      PA=AXP
      PB=AXP
      TT=AXT
      PP=AXP
      AXT=AXT*180.0D0/PI
      AXP=AXP*180.0D0/PI
      VK = 0.0
      VT = 0.0
      VP = 0.0
      VD = 0.0
      VC = 0.0
C
C     DO THE ELLIPSOIDAL TRANSFORMATION FOR THE K(DK/DE) COEFFICIENTS
C
      IF (GAMMB .EQ. 1.0) GO TO 40
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP = DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMB*CT/DSQRT(ST2*(SP2+AB2*CP2)+GB2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP = SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2)+GB2*CT2
      GO TO 41
 40   COEFB = 1.0D0
      T = TT
      P = PP
 41   CALL S2POLY(T,P,B)
      DO 42 I = 1,KE
 42   VD = VD + D(I)*B(I)
      VD = VD * PI/COEFB
C
C     ELLIPSOIDAL TRANSFORMATION FOR THE K**2 COEFFICIENTS
C
      IF (GAMMA .EQ. 1.0) GO TO 45
      IF (JJJJ .NE. 0) CALL OILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
```

```
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = GAMMA*CT/DSQRT(ST2*(SP2+AA2*CP2)+GA2*CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      EPSILN = SP2 + AA2*CP2
      XI = ST2 * EPSILN + GA2 * CT2
      DXIDT = 2.0D0*ST*CT*(EPSILN-GA2)
      DXIDP = 2.0D0*(1.0-AA2)*ST2*SP*CP
      DPDP = ALPHA/EPSILN
      DTDP = GAMMA*(1.0-AA2)*ST*CT*SP*CP
      DTDP = DTDP/(DSQRT(EPSILN) * XI)
      DTDT = GAMMA*DSQRT(EPSILN)/XI
      COEFA = XI
      GO TO 46
45    COEFA = 1.0D0
      EPSILN = 1.0D0
      XI = 1.0D0
      DXIDT = 0.0
      DXIDP = 0.0
      DPDP = 1.0D0
      DTDP = 0.0
      DTDT = 1.0D0
      T = TT
      P = PP
46    CALL S2POLY(T,P,A)
      CALL S2POLT(T,P,B)
      CALL S2POLP(T,P,F)
      DO 52 I = 1,KE
      VC = VC + C(I)*A(I)
      VT = VT + C(I)*B(I)
52    VP = VP + C(I)*F(I)
      VC = VC/COEFA
      RSQ = VC
      VC = DSQRT(VC)
      VK = (VC+VC)/VD
      VTTEMP = (DTDT*VT - RSQ*DXIDT)/XI
      VPTEMP = (DTDP*VT + DPDP*VP - RSQ*DXIDP)/XI
      VT = VTTEMP
      VP = VPTEMP
      VT = -VT/(VD*VC)
      GP = 0.0
      IF (DABS(TT) .GT. 1.0D-06) GP = 1.0D0/(DSIN(TT)*VD*VC)
      VP = -VP * GP
      VABS = VK*VK + VT*VT + VP*VP
      VABS = DSQRT(VABS)
      JJ=(J-1)*NSPACE(II)
      WRITE(6,9) JJ,AXT,AXP,VK,VT,VP,VABS
      WRITE(10,12)  JJ,AXT,AXP,VK,VT,VP,VABS
C
C     SIMPSON'S RULE INTEGRATION FOR THE FERMI AREA AND EFFECTIVE MASS.
C
```

```
      V = VC/VK
      K=J-1
      IF ( MOD(K,2) .EQ. 0) GO TO 30
      AV=V+V+V+V
      AD = RSQ + RSQ + RSQ + RSQ
      GO TO 33
30    AV=V+V
      AD = RSQ + RSQ
      IF(J.EQ.1) GO TO 32
      IF(J.EQ.NTTT) GO TO 32
      GO TO 33
 32   AV = V
      AD = RSQ
33    AREA = AREA  + AD
      DADE = DADE + AV
200   CONTINUE
      DT=NSPACE(II)*PI/180.0D0
      AMULT=NSPACE(II)*NTOT(II)
      AMULT=360.0D0/AMULT
      AREA=AREA*AMULT*DT/6.0D0
      DADE=2.0D0*DADE*AMULT*DT/6.0D0
 182  EM = DADE/PI
      WRITE (6,10) AREA,EM
291   CONTINUE
 139  STOP
      END
```

DENSITY OF STATES PROGRAMS

```
C ***********************************************************************
C *                                                                     *
C *                          J141                                       *
C *                                                                     *
C *   COMPUTATION OF THE DENSITY OF STATES                              *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 KDKDE
      REAL*4 T,P,SYM
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),IN(91),B(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
 1    FORMAT(10X,I4,E17.6)
 2    FORMAT(I4)
 3    FORMAT(15X,I5/4F20.15/4F20.15/4F20.15/4F20.15)
 4    FORMAT(1H1,10X,'DENSTIY OF STATES CALCULATION'//5X,
     $      ' K**2 FIT WITH ',I2,' HARMONICS.'/5X,
     $      ' (1/PI)2K(DK/DE) FIT WITH ',I2,' HARMONICS.'///)
 5    FORMAT(' INTEGRATION STEPS'//)
 6    FORMAT (/' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',6X,
     $      'FERMI VELOCITY',8X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,
     $      'IN PHI',28X,'(K COMPONENT)'/)
 7    FORMAT(//' THFTA INTEGRATION --',I4,' STEPS.  PHI INTEGRATION --',
     $      I4,' STEPS'//)
 8    FORMAT (I7,4X,I7,F15.6,F12.6,F15.8,5X,F15.8)
 9    FORMAT (///15X,' DENSITY OF STATES (IN 1/(BOHR RADII)**3-RY) IS ',
     $      1PD15.8,'.')
 11   FORMAT (F10.6)
 12   FORMAT (///15X,' DENSITY OF STATES (IN 1/ATOM-RY) IS ',1PD15.8,
     $      '.')
 13   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-EV) IS ',
     $      1PD15.8,'.')
 14   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-ERG) IS ',
     $      1PD15.8,'.')
      READ (5,3) (NAM(I),(SYM(I,J),J=1,16),I=1,91)
      DO 10 I = 1,91
      C(I) = 0.0
 10   D(I) = 0.0
      READ (5,2) LC
      READ (5,1) (IN(I),C(I),I=1,LC)
      READ (5,2) LD
      READ (5,1) (IN(I),D(I),I=1,LD)
      CALL COMSET
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,1)
      PI = 3.141593653589793
      READ (5,2) N,M
C
C     READ IN LATTICE CONSTANT AND NUMBER OF ATOMS/UNIT CELL.
C
      READ (5,11) A
```

```
      READ (5,2) NCELL
      DEN = 0.0
      WRITE (6,4) LC,LD
      WRITE(6,7) N,M
      WRITE(6,5)
      WRITE(6,6)
      DP = PI/(4.0D0 * (M-1))
      DT = PI/(2.0D0 * (N-1))
      DO 20 J = 1,M
      BE = 0.0
      PP = (J-1) * DP
      P = PP
      PHI = PP * 180.0D0/PI
      DO 21 K = 1,N
      TT = (K-1) * DT
      THETA = TT * 180.0D0/PI
      T = TT
      R2 = 0.0
      KDKDE = 0.0
      CALL KUPOLY(T,P,B)
      DO 30 I = 1,KE
      R2 = R2 + C(I)*B(I)
30    KDKDE = KDKDE + D(I) * B(I)
      KDKDE = PI * KDKDE
      R = DSQRT(R2)
      VK = R/KDKDE
      VK = VK + VK
      DS = R2 * DSIN(TT)/VK
      WRITE (6,8) K,J,THETA,PHI,VK,R
      IF (MOD(K,2) .EQ. 1) BDS = DS + DS
      IF (MOD(K,2) .EQ. 0) BDS = DS + DS + DS + DS
      IF (K .EQ. 1) BDS = DS
      IF (K .EQ. N) BDS = DS
      BE = BE + BDS
21    CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J .EQ. N) BDEN = BE
      DEN = DEN + BDEN
20    CONTINUE
      DEN = DEN * 16.00D0 * DT * DP/9.00D0
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN/PI2
      DEN = DEN + DEN
      WRITE (6,9) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      ADEN = DEN
      A = A * 1.00000000D-08
      ANCELL = NCELL
      ADEN = ADEN * A * A * A/ANCELL
      WRITE (6,12) ADEN
      DEN = DEN/13.60486
      WRITE (6,13) DEN
      DEN = DEN * 1.60206D-12
      WRITE (6,14) DEN
      STOP
      END
```

```
C ***********************************************************************
C *                                                                     *
C *                          J142                                       *
C *                                                                     *
C *   COMPUTATION OF THE DENSITY OF STATES                              *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  D6H                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 A2,G2,E1,E2,E3
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),IN(91),B(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,10X,' DENSITY OF STATES CALCULATION'//5X,
     $      ' K**2 FIT WITH ',I2,' HARMONICS.'//5X,
     $      ' (1/PI)2K(DK/DE) FIT WITH ',I2,' HARMONICS.'///)
 5    FORMAT (' INTEGRATION STEPS'//)
 6    FORMAT (/' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',6X,
     $      'FERMI VELOCITY',8X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,
     $      'IN PHI',28X,'(K COMPONENT)'/)
 7    FORMAT (//' THETA INTEGRATION --',I4,' STEPS',
     $      '   PHI INTEGRATION --',I4,' STEPS.'//)
 8    FORMAT (I7,4X,I7,F15.6,F12.6,F15.8,5X,F15.8)
 9    FORMAT (///15X,' DENSITY OF STATES (IN 1/(BOHR RADII)**3-RY) IS ',
     $      1PD15.8,'.')
 11   FORMAT (F10.6)
 12   FORMAT (///15X,' DENSITY OF STATES (IN 1/ATOM-RY) IS ',
     $      1PD15.8,'.')
 13   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-EV) IS ',
     $      1PD15.8,'.')
 14   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-ERG) IS ',
     $      1PD15.8,'.')
      DO 10 I = 1,91
      C(I) = 0.0
 10   D(I) = 0.0
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA
      ALPHA = 1.0
      EA1 = 0.0
      EA2 = 0.0
      EA3 = 0.0
      READ (5,2) LC
      READ (5,1) (IN(I),C(I),I=1,LC)
      WRITE(6,1) (IN(I),C(I),I=1,LC)
      WRITE(6,3) GAMMA
      READ (5,3) GAMMB
```

```
ALPHB = 1.0
EB1 = 0.0
EB2 = 0.0
EB3 = 0.0
READ (5,2) LD
READ (5,1) (IN(I),D(I),I=1,LD)
WRITE(6,1) (IN(I),D(I),I=1,LD)
WRITE(6,3) GAMMB
KE = MAX0(LC,LD)
KLN = MTOL(KE,2)
PI = 3.141592653589793
READ (5,2) N,M
READ (5,3) VOL
JJJJ = 0
IIII = 0
IF (EA1 .NE. 0) JJJJ = 1
IF (EA2 .NE. 0) JJJJ = 1
IF (EA3 .NE. 0) JJJJ = 1
IF (EB1 .NE. 0) IIII = 1
IF (EB2 .NE. 0) IIII = 1
IF (EB3 .NE. 0) IIII = 1
AA2 = ALPHA * ALPHA
AB2 = ALPHB * ALPHB
GA2 = GAMMA * GAMMA
GB2 = GAMMB * GAMMB
CALL COMSET
DEN = 0.0
WRITE (6,4) LC,LD
WRITE (6,7) N,M
WRITE (6,5)
WRITE (6,6)
DT = PI/(2.0D00 * (N-1))
DP = PI/(6.0D00 * (M-1))
DO 20 J = 1,M
BE = 0.0
PP = (J-1) * DP
PHI = PP * 180.0D00/PI
DO 21 K = 1,N
TT = (K-1) * DT
THETA = TT * 180.0D00/PI
TA = TT
TB = TT
PA = PP
PB = PP
IF (GAMMA .EQ. 1.0) GO TO 40
IF (JJJJ .NE. 0) CALL OILER(EA1,EA2,EA3,TA,PA,1)
ST = DSIN(TA)
CT = DCOS(TA)
SP = DSIN(PA)
CP = DCOS(PA)
ST2 = ST * ST
CT2 = CT * CT
SP2 = SP * SP
CP2 = CP * CP
TEMP = CT/DSQRT(ST2*(SP2+AA2*CP2)/GA2+CT2)
IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
TA = DARCOS(TEMP)
SPP = SP/ALPHA
CALL DFLOP(SPP,CP,PA)
T = TA
```

```
      P = PA
      COEFA = ST2*(SP2+AA2*CP2) + GA2*CT2
      GO TO 41
40    COEFA = 1.0D0
      T = TT
      P = PP
41    R2 = 0.0
      CALL HXPOLY(T,P,B)
      DO 30 I = 1,KE
30    R2 = R2 + C(I)*B(I)
      R2 = R2/COFFA
      IF (GAMMB .EQ. 1.0) GO TO 50
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP=DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AB2*CP2)/GB2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP=SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2) + GB2*CT2
      GO TO 51
 50   COEFB = 1.0D0
      T = TT
      P = PP
51    ADKDE=0.0
      CALL HXPOLY(T,P,B)
      DO 52 I = 1,KE
52    ADKDE=ADKDE+D(I)*B(I)
      ADKDE=ADKDE*PI/COEFB
      R = DSQRT(R2)
      VK=R/ADKDE
      VK = VK + VK
      DS = R2 * DSIN(TT)/VK
      WRITE (6,8) K,J,THETA,PHI,VK,R
      IF (MOD(K,2) .EQ. 1) BDS = DS + DS
      IF (MOD(K,2) .EQ. 0) BDS = DS + DS + DS + DS
      IF (K .EQ. 1) BDS = DS
      IF (K .EQ. N) BDS = DS
      BE = BE + BDS
 21   CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J.EQ.M) BDEN=BE
      DEN=DEN+BDEN
 20   CONTINUE
      DEN = DEN * 24.00D00 * DT * DP/9.00D00
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
```

```
      WRITE (6,9) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      ADEN = DEN
      VOL = VOL * 1.00D-24
      ADEN = ADEN * VOL
      WRITE (6,12) ADEN
      DEN = DEN/13.60486
      WRITE (6,13) DEN
      DEN = DEN * 1.60206D-12
      WRITE (6,14) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J143                                       *
C *                                                                     *
C *   COMPUTATION OF THE DENSITY OF STATES                              *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP: C6H                                               *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 A2,G2,E1,E2,E3
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),IN(91),B(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARPY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,10X,' DENSITY OF STATES CALCULATION'//5X,
     $     ' K**2 FIT WITH ',I2,' HARMONICS.'//5X,
     $     ' (1/PI)2K(DK/DE) FIT WITH ',I2,' HARMONICS.'///)
 5    FORMAT (' INTEGRATION STEPS'//)
 6    FORMAT (/' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',6X,
     $     'FERMI VELOCITY',8X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,
     $     'IN PHI',28X,'(K COMPONENT)'/)
 7    FORMAT (//' THETA INTEGRATION --',I4,' STEPS',
     $     '   PHI INTEGRATION --',I4,' STEPS.'//)
 8    FORMAT (I7,4X,I7,F15.6,F12.6,F15.8,5X,F15.8)
 9    FORMAT (///15X,' DENSITY OF STATES (IN 1/(BOHR RADII)**3-RY) IS ',
     $     1PD15.8,'.')
 11   FORMAT (F10.6)
 12   FORMAT (///15X,' DENSITY OF STATES (IN 1/ATOM-RY) IS ',
     $     1PD15.8,'.')
 13   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-EV) IS ',
     $     1PD15.8,'.')
 14   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-ERG) IS ',
     $     1PD15.8,'.')
      DO 10 I = 1,91
      C(I) = 0.0
```

```
10    D(I) = 0.0
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA
      ALPHA = 1.0
      EA1 = 0.0
      EA2 = 0.0
      EA3 = 0.0
      READ (5,2) LC
      READ (5,1) (IN(I),C(I),I=1,LC)
      WRITE(6,1) (IN(I),C(I),I=1,LC)
      WRITE(6,3) GAMMA
      READ (5,3) GAMMB
      ALPHB = 1.0
      EB1 = 0.0
      EB2 = 0.0
      EB3 = 0.0
      READ (5,2) LD
      READ (5,1) (IN(I),D(I),I=1,LD)
      WRITE(6,1) (IN(I),D(I),I=1,LD)
      WRITE(6,3) GAMMB
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,3)
      PI = 3.141592653589793
      READ (5,2) N,M
      READ (5,3) VOL
      JJJJ = 0
      IIII = 0
      IF (EA1 .NE. 0) JJJJ = 1
      IF (EA2 .NE. 0) JJJJ = 1
      IF (EA3 .NE. 0) JJJJ = 1
      IF (EB1 .NE. 0) IIII = 1
      IF (EB2 .NE. 0) IIII = 1
      IF (EB3 .NE. 0) IIII = 1
      AA2 = ALPHA * ALPHA
      AB2 = ALPHB * ALPHB
      GA2 = GAMMA * GAMMA
      GB2 = GAMMB * GAMMB
      CALL COMSET
      DEN = 0.0
      WRITE (6,4) LC,LD
      WRITE (6,7) N,M
      WRITE (6,5)
      WRITE (6,6)
      DP = PI/(3.00D00 * (M-1))
      DT = PI/(2.00D00*(N-1))
      DO 20 J = 1,M
      BE = 0.0
      PP = (J-1) * DP
      PHI = PP * 180.00D00/PI
      DO 21 K = 1,N
      TT = (K-1) * DT
      THETA = TT * 180.00D00/PI
      TA = TT
      TB = TT
      PA = PP
      PB = PP
      IF (GAMMA .EQ. 1.0) GO TO 40
      IF (JJJJ .NE. 0) CALL OILFR(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
```

```
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AA2*CP2)/GA2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      COEFA = ST2*(SP2+AA2*CP2) + GA2*CT2
      GO TO 41
 40   COEFA = 1.0D0
      T = TT
      P = PP
 41   R2 = 0.0
      CALL C6POLY(T,P,B)
      DO 30 I = 1,KE
 30   R2 = R2 + C(I)*B(I)
      R2 = R2/COEFA
      IF (GAMMB .EQ. 1.0) GO TO 50
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP=DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AB2*CP2)/GB2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP=SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2) + GB2*CT2
      GO TO 51
 50   COEFB = 1.0D0
      T = TT
      P = PP
51    ADKDE=0.0
      CALL C6POLY(T,P,B)
      DO 52 I = 1,KE
52    ADKDE=ADKDE+D(I)*B(I)
      ADKDE=ADKDE*PI/COEFB
      R = DSQRT(R2)
      VK=R/ADKDE
      VK = VK + VK
      DS = R2 * DSIN(TT)/VK
      WRITE (6,8) K,J,THETA,PHI,VK,R
      IF (MOD(K,2) .EQ. 1) BDS = DS + DS
      IF (MOD(K,2) .EQ. 0) BDS = DS + DS + DS + DS
      IF (K .EQ. 1) BDS = DS
      IF (K .EQ. N) BDS = DS
```

```
      BE = BE + BDS
21    CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J.EQ.M) BDEN=BE
      DEN=DEN+BDEN
20    CONTINUE
      DEN = DEN * 12.00D00 * DT * DP/9.00D00
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,9) DEN
      AO = 5.29172D-09
      DEN = DEN/(AO*AO*AO)
      ADEN = DEN
      VOL = VOL * 1.00D-24
      ADEN = ADEN * VOL
      WRITE (6,12) ADEN
      DEN = DEN/13.60486
      WRITE (6,13) DEN
      DEN = DEN * 1.60206D-12
      WRITE (6,14) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J144                                       *
C *                                                                     *
C *   COMPUTATION OF THE DENSITY OF STATES                              *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  D3D                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 A2,G2,E1,E2,E3
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),IN(91),B(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARPY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,10X,' DENSITY OF STATES CALCULATION'//5X,
     $      ' K**2 FIT WITH ',I2,' HARMONICS.'//5X,
     $      ' (1/PI)2K(DK/DE) FIT WITH ',I2,' HARMONICS.'///)
 5    FORMAT (' INTEGRATION STEPS'//)
 6    FORMAT (/' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',6X,
     $      'FERMI VELOCITY',8X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,
     $      'IN PHI',28X,'(K COMPONENT)'/)
 7    FORMAT (//' THETA INTEGRATION --',I4,' STEPS',
```

```
     $       '   PHI INTEGRATION --',I4,' STEPS.'//)
8     FORMAT (I7,4X,I7,F15.6,F12.6,F15.8,5X,F15.8)
9     FORMAT (///15X,' DENSITY OF STATES (IN 1/(BOHR RADII)**3-RY) IS ',
     $      1PD15.8,'.')
11    FORMAT (F10.6)
12    FORMAT (///15X,' DENSITY OF STATES (IN 1/ATOM-RY) IS ',
     $      1PD15.8,'.')
13    FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-EV) IS ',
     $      1PD15.8,'.')
14    FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-ERG) IS ',
     $      1PD15.8,'.')
      DO 10 I = 1,91
      C(I) = 0.0
10    D(I) = 0.0
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA
      ALPHA = 1.0
      EA1 = 0.0
      EA2 = 0.0
      EA3 = 0.0
      READ (5,2) LC
      READ (5,1) (IN(I),C(I),I=1,LC)
      WRITE(6,1) (IN(I),C(I),I=1,LC)
      WRITE(6,3) GAMMA
      READ (5,3) GAMMB
      ALPHB = 1.0
      EB1 = 0.0
      EB2 = 0.0
      EB3 = 0.0
      READ (5,2) LD
      READ (5,1) (IN(I),D(I),I=1,LD)
      WRITE(6,1) (IN(I),D(I),I=1,LD)
      WRITE(6,3) GAMMB
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,4)
      PI = 3.14159265358979 3
      READ (5,2) N,M
      READ (5,3) VOL
      JJJJ = 0
      IIII = 0
      IF (EA1 .NE. 0) JJJJ = 1
      IF (EA2 .NE. 0) JJJJ = 1
      IF (EA3 .NE. 0) JJJJ = 1
      IF (EB1 .NE. 0) IIII = 1
      IF (EB2 .NE. 0) IIII = 1
      IF (EB3 .NE. 0) IIII = 1
      AA2 = ALPHA * ALPHA
      AB2 = ALPHB * ALPHB
      GA2 = GAMMA * GAMMA
      GB2 = GAMMB * GAMMB
      CALL COMSET
      DEN = 0.0
      WRITE (6,4) LC,LD
      WRITE (6,7) N,M
      WRITE (6,5)
      WRITE (6,6)
      DP = PI/(3.00D00 * (M-1))
      DT = PI/(2.00D00 * (N-1))
      DO 20 J = 1,M
```

```
      BE = 0.0
      PP = (J-1) * DP
      PHI = PP * 180.0D00/PI
      DO 21 K = 1,N
      TT = (K-1) * DT
      THETA = TT * 180.0D00/PI
      TA = TT
      TB = TT
      PA = PP
      PB = PP
      IF (GAMMA .EQ. 1.0) GO TO 40
      IF (JJJJ .NE. 0) CALL OILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AA2*CP2)/GA2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      COEFA = ST2*(SP2+AA2*CP2) + GA2*CT2
      GO TO 41
 40   COEFA = 1.0D0
      T = TT
      P = PP
 41   R2 = 0.0
      CALL D3POLY(T,P,B)
      DO 30 I = 1,KE
 30   R2 = R2 + C(I)*B(I)
      R2 = R2/COEFA
      IF (GAMMB .EQ. 1.0) GO TO 50
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP=DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AB2*CP2)/GB2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP=SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2) + GB2*CT2
      GO TO 51
 50   COEFB = 1.0D0
      T = TT
      P = PP
 51   ADKDE=0.0
```

```
      CALL D3POLY(T,P,B)
      DO 52 I = 1,KE
52    ADKDE=ADKDE+D(I)*B(I)
      ADKDE=ADKDE*PI/COEFB
      R = DSQRT(R2)
      VK=R/ADKDE
      VK = VK + VK
      DS = R2 * DSIN(TT)/VK
      WRITE (6,8) K,J,THETA,PHI,VK,R
      IF (MOD(K,2) .EQ. 1) BDS = DS + DS
      IF (MOD(K,2) .EQ. 0) BDS = DS + DS + DS + DS
      IF (K .EQ. 1) BDS = DS
      IF (K .EQ. N) BDS = DS
      BE = BE + BDS
21    CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J.EQ.M) BDEN=BE
      DEN=DEN+BDEN
20    CONTINUE
      DEN = DEN * 12.00D00 * DT * DP/9.00D00
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,9) DEN
      AO = 5.29172D-09
      DEN = DEN/(AO*AO*AO)
      ADEN = DEN
      VOL = VOL * 1.00D-24
      ADEN = ADEN * VOL
      WRITE (6,12) ADEN
      DEN = DEN/13.60486
      WRITE (6,13) DEN
      DEN = DEN * 1.60206D-12
      WRITE (6,14) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J145                                       *
C *                                                                     *
C *   COMPUTATION OF THE DENSITY OF STATES                              *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  S6  (C3I)                                        *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 A2,G2,E1,E2,E3
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),IN(91),B(91)
```

```
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
1     FORMAT (10X,I4,E17.6)
2     FORMAT (I4)
3     FORMAT (F10.4)
4     FORMAT (1H1,10X,' DENSITY OF STATES CALCULATION'//5X,
     $      ' K**2 FIT WITH ',I2,' HARMONICS.'//5X,
     $      ' (1/PI)2K(DK/DE) FIT WITH ',I2,' HARMONICS.'///)
5     FORMAT (' INTEGRATION STEPS'//)
6     FORMAT (/' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',6X,
     $      'FERMI VELOCITY',8X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,
     $      'IN PHI',28X,'(K COMPONENT)'/)
7     FORMAT (//' THETA INTEGRATION --',I4,' STEPS',
     $      '   PHI INTEGRATION --',I4,' STEPS.'//)
8     FORMAT (I7,4X,I7,F15.6,F12.6,F15.8,5X,F15.8)
9     FORMAT (///15X,' DENSITY OF STATES (IN 1/(BOHR RADII)**3-RY) IS ',
     $      1PD15.8,'.')
11    FORMAT (F10.6)
12    FORMAT (///15X,' DENSITY OF STATES (IN 1/ATOM-RY) IS ',
     $      1PD15.8,'.')
13    FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-EV) IS ',
     $      1PD15.8,'.')
14    FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-ERG) IS ',
     $      1PD15.8,'.')
      DO 10 I = 1,91
      C(I) = 0.0
10    D(I) = 0.0
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA
      ALPHA = 1.0
      EA1 = 0.0
      EA2 = 0.0
      EA3 = 0.0
      READ (5,2) LC
      READ (5,1) (IN(I),C(I),I=1,LC)
      WRITE(6,1) (IN(I),C(I),I=1,LC)
      WRITE(6,3) GAMMA
      READ (5,3) GAMMB
      ALPHB = 1.0
      EB1 = 0.0
      EB2 = 0.0
      EB3 = 0.0
      READ (5,2) LD
      READ (5,1) (IN(I),D(I),I=1,LD)
      WRITE(6,1) (IN(I),D(I),I=1,LD)
      WRITE(6,3) GAMMB
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,5)
      PI = 3.14159265358979 3
      READ (5,2) N,M
      READ (5,3) VOL
      JJJJ = 0
      IIII = 0
      IF (EA1 .NE. 0) JJJJ = 1
      IF (EA2 .NE. 0) JJJJ = 1
      IF (EA3 .NE. 0) JJJJ = 1
      IF (EB1 .NE. 0) IIII = 1
      IF (EB2 .NE. 0) IIII = 1
      IF (EB3 .NE. 0) IIII = 1
```

```
      AA2 = ALPHA * ALPHA
      AB2 = ALPHP * ALPHB
      GA2 = GAMMA * GAMMA
      GB2 = GAMMB * GAMMB
      CALL COMSET
      DEN = 0.0
      WRITE (6,4) LC,LD
      WRITE (6,7) N,M
      WRITE (6,5)
      WRITE (6,6)
      DP = 2.0D00 * PI/(3.00D00 * (M-1))
      DT = PI/(2.00D00 * (N-1))
      DO 20 J = 1,M
      BE = 0.0
      PP = (J-1) * DP
      PHI = PP * 180.00D00/PI
      DO 21 K = 1,N
      TT = (K-1) * DT
      THETA = TT * 180.00D00/PI
      TA = TT
      TB = TT
      PA = PP
      PB = PP
      IF (GAMMA .EQ. 1.0) GO TO 40
      IF (JJJJ .NE. 0) CALL OILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AA2*CP2)/GA2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      COEFA = ST2*(SP2+AA2*CP2) + GA2*CT2
      GO TO 41
40    COEFA = 1.0D0
      T = TT
      P = PP
41    R2 = 0.0
      CALL S6POLY(T,P,B)
      DO 30 I = 1,KE
30    R2 = R2 + C(I)*B(I)
      R2 = R2/COEFA
      IF (GAMMB .EQ. 1.0) GO TO 50
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP=DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
```

```
      TEMP = CT/DSQRT(ST2*(SP2+AB2*CP2)/GB2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP=SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2) + GB2*CT2
      GO TO 51
 50   COEFB = 1.0D0
      T = TT
      P = PP
51    ADKDE=0.0
      CALL S6POLY(T,P,B)
      DO 52 I = 1,KE
52    ADKDE=ADKDE+D(I)*B(I)
      ADKDE=ADKDE*PI/COEFB
      R = DSQRT(R2)
      VK=R/ADKDE
      VK = VK + VK
      DS = R2 * DSIN(TT)/VK
      WRITE (6,8) K,J,THETA,PHI,VK,R
      IF (MOD(K,2) .EQ. 1) BDS = DS + DS
      IF (MOD(K,2) .EQ. 0) BDS = DS + DS + DS + DS
      IF (K .EQ. 1) BDS = DS
      IF (K .EQ. N) BDS = DS
      BE = BE + BDS
 21   CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J.EQ.M) BDEN=BE
      DEN=DEN+BDEN
 20   CONTINUE
      DEN = DEN * 6.00D00 * DT * DP/9.00D00
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,9) DEN
      AO = 5.29172D-09
      DEN = DEN/(AO*AO*AO)
      ADEN = DEN
      VOL = VOL * 1.00D-24
      ADEN = ADEN * VOL
      WRITE (6,12) ADEN
      DEN = DEN/13.60486
      WRITE (6,13) DEN
      DEN = DEN * 1.60206D-12
      WRITE (6,14) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J146                                       *
C *                                                                     *
C *   COMPUTATION OF THE DENSITY OF STATES                              *
C *
```

```
C *    INPUT DATA:                                                      *
C *         1)  COEFFICIENTS C(I) FOR K**2                              *
C *         2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                   *
C *                                                                     *
C *    SYMMETRY GROUP:  D4H                                             *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 A2,G2,E1,E2,E3
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),IN(91),B(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,10X,' DENSITY OF STATES CALCULATION'//5X,
     $      ' K**2 FIT WITH ',I2,' HARMONICS.'//5X,
     $      ' (1/PI)2K(DK/DE) FIT WITH ',I2,' HARMONICS.'///)
 5    FORMAT (' INTEGRATION STEPS'//)
 6    FORMAT (/' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',6X,
     $      'FERMI VELOCITY',8X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,
     $      'IN PHI',28X,'(K COMPONENT)'/)
 7    FORMAT (//' THETA INTEGRATION --',I4,' STEPS',
     $      '   PHI INTEGRATION --',I4,' STEPS.'//)
 8    FORMAT (I7,4X,I7,F15.6,F12.6,F15.8,5X,F15.8)
 9    FORMAT (///15X,' DENSITY OF STATES (IN 1/(BOHR RADII)**3-RY) IS ',
     $      1PD15.8,'.')
 11   FORMAT (F10.6)
 12   FORMAT (///15X,' DENSITY OF STATES (IN 1/ATOM-RY) IS ',
     $      1PD15.8,'.')
 13   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-EV) IS ',
     $      1PD15.8,'.')
 14   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-ERG) IS ',
     $      1PD15.8,'.')
      DO 10 I = 1,91
      C(I) = 0.0
 10   D(I) = 0.0
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA
      ALPHA = 1.0
      EA1 = 0.0
      EA2 = 0.0
      EA3 = 0.0
      READ (5,2) LC
      READ (5,1) (IN(I),C(I),I=1,LC)
      WRITE(6,1) (IN(I),C(I),I=1,LC)
      WRITE(6,3) GAMMA
      READ (5,3) GAMMB
      ALPHB = 1.0
      EB1 = 0.0
      EB2 = 0.0
      EB3 = 0.0
      READ (5,2) LD
      READ (5,1) (IN(I),D(I),I=1,LD)
      WRITE(6,1) (IN(I),D(I),I=1,LD)
      WRITE(6,3) GAMMB
```

```
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,6)
      PI = 3.141592653589793
      READ (5,2) N,M
      READ (5,3) VOL
      JJJJ = 0
      IIII = 0
      IF (EA1 .NE. 0) JJJJ = 1
      IF (EA2 .NE. 0) JJJJ = 1
      IF (EA3 .NE. 0) JJJJ = 1
      IF (EB1 .NE. 0) IIII = 1
      IF (EB2 .NE. 0) IIII = 1
      IF (EB3 .NE. 0) IIII = 1
      AA2 = ALPHA * ALPHA
      AB2 = ALPHB * ALPHB
      GA2 = GAMMA * GAMMA
      GB2 = GAMMB * GAMMB
      CALL COMSET
      DEN = 0.0
      WRITE (6,4) LC,LD
      WRITE (6,7) N,M
      WRITE (6,5)
      WRITE (6,6)
      DP = PI/(4.0D00 * (M-1))
      DT = PI/(2.0D00 * (N-1))
      DO 20 J = 1,M
      BE = 0.0
      PP = (J-1) * DP
      PHI = PP * 180.0D00/PI
      DO 21 K = 1,N
      TT = (K-1) * DT
      THETA = TT * 180.0D00/PI
      TA = TT
      TB = TT
      PA = PP
      PB = PP
      IF (GAMMA .EQ. 1.0) GO TO 40
      IF (JJJJ .NE. 0) CALL OILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AA2*CP2)/GA2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      COEFA = ST2*(SP2+AA2*CP2) + GA2*CT2
      GO TO 41
40    COEFA = 1.0D0
      T = TT
      P = PP
41    R2 = 0.0
      CALL SQPOLY(T,P,B)
```

```
      DO 30 I = 1,KE
 30   R2 = R2 + C(I)*B(I)
      R2 = R2/COEFA
      IF (GAMMB .EQ. 1.0) GO TO 50
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP=DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AB2*CP2)/GB2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP=SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2) + GB2*CT2
      GO TO 51
 50   COEFB = 1.0D0
      T = TT
      P = PP
 51   ADKDE=0.0
      CALL SQPOLY(T,P,B)
      DO 52 I = 1,KE
 52   ADKDE=ADKDE+D(I)*B(I)
      ADKDE=ADKDE*PI/COEFB
      R = DSQRT(R2)
      VK=R/ADKDE
      VK = VK + VK
      DS = R2 * DSIN(TT)/VK
      WRITE (6,8) K,J,THETA,PHI,VK,R
      IF (MOD(K,2) .EQ. 1) BDS = DS + DS
      IF (MOD(K,2) .EQ. 0) BDS = DS + DS + DS + DS
      IF (K .EQ. 1) BDS = DS
      IF (K .EQ. N) BDS = DS
      BE = BE + BDS
 21   CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J.EQ.M) BDEN=BE
      DEN=DEN+BDEN
 20   CONTINUE
      DEN = DEN * 16.00D00 * DT * DP/9.00D00
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,9) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      ADEN = DEN
      VOL = VOL * 1.00D-24
      ADEN = ADEN * VOL
      WRITE (6,12) ADEN
      DEN = DEN/13.60486
```

```
      WRITE (6,13) DEN
      DEN = DEN * 1.60206D-12
      WRITE (6,14) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J147                                       *
C *                                                                     *
C *   COMPUTATION OF THE DENSITY OF STATES                              *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  C4H                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 A2,G2,E1,E2,E3
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),IN(91),B(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,10X,' DENSITY OF STATES CALCULATION'//5X,
     $      ' K**2 FIT WITH ',I2,' HARMONICS.'//5X,
     $      ' (1/PI)2K(DK/DE) FIT WITH ',I2,' HARMONICS.'///)
 5    FORMAT (' INTEGRATION STEPS'//)
 6    FORMAT (/' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',6X,
     $      'FERMI VELOCITY',8X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,
     $      'IN PHI',28X,'(K COMPONENT)'/)
 7    FORMAT (//' THETA INTEGRATION --',I4,' STEPS',
     $      '   PHI INTEGRATION --',I4,' STEPS.'//)
 8    FORMAT (I7,4X,I7,F15.6,F12.6,F15.8,5X,F15.8)
 9    FORMAT (///15X,' DENSITY OF STATES (IN 1/(BOHR RADII)**3-RY) IS ',
     $      1PD15.8,'.')
 11   FORMAT (F10.6)
 12   FORMAT (///15X,' DENSITY OF STATES (IN 1/ATOM-RY) IS ',
     $      1PD15.8,'.')
 13   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-EV) IS ',
     $      1PD15.8,'.')
 14   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-ERG) IS ',
     $      1PD15.8,'.')
      DO 10 I = 1,91
      C(I) = 0.0
 10   D(I) = 0.0
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA
      ALPHA = 1.0
      EA1 = 0.0
      EA2 = 0.0
      EA3 = 0.0
```

```
READ (5,2) LC
READ (5,1) (IN(I),C(I),I=1,LC)
WRITE(6,1) (IN(I),C(I),I=1,LC)
WRITE(6,3) GAMMA
READ (5,3) GAMMB
ALPHB = 1.0
EB1 = 0.0
EB2 = 0.0
EB3 = 0.0
READ (5,2) LD
READ (5,1) (IN(I),D(I),I=1,LD)
WRITE(6,1) (IN(I),D(I),I=1,LD)
WRITE(6,3) GAMMB
KE = MAX0(LC,LD)
KLN = MTOL(KE,7)
PI = 3.14159265358979 3
READ (5,2) N,M
READ (5,3) VOL
JJJJ = 0
IIII = 0
IF (EA1 .NE. 0) JJJJ = 1
IF (EA2 .NE. 0) JJJJ = 1
IF (EA3 .NE. 0) JJJJ = 1
IF (EB1 .NE. 0) IIII = 1
IF (EB2 .NE. 0) IIII = 1
IF (EB3 .NE. 0) IIII = 1
AA2 = ALPHA * ALPHA
AB2 = ALPHB * ALPHB
GA2 = GAMMA * GAMMA
GB2 = GAMMB * GAMMB
CALL COMSET
DEN = 0.0
WRITE (6,4) LC,LD
WRITE (6,7) N,M
WRITE (6,5)
WRITE (6,6)
DP = PI/(2.00D00 * (M-1))
DT = PI/(2.00D00 * (N-1))
DO 20 J = 1,M
BE = 0.0
PP = (J-1) * DP
PHI = PP * 180.00D00/PI
DO 21 K = 1,N
TT = (K-1) * DT
THETA = TT * 180.00D00/PI
TA = TT
TB = TT
PA = PP
PB = PP
IF (GAMMA .EQ. 1.0) GO TO 40
IF (JJJJ .NE. 0) CALL OILER(EA1,EA2,EA3,TA,PA,1)
ST = DSIN(TA)
CT = DCOS(TA)
SP = DSIN(PA)
CP = DCOS(PA)
ST2 = ST * ST
CT2 = CT * CT
SP2 = SP * SP
CP2 = CP * CP
TEMP = CT/DSQRT(ST2*(SP2+AA2*CP2)/GA2+CT2)
```

```
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      COEFA = ST2*(SP2+AA2*CP2) + GA2*CT2
      GO TO 41
40    COEFA = 1.0D0
      T = TT
      P = PP
41    R2 = 0.0
      CALL C4POLY(T,P,B)
      DO 30 I = 1,KE
30    R2 = R2 + C(I)*B(I)
      R2 = R2/COEFA
      IF (GAMMB .EQ. 1.0) GO TO 50
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP=DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AB2*CP2)/GB2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP=SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2) + GB2*CT2
      GO TO 51
50    COEFB = 1.0D0
      T = TT
      P = PP
51    ADKDE=0.0
      CALL C4POLY(T,P,B)
      DO 52 I = 1,KE
52    ADKDE=ADKDE+D(I)*B(I)
      ADKDE=ADKDE*PI/COEFB
      R = DSQRT(R2)
      VK=R/ADKDE
      VK = VK + VK
      DS = R2 * DSIN(TT)/VK
      WRITE (6,8) K,J,THETA,PHI,VK,R
      IF (MOD(K,2) .EQ. 1) BDS = DS + DS
      IF (MOD(K,2) .EQ. 0) BDS = DS + DS + DS + DS
      IF (K .EQ. 1) BDS = DS
      IF (K .EQ. N) BDS = DS
      BE = BE + BDS
21    CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J.EQ.M) BDEN=BE
      DEN=DEN+BDEN
20    CONTINUE
```

```
      DEN = DEN * 8.00D00 * DT * DP/9.00D00
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,9) DEN
      AO = 5.29172D-09
      DEN = DEN/(AO*AO*AO)
      ADEN = DEN
      VOL = VOL * 1.00D-24
      ADEN = ADEN * VOL
      WRITE (6,12) ADEN
      DEN = DEN/13.60486
      WRITE (6,13) DEN
      DEN = DEN * 1.60206D-12
      WRITE (6,14) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J148                                       *
C *                                                                     *
C *   COMPUTATION OF THE DENSITY OF STATES                              *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  VH                                               *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 A2,G2,E1,E2,E3
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),IN(91),B(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,10X,' DENSITY OF STATES CALCULATION'//5X,
     $     ' K**2 FIT WITH ',I2,' HARMONICS.'//5X,
     $     ' (1/PI)2K(DK/DE) FIT WITH ',I2,' HARMONICS.'///)
 5    FORMAT (' INTEGRATION STEPS'//)
 6    FORMAT (/' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',6X,
     $     'FERMI VELOCITY',8X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,
     $     'IN PHI',28X,'(K COMPONENT)'/)
 7    FORMAT (//' THETA INTEGRATION --',I4,' STEPS',
     $     '   PHI INTEGRATION --',I4,' STEPS.'//)
 8    FORMAT (I7,4X,I7,F15.6,F12.6,F15.8,5X,F15.8)
 9    FORMAT (///15X,' DENSITY OF STATES (IN 1/(BOHR RADII)**3-RY) IS ',
     $     1PD15.8,'.')
 11   FORMAT (F10.6)
 12   FORMAT (///15X,' DENSITY OF STATES (IN 1/ATOM-RY) IS ',
     $     1PD15.8,'.')
 13   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-EV) IS ',
```

```
     $      1PD15.8,'.')
14   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-ERG) IS ',
     $      1PD15.8,'.')
      DO 10 I = 1,91
      C(I) = 0.0
10    D(I) = 0.0
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA,ALPHA
      EA1 = 0.0
      EA2 = 0.0
      EA3 = 0.0
      READ (5,2) LC
      READ (5,1) (IN(I),C(I),I=1,LC)
      WRITE(6,1) (IN(I),C(I),I=1,LC)
      WRITE(6,3) GAMMA,ALPHA
      READ (5,3) GAMMB,ALPHB
      EB1 = 0.0
      EB2 = 0.0
      EB3 = 0.0
      READ (5,2) LD
      READ (5,1) (IN(I),D(I),I=1,LD)
      WRITE(6,1) (IN(I),D(I),I=1,LD)
      WRITE(6,3) GAMMB,ALPHB
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,8)
      PI = 3.141592653589793
      READ (5,2) N,M
      READ (5,3) VOL
      JJJJ = 0
      IIII = 0
      IF (EA1 .NE. 0) JJJJ = 1
      IF (EA2 .NE. 0) JJJJ = 1
      IF (EA3 .NE. 0) JJJJ = 1
      IF (EB1 .NE. 0) IIII = 1
      IF (EB2 .NE. 0) IIII = 1
      IF (EB3 .NE. 0) IIII = 1
      AA2 = ALPHA * ALPHA
      AB2 = ALPHB * ALPHB
      GA2 = GAMMA * GAMMA
      GB2 = GAMMB * GAMMB
      CALL COMSET
      DEN = 0.0
      WRITE (6,4) LC,LD
      WRITE (6,7) N,M
      WRITE (6,5)
      WRITE (6,6)
      DP = PI/(2.00D00 * (M-1))
      DT = PI/(2.00D00 * (N-1))
      DO 20 J = 1,M
      BE = 0.0
      PP = (J-1) * DP
      PHI = PP * 180.00D00/PI
      DO 21 K = 1,N
      TT = (K-1) * DT
      THETA = TT * 180.00D00/PI
      TA = TT
      TB = TT
      PA = PP
      PB = PP
```

```
      IF (GAMMA .EQ. 1.0) GO TO 40
      IF (JJJJ .NE. 0) CALL OILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AA2*CP2)/GA2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      COEFA = ST2*(SP2+AA2*CP2) + GA2*CT2
      GO TO 41
 40   COEFA = 1.0D0
      T = TT
      P = PP
 41   R2 = 0.0
      CALL VHPOLY(T,P,B)
      DO 30 I = 1,KE
 30   R2 = R2 + C(I)*B(I)
      R2 = R2/COEFA
      IF (GAMMB .EQ. 1.0) GO TO 50
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP=DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AB2*CP2)/GB2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP=SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2) + GB2*CT2
      GO TO 51
 50   COEFB = 1.0D0
      T = TT
      P = PP
51    ADKDE=0.0
      CALL VHPOLY(T,P,B)
      DO 52 I = 1,KE
52    ADKDE=ADKDE+D(I)*B(I)
      ADKDE=ADKDE*PI/COEFB
      R = DSQRT(R2)
      VK=R/ADKDE
      VK = VK + VK
      DS = R2 * DSIN(TT)/VK
      WRITE (6,8) K,J,THETA,PHI,VK,R
      IF (MOD(K,2) .EQ. 1) BDS = DS + DS
```

```
      IF (MOD(K,2) .EQ. 0) BDS = DS + DS + DS + DS
      IF (K .EQ. 1) BDS = DS
      IF (K .EQ. N) BDS = DS
      BE = BE + BDS
 21   CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J.EQ.M) BDEN=BE
      DEN=DEN+BDEN
 20   CONTINUE
      DEN = DEN * 8.00D00 * DT * DP/9.00D00
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,9) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      ADEN = DEN
      VOL = VOL * 1.00D-24
      ADEN = ADEN * VOL
      WRITE (6,12) ADEN
      DEN = DEN/13.60486
      WRITE (6,13) DEN
      DEN = DEN * 1.60206D-12
      WRITE (6,14) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J149                                       *
C *                                                                     *
C *   COMPUTATION OF THE DENSITY OF STATES                              *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:  C2H                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 A2,G2,E1,E2,E3
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),IN(91),B(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARPY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,10X,' DENSITY OF STATES CALCULATION'//5X,
     $      ' K**2 FIT WITH ',I2,' HARMONICS.'//5X,
     $      ' (1/PI)2K(DK/DE) FIT WITH ',I2,' HARMONICS.'///)
 5    FORMAT (' INTEGRATION STEPS'//)
 6    FORMAT (/' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',6X,
```

```
     $      'FERMI VELOCITY',8X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,
     $      'IN PHI',28X,'(K COMPONENT)'/)
7    FORMAT (//' THETA INTEGRATION --',I4,' STEPS',
     $      '   PHI INTEGRATION --',I4,' STEPS.'//)
8    FORMAT (I7,4X,I7,F15.6,F12.6,F15.8,5X,F15.8)
9    FORMAT (///15X,' DENSITY OF STATES (IN 1/(BOHR RADII)**3-RY) IS ',
     $      1PD15.8,'.')
11   FORMAT (F10.6)
12   FORMAT (///15X,' DENSITY OF STATES (IN 1/ATOM-RY) IS ',
     $      1PD15.8,'.')
13   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-EV) IS ',
     $      1PD15.8,'.')
14   FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-ERG) IS ',
     $      1PD15.8,'.')
     DO 10 I = 1,91
     C(I) = 0.0
10   D(I) = 0.0
     NUT = 0
     LUMP = 1
     READ (5,3) GAMMA,ALPHA,EA2
     EA1 = 0.0
     EA3 = 0.0
     READ (5,2) LC
     READ (5,1) (IN(I),C(I),I=1,LC)
     WRITE(6,1) (IN(I),C(I),I=1,LC)
     WRITE(6,3) GAMMA,ALPHA,EA2
     READ (5,3) GAMMB,ALPHB,EB2
     EB1 = 0.0
     EB3 = 0.0
     READ (5,2) LD
     READ (5,1) (IN(I),D(I),I=1,LD)
     WRITE(6,1) (IN(I),D(I),I=1,LD)
     WRITE(6,3) GAMMB,ALPHB,EB2
     KE = MAX0(LC,LD)
     KLN = MTOL(KE,9)
     PI = 3.141592653589793
     EA1=EA1*PI/180.0D0
     EA2=EA2*PI/180.0D0
     EA3=EA3*PI/180.0D0
     EB1=EB1*PI/180.0D0
     EB2=EB2*PI/180.0D0
     EB3=EB3*PI/180.0D0
     READ (5,2) N,M
     READ (5,3) VOL
     JJJJ = 0
     IIII = 0
     IF (EA1 .NE. 0) JJJJ = 1
     IF (EA2 .NE. 0) JJJJ = 1
     IF (EA3 .NE. 0) JJJJ = 1
     IF (EB1 .NE. 0) IIII = 1
     IF (EB2 .NE. 0) IIII = 1
     IF (EB3 .NE. 0) IIII = 1
     AA2 = ALPHA * ALPHA
     AB2 = ALPHB * ALPHB
     GA2 = GAMMA * GAMMA
     GB2 = GAMMB * GAMMB
     CALL COMSET
     DEN = 0.0
     WRITE (6,4) LC,LD
     WRITE (6,7) N,M
```

```
      WRITE (6,5)
      WRITE (6,6)
      DP = PI/(M-1)
      DT = PI/(2.00D00 * (N-1))
      DO 20 J = 1,M
      BE = 0.0
      PP = (J-1) * DP
      PHI = PP * 180.00D00/PI
      DO 21 K = 1,N
      TT = (K-1) * DT
      THETA = TT * 180.00D00/PI
      TA = TT
      TB = TT
      PA = PP
      PB = PP
      IF (GAMMA .EQ. 1.0) GO TO 40
      IF (JJJJ .NE. 0) CALL OILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AA2*CP2)/GA2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      COEFA = ST2*(SP2+AA2*CP2) + GA2*CT2
      GO TO 41
40    COEFA = 1.0D0
      T = TT
      P = PP
41    R2 = 0.0
      CALL C2POLY(T,P,B)
      DO 30 I = 1,KE
30    R2 = R2 + C(I)*B(I)
      R2 = R2/COEFA
      IF (GAMMB .EQ. 1.0) GO TO 50
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP=DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AB2*CP2)/GB2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP=SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2) + GB2*CT2
```

```
      GO TO 51
 50   COEFB = 1.CDO
      T = TT
      P = PP
51    ADKDE=0.0
      CALL C2POLY(T,P,B)
      DO 52 I = 1,KE
52    ADKDE=ADKDE+D(I)*B(I)
      ADKDE=ADKDE*PI/COEFB
      R = DSQRT(R2)
      VK=R/ADKDE
      VK = VK + VK
      DS = R2 * DSIN(TT)/VK
      WRITE (6,8) K,J,THETA,PHI,VK,R
      IF (MOD(K,2) .EQ. 1) BDS = DS + DS
      IF (MOD(K,2) .EQ. 0) BDS = DS + DS + DS + DS
      IF (K .EQ. 1) BDS = DS
      IF (K .EQ. N) BDS = DS
      BE = BE + BDS
 21   CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J.EQ.M) BDEN=BE
      DEN=DEN+BDEN
 20   CONTINUE
      DEN = DEN * 4.00D00 * DT * DP/9.00D00
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,9) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      ADEN = DEN
      VOL = VOL * 1.00D-24
      ADEN = ADEN * VOL
      WRITE (6,12) ADEN
      DEN = DEN/13.60486
      WRITE (6,13) DEN
      DEN = DEN * 1.60206D-12
      WRITE (6,14) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J1410                                      *
C *                                                                     *
C *   COMPUTATION OF THE DENSITY OF STATES                              *
C *                                                                     *
C *   INPUT DATA:                                                       *
C *        1)  COEFFICIENTS C(I) FOR K**2                               *
C *        2)  COEFFICIENTS D(I) FOR (1/PI)2K(DK/DE)                    *
C *                                                                     *
C *   SYMMETRY GROUP:   S2                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
```

```
      REAL*4 A2,G2,E1,E2,E3
      REAL*4 T,P,SYM
      REAL*4 ALPHA,AA2,GAMMA,GA2,EA1,EA2,EA3
      REAL*4 ALPHB,AB2,GAMMB,GB2,EB1,EB2,EB3
      DIMENSION NAM(91),SYM(91,16),C(91),D(91),IN(91),B(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
1     FORMAT (10X,I4,E17.6)
2     FORMAT (I4)
3     FORMAT (F10.4)
4     FORMAT (1H1,10X,' DENSITY OF STATES CALCULATION'//5X,
     $      ' K**2 FIT WITH ',I2,' HARMONICS.'//5X,
     $      ' (1/PI)2K(DK/DE) FIT WITH ',I2,' HARMONICS.'///)
5     FORMAT (' INTEGRATION STEPS'//)
6     FORMAT (/' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',6X,
     $      'FERMI VELOCITY',8X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,
     $      'IN PHI',28X,'(K COMPONENT)'/)
7     FORMAT (//' THETA INTEGRATION --',I4,' STEPS',
     $      '   PHI INTEGRATION --',I4,' STEPS.'//)
8     FORMAT (I7,4X,I7,F15.6,F12.6,F15.8,5X,F15.8)
9     FORMAT (///15X,' DENSITY OF STATES (IN 1/(BOHR RADII)**3-RY) IS ',
     $      1PD15.8,'.')
11    FORMAT (F10.6)
12    FORMAT (///15X,' DENSITY OF STATES (IN 1/ATOM-RY) IS ',
     $      1PD15.8,'.')
13    FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-EV) IS ',
     $      1PD15.8,'.')
14    FORMAT (///15X,' DENSITY OF STATES (IN 1/CC-ERG) IS ',
     $      1PD15.8,'.')
      DO 10 I = 1,91
      C(I) = 0.0
10    D(I) = 0.0
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA,ALPHA,EA1,EA2,EA3
      READ (5,2) LC
      READ (5,1) (IN(I),C(I),I=1,LC)
      WRITE(6,1) (IN(I),C(I),I=1,LC)
      WRITE(6,3) GAMMA,ALPHA,EA1,EA2,EA3
      READ (5,3) GAMMB,ALPHB,EB1,EB2,EB3
      READ (5,2) LD
      READ (5,1) (IN(I),D(I),I=1,LD)
      WRITE(6,1) (IN(I),D(I),I=1,LD)
      WRITE(6,3) GAMMB,ALPHB,EB1,EB2,EB3
      KE = MAX0(LC,LD)
      KLN = MTOL(KE,10)
      PI = 3.14159265358979 3
      EA1=EA1*PI/180.0D0
      EA2=EA2*PI/180.0D0
      EA3=EA3*PI/180.0D0
      EB1=EB1*PI/180.0D0
      EB2=EB2*PI/180.0D0
      EB3=EB3*PI/180.0D0
      READ (5,2) N,M
      READ (5,3) VOL
      JJJJ = 0
      IIII = 0
      IF (EA1 .NE. 0) JJJJ = 1
      IF (EA2 .NE. 0) JJJJ = 1
      IF (EA3 .NE. 0) JJJJ = 1
```

```
      IF (EB1 .NE. 0) IIII = 1
      IF (EB2 .NE. 0) IIII = 1
      IF (EB3 .NE. 0) IIII = 1
      AA2 = ALPHA * ALPHA
      AB2 = ALPHB * ALPHB
      GA2 = GAMMA * GAMMA
      GB2 = GAMMB * GAMMB
      CALL COMSET
      DEN = 0.0
      WRITE (6,4) LC,LD
      WRITE (6,7) N,M
      WRITE (6,5)
      WRITE (6,6)
      DP = 2.0D00 * PI/(M-1)
      DT = PI/(2.00D00 * (N-1))
      DO 20 J = 1,M
      BE = 0.0
      PP = (J-1) * DP
      PHI = PP * 180.00D00/PI
      DO 21 K = 1,N
      TT = (K-1) * DT
      THETA = TT * 180.00D00/PI
      TA = TT
      TB = TT
      PA = PP
      PB = PP
      IF (GAMMA .EQ. 1.0) GO TO 40
      IF (JJJJ .NE. 0) CALL OILER(EA1,EA2,EA3,TA,PA,1)
      ST = DSIN(TA)
      CT = DCOS(TA)
      SP = DSIN(PA)
      CP = DCOS(PA)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AA2*CP2)/GA2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TA = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PA)
      T = TA
      P = PA
      COEFA = ST2*(SP2+AA2*CP2) + GA2*CT2
      GO TO 41
40    COEFA = 1.0D0
      T = TT
      P = PP
41    R2 = 0.0
      CALL S2POLY(T,P,B)
      DO 30 I = 1,KF
30    R2 = R2 + C(I)*B(I)
      R2 = R2/COEFA
      IF (GAMMB .EQ. 1.0) GO TO 50
      IF (IIII .NE. 0) CALL OILER(EB1,EB2,EB3,TB,PB,1)
      ST = DSIN(TB)
      CT = DCOS(TB)
      SP=DSIN(PB)
      CP = DCOS(PB)
      ST2 = ST * ST
```

```
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+AB2*CP2)/GB2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TB = DARCOS(TEMP)
      SPP=SP/ALPHB
      CALL DFLOP(SPP,CP,PB)
      T = TB
      P = PB
      COEFB = ST2*(SP2+AB2*CP2) + GB2*CT2
      GO TO 51
 50   COEFB = 1.0D0
      T = TT
      P = PP
51    ADKDE=0.0
      CALL S2POLY(T,P,B)
      DO 52 I = 1,KE
52    ADKDE=ADKDE+D(I)*B(I)
      ADKDE=ADKDE*PI/COEFB
      R = DSQRT(R2)
      VK=R/ADKDE
      VK = VK + VK
      DS = R2 * DSIN(TT)/VK
      WRITE (6,8) K,J,THETA,PHI,VK,R
      IF (MOD(K,2) .EQ. 1) BDS = DS + DS
      IF (MOD(K,2) .EQ. 0) BDS = DS + DS + DS + DS
      IF (K .EQ. 1) BDS = DS
      IF (K .EQ. N) BDS = DS
      BE = BE + BDS
 21   CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J.EQ.M) BDEN=BE
      DEN=DEN+BDEN
 20   CONTINUE
      DEN = DEN*2.0D0*DT*DP/9.00D0
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,9) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      ADEN = DEN
      VOL = VOL * 1.00D-24
      ADEN = ADEN * VOL
      WRITE (6,12) ADEN
      DEN = DEN/13.60486
      WRITE (6,13) DEN
      DEN = DEN * 1.60206D-12
      WRITE (6,14) DEN
      STOP
      END
```

FERMI VOLUME PROGRAMS

```
C ***********************************************************************
C *                                                                     *
C *                          J151                                       *
C *                                                                     *
C *   COMPUTATION OF THE VOLUME OF THE FERMI SURFACE.                   *
C *                                                                     *
C *   INPUT DATA:  THE C(I) ARE THE RADIUS**2 COEFFICIENTS.             *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      DIMENSION NAM(91),SYM(91,16),C(91),IN(91),B(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (15X,I5/4F20.15/4F20.15/4F20.15/4F20.15)
 4    FORMAT (1H1,20X,' VOLUME OF THE FERMI SURFACE'//
     $     10X,' K**2 FIT WITH ',I2,' HARMONICS.'///' THETA',
     $     ' INTEGRATION -- ',I4,' STEPS.  PHI INTEGRATION --',
     $     I4,' STEPS.'///30X,' INTEGRATION STEPS'//' ',4X,
     $     'STEP',7X,'STEP',6X,'THETA',8X,'PHI',6X,'FERMI RADIUS'/
     $     ' ',2X,'IN THETA',4X,'IN PHI'/)
 5    FORMAT (I7,4X,I7,F15.6,F12.6,F14.8)
 6    FORMAT (///15X,' VOLUME OF THE FERMI SURFACE IS ',1PD15.8,
     $     ' (IN BOHR RADII**3).')
 7    FORMAT (F10.6)
 8    FORMAT (///15X,' NUMBER OF STATES/BOHR RADIUS**3  IS ',
     $     1PD15.8,'.')
 9    FORMAT (///15X,' NUMBER OF STATES/CC IS ',1PD15.8,'.')
 10   FORMAT (///15X,' NUMBER OF STATES/ATOM IS ',1PD15.8,'.')
      READ (5,3) (NAM(I),(SYM(I,J),J=1,16),I=1,91)
      READ (5,2) L
      READ (5,1) (IN(I),C(I),I=1,L)
      READ (5,2) N,M
C
C     READ LATTICE CONSTANT AND NUMBER OF ATOMS/UNIT CELL
C
      READ (5,7) A
      READ (5,2) NCELL
      CALL COMSET
      KE = L
      KLN = MTOL(KE,1)
      WRITE (6,4) L,N,M
      DEN = 0.0
      PI = 3.141592653589793
      DP = PI/(4.0D0*(M - 1))
      DT = PI/(2.0D0*(N-1))
      DO 80 J = 1,M
      BE = 0.0
      PP = (J-1) * DP
      PHI = PP * 180.0D0/PI
      P = PP
      DO 51 K = 1,N
      TT = (K-1) * DT
      THETA = TT * 180.0D0/PI
      T = TT
      R2 = 0.0
      CALL KUPOLY(T,P,B)
      DO 30 I = 1,KE
 30   R2 = R2 + C(I)*B(I)
```

```
      R = DSQRT(R2)
      DR = R * R * R * DSIN(TT)
      WRITE (6,5) K,J,THETA,PHI,R
      IF (MOD(K,2) .EQ. 1) BDR = DR + DR
      IF (MOD(K,2) .EQ. 0) BDR = DR + DR + DR + DR
      IF (K .EQ. 1) BDR = DR
      IF (K .EQ. N) BDR = DR
      BE = BE + BDR
51    CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J .EQ. M) BDEN = BE
      DEN = DEN + BDEN
80    CONTINUE
      DEN = DEN * 16.00D0 * DT * DP/27.00D0
      WRITE (6,6) DEN
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN/PI2
      DEN = DEN + DEN
      WRITE (6,8) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      WRITE (6,9) DEN
      A = A * 1.00000000D-08
      ANCELL = NCELL
      DEN = DEN * A * A * A/ANCELL
      WRITE (6,10) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J152                                       *
C *                                                                     *
C *   COMPUTATION OF THE VOLUME OF THE FERMI SURFACE.                   *
C *                                                                     *
C *   INPUT DATA:   THE C(I) ARE THE K**2 COEFFICIENTS.                 *
C *                                                                     *
C *   SYMMETRY GROUP:  D6H                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM,ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16),C(91),B(91),IN(91)
      COMMON/KUBE/NAM,SYM,KF,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,20X,'VOLUME OF THE FERMI SURFACE'//
     $      10X,' K**2 IS FIT WITH ',I2,' HARMONICS.'///' THETA',
     $      ' INTEGRATION --',I4,' STEPS.',
     $      '   PHI INTEGRATION --',I4,' STEPS.'///30X,' INTEGRATION',
     $      ' STEPS'//' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',
     $      6X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,'IN PHI'/)
 5    FORMAT (I7,4X,I7,F15.6,F12.6,F14.8)
 6    FORMAT (///15X,' VOLUME OF THE FERMI SURFACE ',
```

```
     $      '(IN BOHR RADII**3) IS ',1PD15.8,'.')
 7    FORMAT (F10.6)
 8    FORMAT (///15X,' NUMBER OF STATES/BOHR RADIUS**3 IS ',
     $      1PD15.8,'.')
 9    FORMAT (///15X,' NUMBER OF STATES/CC IS ',1PD15.8,'.')
 10   FORMAT (///15X,' NUMBER OF STATES/ATOM IS ',1PD15.8,'.')
C
C     READ INPUT DATA
C
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA
      ALPHA = 1.0
      E1 = 0.0
      E2 = 0.0
      E3 = 0.0
      READ (5,2) L
      READ (5,1) (IN(I),C(I),I=1,L)
      WRITE(6,1) (IN(I),C(I),I=1,L)
      WRITE(6,3)  GAMMA
      READ (5,2) N,M
      READ (5,3) VOL
      CALL COMSET
      KE = L
      KLN = MTOL(KE,2)
      A2 = ALPHA * ALPHA
      G2 = GAMMA * GAMMA
      JJJJ = 0
      IF (E1 .NE. 0) JJJJ = 1
      IF (E2 .NE. 0) JJJJ = 1
      IF (E3 .NE. 0) JJJJ = 1
      WRITE (6,4) L,N,M
      DEN = 0.0
      PI = 3.141592653589793
      DP = PI/(6.00D00 * (M-1))
      DT = PI/(2.00D00 * (N-1))
      DO 80 J = 1,M
      BE = 0.0
      DO 51 K = 1,N
      PP = (J-1) * DP
      PHI = PP * 180.0D0/PI
      TT = (K-1) * DT
      THETA = TT * 180.00D00/PI
      TSAVE = TT
      IF (JJJJ .NE. 0) CALL OILER(E1,E2,E3,TT,PP,1)
      IF (GAMMA .EQ. 1.0) GO TO 40
      ST = DSIN(TT)
      CT = DCOS(TT)
      SP = DSIN(PP)
      CP = DCOS(PP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+A2*CP2)/G2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TT = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PP)
      T = TT
```

```
      P = PP
      COEF = ST2 * (SP2 + A2*CP2) + G2 * CT2
      GO TO 41
40    COEF = 1.0000000000000000D00
      T = TT
      P = PP
41    R2 = 0.0
      CALL HXPOLY(T,P,B)
      DO 30 I = 1,KF
30    R2 = R2 + C(I)*B(I)
      R2 = R2/COEF
      R = DSQRT(R2)
      DR = R * R * R * DSIN(TSAVE)
      WRITE (6,5) K,J,THETA,PHI,R
      IF (MOD(K,2) .EQ. 1) BDR = DR + DR
      IF (MOD(K,2) .EQ. 0) BDR = DR + DR + DR + DR
      IF (K .EQ. 1) BDR = DR
      IF (K .EQ. N) BDR = DR
      BE = BE + BDR
51    CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J .EQ. M) BDEN = BE
      DEN = DEN + BDEN
80    CONTINUE
      DEN = DEN * 24.00D00 * DT * DP/27.00D00
      WRITE (6,6) DEN
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,8) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      WRITE (6,9) DEN
      VOL = VOL * 1.00D-24
      DEN = DEN * VOL
      WRITE (6,10) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J153                                       *
C *                                                                     *
C *   COMPUTATION OF THE VOLUME OF THE FERMI SURFACE.                   *
C *                                                                     *
C *   INPUT DATA:   THE C(I) ARE THE K**2 COEFFICIENTS.                 *
C *                                                                     *
C *   SYMMETRY GROUP:  C6H                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM,ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16),C(91),B(91),IN(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
```

```
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,20X,'VOLUME OF THE FERMI SURFACE'//
     $     10X,' K**2 IS FIT WITH ',I2,' HARMONICS.'///' THETA',
     $     ' INTEGRATION --',I4,' STEPS.',
     $     '   PHI INTEGRATION --',I4,' STEPS.'///30X,' INTEGRATION',
     $     ' STEPS'//' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',
     $     6X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,'IN PHI'/)
 5    FORMAT (I7,4X,I7,F15.6,F12.6,F14.8)
 6    FORMAT (///15X,' VOLUME OF THE FERMI SURFACE ',
     $     '(IN BOHR RADII**3) IS ',1PD15.8,'.')
 7    FORMAT (F10.6)
 8    FORMAT (///15X,' NUMBER OF STATES/BOHR RADIUS**3 IS ',
     $     1PD15.8,'.')
 9    FORMAT (///15X,' NUMBER OF STATES/CC IS ',1PD15.8,'.')
 10   FORMAT (///15X,' NUMBER OF STATES/ATOM IS ',1PD15.8,'.')
C
C     READ INPUT DATA
C
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA
      ALPHA = 1.0
      E1 = 0.0
      E2 = 0.00
      E3 = 0.0
      READ (5,2) L
      READ (5,1) (IN(I),C(I),I=1,L)
      WRITE(6,1) (IN(I),C(I),I=1,L)
      WRITE(6,3)  GAMMA
      READ (5,2) N,M
      READ (5,3) VOL
      CALL COMSET
      KE = L
      KLN = MTOL(KE,10)
      A2 = ALPHA * ALPHA
      G2 = GAMMA * GAMMA
      JJJJ = 0
      IF (E1 .NE. 0) JJJJ = 1
      IF (E2 .NE. 0) JJJJ = 1
      IF (E3 .NE. 0) JJJJ = 1
      WRITE (6,4) L,N,M
      DEN = 0.0
      PI = 3.141592653589793
      DP = PI/(3.00D00 * (M-1))
      DT = PI/(2.00D00 * (N-1))
      DO 80 J = 1,M
      BE = 0.0
      DO 51 K = 1,N
      PP = (J-1) * DP
      PHI = PP * 180.0D0/PI
      TT = (K-1) * DT
      THETA = TT * 180.00D00/PI
      IF (GAMMA .EQ. 1.0) GO TO 40
      TSAVE = TT
      IF (JJJJ .NE. 0) CALL CILER(E1,E2,E3,TT,PP,1)
      ST = DSIN(TT)
      CT = DCOS(TT)
      SP = DSIN(PP)
      CP = DCOS(PP)
```

```
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+A2*CP2)/G2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TT = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PP)
      T = TT
      P = PP
      COEF = ST2 * (SP2 + A2*CP2) + G2 * CT2
      GO TO 41
40    COEF = 1.000000000000000D00
      T = TT
      P = PP
41    R2 = 0.0
      CALL C6POLY(T,P,B)
      DO 30 I = 1,KE
30    R2 = R2 + C(I)*B(I)
      R2 = R2/COEF
      R = DSQRT(R2)
      DR = R * R * R * DSIN(TSAVE)
      WRITE (6,5) K,J,THETA,PHI,R
      IF (MOD(K,2) .EQ. 1) BDR = DR + DR
      IF (MOD(K,2) .EQ. 0) BDR = DR + DR + DR + DR
      IF (K .EQ. 1) BDR = DR
      IF (K .EQ. N) BDR = DR
      BE = BE + BDR
51    CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J .EQ. M) BDEN = BE
      DEN = DEN + BDEN
80    CONTINUE
      DEN = DEN * 12.00D00 * DT * DP/27.00D00
      WRITE (6,6) DEN
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,8) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      WRITE (6,9) DEN
      VOL = VOL * 1.00D-24
      DEN = DEN * VOL
      WRITE (6,10) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                              J154                                   *
C *                                                                     *
C *   COMPUTATION OF THE VOLUME OF THE FERMI SURFACE.                   *
C *                                                                     *
C *   INPUT DATA:   THE C(I) ARE THE K**2 COEFFICIENTS.                 *
```

```
C *                                                                    *
C *   SYMMETRY GROUP:  D3D                                             *
C *                                                                    *
C ********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM,ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16),C(91),B(91),IN(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,20X,'VOLUME OF THE FERMI SURFACE'//
     $     10X,' K**2 IS FIT WITH ',I2,' HARMONICS.'///' THETA',
     $     ' INTEGRATION --',I4,' STEPS.',
     $     '   PHI INTEGRATION --',I4,' STEPS.'///30X,' INTEGRATION',
     $     ' STEPS'//' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',
     $     6X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,'IN PHI'/)
 5    FORMAT (I7,4X,I7,F15.6,F12.6,F14.8)
 6    FORMAT (///15X,' VOLUME OF THE FERMI SURFACE ',
     $     '(IN BOHR RADII**3) IS ',1PD15.8,'.')
 7    FORMAT (F10.6)
 8    FORMAT (///15X,' NUMBER OF STATES/BOHR RADIUS**3 IS ',
     $     1PD15.8,'.')
 9    FORMAT (///15X,' NUMBER OF STATES/CC IS ',1PD15.8,'.')
 10   FORMAT (///15X,' NUMBER OF STATES/ATOM IS ',1PD15.8,'.')
C
C     READ INPUT DATA
C
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA
      ALPHA = 1.0
      E1 = 0.0
      E2 = 0.0
      E3 = 0.0
      READ (5,2) L
      READ (5,1) (IN(I),C(I),I=1,L)
      WRITE(6,1) (IN(I),C(I),I=1,L)
      WRITE(6,3)  GAMMA
      READ (5,2) N,M
      READ (5,3) VOL
      CALL COMSET
      KE = L
      KLN = MTOL(KE,4)
      A2 = ALPHA * ALPHA
      G2 = GAMMA * GAMMA
      JJJJ = 0
      IF (E1 .NE. 0) JJJJ = 1
      IF (E2 .NE. 0) JJJJ = 1
      IF (E3 .NE. 0) JJJJ = 1
      WRITE (6,4) L,N,M
      DEN = 0.0
      PI = 3.141592653589793
      DP = PI/(3.00D00 * (M-1))
      DT = PI/(2.00D00 * (N-1))
      DO 80 J = 1,M
      BE = 0.0
      DO 51 K = 1,N
      PP = (J-1) * DP
```

```
      PHI = PP * 180.0D0/PI
      TT = (K-1) * DT
      THETA = TT * 180.00D00/PI
      IF (GAMMA .EQ. 1.0) GO TO 40
      TSAVE = TT
      IF (JJJJ .NE. 0) CALL OILER(E1,E2,E3,TT,PP,1)
      ST = DSIN(TT)
      CT = DCOS(TT)
      SP = DSIN(PP)
      CP = DCOS(PP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+A2*CP2)/G2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TT = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PP)
      T = TT
      P = PP
      COEF = ST2 * (SP2 + A2*CP2) + G2 * CT2
      GO TO 41
40    COEF = 1.0000000000000000D00
      T = TT
      P = PP
41    R2 = 0.0
      CALL D3POLY(T,P,B)
      DO 30 I = 1,KE
30    R2 = R2 + C(I)*B(I)
      R2 = R2/COFF
      R = DSQRT(R2)
      DR = R * R * R * DSIN(TSAVE)
      WRITE (6,5) K,J,THETA,PHI,R
      IF (MOD(K,2) .EQ. 1) BDR = DR + DR
      IF (MOD(K,2) .EQ. 0) BDR = DR + DR + DR + DR
      IF (K .EQ. 1) BDR = DR
      IF (K .EQ. N) BDR = DR
      BE = BE + BDR
51    CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J .EQ. M) BDEN = BE
      DEN = DEN + BDEN
80    CONTINUE
      DEN = DEN * 12.00D00 * DT * DP/27.00D00
      WRITE (6,6) DEN
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,8) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      WRITE (6,9) DEN
      VOL = VOL * 1.00D-24
      DEN = DEN * VOL
      WRITE (6,10) DEN
      STOP
```

```
      END

C ***********************************************************************
C *                                                                     *
C *                          J155                                       *
C *                                                                     *
C *   COMPUTATION OF THE VOLUME OF THE FERMI SURFACE.                   *
C *                                                                     *
C *   INPUT DATA:   THE C(I) ARE THE K**2 COEFFICIENTS.                 *
C *                                                                     *
C *   SYMMETRY GROUP:  S6  (C3I)                                        *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM,ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16),C(91),B(91),IN(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,20X,'VOLUME OF THE FERMI SURFACE'//
     $      10X,' K**2 IS FIT WITH ',I2,' HARMONICS.'///' THETA',
     $      ' INTEGRATION --',I4,' STEPS.',
     $      '   PHI INTEGRATION --',I4,' STEPS.'///30X,' INTEGRATION',
     $      ' STEPS'//' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',
     $      6X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,'IN PHI'/)
 5    FORMAT (I7,4X,I7,F15.6,F12.6,F14.8)
 6    FORMAT (///15X,' VOLUME OF THE FERMI SURFACE ',
     $      '(IN BOHR RADII**3) IS ',1PD15.8,'.')
 7    FORMAT (F10.6)
 8    FORMAT (///15X,' NUMBER OF STATES/BOHR RADIUS**3 IS ',
     $      1PD15.8,'.')
 9    FORMAT (///15X,' NUMBER OF STATES/CC IS ',1PD15.8,'.')
 10   FORMAT (///15X,' NUMBER OF STATES/ATOM IS ',1PD15.8,'.')
C
C     READ INPUT DATA
C
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA
      ALPHA = 1.0
      E1 = 0.0
      E2 = 0.0
      E3 = 0.0
      READ (5,2) L
      READ (5,1) (IN(I),C(I),I=1,L)
      WRITE(6,1) (IN(I),C(I),I=1,L)
      WRITE(6,3)  GAMMA
      READ (5,2) N,M
      READ (5,3) VOL
      CALL COMSET
      KE = L
      KLN = MTOL(KE,5)
      A2 = ALPHA * ALPHA
      G2 = GAMMA * GAMMA
      JJJJ = 0
      IF (E1 .NE. 0) JJJJ = 1
      IF (E2 .NE. 0) JJJJ = 1
```

```
      IF (E3 .NE. 0) JJJJ = 1
      WRITE (6,4) L,N,M
      DEN = 0.0
      PI = 3.141592653589793
      DP = 2.00D00 * PI/(3.00D00 * (M-1))
      DT = PI/(2.00D00 * (N-1))
      DO 80 J = 1,M
      BE = 0.0
      DO 51 K = 1,N
      PP = (J-1) * DP
      PHI = PP * 180.0D0/PI
      TT = (K-1) * DT
      THETA = TT * 180.00D00/PI
      IF (GAMMA .EQ. 1.0) GO TO 40
      TSAVE = TT
      IF (JJJJ .NE. 0) CALL CILER(E1,E2,E3,TT,PP,1)
      ST = DSIN(TT)
      CT = DCOS(TT)
      SP = DSIN(PP)
      CP = DCOS(PP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+A2*CP2)/G2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TT = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PP)
      T = TT
      P = PP
      COEF = ST2 * (SP2 + A2*CP2) + G2 * CT2
      GO TO 41
40    COEF = 1.000000000000000D00
      T = TT
      P = PP
41    R2 = 0.0
      CALL S6POLY(T,P,B)
      DO 30 I = 1,KE
30    R2 = R2 + C(I)*B(I)
      R2 = R2/COEF
      R = DSQRT(R2)
      DR = R * R * R * DSIN(TSAVE)
      WRITE (6,5) K,J,THETA,PHI,R
      IF (MOD(K,2) .EQ. 1) BDR = DR + DR
      IF (MOD(K,2) .EQ. 0) BDR = DR + DR + DR + DR
      IF (K .EQ. 1) BDR = DR
      IF (K .EQ. N) BDR = DR
      BE = BE + BDR
51    CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J .EQ. M) BDEN = BE
      DEN = DEN + BDEN
80    CONTINUE
      DEN = DEN * 6.00D00 * DT * DP/27.00D00
      WRITE (6,6) DEN
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
```

```
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,8) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      WRITE (6,9) DEN
      VOL = VOL * 1.00D-24
      DEN = DEN * VOL
      WRITE (6,10) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J156                                       *
C *                                                                     *
C *   COMPUTATION OF THE VOLUME OF THE FERMI SURFACE.                   *
C *                                                                     *
C *   INPUT DATA:   THE C(I) ARE THE K**2 COEFFICIENTS.                 *
C *                                                                     *
C *   SYMMETRY GROUP:  D4H                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM,ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16),C(91),B(91),IN(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,20X,'VOLUME OF THE FERMI SURFACE'//
     $     10X,' K**2 IS FIT WITH ',I2,' HARMONICS.'///' THETA',
     $     ' INTEGRATION --',I4,' STEPS.',
     $     '   PHI INTEGRATION --',I4,' STEPS.'///30X,' INTEGRATION',
     $     ' STEPS'//' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',
     $     6X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,'IN PHI'/)
 5    FORMAT (I7,4X,I7,F15.6,F12.6,F14.8)
 6    FORMAT (///15X,' VOLUME OF THE FERMI SURFACE ',
     $     '(IN BOHR RADII**3) IS ',1PD15.8,'.')
 7    FORMAT (F10.6)
 8    FORMAT (///15X,' NUMBER OF STATES/BOHR RADIUS**3 IS ',
     $     1PD15.8,'.')
 9    FORMAT (///15X,' NUMBER OF STATES/CC IS ',1PD15.8,'.')
 10   FORMAT (///15X,' NUMBER OF STATES/ATOM IS ',1PD15.8,'.')
C
C     READ INPUT DATA
C
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA
      ALPHA = 1.0
      E1 = 0.0
      E2 = 0.0
      E3 = 0.0
      READ (5,2) L
      READ (5,1) (IN(I),C(I),I=1,L)
      WRITE(6,1) (IN(I),C(I),I=1,L)
      WRITE(6,3)  GAMMA
```

```
      READ (5,2) N,M
      READ (5,3) VOL
      CALL COMSET
      KE = L
      KLN = MTOL(KE,6)
      A2 = ALPHA * ALPHA
      G2 = GAMMA * GAMMA
      JJJJ = 0
      IF (E1 .NE. 0) JJJJ = 1
      IF (E2 .NE. 0) JJJJ = 1
      IF (E3 .NE. 0) JJJJ = 1
      WRITE (6,4) L,N,M
      DEN = 0.0
      PI = 3.141592653589793
      DP = PI/(4.0D00 * (M-1))
      DT = PI/(2.0D00 * (N-1))
      DO 80 J = 1,M
      BE = 0.0
      DO 51 K = 1,N
      PP = (J-1) * DP
      PHI = PP * 180.0D0/PI
      TT = (K-1) * DT
      THETA = TT * 180.0D00/PI
      IF (GAMMA .EQ. 1.0) GO TO 40
      TSAVE = TT
      IF (JJJJ .NE. 0) CALL CILER(E1,E2,E3,TT,PP,1)
      ST = DSIN(TT)
      CT = DCOS(TT)
      SP = DSIN(PP)
      CP = DCOS(PP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+A2*CP2)/G2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TT = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PP)
      T = TT
      P = PP
      COEF = ST2 * (SP2 + A2*CP2) + G2 * CT2
      GO TO 41
40    COEF = 1.0C0000000000000D00
      T = TT
      P = PP
41    R2 = 0.0
      CALL SOPOLY(T,P,B)
      DO 30 I = 1,KE
30    R2 = R2 + C(I)*B(I)
      R2 = R2/COEF
      R = DSQRT(R2)
      DR = R * R * R * DSIN(TSAVE)
      WRITE (6,5) K,J,THETA,PHI,R
      IF (MOD(K,2) .EQ. 1) BDR = DR + DR
      IF (MOD(K,2) .EQ. 0) BDR = DR + DR + DR + DR
      IF (K .EQ. 1) BDR = DR
      IF (K .EQ. N) BDR = DR
      BE = BE + BDR
51    CONTINUE
```

```
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J .EQ. M) BDEN = BE
      DEN = DEN + BDEN
 80   CONTINUE
      DEN = DEN * 16.00D00 * DT * DP/27.00D00
      WRITE (6,6) DEN
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,8) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      WRITE (6,9) DEN
      VOL = VOL * 1.00D-24
      DEN = DEN * VOL
      WRITE (6,10) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J157                                       *
C *                                                                     *
C *   COMPUTATION OF THE VOLUME OF THE FERMI SURFACE.                   *
C *                                                                     *
C *   INPUT DATA:   THE C(I) ARE THE K**2 COEFFICIENTS.                 *
C *                                                                     *
C *   SYMMETRY GROUP:  C4H                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM,ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16),C(91),B(91),IN(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,20X,'VOLUME OF THE FERMI SURFACE'//
     $     10X,' K**2 IS FIT WITH ',I2,' HARMONICS.'///' THETA',
     $     ' INTEGRATION --',I4,' STEPS.',
     $     '   PHI INTEGRATION --',I4,' STEPS.'///30X,' INTEGRATION',
     $     ' STEPS'//' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',
     $     6X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,'IN PHI'/)
 5    FORMAT (I7,4X,I7,F15.6,F12.6,F14.8)
 6    FORMAT (///15X,' VOLUME OF THE FERMI SURFACE ',
     $     '(IN BOHR RADII**3) IS ',1PD15.8,'.')
 7    FORMAT (F10.6)
 8    FORMAT (///15X,' NUMBER OF STATES/BOHR RADIUS**3 IS ',
     $     1PD15.8,'.')
 9    FORMAT (///15X,' NUMBER OF STATES/CC IS ',1PD15.8,'.')
 10   FORMAT (///15X,' NUMBER OF STATES/ATOM IS ',1PD15.8,'.')
C
C     READ INPUT DATA
C
      NUT = 0
```

```
      LUMP = 1
      READ (5,3) GAMMA
      ALPHA = 1.0
      E1 = 0.0
      E2 = 0.0
      E3 = 0.0
      READ (5,2) L
      READ (5,1) (IN(I),C(I),I=1,L)
      WRITE(6,1) (IN(I),C(I),I=1,L)
      WRITE(6,3)  GAMMA
      READ (5,2) N,M
      READ (5,3) VOL
      CALL COMSET
      KE = L
      KLN = MTOL(KE,7)
      A2 = ALPHA * ALPHA
      G2 = GAMMA * GAMMA
      JJJJ = 0
      IF (E1 .NE. 0) JJJJ = 1
      IF (E2 .NE. 0) JJJJ = 1
      IF (E3 .NE. 0) JJJJ = 1
      WRITE (6,4) L,N,M
      DEN = 0.0
      PI = 3.141592653589793
      DP = PI/(2.0D00 * (M-1))
      DT = PI/(2.0D00 * (N-1))
      DO 80 J = 1,M
      BE = 0.0
      DO 51 K = 1,N
      PP = (J-1) * DP
      PHI = PP * 180.0D0/PI
      TT = (K-1) * DT
      THETA = TT * 180.0D00/PI
      IF (GAMMA .EQ. 1.0) GO TO 40
      TSAVE = TT
      IF (JJJJ .NE. 0) CALL OILER(E1,E2,E3,TT,PP,1)
      ST = DSIN(TT)
      CT = DCOS(TT)
      SP = DSIN(PP)
      CP = DCOS(PP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+A2*CP2)/G2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TT = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PP)
      T = TT
      P = PP
      COEF = ST2 * (SP2 + A2*CP2) + G2 * CT2
      GO TO 41
40    COEF = 1.0000000000000000D00
      T = TT
      P = PP
41    R2 = 0.0
      CALL C4POLY(T,P,B)
      DO 30 I = 1,KE
30    R2 = R2 + C(I)*B(I)
```

```
      R2 = R2/COEF
      R = DSQRT(R2)
      DR = R * R * R * DSIN(TSAVE)
      WRITE (6,5) K,J,THETA,PHI,R
      IF (MOD(K,2) .EQ. 1) BDR = DR + DR
      IF (MOD(K,2) .EQ. 0) BDR = DR + DR + DR + DR
      IF (K .EQ. 1) BDR = DR
      IF (K .EQ. N) BDR = DR
      BE = BE + BDR
 51   CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J .EQ. M) BDEN = BE
      DEN = DEN + BDEN
 80   CONTINUE
      DEN = DEN * 8.00D00 * DT * DP/27.00D00
      WRITE (6,6) DEN
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,8) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      WRITE (6,9) DEN
      VOL = VOL * 1.00D-24
      DEN = DEN * VOL
      WRITE (6,10) DEN
      STOP
      END


C ***********************************************************************
C *                                                                     *
C *                          J158                                       *
C *                                                                     *
C *   COMPUTATION OF THE VOLUME OF THE FERMI SURFACE.                   *
C *                                                                     *
C *   INPUT DATA:   THE C(I) ARE THF K**2 COEFFICIENTS.                 *
C *                                                                     *
C *   SYMMETRY GROUP:  VH                                               *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM,ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16),C(91),B(91),IN(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,20X,'VOLUME OF THE FERMI SURFACE'//
     $      10X,' K**2 IS FIT WITH ',I2,' HARMONICS.'///' THETA',
     $      ' INTEGRATION --',I4,' STEPS.',
     $      '   PHI INTEGRATION --',I4,' STEPS.'///30X,' INTEGRATION',
     $      ' STEPS'//' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',
     $      6X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,'IN PHI'/)
 5    FORMAT (I7,4X,I7,F15.6,F12.6,F14.8)
 6    FORMAT (///15X,' VOLUME OF THE FERMI SURFACE ',
```

```
     $      '(IN BOHR RADII**3) IS ',1PD15.8,'.')
 7    FORMAT (F10.6)
 8    FORMAT (///15X,' NUMBER OF STATES/BOHR RADIUS**3 IS ',
     $      1PD15.8,'.')
 9    FORMAT (///15X,' NUMBER OF STATES/CC IS ',1PD15.8,'.')
 10   FORMAT (///15X,' NUMBER OF STATES/ATOM IS ',1PD15.8,'.')
C
C     READ INPUT DATA
C
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA,ALPHA
      E1 = 0.0
      E2 = 0.0
      E3 = 0.0
      READ (5,2) L
      READ (5,1) (IN(I),C(I),I=1,L)
      WRITE(6,1) (IN(I),C(I),I=1,L)
      WRITE(6,3)  GAMMA,ALPHA
      READ (5,2) N,M
      READ (5,3) VOL
      CALL COMSET
      KE = L
      KLN = MTOL(KE,8)
      A2 = ALPHA * ALPHA
      G2 = GAMMA * GAMMA
      JJJJ = 0
      IF (E1 .NE. 0) JJJJ = 1
      IF (E2 .NE. 0) JJJJ = 1
      IF (E3 .NE. 0) JJJJ = 1
      WRITE (6,4) L,N,M
      DEN = 0.0
      PI = 3.141592653589793
      DP = PI/(2.0D00 * (M-1))
      DT = PI/(2.0D00 * (N-1))
      DO 80 J = 1,M
      BE = 0.0
      DO 51 K = 1,N
      PP = (J-1) * DP
      PHI = PP * 180.0D0/PI
      TT = (K-1) * DT
      THETA = TT * 180.00D00/PI
      IF (GAMMA .EQ. 1.0) GO TO 40
      TSAVE = TT
      IF (JJJJ .NE. 0) CALL CILER(E1,E2,E3,TT,PP,1)
      ST = DSIN(TT)
      CT = DCOS(TT)
      SP = DSIN(PP)
      CP = DCOS(PP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+A2*CP2)/G2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TT = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PP)
      T = TT
      P = PP
```

```
      COEF = ST2 * (SP2 + A2*CP2) + G2 * CT2
      GO TO 41
 40   COEF = 1.0000000000000000D00
      T = TT
      P = PP
 41   R2 = 0.0
      CALL VHPOLY(T,P,B)
      DO 30 I = 1,KE
 30   R2 = R2 + C(I)*B(I)
      R2 = R2/COEF
      R = DSQRT(R2)
      DR = R * R * R * DSIN(TSAVE)
      WRITE (6,5) K,J,THETA,PHI,R
      IF (MOD(K,2) .EQ. 1) BDR = DR + DR
      IF (MOD(K,2) .EQ. 0) BDR = DR + DR + DR + DR
      IF (K .EQ. 1) BDR = DR
      IF (K .EQ. N) BDR = DR
      BE = BE + BDR
 51   CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J .EQ. M) BDEN = BE
      DEN = DEN + BDEN
 80   CONTINUE
      DEN = DEN * 8.00D00 * DT * DP/27.00D00
      WRITE (6,6) DEN
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,8) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      WRITE (6,9) DEN
      VOL = VOL * 1.00D-24
      DEN = DEN * VOL
      WRITE (6,10) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J159                                       *
C *                                                                     *
C *   COMPUTATION OF THE VOLUME OF THE FERMI SURFACE.                   *
C *                                                                     *
C *   INPUT DATA:   THE C(I) ARE THE K**2 COEFFICIENTS.                 *
C *                                                                     *
C *   SYMMETRY GROUP:  C2H                                              *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM,ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16),C(91),B(91),IN(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
```

```
3     FORMAT (F10.4)
4     FORMAT (1H1,20X,'VOLUME OF THE FERMI SURFACE'//
     $      10X,' K**2 IS FIT WITH ',I2,' HARMONICS.'///' THETA',
     $      ' INTEGRATION --',I4,' STEPS.',
     $      '   PHI INTEGRATION --',I4,' STEPS.'///30X,' INTEGRATION',
     $      ' STEPS'//' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',
     $      6X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,'IN PHI'/)
5     FORMAT (I7,4X,I7,F15.6,F12.6,F14.8)
6     FORMAT (///15X,' VOLUME OF THE FERMI SURFACE ',
     $      '(IN BOHR RADII**3) IS ',1PD15.8,'.')
7     FORMAT (F10.6)
8     FORMAT (///15X,' NUMBER OF STATES/BOHR RADIUS**3 IS ',
     $      1PD15.8,'.')
9     FORMAT (///15X,' NUMBER OF STATES/CC IS ',1PD15.8,'.')
10    FORMAT (///15X,' NUMBER OF STATES/ATOM IS ',1PD15.8,'.')
C
C     READ INPUT DATA
C
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA,ALPHA,E2
      E1 = 0.0
      E3 = 0.0
      READ (5,2) L
      READ (5,1) (IN(I),C(I),I=1,L)
      WRITE(6,1) (IN(I),C(I),I=1,L)
      WRITE(6,3) GAMMA, ALPHA, E2
      READ (5,2) N,M
      READ (5,3) VOL
      CALL COMSET
      KE = L
      KLN = MTOL(KE,09)
      A2 = ALPHA * ALPHA
      G2 = GAMMA * GAMMA
      JJJJ = 0
      IF (E1 .NE. 0) JJJJ = 1
      IF (E2 .NE. 0) JJJJ = 1
      IF (E3 .NE. 0) JJJJ = 1
      WRITE (6,4) L,N,M
      DEN = 0.0
      PI = 3.141592653589793
      E1 = E1 * PI/180.
      E2 = E2 * PI/180.
      E3 = E3 * PI/180.
      DP = PI/(M-1)
      DT = PI/(2.00D00 * (N-1))
      DO 80 J = 1,M
      BE = 0.0
      DO 51 K = 1,N
      PP = (J-1) * DP
      PHI = PP * 180.0D0/PI
      TT = (K-1) * DT
      THETA = TT * 180.00D00/PI
      IF (GAMMA .EQ. 1.0) GO TO 40
      TSAVE = TT
      IF (JJJJ .NE. 0) CALL OILER(E1,E2,E3,TT,PP,1)
      ST = DSIN(TT)
      CT = DCOS(TT)
      SP = DSIN(PP)
      CP = DCOS(PP)
```

```
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+A2*CP2)/G2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TT = DARCOS(TEMP)
      CALL DFLOP(SPP,CP,PP)
      T = TT
      P = PP
      COEF = ST2 * (SP2 + A2*CP2) + G2 * CT2
      GO TO 41
40    COEF = 1.0000000000000000D00
      T = TT
      P = PP
41    P2 = 0.0
      CALL C2POLY(T,P,B)
      DO 30 I = 1,KE
30    R2 = R2 + C(I)*B(I)
      R2 = R2/COEF
      R = DSQRT(R2)
      DR = R * R * R * DSIN(TSAVE)
      WRITE (6,5) K,J,THETA,PHI,R
      IF (MOD(K,2) .EQ. 1) BDR = DR + DR
      IF (MOD(K,2) .EQ. 0) BDR = DR + DR + DR + DR
      IF (K .EQ. 1) BDR = DR
      IF (K .EQ. N) BDR = DR
      BE = BE + BDR
51    CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J .EQ. M) BDEN = BE
      DEN = DEN + BDEN
80    CONTINUE
      DEN = DEN * 4.00D00 * DT * DP/27.00D00
      WRITE (6,6) DEN
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,8) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      WRITE (6,9) DEN
      VOL = VOL * 1.00D-24
      DEN = DEN * VOL
      WRITE (6,10) DEN
      STOP
      END

C ***********************************************************************
C *                                                                     *
C *                          J1510                                      *
C *                                                                     *
C *   COMPUTATION OF THE VOLUME OF THE FERMI SURFACE.                   *
C *                                                                     *
C *   INPUT DATA:   THE C(I) ARE THE K**2 COEFFICIENTS.                 *
C *                                                                     *
```

```
C *    SYMMETRY GROUP:    S2                                                *
C *                                                                         *
C ***************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM,ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION NAM(91),SYM(91,16),C(91),B(91),IN(91)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
 1    FORMAT (10X,I4,E17.6)
 2    FORMAT (I4)
 3    FORMAT (F10.4)
 4    FORMAT (1H1,20X,'VOLUME OF THE FERMI SURFACE'//
     $      10X,' K**2 IS FIT WITH ',I2,' HARMONICS.'///' THETA',
     $      ' INTEGRATION --',I4,' STEPS.',
     $      '   PHI INTEGRATION --',I4,' STEPS.'///30X,' INTEGRATION',
     $      ' STEPS'//' ',4X,'STEP',7X,'STEP',6X,'THETA',8X,'PHI',
     $      6X,'FERMI RADIUS'/' ',2X,'IN THETA',4X,'IN PHI'/)
 5    FORMAT (I7,4X,I7,F15.6,F12.6,F14.8)
 6    FORMAT (///15X,' VOLUME OF THE FERMI SURFACE ',
     $      '(IN BOHR RADII**3) IS ',1PD15.8,'.')
 7    FORMAT (F10.6)
 8    FORMAT (///15X,' NUMBER OF STATES/BOHR RADIUS**3 IS ',
     $      1PD15.8,'.')
 9    FORMAT (///15X,' NUMBER OF STATES/CC IS ',1PD15.8,'.')
 10   FORMAT (///15X,' NUMBER OF STATES/ATOM IS ',1PD15.8,'.')
C
C     READ INPUT DATA
C
      NUT = 0
      LUMP = 1
      READ (5,3) GAMMA,ALPHA,E1,E2,E3
      READ (5,2) L
      READ (5,1) (IN(I),C(I),I=1,L)
      WRITE(6,1) (IN(I),C(I),I=1,L)
      WRITE(6,3) GAMMA, ALPHA,E1,E2,E3
      READ (5,2) N,M
      READ (5,3) VOL
      CALL COMSET
      KE = L
      KLN = MTOL(KE,10)
      A2 = ALPHA * ALPHA
      G2 = GAMMA * GAMMA
      JJJJ = 0
      IF (E1 .NE. 0) JJJJ = 1
      IF (E2 .NE. 0) JJJJ = 1
      IF (E3 .NE. 0) JJJJ = 1
      WRITE (6,4) L,N,M
      DEN = 0.0
      PI = 3.141592653589793
      E1 = E1 * PI/180.0D0
      E2 = E2 * PI/180.0D0
      E3 = E3 * PI/180.0D0
      DP = 2.0D0 * PI/(M-1)
      DT = PI/(2.0D0*(N-1))
      DO 80 J = 1,M
      BE = 0.0
      DO 51 K = 1,N
      PP = (J-1) * DP
      PHI = PP * 180.0D0/PI
      TT = (K-1) * DT
```

```
      THETA = TT * 180.0D00/PI
      IF (GAMMA .EQ. 1.0) GO TO 40
      TSAVE = TT
      IF (JJJJ .NE. 0) CALL CILER(E1,E2,E3,TT,PP,1)
      ST = DSIN(TT)
      CT = DCOS(TT)
      SP = DSIN(PP)
      CP = DCOS(PP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      TEMP = CT/DSQRT(ST2*(SP2+A2*CP2)/G2+CT2)
      IF (TEMP .GT. 1.0D0) TEMP = 1.0D0
      TT = DARCOS(TEMP)
      SPP = SP/ALPHA
      CALL DFLOP(SPP,CP,PP)
      T = TT
      P = PP
      COEF = ST2 * (SP2 + A2*CP2) + G2 * CT2
      GO TO 41
40    COEF = 1.0000000000000000D00
      T = TT
      P = PP
41    R2 = 0.0
      CALL S2POLY(T,P,B)
      DO 30 I = 1,KE
30    R2 = R2 + C(I)*B(I)
      R2 = R2/COEF
      R = DSQRT(R2)
      DR = R * R * R * DSIN(TSAVE)
      WRITE (6,5) K,J,THETA,PHI,R
      IF (MOD(K,2) .EQ. 1) BDR = DR + DR
      IF (MOD(K,2) .EQ. 0) BDR = DR + DR + DR + DR
      IF (K .EQ. 1) BDR = DR
      IF (K .EQ. N) BDR = DR
      BE = BE + BDR
51    CONTINUE
      IF (MOD(J,2) .EQ. 1) BDEN = BE + BE
      IF (MOD(J,2) .EQ. 0) BDEN = BE + BE + BE + BE
      IF (J .EQ. 1) BDEN = BE
      IF (J .EQ. M) BDEN = BE
      DEN = DEN + BDEN
80    CONTINUE
      DEN = DEN*2.00D00*DT*DP/27.00D0
      WRITE (6,6) DEN
      PI2 = PI + PI
      PI2 = PI2 * PI2 * PI2
      DEN = DEN + DEN
      DEN = DEN/PI2
      WRITE (6,8) DEN
      A0 = 5.29172D-09
      DEN = DEN/(A0*A0*A0)
      WRITE (6,9) DEN
      VOL = VOL * 1.00D-24
      DEN = DEN * VOL
      WRITE (6,10) DEN
      STOP
      END
```

KUBIC HARMONIC GENERATOR

```
C ***********************************************************************
C *                                                                     *
C *                          HAGEN                                      *
C *                                                                     *
C *   THE KUBIC HARMONIC COEFFICIENT GENERATOR.                         *
C *                                                                     *
C *   THIS PROGRAM GENERATES THE COEFFICIENTS FOR THE                   *
C *   SPHERICAL HARMONIC EXPANSION OF THE KUBIC HARMONICS.              *
C *   MAXIMUM ORDER OF THIS EXPANSION IS L = 60                         *
C *   ACTUAL ORDER DESIRED IS READ INTO THE PROGRAM IN I3 FORMAT.       *
C *                                                                     *
C ***********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION B(35,35), C(35), D(35,35)
      DIMENSION ERRVEC(35), VECTOR(35,35)
      DIMENSION DIAG(100),SUPERD(100),WVEC(100),PVEC(100),T(100,3)
      COMMON/F202/DIAG,SUPERD,WVEC,PVEC,T
 1    FORMAT (///10X,' # =',I3,5X,'J =',I3,5X,'I =',I3)
 2    FORMAT (10X,3I5)
 3    FORMAT (4F20.15)
 4    FORMAT (/5F26.14)
 5    FORMAT (1H1,'THE D MATRIX FOR L =',I3///)
 6    FORMAT (/8F15.8)
 7    FORMAT (////)
 8    FORMAT (//' THE ERROR VECTOR -- (DA-IA) -- FOR VECTOR',I3//)
 9    FORMAT (//' THE DEVIATION PER COMPONENT IS',1PD15.8)
 10   FORMAT (I3)
 18   FORMAT (/1P6D20.4)
 19   FORMAT(/10X,8F15.8)
      READ (5,10) LNUM
      IF (LNUM .GT. 60) LNUM = 60
      LNUM = LNUM + 1
      J3 = 1
      DO 100 J1 = 1,LNUM,2
      DO 11 J2 = 1,35
      DO 11 I2 = 1,35
      D(I2,J2) = 0.0
 11   CONTINUE
      J = J1 - 1
      MO = J/4 + 1
      IF (J .NE. 0) GO TO 13
      JEN = 1
      VECTOR(1,1) = 1.00000000D00
      DO 12 K = 2,35
 12   VECTOR(1,K) = 0.0
      GO TO 90
 13   IF (J .EQ. 2) GO TO 100
C
C     CALCULATION OF THE D MATRIX
C
      SQ2 = DSQRT(2.00000000D00)
      DO 14 I = 2,MO
      M1 = 4 * (I-1)
      CALL WIG(J,M1,0,AN1)
      D(1,I) = SQ2 * AN1
```

```
      D(I,1) = D(1,I)
      DO 14 K = I,MO
      M = 4 * (K - 1)
      CALL WIG(J,M1,M,AN1)
      CALL WIG(J,M1,-M,AN2)
      D(K,I) = AN1 + AN2
 14   D(I,K) = D(K,I)
      CALL WIG(J,0,0,AN1)
      D(1,1) = AN1
      WRITE (6,5) J
      IF (MO .GT. 8) GO TO 20
      DO 15 I = 1,MO
 15   WRITE (6,6) (D(I,K),K=1,MO)
      WRITE (6,7)
      GO TO 21
 20   DO 16 I = 1,MO
      WRITE (6,6) (D(I,K),K=1,8)
 16   WRITE (6,19) (D(I,K),K=9,MO)
      WRITE (6,7)
C
C     CALCULATION OF THE DEGENERACY PARAMETER FOR THE EIGENVALUE 1.
C
 21   I2 = 0
      IZZ = J - 2
      I1 = MOD(IZZ,12)
      IF (I1 .EQ. 0) I2 = -1
      JEN = J/12 + I2 + 1
C
C     CLEAR MATRICES
C
      DO 22 K = 1,35
      C(K) = 0.0
      DO 22 L = 1,35
      B(K,L) = 0.0
 22   VECTOR(K,L) = 0.0
C
C
C     COMPUTE EIGENVECTORS.
C     METHOD USED:
C          HOUSEHOLDER REDUCTION TO TRIDIAGONAL FORM,
C          FOLLOWED BY GIVEN'S METHOD FOR DETERMINING
C          THE EIGENVALUES OF A TRIDIAGONAL MATRIX.
C          THE EIGENVECTORS ARE COMPUTED BY INVERSE ITERATION.
C          FOR MULTIPLE ROOTS, ORTHOGONALITY IS ESTABLISHED BY THE
C          GRAM-SCHMIDT PROCESS.
C     ONLY THE EIGENVALUES AND EIGENVECTORS OF INTEREST ARE COMPUTED.
C
C     THIS IS AMDLIB SUBROUTINE F202S.
C
      DO 23 K = 1,MO
      DO 23 L = 1,MO
 23   B(K,L) = D(K,L)
      CALL EIGEN(B,VECTOR,MO,C,JEN,ANORM,35)
      DO 25 K = 1,JEN
      DO 25 L = 1,MO
 25   VECTOR(K,L) = B(L,K)
C
C     PRODUCTION OF THE PROPER LINEAR COMBINATIONS OF
C          THE EIGENVECTORS TO CORRESPOND WITH THE VECTORS
C          PRODUCED BY MUELLER AND PRIESTLEY.
```

```
C
      DO 60 I = 1,JEN
      TEMP = VECTOR(I,1)
      DO 60 K = 1,MO
      VECTOR(I,K) = VECTOR(I,K)/TEMP
 60   CONTINUE
      IF (JEN .EQ. 1) GO TO 65
      I = 2
 66   II = 1
      KK = 2
 62   DO 63 K = 1,MO
      VECTOR(I,K) = VECTOR(I,K) - VECTOR(II,K)
 63   CONTINUE
      TEMP = VECTOR(I,KK)
      DO 64 K = 1,MO
      VECTOR(I,K) = VECTOR(I,K)/TEMP
 64   CONTINUE
      II = II + 1
      KK = KK + 1
      IF (II .EQ. I) GO TO 61
      GO TO 62
 61   CONTINUE
      IF (I .EQ. JEN) GO TO 65
      I = I + 1
      GO TO 66
 65   CONTINUE
C
C     GRAM/SCHMIDT ORTHONORMALIZATION PROCESS
C
      DO 30 I2 = 1,JEN
      I3 = JEN - I2 + 1
      I4 = I3 - 1
      SUM = 0.C
      DO 31 K = 1,MO
 31   SUM = SUM + VECTOR(I3,K)*VECTOR(I3,K)
      SUM = DSQRT(SUM)
      DO 32 K = 1,MO
 32   VECTOR(I3,K) = VECTOR(I3,K)/SUM
      IF (I3 .LE. 1) GO TO 30
      DO 33 L = 1,I4
      SUM = 0.0
      DO 34 K = 1,MO
 34   SUM = SUM + VECTOR(I3,K)*VECTOR(L,K)
      DO 35 K = 1,MO
 35   VECTOR(L,K) = VECTOR(L,K) - SUM*VECTOR(I3,K)
 33   CONTINUE
 30   CONTINUE
C
C     COMPUTE RESIDUAL VECTOR D*A - I*A
C
      DO 40 I = 1,JEN
      DO 41 K = 1,35
 41   ERRVEC(K) = 0.0
      DO 42 K = 1,MO
      DO 43 L = 1,MO
 43   ERRVEC(K) = ERRVEC(K) + D(K,L)*VECTOR(I,L)
 42   ERRVEC(K) = ERRVEC(K) - VECTOR(I,K)
      DEV = 0.0
      DO 44 K = 1,MO
 44   DEV = DEV + ERRVEC(K)*ERRVEC(K)
```

```
      XMO = MO
      DEV = DEV/XMO
      DEV = DSQRT(DEV)
      WRITE (6,8) I
      WRITE (6,18) (ERRVEC(K),K=1,MO)
      WRITE (6,9) DEV
 40   CONTINUE
C
C     OUTPUT OF COEFFICIENTS
C
 90   DO 95 I = 1,JEN
      WRITE (6,1) J3,J,I
      WRITE (6,4) (VECTOR(I,K),K=1,MO)
      WRITE (7,2) J3,J,I
      WRITE (7,3) (VECTOR(I,K),K=1,16)
 95   J3 = J3 + 1
 100  CONTINUE
      STOP
      END
```

SPHERICAL HARMONIC GENERATOR PACKAGE

```
      INTEGER FUNCTION IYLM*4 (LL,MM)
C ***********************************************************************
C *                                                                     *
C *   THIS FUNCTION SUBROUTINE RETURNS THE INDEX OF THE                 *
C *        XPLM OR DYLM MATRIX THAT CORRESPONDS TO L = LL AND M = MM    *
C *                                                                     *
C ***********************************************************************
      L = LL
      M = MM
      IF (M .LT. 0) M = -M
      IF (L .LT. 0) GO TO 1
      IF (L .GT. 60) GO TO 1
      IF (M .GT. L) GO TO 1
      L = L/2
      I = L * L
      I = I + 1
      N = I + M
      IF (N .GT. 961) GO TO 1
      IYLM = N
      RETURN
 1    IYLM = 0
      RETURN
      END


      REAL FUNCTION YLM*8 (LL,MM,J)
C ***********************************************************************
C *                                                                     *
C *   IF J = 1, THE CURRENT VALUE OF THE SPHERICAL HARMONIC             *
C *        OF ORDER LL AND INDEX MM IS RETURNED.                        *
C *                                                                     *
C *   IF J = -1, THE CURRENT VALUE OF THE THETA DERIVATIVE              *
C *        OF THE SPHERICAL HARMONIC OF ORDER LL AND INDEX MM           *
C *        IS RETURNED.                                                 *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XPLM(961),DYLM(961)
      COMMON/YMATRX/XPLM,DYLM
      L = LL
      M = MM
      IF (M .LT. 0) M = -M
      IF (L .LT. 0) GO TO 10
      IF (L .GT. 60) GO TO 10
      IF (M .GT. L) GO TO 10
      L = L/2
      I = L * L
      I = I + 1
      N = I + M
      IF (N .GT. 961) GO TO 10
      IF (J .GT. 0) YLM = XPLM(N)
      IF (J .LT. 0) YLM = DYLM(N)
      RETURN
 10   YLM = 0.0
      RETURN
```

```
      END

      SUBROUTINE BLKDAT
C ***********************************************************************
C *                                                                     *
C *   BLOCK DATA SUBPROGRAM USED FOR INITIALIZATION OF AN ARRAY USED    *
C *   BY SUBROUTINES SPHERE AND GRAD.                                   *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/FACTOR/FL(121),OMEGAC(961),TO(961),COOF(61)
            FL(  1)=   0.000000000000000
            FL(  2)=   0.000000000000000
            FL(  3)=   0.693147180558299
            FL(  4)=   1.791759469226236
            FL(  5)=   3.178053830342833
            FL(  6)=   4.787491742754355
            FL(  7)=   6.579251211951487
            FL(  8)=   8.525161360856146
            FL(  9)=  10.604602902662009
            FL( 10)=  12.801827479852363
            FL( 11)=  15.104412572924048
            FL( 12)=  17.502307845745236
            FL( 13)=  19.987214495427907
            FL( 14)=  22.552163852844387
            FL( 15)=  25.191221182700247
            FL( 16)=  27.899271383415908
            FL( 17)=  30.671860105823725
            FL( 18)=  33.505073449574411
            FL( 19)=  36.395445207133889
            FL( 20)=  39.339884187094867
            FL( 21)=  42.335616460070014
            FL( 22)=  45.380138898268342
            FL( 23)=  48.471181351691484
            FL( 24)=  51.606675567105412
            FL( 25)=  54.784729397855699
            FL( 26)=  58.003605222329497
            FL( 27)=  61.261701760813594
            FL( 28)=  64.557538626715541
            FL( 29)=  67.889743136242032
            FL( 30)=  71.257038965821266
            FL( 31)=  74.658236347138882
            FL( 32)=  78.092223552986979
            FL( 33)=  81.557959455996752
            FL( 34)=  85.054467016831040
            FL( 35)=  88.580827541649342
            FL( 36)=  92.136175602674484
            FL( 37)=  95.719694541767240
            FL( 38)=  99.330612454563379
            FL( 39)=  102.96819861419499
            FL( 40)=  106.63176026009023
            FL( 41)=  110.32063971459866
            FL( 42)=  114.03421178087592
            FL( 43)=  117.77188139967620
            FL( 44)=  121.53308151476085
            FL( 45)=  125.31727114878595
            FL( 46)=  129.12393363565207
            FL( 47)=  132.95257503539324
            FL( 48)=  136.80272263661027
```

```
FL( 49)=  140.67392364516854
FL( 50)=  144.56574394553900
FL( 51)=  148.47776694968343
FL( 52)=  152.40959258377552
FL( 53)=  156.36083630099893
FL( 54)=  160.33112821355462
FL( 55)=  164.32011226192117
FL( 56)=  168.32744544744492
FL( 57)=  172.35279713571072
FL( 58)=  176.39584840461612
FL( 59)=  180.45629141479731
FL( 60)=  184.53382885828614
FL( 61)=  188.62817342206836
FL( 62)=  192.73904728516936
FL( 63)=  196.86618167161942
FL( 64)=  201.00931639596820
FL( 65)=  205.16819947957993
FL( 66)=  209.34258675202727
FL( 67)=  213.53224149346352
FL( 68)=  217.73693411052227
FL( 69)=  221.95644181594253
FL( 70)=  226.19054832309484
FL( 71)=  230.43904356285930
FL( 72)=  234.70172344148159
FL( 73)=  238.97838956117630
FL( 74)=  243.26884900033474
FL( 75)=  247.57291409373283
FL( 76)=  251.89040220901370
FL( 77)=  256.22113554924726
FL( 78)=  260.56494096666574
FL( 79)=  264.92164979130030
FL( 80)=  269.29109764844179
FL( 81)=  273.67312428355217
FL( 82)=  278.06757343560457
FL( 83)=  282.47429268062115
FL( 84)=  286.89313329011202
FL( 85)=  291.32395008951426
FL( 86)=  295.76660134643316
FL( 87)=  300.22094864398241
FL( 88)=  304.68685676157475
FL( 89)=  309.16419357806444
FL( 90)=  313.65282994508743
FL( 91)=  318.15263962000608
FL( 92)=  322.66349912434816
FL( 93)=  327.18528769910336
FL( 94)=  331.71788719296455
FL( 95)=  336.26118197292089
FL( 96)=  340.81505886465311
FL( 97)=  345.37940705567598
FL( 98)=  349.95411803573370
FL( 99)=  354.53908551484346
FL(100)=  359.13420536369085
FL(101)=  363.73937555402517
FL(102)=  368.35449606925249
FL(103)=  372.97946888208389
FL(104)= 377.61419787257910
FL(105)=  382.25858876854181
FL(106)=  386.91254912316799
FL(107)=  391.57598821073771
FL(108)=  396.24881704896688
```

```
      FL(109)=  400.93094827234745
      FL(110)=  405.62229615449905
      FL(111)=  410.32277652621269
      FL(112)=  415.03230672329664
      FL(113)=  419.75080559402704
      FL(114)=  424.47819341719151
      FL(115)=  429.21439186483622
      FL(116)=  433.95932399481535
      FL(117)=  438.71291418373585
      FL(118)=  443.47508811950684
      FL(119)=  448.24577274173498
      FL(120)=  453.02489623427391
      FL(121)=  457.81238798052073
      RETURN
      END


      SUBROUTINE COMSET
C ***********************************************************************
C *                                                                     *
C *   SUBROUTINE USED FOR INITIALIZATION OF TWO ARRAYS USED BY          *
C *   SUBROUTINES SPHERE AND GRAD.                                      *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/FACTOR/FL(121),OMEGAO(961),TO(961),COOF(61)
      CALL BLKDAT
      TWO = DLOG(2.0000000000D00)
      PI = 3.141592653589793
      FPI = PI + PI + PI + PI
      FPI = DLOG(FPI)
      DO 10 L1 = 1,61,2
      L = L1 - 1
      XL2P1 = L + L + 1
      XL2P1 = DLOG(XL2P1)
      XTWOL = L * TWO
      DO 20 M1 = 1,L1
      M = M1 - 1
      I = IYLM(L,M)
      COEF = XL2P1 + FL(L-M+1) - FPI - FL(L+M+1)
      COEF = COEF/2.0000000000D00
      Z = FL(L+L+1) - FL(L+1) - FL(L-M+1) - XTWOL + COEF
      XTWOM = M * TWO
      ZZ = FL(L+M+1) - FL(L-M+1) - FL(M+1) - XTWOM + COEF
      OMEGAO(I) = DEXP(Z)
 20   TO(I) = DEXP(ZZ)
 10   CONTINUE
      RETURN
      END


      SUBROUTINE SPHERE(THETA,PHI,LMAX)
C ***********************************************************************
C *                                                                     *
C *   THIS SUBROUTINE GENERATES THE SPHERICAL HARMONICS Y(L,M)          *
C *   FOR ALL EVEN ORDERS OF L,  L <= LMAX,  AND ALL NON-NEGATIVE       *
C *   INDEXES M,  0 <= M <= L.
C *                                                                     *
C *   NOTE:   THE PHI DEPENDENCE OF THE Y(L,M)'S,  EXP(I*M*PHI)         *
C *   IS NOT CALCULATED.                                                *
```

```
C *                                                                    *
C *   THE HARMONICS CALCULATED BY THIS ROUTINE CAN BEST BE REFERENCED  *
C *   BY USE OF FUNCTION SUBPROGRAMS YLM AND IYLM.                     *
C *                                                                    *
C *   DOUBLE PRECISION ROUTINE.                                        *
C *                                                                    *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 THETA,PHI
      COMMON/YMATRX/XPLM(961),DYLM(961)
      COMMON/FACTOR/FL(121),OMEGA(961),T(961),COEF(61)
      DO 1 I = 1,961
 1    XPLM(I) = 0.0
      IF (LMAX .GT. 60) LMAX = 60
      IU = LMAX + 1
      TT = THETA
      X = DCOS(TT)
      XX = DABS(X)
      X2 = X * X
      Y = DSIN(TT)
      Z = 1.0 - X
      KYLM = 0
      DO 180 L1 = 1,IU,2
      L = L1 - 1
      F = 1.0000000000000000D00
      DO 180 M1 = 1,L1
      M = M1 - 1
      LMM = L - M
      KYLM = KYLM + 1
C
C     DIVISION OF PROBLEM INTO FOUR CASES.
C
      IF (XX .LT. 1.0D-06) GO TO 110
      IF ((1.0 - XX) .LT. 1.0D-06) GO TO 150
      IF (XX .GT. 0.77)GO TO 70
C
C     CASE WHERE COS(THETA) <= 0.77
C
      COEF(1) = OMEGA(KYLM)
      LIM = LMM/2
      DO 10 NU = 1,LIM
      NU2 = NU + NU
      J = NU + 1
      A = (LMM - NU2 + 2) * (LMM - NU2 + 1)
      B = NU2 * (L + L - NU2 + 1)
      A = A/B
 10   COEF(J) = -COEF(J-1) * A
      IF (LIM .EQ. 0) GO TO 30
      SUM = COEF(1)
      DO 20 I = 1,LIM
      J = I + 1
 20   SUM = SUM * X2 + COEF(J)
      IF (MOD(LMM,2) .NE. 0) SUM = SUM * X
      GO TO 90
 30   SUM = COEF(1)
      IF (LMM .EQ. 1) SUM = SUM * X
      GO TO 90
C
C     CASE WHERE COS(THETA) > 0.77
C
```

```
 70   COEF(1) = T(KYLM)
      DO 40 NU = 1,LMM
      NU2 = NU + NU
      J = NU + 1
      A = (LMM - NU + 1) * (L + M + NU)
      B = NU2 * (M + NU)
      A = A/B
 40   COEF(J) = -COEF(J - 1) * A
      IF (LMM .EQ. 0) GO TO 60
      SUM = COEF(LMM + 1)
      DO 50 I = 1,LMM
      J = LMM + 1 - I
 50   SUM = SUM * Z  +  COEF(J)
      GO TO 90
 60   SUM = COEF(1)
 90   IF (M .EQ. 0) GO TO 100
      F = Y * F
      IF (F .LT. 1.0D-30) F = 0.0
 100  XPLM(KYLM) = SUM * F
 180  CONTINUE
      RETURN
C
C     CASE WHERE COS(THETA) = 0
C
 110  IF (MOD(LMM,2) .NE. 0) GO TO 180
      TSUM = OMEGA(KYLM)
      LIM = LMM/2
      IF (LIM .EQ. 0) GO TO 140
      DO 140 NU = 1,LIM
      NU2 = NU + NU
      A = (LMM - NU2 + 2) * (LMM - NU2 + 1)
      B = NU2 * (L + L + 1 - NU2)
      A = A/B
      TSUM = -TSUM * A
 140  SUM = TSUM
      GO TO 90
C
C     CASE WHERE COS(THETA) = 1.0
C
 150  IF (M .NE. 0) GO TO 180
      XPLM(KYLM) = T(KYLM)
      GO TO 180
      END


      SUBROUTINE GRAD(THETA,PHI,LMAX)
C ***********************************************************************
C *                                                                     *
C *   THIS SUBROUTINE GENERATES THE PARTIAL DERIVATIVES OF THE          *
C *   SPHERICAL HARMONICS WITH RESPECT TO THETA FOR ALL EVEN ORDERS OF  *
C *   L, L<= LMAX, AND ALL NON-NEGATIVE INDEXES M, 0 <= M <= L.         *
C *                                                                     *
C *   NOTE:   THE PHI DEPENDENCE OF THE HARMONIC DERIVATIVES,           *
C *   EXP(I*M*PHI)  IS NOT CALCULATED.                                  *
C *                                                                     *
C *   THE HARMONICS CALCULATED BY THIS ROUTINE CAN BEST BE REFERENCED   *
C *   BY USE OF FUNCTION SUBPROGRAMS YLM AND IYLM.                      *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
```

```
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 THETA,PHI
      COMMON/YMATRX/XPLM(961),DYLM(961)
      COMMON/FACTOR/FL(121),OMEGA(961),T(961),COEF(61)
      DO 1 I = 1,961
 1    DYLM(I) = 0.0
      IF (LMAX .GT. 60) LMAX = 60
      IU = LMAX + 1
      TT = THETA
      X = DCOS(TT)
      XX = DABS(X)
      X2 = X * X
      Y = DSIN(TT)
      Z = 1.0 - X
      KYLM = 0
      DO 180 L1 = 1,IU,2
      L = L1 - 1
      F = 1.000000000000000D00
      IF ((1.0 - XX) .GT. 1.0D-06) F = F/Y
      DO 180 M1 = 1,L1
      M = M1 - 1
      AM = M
      LMM = L - M
      KYLM = KYLM + 1
C
C     DIVISION OF PROBLEM INTO FOUR CASES
C
      IF (XX .LT. 1.0D-06) GO TO 110
      IF ((1.0 - XX) .LT. 1.0D-06) GO TO 150
      IF (XX .GT. 0.77) GO TO 70
C
C     CASE WHERE COS(THETA) <= 0.77
C
      COEF(1) = OMEGA(KYLM)
      LIM = LMM/2
      DO 10 NU = 1,LIM
      NU2 = NU + NU
      J = NU + 1
      A = (LMM - NU2 + 2) * (LMM - NU2 + 1)
      B = NU2 * (L + L - NU2 + 1)
      A = A/B
 10   COEF(J) = -COEF(J-1) * A
      IF (LIM .EQ. 0) GO TO 30
      XK = LMM
      SUM1 = COEF(1) * XK
      SUM2 = COEF(1)
      DO 20 I = 1,LIM
      J = I + 1
      XK = LMM - I - I
      SUM1 = SUM1 * X2 + COEF(J)*XK
 20   SUM2 = SUM2 * X2 + COEF(J)
      IF (MOD(LMM,2) .EQ. 0) GO TO 40
      SUM1 = SUM1 * X
      SUM2 = SUM2 * X
      GO TO 40
 30   XK = LMM
      SUM1 = COEF(1) * XK
      SUM2 = COEF(1)
      IF (LMM .EQ. 0) GO TO 40
```

```
      SUM1 = SUM1 * X
      SUM2 = SUM2 * X
 40   SUM = -Y*Y*SUM1/X + AM*X*SUM2
      GO TO 90
C
C     CASE WHERE COS(THETA) > 0.77
C
 70   COEF(1) = T(KYLM)
      DO 50 NU = 1,LMM
      NU2 = NU + NU
      J = NU + 1
      A = (LMM - NU + 1) * (L + M + NU)
      B = NU2 * (M + NU)
      A = A/B
 50   COEF(J) = -COEF(J-1) * A
      IF (LMM .EQ. 0) GO TO 80
      XK = LMM
      SUM1 = COEF(LMM+1) * XK
      SUM2 = COEF(LMM+1)
      DO 60 I = 1,LMM
      J = LMM + 1 - I
      XK = LMM - I
      SUM1 = SUM1 * Z  + COEF(J) * XK
 60   SUM2 = SUM2 * Z  +  COEF(J)
      GO TO 85
 80   SUM1 = 0.0
      SUM2 = COEF(1)
 85   SUM = (1.0D0+X)*SUM1 + AM*X*SUM2
 90   IF (M .EQ. 0) GO TO 100
      F = Y * F
      IF (F .LT. 1.0D-30) F = 0.0
 100  DYLM(KYLM) = SUM * F
 180  CONTINUE
      RETURN
C
C     CASE WHERE COS(THETA) = 0
C
 110  IF (MOD(LMM,2) .EQ. 0) GO TO 180
      TSUM = OMEGA(KYLM)
      LIM = LMM/2
      IF (LIM .EQ. 0) GO TO 140
      DO 140 NU = 1,LIM
      NU2 = NU + NU
      A = (LMM - NU2 + 2) * (LMM - NU2 + 1)
      B = NU2 * (L + L - NU2 + 1)
      A = A/B
      TSUM = -TSUM * A
 140  SUM = -TSUM
      GO TO 90
C
C     CASE WHERE COS(THETA) = 1.0
C
 150  IF (M .NE. 1) GO TO 180
      DYLM(KYLM) = -T(KYLM)
      GO TO 180
      END
```

HARMONIC FUNCTION GENERATORS

```
      SUBROUTINE KUPOLY(T,P,B)
C ***********************************************************************
C *                                                                     *
C $   THIS SUBROUTINE USES LINEAR COMBINATIONS OF                       *
C *   SPHERICAL HARMONICS TO GENERATE KUBIC HARMONICS.                  *
C *                                                                     *
C *   DOUBLE PRECISION                                                  *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P
      REAL*4 SYM
      DIMENSION B(91),NAM(91),SYM(91,16),XPLM(961),DYLM(961)
      COMMON/YMATRX/XPLM,DYLM
      COMMON/KUBE/NAM,SYM,KE,KLN
      SQ2 = DSQRT(2.0000000000000000D00)
      CALL SPHERE(T,P,KLN)
      PHI = P
      DO 1 J = 1,KE
      AS = 0.0
      L = NAM(J)
      ML = L/4 + 1
      DO 2 K = 1,ML
      M = 4 * (K - 1)
      AM = M
      CP = DCOS(AM * PHI)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
 2    AS = AS + SYM(J,K) * CP * AN
 1    B(J) = AS
      RETURN
      END


      SUBROUTINE HXPOLY(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP D6H.               *
C *                                                                     *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,6
      M = M1 - 1
```

```
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE C6POLY(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP C6H.               *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,6
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * SP
      IF (NUT .NE. 69) NAM(N) = L
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE D3POLY(T,P,B)
C ***********************************************************************
```

```
C *                                                                    *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP D3H.              *
C *                                                                    *
C *   DOUBLE PRECISION ROUTINE.                                        *
C *                                                                    *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO M1 = 1,L1,3
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END

      SUBROUTINE S6POLY(T,P,B)
C ***********************************************************************
C *                                                                    *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP S6.               *
C *                                                                    *
C *   DOUBLE PRECISION ROUTINE.                                        *
C *                                                                    *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO M1 = 1,L1,3
      M = M1 - 1
      N = N + 1
```

```
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * SP
      IF (NUT .NE. 69) NAM(N) = L
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE SQPOLY(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP D4H.               *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,4
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE C4POLY(T,P,B)
C ***********************************************************************
C *                                                                     *
```

```
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP C4H.                *
C *                                                                      *
C *   DOUBLE PRECISION ROUTINE.                                          *
C *                                                                      *
C ************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,4
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * SP
      IF (NUT .NE. 69) NAM(N) = L
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE VHPOLY(T,P,B)
C ************************************************************************
C *                                                                      *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP VH.                 *
C *                                                                      *
C *   DOUBLE PRECISION ROUTINE.                                          *
C *                                                                      *
C ************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
```

```
      L = L1 - 1
      DO 50 M1 = 1,L1,2
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE C2POLY(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP C2H.               *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,2
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * SP
      IF (NUT .NE. 69) NAM(N) = L
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END
```

```
      SUBROUTINE S2POLY(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP S2.                *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * SP
      IF (NUT .NE. 69) NAM(N) = L
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END
```

HARMONIC FUNCTION THETA-DERIVATIVE GENERATOR

```
      SUBROUTINE KUPOLT(T,P,B)
C ***********************************************************************
C *                                                                     *
C $   THIS SUBROUTINE USES LINEAR COMBINATIONS OF                       *
C *   SPHERICAL HARMONICS TO GENERATE KUBIC HARMONICS.                  *
C *                                                                     *
C *   THETA DERIVATIVE                                                  *
C *                                                                     *
C *   DOUBLE PRECISION                                                  *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P
```

```
      REAL*4 SYM
      DIMENSION B(91),NAM(91),SYM(91,16),XPLM(961),DYLM(961)
      COMMON/YMATRX/XPLM,DYLM
      COMMON/KUBE/NAM,SYM,KE,KLN
      SQ2 = DSQRT(2.0000000000000000D00)
      CALL GRAD(T,P,KLN)
      PHI = P
      DO 1 J = 1,KE
      AS = 0.0
      L = NAM(J)
      ML = L/4 + 1
      DO 2 K = 1,ML
      M = 4 * (K - 1)
      AM = M
      CP = DCOS(AM * PHI)
      AN = YLM(L,M,-1)
      IF (M .NE. 0) AN = SQ2 * AN
 2    AS = AS + SYM(J,K) * CP * AN
 1    B(J) = AS
      RETURN
      END


      SUBROUTINE HXPOLT(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP D6H.               *
C *                                                                     *
C *   THETA DERIVATIVE                                                  *
C *                                                                     *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,F1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL GRAD(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,6
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      AN = YLM(L,M,-1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END
```

```
      SUBROUTINE C6POLT(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP C6H.               *
C *                                                                     *
C *   THETA DERIVATIVE                                                  *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL GRAD(T,P,KLN)
      SQ2 = DSQRT(2.000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,6
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,-1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * SP
      IF (NUT .NE. 69) NAM(N) = L
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END

      SUBROUTINE D3POLT(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP D3H.               *
C *                                                                     *
C *   THETA DERIVATIVE                                                  *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
```

```
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL GRAD(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO M1 = 1,L1,3
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      AN = YLM(L,M,-1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE S6POLT(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP S6.                *
C *                                                                     *
C *   THETA DERIVATIVE                                                  *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL GRAD(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO M1 = 1,L1,3
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,-1)
      IF (M .NE. 0) AN = SQ2 * AN
```

```
      B(N) = AN * CP
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * SP
      IF (NUT .NE. 69) NAM(N) = L
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE SQPOLT(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP D4H.               *
C *                                                                     *
C *   THETA DERIVATIVE                                                  *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL GRAD(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,4
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      AN = YLM(L,M,-1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE C4POLT(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP C4H.               *
C *                                                                     *
C *   THETA DERIVATIVE                                                  *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
```

```
C *                                                                    *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL GRAD(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,4
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,-1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * SP
      IF (NUT .NE. 69) NAM(N) = L
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE VHPOLT(T,P,B)
C ***********************************************************************
C *                                                                    *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP VH.               *
C *                                                                    *
C *   THETA DERIVATIVE                                                 *
C *                                                                    *
C *   DOUBLE PRECISION ROUTINE.                                        *
C *                                                                    *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL GRAD(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
```

```
      DO 50 M1 = 1,L1,2
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      AN = YLM(L,M,-1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE C2POLT(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP C2H.               *
C *                                                                     *
C *   THETA DERIVATIVE                                                  *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL GRAD(T,P,KLN)
      SQ2 = DSQRT(2.000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,2
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,-1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * SP
      IF (NUT .NE. 69) NAM(N) = L
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END
```

```
      SUBROUTINE S2POLT(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP S2.                *
C *                                                                     *
C *   THETA DERIVATIVE                                                  *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL GRAD(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,-1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = AN * CP
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * SP
      IF (NUT .NE. 69) NAM(N) = L
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END
```

HARMONIC FUNCTION PHI-DERIVATIVE GENERATORS

```
      SUBROUTINE KUPOLP(T,P,B)
C ***********************************************************************
C *                                                                     *
C $   THIS SUBROUTINE USES LINEAR COMBINATIONS OF                       *
C *   SPHERICAL HARMONICS TO GENERATE KUBIC HARMONICS.                  *
C *                                                                     *
C *   PHI DERIVATIVE                                                    *
C *                                                                     *
C *   DOUBLE PRECISION                                                  *
C *                                                                     *
```

```
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P
      REAL*4 SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16),XPLM(961),DYLM(961)
      COMMON/YMATRX/XPLM,DYLM
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      SQ2 = DSQRT(2.0000000000000000D00)
      CALL SPHERE(T,P,KLN)
      PHI = P
      DO 1 J = 1,KE
      AS = 0.0
      L = NAM(J)
      ML = L/4 + 1
      DO 2 K = 1,ML
      M = 4 * (K - 1)
      AM = M
      SP = DSIN(AM * PHI)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
 2    AS = AS - SYM(J,K) * SP * AN * M
 1    B(J) = AS
      RETURN
      END


      SUBROUTINE HXPOLP(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP D6H.               *
C *                                                                     *
C *   PHI DERIVATIVE                                                    *
C *                                                                     *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,6
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = -AN * SP * M
```

```
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE C6POLP(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP C6H.               *
C *                                                                     *
C *   PHI DERIVATIVE                                                    *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,6
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = -AN * SP * M
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * CP * M
      IF (NUT .NE. 69) NAM(N) = L
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE D3POLP(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP D3H.               *
C *                                                                     *
C *   PHI DERIVATIVE                                                    *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
```

```
C *                                                                    *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO M1 = 1,L1,3
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = -AN * SP * M
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE S6POLP(T,P,B)
C ***********************************************************************
C *                                                                    *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP S6.               *
C *                                                                    *
C *   PHI DERIVATIVE                                                   *
C *                                                                    *
C *   DOUBLE PRECISION ROUTINE.                                        *
C *                                                                    *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO M1 = 1,L1,3
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
```

```
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = -AN * SP * M
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * CP * M
      IF (NUT .NE. 69) NAM(N) = L
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE SQPOLP(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP D4H.               *
C *                                                                     *
C *   PHI DERIVATIVE                                                    *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,4
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = -AN * SP * M
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE C4POLP(T,P,B)
C ***********************************************************************
C *                                                                     *
```

```
C *    HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP C4H.                 *
C *                                                                        *
C *    PHI DERIVATIVE                                                      *
C *                                                                        *
C *    DOUBLE PRECISION ROUTINE.                                           *
C *                                                                        *
C ************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,4
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = -AN * SP * M
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * CP * M
      IF (NUT .NE. 69) NAM(N) = L
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END

      SUBROUTINE VHPOLP(T,P,B)
C ************************************************************************
C *                                                                      *
C *    HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP VH.                *
C *                                                                      *
C *    PHI DERIVATIVE                                                    *
C *                                                                      *
C *    DOUBLE PRECISION ROUTINE.                                         *
C *                                                                      *
C ************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
```

```
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,2
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = -AN * SP * M
 50   CONTINUE
 100  CONTINUE
 69   NUT = 69
      RETURN
      END


      SUBROUTINE C2POLP(T,P,B)
C ***********************************************************************
C *                                                                     *
C *   HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP C2H.               *
C *                                                                     *
C *   PHI DERIVATIVE                                                    *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1,2
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = -AN * SP*M
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * CP*M
      IF (NUT .NE. 69) NAM(N) = L
```

```
50    CONTINUE
100   CONTINUE
69    NUT = 69
      RETURN
      END

      SUBROUTINE S2POLP(T,P,B)
C ***********************************************************************
C *                                                                     *
C *    HARMONIC FUNCTION GENERATOR FOR SYMMETRY GROUP S2.               *
C *                                                                     *
C *    PHI DERIVATIVE                                                   *
C *                                                                     *
C *    DOUBLE PRECISION ROUTINE.                                        *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 T,P,SYM
      REAL*4 ALPHA,A2,GAMMA,G2,E1,E2,E3
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      KLN1 = KLN + 1
      PP = P
      N = 0
      CALL SPHERE(T,P,KLN)
      SQ2 = DSQRT(2.0000000000000000D00)
      DO 100 L1 = 1,KLN1,2
      L = L1 - 1
      DO 50 M1 = 1,L1
      M = M1 - 1
      N = N + 1
      IF (N .GT. KE) GO TO 69
      IF (NUT .NE. 69) NAM(N) = L
      AM = M
      CP = DCOS(AM*PP)
      SP = DSIN(AM*PP)
      AN = YLM(L,M,1)
      IF (M .NE. 0) AN = SQ2 * AN
      B(N) = -AN * SP * M
      IF (M .EQ. 0) GO TO 50
      N = N + 1
      B(N) = AN * CP * M
      IF (NUT .NE. 69) NAM(N) = L
50    CONTINUE
100   CONTINUE
69    NUT = 69
      RETURN
      END
```

LEGENDRE POLYNOMIAL GENERATOR

```
      SUBROUTINE COFT(B)
C ***********************************************************************
C *                                                                     *
C *    THIS ROUTINE CALCULATES THE LEGENDRE POLYNOMIALS P<L>(X)         *
C *    FOR THE VALUE X = 0.                                             *
```

```
C *              P<L>(0)                                                  *
C *                                                                       *
C *   DOUBLE PRECISION ROUTINE.                                           *
C *                                                                       *
C ***********************************************************************
      IMPLICIT REAL*8 (A-E)
      DIMENSION B(91),NAM(91),SYM(91,16)
      COMMON/KUBE/NAM,SYM,KE,KLN
      DO 1 J3 = 1,91
 1    B(J3) = 0.0
      DO 2 J3 = 1,KE
      N = NAM(J3)/2
      JJ = 2*N - 1
      A = JJ
      C = 2*N
      AR = 1.
 3    IF (JJ .LT. 1) GO TO 4
      AR = AR * A/C
      A = A - 2.
      C = C - 2.
      JJ = JJ - 2
      GO TO 3
 4    I = MOD(N,2)
      IF (I .GT. 0) AR = -AR
 2    B(J3) = AR
      RETURN
      END
```

MULTIPURPOSE SUBROUTINES

```
      SUBROUTINE KLSQ (NUM,KE,CAL,DELTA,FUNC,IDELTA)
C ***********************************************************************
C *                                                                       *
C *   LEAST SQUARES FITTING PROCEDURE.                                    *
C *                                                                       *
C ***********************************************************************
      IMPLICIT REAL*8 (A-E)
      REAL*8 PI,T,P
      DIMENSION T(600),P(600)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NT(10),NL(10),NZ(10),TL(10)
      DIMENSION A(91,91),B(91),C(91),BE(91)
      DIMENSION BC(91)
      DIMENSION AA(91,91),BB(91)
      COMMON/LIMIT/MID,NIT,BC ,LNIT
      COMMON/WORK/BE,B,C
      COMMON/OVER/A,N
      COMMON/INTO/IN,R
      COMMON/TEEPEE/T,P
      COMMON/ORIENT/NT,NL,NZ,TL
 1    FORMAT (//7X,' LAB ANGLE',7X,'THETA',11X,'PHI',11X,
     $     'EXP''TL VALUE',7X,'TTHETA',10X,'TPHI',16X,'RS' )
 2    FORMAT (9X,I6,6X,F11.6,4X,F11.6,7X,E13.6,6X,F11.6,4X,F11.6,7X,E13
     ,.6)
 3    FORMAT (/' PLANE # ',I2)
4     FORMAT(I5,2F10.4,E17.6,2F10.4,E17.6,1X)
 15   FORMAT (//10X,' STANDARD DEVIATION =',F10.6)
```

```
      PI = 3.141592653589793
      DO 10 I = 1,91
      B(I) = 0.0
      C(I) = 0.0
      BE(I) = 0.0
      DO 10 J = 1,91
 10   A(I,J) = 0.0
      DO 20 I = 1,NUM
      IF (DELTA.NE.0.0)     GO TO 37
      WRITE (6,3) I
      WRITE (6,1)
 37   CONTINUE
      CALL LAB(I,CAL)
      NTI = NT(I)
      DO 20 J = 1,NTI
      CALL ELLIPS(I,J,TT,PP,RS)
      CALL FUNC(TT,PP,BE)
      TTT=TT*180.0D0/PI
      PPP=PP*180.0D0/PI
      TT = T(J) * 180.0D0/PI
      PP = P(J) * 180.0D0/PI
      IF (DELTA.NE. 0.0)  GO TO 36
      WRITE (6,2) IN(J,I),TT,PP,R(J,I),TTT,PPP,RS
      WRITE(10,4) IN(J,I),TT,PP,R(J,I),TTT,PPP,RS
 36   CONTINUE
      DO 30 L = 1,KE
      B(L) = B(L) + RS * BE(L)
      DO 30 K = 1,KE
 30   A(L,K) = A(L,K) + BE(L)*BE(K)
 20   CONTINUE
      DO 31 L=1,KE
      BB(L)=B(L)
      DO 31  K=1,KE
      AA(L,K)=A(L,K)
 31   CONTINUE
      DO 39 II=LNIT,NIT
      IF (MOD(II,2).EQ.1) GO TO 32
      IF (MOD(II,2).EQ.0)  GO TO 33
 32   CONTINUE
      NLL=1
      NUL=MID
      GO TO 34
 33   NLL=MID+1
      NUL=KE
 34   CONTINUE
      LO=NLL-1
      DO 23  L=NLL,NUL
      IF (NLL.LE.1) GO TO 24
      DO 21  K=1,LO
 21   B(L)=B(L)-BC(K)*A(K,L)
 24   KO=NUL+1
      IF (NUL.GE. KE) GO TO 23
      DO 22 K=KO,KE
 22   B(L)=B(L)-BC(K)*A(K,L)
 23   CONTINUE
      DO 25 L = NLL,NUL
      LL = L - NLL + 1
      DO 25 K = NLL,NUL
      KK = K - NLL + 1
      A(LL,KK) = A(L,K)
```

```
 25   CONTINUE
      N = NUL - NLL + 1
      CALL MATINV
      DO 40 I = NLL,NUL
      IJ = I - NLL + 1
      C(I)=0.0
      DO 40 J = NLL,NUL
      JJ = J - NLL + 1
      C(I) = C(I) + A(IJ,JJ)*B(J)
 40   CONTINUE
      DO 46  I=1,KE
      IF (I.GE.NLL) GO TO 43
      GO TO 42
 43   IF(I.LE.NUL) GO TO 45
 42   C(I) = BC(I)
      GO TO 46
45    BC(I)=C(I)
46    CONTINUE
      DO 38  L=1,KE
      B(L)=BB(L)
      DO 38  K=1,KE
38    A(L,K)=AA(L,K)
C
C     CALCULATION OF STANDARD DEVIATION
C
      DELTA = 0.0
      IF (IDELTA .EQ. 0) RETURN
      DO 50 I = 1,NUM
      CALL LAB(I,CAL)
      NTI = NT(I)
      DO 50 J = 1,NTI
      CALL ELLIPS(I,J,TT,PP,RS)
      CALL FUNC(TT,PP,BE)
      AREA = 0.0
      DO 60 K = 1,KE
 60   AREA = AREA + BE(K) * C(K)
      DIFF = R(J,I)*(1.0D0-AREA/RS)
 50   DELTA = DIFF*DIFF + DELTA
      NTI = 0
      DO 70 I = 1,NUM
 70   NTI = NTI + NT(I)
      DELTA = DELTA/NTI
      DELTA = DSQRT(DELTA)
      WRITE (6,15) DELTA
39    CONTINUE
      RETURN
      END


      SUBROUTINE LAB(I,CAL)
C ***********************************************************************
C *                                                                     *
C *   THIS SUBROUTINE CONVERTS FROM THE LAB TO THE                      *
C *   CRYSTAL-AXIS DIRECTIONS, WHERE WE ALLOW FOR A CONE ANGLE.         *
C *                                                                     *
C *   ONLY THE ANGLES FOR THE I-TH PLANE ARE COMPUTED.                  *
C *   FUTHER TRANSFORMATION IS ACCOMPLISHED IN SUBROUTINE TRANS.        *
C *   IF FURTHER TRANSFORMATION IS NOT WANTED,                          *
C *        A DUMMY SUBROUTINE MUST BE PROVIDED.                         *
C *                                                                     *
```

```
C ***********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 R
      REAL*4 T1,T2,P1,P2,TL
      DIMENSION THETA(600),PHI(600)
      DIMENSION IN(600,10),R(600,10)
      DIMENSION X(3),Y(3),Z(3),S(3),W(3)
      DIMENSION T1(10),T2(10),P1(10),P2(10)
      DIMENSION NL(10),NT(10),NZ(10),TL(10)
      COMMON/INTO/IN,R
      COMMON/TEEPEE/THETA,PHI
      COMMON/ANGLE/T1,T2,P1,P2
      COMMON/ORIENT/NT,NL,NZ,TL
      PI = 3.141592653589793
      T11 = T1(I)
      T22 = T2(I)
      P11 = P1(I)
      P22 = P2(I)
      CALL DGOTTA(X,T11,P11)
      CALL DGOTTA(Y,T22,P22)
      CALL DCRISS(X,Y,Z)
      CALL DCRISS (Z,X,Y)
      TLI = TL(I) * PI/180.
      YC = DCOS(TLI)
      YS = DSIN(TLI)
      NTI = NT(I)
      DO 60 J = 1,NTI
      PR = IN(J,I) - NZ(I)
      PR = PR * 0.1/CAL
      PR = PR * PI/180.
      ZC = DCOS(PR)
      ZS = DSIN(PR)
      DO 61 K = 1,3
      S(K) = X(K)*ZC + Y(K)*ZS
 61   W(K) = S(K)*YS + Z(K)*YC
      CALL TRANS(W,I,J)
      THETA(J) = DARCOS(W(3))
      PHI(J) = DATAN2(W(2),W(1))
 60   CONTINUE
      RETURN
      END

      SUBROUTINE MATINV
C ***********************************************************************
C *                                                                     *
C *   MATRIX INVERSION ROUTINE.                                         *
C *   USABLE ON ANY SQUARE MATRIX.                                      *
C *   BASED ON ANL F402S  --  AN AMDLIB PROGRAM                         *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 PIVOT
      DIMENSION A(91,91),PIVOT(100),INDEX(100)
      COMMON/OVEP/A,N
C
C     INITIALIZE DETERMINANT AND PIVOT ELEMENT ARRAY
```

```
C
      DETERM = 1.0
      DO 20 I = 1,N
      PIVOT(I) = 0.0
 20   INDEX(I) = 0.0
C
C     PERFORM SUCCESSIVE PIVOT OPERATIONS (GRAND LOOP)
C
      DO 550 I = 1,N
C
C     SEARCH FOR PIVOT FLEMENT AND EXTEND DETERMINANT PARTIAL PORDUCTS
C
      AMAX = 0.0
      DO 105 J = 1,N
      IF (PIVOT(J) .NE. 0.0) GO TO 105
      DO 100 K = 1,N
      IF (PIVOT(K) .NE. 0.0) GO TO 100
      TEMP = DABS(A(J,K))
      IF (TEMP .LT. AMAX) GO TO 100
      IROW = J
      ICOLUM = K
      AMAX = TEMP
 100  CONTINUE
 105  CONTINUE
 106  INDEX(I) = 4096 * IROW + ICOLUM
 107  J = IROW
 108  AMAX = A(J,ICOLUM)
      DETERM =  AMAX
C     DETERM = DETERM * AMAX
C
C     RETURN IF MATRIX IS SINGULAR (ZERO PIVOT) AFTER COLUMN INTERCHANGE
C
      IF (DETERM .EQ. 0.0) GO TO 600
C
      PIVOT(ICOLUM) = AMAX
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
      IF (IROW .EQ. ICOLUM) GO TO 260
      DETERM = -DETERM
      DO 200 K = 1,N
      SWAP = A(J,K)
      A(J,K) = A(ICOLUM,K)
      A(ICOLUM,K) = SWAP
 200  CONTINUE
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
 260  K = ICOLUM
      A(ICOLUM,K) = 1.0
      DO 350 K = 1,N
      A(ICOLUM,K) = A(ICOLUM,K)/AMAX
 350  CONTINUE
C
C     REDUCE NON-PIVOT ROWS
C
 380  DO 550 J = 1,N
      IF (J .EQ. ICOLUM) GO TO 550
      T = A(J,ICOLUM)
      A(J,ICOLUM) = 0.0
```

```
      DO 450 K = 1,N
      A(J,K) = A(J,K) - A(ICOLUM,K) * T
 450  CONTINUE
 550  CONTINUE
C
C     INTERCHANGE COLUMNS AFTER ALL PIVOT OPERATIONS HAVE BEEN PERFORMED
C
  600 DO 710 I = 1,N
      I1 = N + 1 - I
      K = INDEX(I1)/4096
      ICOLUM = INDEX(I1) - 4096 * K
      IF (K .EQ. ICOLUM) GO TO 710
      DO 705 J = 1,N
      SWAP = A(J,K)
      A(J,K) = A(J,ICOLUM)
      A(J,ICOLUM) = SWAP
 705  CONTINUE
 710  CONTINUE
      IF (DETERM .EQ. 0.0) WRITE (6,1)
 1    FORMAT (//' MATRIX IS TOO SINGULAR.'//)
      RETURN
      END


      INTEGER FUNCTION LTOM*4 (IJ,II)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE CALCULATES THE NUMBER OF TERMS IN AN EXPANSION       *
C *   WHICH INCLUDES TERMS OF ORDERS UP TO L = IJ FOR THE SYMMETRY      *
C *   GROUP II.                                                         *
C *                                                                     *
C *          CODE NUMBERS FOR SYMMETRY GROUPS:                          *
C *    1  --  OH                                                        *
C *    2  --  D6H                                                       *
C *    3  --  C6H                                                       *
C *    4  --  D3H                                                       *
C *    5  --  S6                                                        *
C *    6  --  D4H                                                       *
C *    7  --  C4H                                                       *
C *    8  --  VH                                                        *
C *    9  --  C2H                                                       *
C *   10  --  S2                                                        *
C *                                                                     *
C ***********************************************************************
      I = IJ
C
C     SYMMETRY GROUP OH.
C
      IF (II .GT. 1) GO TO 2
      IF (I .GT. 60) I = 60
      J = I + 1
      LTOM = 1
      IF (I .LT. 4) RETURN
      DO 1 K = 5,J,2
      L = K  - 1
      I2 = 0
      IZZ = L - 2
      I1 = MOD(IZZ,12)
      IF (I1 .EQ. 0) I2 = -1
      JEN = L/12 + I2 + 1
```

```
      LTOM = LTOM + JEN
 1    CONTINUE
      RETURN
C
C     SYMMETRY GROUP D6H
C
 2    IF (II .GT. 2) GO TO 3
      IF (I .GT. 40) I = 40
      LTOM = 0
      K = I + 1
      DO 19 L1 = 1,K,2
      L = L1 - 1
      M = L/6 + 1
 19   LTOM = LTOM + M
      RETURN
C
C     SYMMETRY GROUP C6H
C
 3    IF (II .GT. 3) GO TO 6
      IF (I .GT. 30) I = 30
      LTOM = 0
      K = I + 1
      DO 17 L1 = 1,K,2
      L = L1 - 1
      M = 2 * (L/6) + 1
 17   LTOM = LTOM + M
      RETURN
C
C     SYMMETRY GROUP D3H
C
 6    IF (II .GT. 4) GO TO 8
      IF (I .GT. 30) I = 30
      LTOM = 0
      K = I + 1
      DO 15 L1 = 1,K,2
      L = L1 - 1
      M = L/3 + 1
 15   LTOM = LTOM + M
      RETURN
C
C     SYMMETRY GROUP S6
C
 8    IF (II .GT. 5) GO TO 10
      IF (I .GT. 20) I = 20
      LTOM = 0
      K = I + 1
      DO 13 L1 = 1,K,2
      L = L1 - 1
      M = 2 * (L/3) + 1
 13   LTOM = LTOM + M
      RETURN
C
C     SYMMETRY GROUP D4H
C
 10   IF (II .GT. 6) GO TO 12
      IF (I .GT. 34) I = 34
      LTOM = 0
      K = I + 1
      DO 4 L1 = 1,K,2
      L = L1 - 1
```

```
      M = L/4 + 1
 4    LTOM = LTOM + M
      RETURN
C
C     SYMMETRY GROUP C4H
C
 12   IF (II .GT. 7) GO TO 14
      IF (I .GT. 24) I = 24
      LTOM = 0
      K = I + 1
      DO 11 L1 = 1,K,2
      L = L1 - 1
      M = 2 * (L/4) + 1
 11   LTOM = LTOM + M
      RETURN
C
C     SYMMETRY GROUP VH
C
 14   IF (II .GT. 8) GO TO 16
      IF (I .GT. 24) I = 24
      LTOM = 0
      K = I + 1
      DO 9 L1 = 1,K,2
      L = L1 - 1
      M = L/2 + 1
 9    LTOM = LTOM + M
      RETURN
C
C     SYMMETRY GROUP C2H
C
 16   IF (II .GT. 9) GO TO 18
      IF (I .GT. 16) I = 16
      LTOM = 0
      K = I + 1
      DO 5 L1 = 1,K,2
      L = L1 - 1
      M = L + 1
 5    LTOM = LTOM + M
      RETURN
C
C     SYMMETRY GROUP S2
C
 18   IF (II .GT. 10) GO TO 20
      IF (I .GT. 12) I = 12
      LTOM = 0
      K = I + 1
      DO 7 L1 = 1,K,2
      L = L1 - 1
      M = L + L + 1
 7    LTOM = LTOM + M
      RETURN
 20   LTOM = 0
      RETURN
      END

      INTEGER FUNCTION MTOL*4 (IJ,II)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE CALCULATES THE L VALUE WHICH WOULD PRODUCE A TOTAL   *
```

```
C *   OF IJ HARMONICS.  THE SYMMETRY GROUP IS II.                          *
C *   THIS IS THE INVERSE ROUTINE TO LTOM.                                 *
C *                                                                        *
C ************************************************************************
      I = IJ
      IF (I .GT. 91) I = 91
C
C     SYMMETRY GROUP OH
C
      IF (II .GT. 1) GO TO 4
      IF (I .GT. 1) GO TO 1
      MTOL = 0
      RETURN
 1    I = I - 1
      L = 4
 2    I2 = 0
      IZZ = L - 2
      I1 = MOD(IZZ,12)
      IF (I1 .EQ. 0) I2 = -1
      JEN = L/12 + I2 + 1
      I = I - JEN
      IF (I .LT. 0) GO TO 3
      L = L + 2
      GO TO 2
 3    MTOL = L - 2
      RETURN
C
C     SYMMETRY GROUP D6H
C
 4    IF (II .GT. 2) GO TO 7
      L = 0
 29   M = L/6 + 1
      I = I - M
      IF (I .LT. 0) GO TO 30
      L = L + 2
      GO TO 29
 30   MTOL = L - 2
      IF (MTOL .LT. 0) MTOL = 0
      RETURN
C
C     SYMMETRY GROUP C6H
C
 7    IF (II .GT. 3) GO TO 10
      L = 0
 26   M = 2 * (L/6) + 1
      I = I - M
      IF (I .LT. 0) GO TO 27
      L = L + 2
      GO TO 26
 27   MTOL = L - 2
      IF (MTOL .LT. 0) MTOL = 0
      RETURN
C
C     SYMMETRY GROUP D3H
C
 10   IF (II .GT. 4) GO TO 13
      L = 0
 23   M = L/3 + 1
      I = I - M
      IF (I .LT. 0) GO TO 24
```

```
      L = L + 2
      GO TO 23
 24   MTOL = L - 2
      IF (MTOL .LT. 0) MTOL = 0
      RETURN
C
C     SYMMETRY GROUP S6
C
 13   IF (II .GT. 5) GO TO 16
      L = 0
 20   M = 2 * (L/3) + 1
      I = I - M
      IF (I .LT. 0) GO TO 21
      L = L + 2
      GO TO 20
 21   MTOL = L - 2
      IF (MTOL .LT. 0) MTOL = 0
      RETURN
C
C     SYMMETRY GROUP D4H
C
 16   IF (II .GT. 6) GO TO 19
      L = 0
 5    M = L/4 + 1
      I = I - M
      IF (I .LT. 0) GO TO 6
      L = L + 2
      GO TO 5
 6    MTOL = L - 2
      IF (MTOL .LT. 0) MTOL = 0
      RETURN
C
C     SYMMETRY GROUP C4H
C
 19   IF (II .GT. 7) GO TO 22
      L = 0
 17   M = 2 * (L/4) + 1
      I = I - M
      IF (I .LT. 0) GO TO 18
      L = L + 2
      GO TO 17
 18   MTOL = L - 2
      IF (MTOL .LT. 0) MTOL = 0
      RETURN
C
C     SYMMETRY GROUP VH
C
 22   IF (II .GT. 8) GO TO 25
      L = 0
 14   M = L/2 + 1
      I = I - M
      IF (I .LT. 0) GO TO 15
      L = L + 2
      GO TO 14
 15   MTOL = L - 2
      IF (MTOL .LT. 0) MTOL = 0
      RETURN
C
C     SYMMETRY GROUP C2H
C
```

```
 25   IF (II .GT. 9) GO TO 28
      L = 0
 8    M = L + 1
      I = I - M
      IF (I .LT. 0) GO TO 9
      L = L + 2
      GO TO 8
 9    MTOL = L - 2
      IF (MTOL .LT. 0) MTOL = 0
      RETURN
C
C     SYMMETRY GROUP S2
C
 28   IF (II .GT. 10) GO TO 31
      L = 0
 11   M = L + L + 1
      I = I - M
      IF (I .LT. 0) GO TO 12
      L = L + 2
      GO TO 11
 12   MTOL = L - 2
      IF (MTOL .LT. 0) MTOL = 0
      RETURN
 31   MTOL = 0
      RETURN
      END


      SUBROUTINE TRANS(W,I,J)
C ***********************************************************************
C *                                                                     *
C *   DUMMY PROCEDURE FOR USE WHEN NO ADDITIONAL TRANSFORMATION         *
C *   IS NEEDED.                                                        *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION W(3),NCODE(600,15)
      COMMON/ROTATE/NCODE
      RETURN
      END


      SUBROUTINE TRANS(W,I,J)
C ***********************************************************************
C *   THIS ROUTINE PROVIDES A SPECIAL TRANSFORMATION TO BE USED WHEN    *
C *   D4H POCKETS OF THE FERMI SURFACE OF A CUBIC MATERIAL ARE TO       *
C *   BE ANALYZED.                                                      *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION W(3),NCODE(600,10)
      COMMON/ROTATE/NCODE
C
C     MAP ALL DATA TO A COMMON COORDINATE SYSTEM
C
      NCOD = NCODE(J,I)
      IF (NCOD .LE. 0) RETURN
C
C     ROTATE DATA ABOUT Y AXIS
C
```

```
      IF (NCOD .GT. 1) GO TO 64
      SAVE = W(3)
      W(3) = W(1)
      W(1) = -SAVE
      RETURN
C
C     ROTATE DATA ABOUT X AXIS
C
 64   SAVE = W(3)
      W(3) = W(2)
      W(2) = -SAVE
      RETURN
      END


      SUBROUTINE TRANS(W,I,J)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE IS USED FOR C2H POCKETS OF THE FERMI SURFACE OF      *
C *    IN THE ARSENIC STRUCTURE                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 EE1,EE2,EE3
      DIMENSION NCODE(600,10)
      DIMENSION W(3)
      COMMON/ROTATE/NCODE
      PI = 3.141592653589793
      EE1=2.0D0*PI/3.0D0
      EE2=0.0
      EE3=0.0
C
C     MAP ALL DATA TO A COMMON COORDINATE SYSTEM.
C
      NCOD = NCODE(J,I)
      IF (NCOD.EQ.0) GO TO 40
      IF (NCOD.EQ.2) EE1=-EE1
      CALL EULER(EE1,EE2,EE3,W,1)
 40   CONTINUE
      RETURN
      END


      SUBROUTINE ELLIPS(I,J,T,P,RS)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE IS DESIGNED FOR USE WITH THE OH CRYSTAL GROUP.       *
C *                                                                     *
C ***********************************************************************
      REAL*8 THETA(600),PHI(600)
      DIMENSION IN(600,10),R(600,10)
      COMMON/TEEPEE/THETA,PHI
      COMMON/INTO/IN,R
      RS = R(J,I)
      T = THETA(J)
      P = PHI(J)
      RETURN
      END
```

```
      SUBROUTINE ELLIPS(I,J,T,P,RS)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE IS FOR USE WHENEVER AN ELLIPSOIDAL TRANSFORMATION    *
C *   IS CONTEMPLATED.                                                  *
C *                                                                     *
C ***********************************************************************
      REAL*8 TT,PP,ST,CT,SP,CP,ST2,CT2,SP2,CP2,THETA,PHI
      DIMENSION THETA(600),PHI(600)
      DIMENSION IN(600,10),R(600,10)
      COMMON/TEEPEE/THETA,PHI
      COMMON/INTO/IN,R
      COMMON/CARRY/NUT,LUMP,ALPHA,A2,GAMMA,G2,E1,E2,E3,JJJJ
      TT = THETA(J)
      PP = PHI(J)
      IF (LUMP .EQ. 0) GO TO 93
      IF (GAMMA .EQ. 1.0) GO TO 151
      IF (JJJJ.NE.0) CALL OILER(E1,E2,E3,TT,PP,1)
      ST = DSIN(TT)
      CT = DCOS(TT)
      SP = DSIN(PP)
      CP = DCOS(PP)
      ST2 = ST * ST
      CT2 = CT * CT
      SP2 = SP * SP
      CP2 = CP * CP
      RS = GAMMA * ALPHA * R(J,I) * DSQRT(ST2*(SP2+CP2/A2)+CT2/G2)
      TT = DARCOS(CT/DSQRT(ST2*G2*(SP2+CP2/A2)+CT2))
      SP = SP * ALPHA
      CALL DFLOP(SP,CP,PP)
      T = TT
      P = PP
      RETURN
 93   RS = 1.0/(R(J,I) * R(J,I))
      T = TT
      P = PP
      RETURN
 151  RS = R(J,I)
      T = TT
      P = PP
      RETURN
      END
```

VECTOR SUBROUTINES

```
      SUBROUTINE GOTTA(X,T,P)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE PRODUCES A CARTESIAN UNIT VECTOR X THAT IS           *
C *   IN THE DIRECTION (T,P).                                           *
C *                                                                     *
C ***********************************************************************
      DIMENSION X(3)
      ST = SIN(T)
      SP = SIN(P)
      CT = COS(T)
      CP = COS(P)
      X(1) = ST * CP
```

```
      X(2) = ST * SP
      X(3) = CT
      RETURN
      END

      SUBROUTINE DGOTTA(X,T,P)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE PRODUCES A CARTESIAN UNIT VECTOR THAT IS IN THE      *
C *   DIRECTION (T,P)                                                   *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(3)
      ST = DSIN(T)
      SP = DSIN(P)
      CT = DCOS(T)
      CP = DCOS(P)
      X(1) = ST * CP
      X(2) = ST * SP
      X(3) = CT
      RETURN
      END

      SUBROUTINE CRISS(A,B,C)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE PRODUCES A UNIT VECTOR THAT IS IN THE DIRECTION      *
C *   OF THE CROSS PRODUCT OF A AND B.                                  *
C *                                                                     *
C ***********************************************************************
      DIMENSION A(3),B(3),C(3)
      DO 10 L = 1,3
      K = L + 1
      M = MOD(L,3) + 1
      N = MOD(K,3) + 1
 10   C(L) = A(M) * B(N)  -  A(N) * B(M)
      CALL WOW(C)
      RETURN
      END

      SUBROUTINE DCRISS(A,B,C)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE PRODUCES A UNIT VECTOR IN THE DIRECTION OF THE       *
C *   CROSS PRODUCT OF A AND B.                                         *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      REAL*8 A(3),B(3),C(3)
      DO 10 L = 1,3
      K = L + 1
      M = MOD(L,3) + 1
      N = MOD(K,3) + 1
```

```
 10   C(L) = A(M) * B(N)  -  A(N) * B(M)
      CALL DWOW(C)
      RETURN
      END


      SUBROUTINE WOW(T)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE CONVERTS THE VECTOR T INTO A UNIT VECTOR IN THE      *
C *   SAME DIRECTION.                                                   *
C *                                                                     *
C ***********************************************************************
      DIMENSION T(3)
      S = 0.0
      DO 10 I = 1,3
 10   S = T(I)*T(I) + S
      S = SQRT(S)
      DO 20 I = 1,3
 20   T(I) = T(I)/S
      RETURN
      END


      SUBROUTINE DWOW(T)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE CONVERTS THE VECTOR T INTO A UNIT VECTOR IN THE      *
C *   SAME DIRECTION.                                                   *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      REAL*8 T(3), S
      S = 0.0
      DO 10 I = 1,3
 10   S = T(I) * T(I)  +  S
      S = DSQRT(S)
      DO 20 I = 1,3
 20   T(I) = T(I)/S
      RETURN
      END


      SUBROUTINE ORD(X)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE ORDERS THE THREE COMPONENTS OF X IN DECREASING       *
C *   ORDER OF MAGNITUDE.                                               *
C *                                                                     *
C *              X  >  Y  >  Z                                          *
C *                                                                     *
C ***********************************************************************
      DIMENSION X(3)
      DO 10 I = 1,3
 10   X(I) = ABS(X(I))
      DO 20 I = 1,2
      I1 = I + 1
      DO 20 J = I1,3
      DIF = X(I) - X(J)
```

```
      IF (DIF .GT. 0.0) GO TO 20
      XS = X(I)
      X(I) = X(J)
      X(J) = XS
 20   CONTINUE
      RETURN
      END


      SUBROUTINE DORD(X)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE ORDERS THE THREE COMPONENTS OF X IN DECREASING       *
C *   ORDER OF MAGNITUDE.                                               *
C *                                                                     *
C *              X  >  Y  >  Z                                          *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(3)
      DO 1 I = 1,3
 1    X(I) = DABS(X(I))
      DO 2 I = 1,2
      I1 = I + 1
      DO 2 J = I1,3
      DIF = X(I) - X(J)
      IF (DIF .GT. 0.0) GO TO 2
      XS = X(I)
      X(I) = X(J)
      X(J) = XS
 2    CONTINUE
      RETURN
      END


      SUBROUTINE FLOP(S,C,T)
C ***********************************************************************
C *                                                                     *
C *   THIS SUBROUTINE TAKES THE SINE AND COSINE OF AN ANGLE             *
C *   T AND PRODUCES THE ANGLE T, WHERE T IS IN THE RANGE               *
C *                  0  <=  T  <=  2*PI.                                *
C *                                                                     *
C ***********************************************************************
      REAL*8 SS,CC,TT,PI
      PI = 3.141592653589793
      SS = S
      CC = C
      IF (CC .NE. 0.0) GO TO 10
      IF (SS .GE. 0.0) GO TO 22
      TT = PI * 1.500000000000
      T = TT
      RETURN
 22   TT = PI * 0.500000000000
      T = TT
      RETURN
 10   IF (CC .LT. 0.0) GO TO 30
      TT = DATAN(SS/CC)
      IF (SS .GE. 0.0) GO TO 12
```

```
      TT = PI + PI + TT
 12   T = TT
      RETURN
 30   TT = DATAN(SS/CC)
      TT = TT + PI
      T = TT
      RETURN
      END


      SUBROUTINE DFLOP(S,C,T)
C ***********************************************************************
C *                                                                     *
C *   THIS SUBROUTINE TAKES THE SINE AND COSINE OF AN ANGLE T           *
C *   AND PRODUCES THE ANGLE T, WHERE T IS IN THE RANGE                 *
C *              0 <= T <= 2*PI.                                        *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      PI = 3.141592653589793
      IF (C .NE. 0.0) GO TO 10
      IF (S .GE. 0.0) GO TO 22
      T = PI * 1.500000000000D00
      RETURN
 22   T = PI * 0.500000000000
      RETURN
 10   IF (C .LT. 0.0) GO TO 30
      T = DATAN(S/C)
      IF (S .GE. 0.0) GO TO 12
      T = T + PI + PI
 12   RETURN
 30   T = DATAN(S/C)
      T = T + PI
      RETURN
      END


      SUBROUTINE GETTA(W,THETA,PHI)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE CALCULATES THE SPHERICAL POLAR DIRECTION             *
C *   (THETA,PHI) IN WHICH A VECOTR W LIES.                             *
C *                                                                     *
C ***********************************************************************
      REAL*8 X,Y,Z,T
      REAL*4 W(3),S(3)
      DO 1 I = 1,3
 1    S(I) = W(I)
      CALL WOW(S)
      X = S(1)
      Y = S(2)
      Z = S(3)
      CALL FLOP(Y,X,PHI)
      T = DARCOS(Z)
      THETA = T
      RETURN
      END
```

```
      SUBROUTINE DGETTA(W,T,P)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE CALCULATES THE SPHERICAL POLAR DIRECTION (T,P)       *
C *   IN WHICH THE VECTOR W LIES.                                       *
C *                                                                     *
C *   DOUBLE PRECISION ROUTINE.                                         *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION W(3),S(3)
      DO 1 I = 1,3
 1    S(I) = W(I)
      CALL DWOW(S)
      X = S(1)
      Y = S(2)
      Z = S(3)
      CALL DFLOP(Y,X,P)
      T = DARCOS(Z)
      RETURN
      END

      SUBROUTINE OILER(THETA,PHI,PSI,T,P,I)
C ***********************************************************************
C *                                                                     *
C *   THIS ROUTINE ROTATES A VECTOR IN THE DIRECTION (T,P) BY THE       *
C *   EULER ANGLES (THETA,PHI,PSI).                                     *
C *                                                                     *
C ***********************************************************************
      REAL*8 T,P,X(3)
      REAL*4 THETA,PHI,PSI
      CALL DGOTTA(X,T,P)
      CALL EULER(THETA,PHI,PSI,X,I)
      CALL DGETTA(X,T,P)
      RETURN
      END

      SUBROUTINE EULER(THETA,PHI,PSI,X,IZ)
C ***********************************************************************
C *                                                                     *
C *   THIS SUBROUTINE ROTATES THE CARTESIAN VECTOR X BY THE             *
C *   EULER ANGLES (THETA,PHI,PSI).                                     *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 THETA,PHI,PSI
      DIMENSION X(3),Y(3),A(3,3)
      TT = THETA
      PH = PHI
      PS = PSI
      CTT = DCOS(TT)
      CPH = DCOS(PH)
      CPS = DCOS(PS)
      STT = DSIN(TT)
      SPH = DSIN(PH)
      SPS = DSIN(PS)
      DO 1 I = 1,3
```

```
 1    Y(I) = X(I)
      A(1,1) = CPS * CPH - CTT * SPH * SPS
      A(1,2) = -SPS * CPH - CTT * SPH * CPS
      A(1,3) = STT * SPH
      A(2,1) = CPS * SPH + CTT * CPH * SPS
      A(2,2) = -SPS * SPH + CTT * CPH * CPS
      A(2,3) = -STT * CPH
      A(3,1) = STT * SPS
      A(3,2) = STT * CPS
      A(3,3) = CTT
      IF (IZ .LT. 0) GO TO 3
 2    DO 10 I = 1,3
      X(I) = 0
      DO 10 J = 1,3
10    X(I)=A(I,J)*Y(J)+X(I)
      RETURN
 3    DO 4 I = 1,2
      K = I + 1
      DO 4 J = K,3
      SAVE = A(I,J)
      A(I,J) = A(J,I)
 4    A(J,I) = SAVE
      GO TO 2
      END
```

SPECIAL PURPOSE SUBROUTINES

```
      SUBROUTINE WIG(J1,N1,N,AN)
C ***********************************************************************
C *                                                                     *
C *   SUBROUTINE TO CALCULATE THE MATRIX ELEMENTS FOR A                 *
C *   ROTATION OF PI/2 ABOUT THE Y-AXIS.    VERSION II.                 *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      M1 = N1
      M = N
      J = J1
      IF (M1 .GE. M) GO TO 2
      MS = M1
      M1 = M
      M = MS
    2 CONTINUE
C
C     CALCULATION OF THE HYPERGEOMETRIC FUNCTION.
C
      NA = M1 - J - 1
      NB = -M - J - 1
      NC = M1 - M
      AN1 = 1.
      R = 1.
      MS = - MAX0(NA,NB) - 1
      NRE = 0
      DO 100 I = 1,MS
      NSE = 0
      NA = NA + 1
      NB = NB + 1
      NC = NC + 1
```

```
      A = NA
      B = NB
      C = NC
      R = R * A * B/C
      IF (R .LE. 1.0D20) GO TO 60
      NRE = NRE + 1
      R = R * 1.0D-20
   60 S = R
      DO 90 K = 1,I
      D = K
      S = S/D
      IF (S .GT. 1.0D-20) GO TO 80
      NSE = NSE - 1
      S = S * 1.0D20
   80 CONTINUE
   90 CONTINUE
      NE = NRE + NSE
      NSE = IABS(NE)
      IF (NE) 110,120,130
  110 DO 140 K = 1,NSE
  140 S = S * 1.0D-20
      GO TO 120
  130 DO 150 K = 1,NSE
  150 S = S * 1.0D20
  120 CONTINUE
      NE = MOD(I,2)
      IF (NE) 160,170,160
  160 AN1 = AN1 - S
      GO TO 180
  170 AN1 = AN1 + S
  180 CONTINUE
  100 CONTINUE
      AN = AN1
C
C     COMPUTATION OF THE CLEBSCH-GORDON - LIKE FACTOR.
C
      AN1 = 1.
      NA = J - M
      NB = J + M1
      NC = J + M
      ND = J - M1
      NE = M1 - M
      J2 = 2 * J
      DO 200 I = 1,J2
      A = NA
      B = NB
      C = NC
      D = ND
      E = NE
      IF (NA .LT. 1) A = 1.
      IF (NB .LT. 1) B = 1.
      IF (NC .LT. 1) C = 1.
      IF (ND .LT. 1) D = 1.
      IF (NE .LT. 1) E = 1.
      Z = A * B/(C * D * E * E * 2.0000000000000)
      Z = DSQRT(Z)
      AN1 = AN1 * Z
      NA = NA - 1
      NB = NB - 1
      NC = NC - 1
```

```
      ND = ND - 1
      NE = NE - 1
  200 CONTINUE
      AN = AN * AN1
      RETURN
      END


      SUBROUTINE EIGEN (A,B,NSUB,VALU,MSUB,ANORM,NMAX)
C ***********************************************************************
C *                                                                     *
C *   EIGENVALUES AND EIGENVECTORS OF A REAL SYMMETRIC MATRIX           *
C *                                                                     *
C ***********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2 INDEX
      DIMENSION A(NMAX,NSUB), B(NMAX,NSUB), VALU(MSUB)
      DIMENSION     DIAG(100), SUPERD(100), WVEC(100), PVEC(100),
     1              QVEC(100), VALL(100), Q(100), U(100),
     2              INDEX(100), FACTOR(100), V(100), T(100,3)
      COMMON /F202/ DIAG, SUPERD, WVEC, PVEC, T
      EQUIVALENCE (WVEC(1),VALL(1),FACTOR(1),U(1))
      EQUIVALENCE (PVEC(1),QVEC(1),Q(1),V(1))
      DATA E1/Z3420000000000000/
C
C     HOUSEHOLDER SIMILARITY TRANSFORMATION TO CO-DIAGONAL FORM
C
      N=NSUB
      M=MSUB
      IF (N.LT.0) GO TO 220
      IF (M.LE.0.OR.N.LE.2) GO TO 50
C
C     GENERATE IDENTITY MATRIX
C
      DO 40 I=2,N
      DO 40 J=2,N
      B(J,I)=0.0
      IF (I.EQ.J) B(J,I)=1.0
   40 CONTINUE
C
   50 DO 200 I=1,N
C
C     REDUCE COLUMN OF MATRIX
C
      I1=I+1
      I2=I1+1
      IF (I2.GT.N) GO TO 160
      SUM=0.0
      DO 70 J=I2,N
   70 SUM=SUM+A(J,I)**2
      IF (SUM.EQ.0.0) GO TO 160
      J=I1
      TEMP=A(J,I)
      SUM=DSQRT(SUM+ TEMP  **2)
      A(J,I )=-DSIGN(SUM, TEMP  )
      WVEC(J )=DSQRT(      1.0+DABS( TEMP  )/SUM)
      DIV=DSIGN(     WVEC(J )*SUM, TEMP  )
      DO 85 J=I2,N
   85 WVEC(J)=A(J,I)/DIV
      SCALAR=0.0
```

```
      DO 95 J=I1,N
      PVEC(J)=0.0
      DO 90 K=I1,N
   90 PVEC(J)=PVEC(J)+A(K,J)*WVEC(K)
      SCALAR=SCALAR+PVEC(J)*WVEC(J)
   95 CONTINUE
      SCALAR=SCALAR/2.0
      DO 120 J=I1,N
      QVEC(J)=PVEC(J)-SCALAR*WVEC(J)
      DO 120 K=I1,J
      A(K,J)=A(K,J)-(WVEC(K)*QVEC(J)+WVEC(J)*QVEC(K))
      A(J,K)=A(K,J)
  120 CONTINUE
      IF (M.LE.0) GO TO 160
C
C     SAVE ROTATION FOR LATER APPLICATION TO CO-DIAGONAL VECTORS
C
      DO 150 K=2,N
      TEMP=0.0
      DO 140 J=I1,N
  140 TEMP=TEMP+WVEC(J)*B(J,K)
      DO 150 J=I1,N
      B(J,K)=B(J,K)-WVEC(J)*TEMP
  150 CONTINUE
C
C     MOVE CO-DIAGONAL FORM ELEMENTS FOR ITERATIVE PROCEDURE
C
  160 J=I
      DIAG(I)=A(J,I)
      IF (I.NE.N) SUPERD(I)=A(J+1,I)
  200 CONTINUE
C
C     GIVENS EIGENVALUE ITERATION FROM STURM CHAIN OF CO-DIAGONAL MINORS
C
  220 N=IABS (N)
      M=IABS (M)
C
C     CALCULATE NORM OF MATRIX AND INITIALIZE EIGENVALUE BOUNDS
C
      ANORM2=DIAG(1)**2
      IF (N.EQ.1) GO TO 235
      DO 230 L=2,N
      Q(L-1)=SUPERD(L-1)**2
      ANORM2=DIAG(L)**2+Q(L-1)+Q(L-1)+ANORM2
  230 CONTINUE
  235 ANORM=DSQRT(ANORM2)
      IF (M.EQ.0) RETURN
      DO 240 L=1,M
      VALU(L)=ANORM
      VALL(L)=-ANORM
  240 CONTINUE
      EPS1=ANORM*E1
      IF (EPS1.EQ.0.0) RETURN
C
  250 DO 570 L=1,M
C
C     CHOOSE NEW TRIAL VALUE WHILE TESTING BOUNDS FOR CONVERGENCE
C
  260 TAU=(VALU(L)+VALL(L))/2.0
      IF (2.0*(TAU-VALL(L)).LE.E1*DABS(TAU)) GO TO 570
```

```
C
C     DETERMINE SIGNS OF PRINCIPAL MINORS
C
      MATCH=0
      T2=0.0
      T1=1.0
      DO 450 L1=1,N
      P=DIAG(L1)-TAU
      IF (T2.NE.0.0) GO TO 330
  300 T1=DSIGN(1.0D0,T1)
  330 IF (T1.NE.0.0) GO TO 400
      T0=-DSIGN(1.0D0,T2)
      T2=0.0
      IF (Q(L1-1).NE.0.0) GO TO 410
      GO TO 300
  400 T0=P-Q(L1-1)*T2/T1
      T2=1.0
C
C     COUNT AGREEMENTS IN SIGN (ZERO CONSIDERED POSITIVE)
C
  410 IF (T0) 440, 420, 430
  420 T2=T1
      IF (T2) 440, 430, 430
  430 MATCH=MATCH+1
  440 T1=T0
  450 CONTINUE
C
C     ESTABLISH TIGHTER BOUNDS ON EIGENVALUES
C
      DO 530 L1=L,M
      IF (L1.LE.MATCH) GO TO 500
      IF (VALU(L1).LE.TAU) GO TO 260
      VALU(L1)=TAU
      GO TO 530
  500 VALL(L1)=TAU
  530 CONTINUE
      GO TO 260
C
  570 CONTINUE
C
C     EIGENVECTORS OF CO-DIAGONAL SYMMETRIC MATRIX--INVERSE ITERATION
C
      M=MSUB
      IF (M.LT.0) RETURN
C
      DO 970 I=1,M
C
C     CHECK FOR REPEATED VALUE
C
      IF (I.EQ.1) GO TO 725
      IF (VALU(I-1)-VALU(I).LT.1.0E+4*EPS1) GO TO 730
  725 I1=-1
  730 I1=I1+1
C
C     TRIANGULARIZE CO-DIAGONAL FORM AFTER EIGENVALUE SUBTRACTION
C
      DO 760 L=1,N
      V(L)=EPS1*(I1*L+1)
      T(L,2)=DIAG(L)-VALU(I)
      IF (L.LT.N) GO TO 740
```

```
      T(L,3)=0.0
      GO TO 760
  740 T(L,3)=SUPERD(L)
      IF (T(L,3).EQ.0.0) T(L,3)=EPS1
      T(L+1,1)=T(L,3)
  760 CONTINUE
      DO 820 J=1,N
      T(J,1)=T(J,2)
      T(J,2)=T(J,3)
      T(J,3)=0.0
      GO TO 780
  770 T(J,1)=EPS1
  780 VTEMP=DABS(T(J,1))
      IF (VTEMP.LT.EPS1) GO TO 770
      IF (J.EQ.N) GO TO 820
      INDEX(J)=0
      IF (DABS(T(J+1,1)).LE.VTEMP) GO TO 810
      INDEX(J)=1
      DO 800 K=1,3
      VTEMP=T(J,K)
      T(J,K)=T(J+1,K)
      T(J+1,K)=VTEMP
  800 CONTINUE
  810 VTEMP    =T(J+1,1)/T(J,1)
      FACTOR(J)=VTEMP
      T(J+1,2)=T(J+1,2)-  VTEMP  *T(J,2)
      T(J+1,3)=T(J+1,3)-  VTEMP  *T(J,3)
  820 CONTINUE
      ITER=1
      IF (I1.GT.0) GO TO 920
C
C     BACK SUBSTITUTE TO OBTAIN EIGENVECTOR
C
  860 DO 870 L1=1,N
      L=N+1-L1
      V( L )=(V( L )-T(L,2)*V( L+1 )-T(L,3)*V( L+2 ))/T(L,1)
  870 CONTINUE
      GO TO (875,920), ITER
C
C     PERFORM SECOND ITERATION
C
  875 ITER=2
      IF (N.EQ.1) GO TO 860
  880 DO 910 L=2,N
      IF (INDEX(L-1).EQ.0) GO TO 900
      VTEMP=V(L-1)
      V(L)=VTEMP
      V(L-1)=V(L)
  900 V(L)=V(L)-FACTOR(L-1)*V(L-1)
  910 CONTINUE
      GO TO 860
C
C     ORTHOGONALIZE VECTOR TO OTHERS ASSOCIATED WITH REPEATED VALUE
C
  920 IF (I1.EQ.0) GO TO 945
      DO 940 L1=1,I1
      K=I-L1
      VTEMP=0.0
      DO 930 J=1,N
  930 VTEMP=VTEMP+A(J,K)*V(J)
```

```
      DO 940 J=1,N
  940 V(J)=V(J)-A(J,K)*VTEMP
      GO TO (880,945), ITER
C
C     NORMALIZE VECTOR
C
  945 VNORM2=0.0
      DO 950 L=1,N
  950 VNORM2=VNORM2+V( L )**2
      VNORM=DSQRT(VNORM2)
      DO 960 J=1,N
  960 A(J,I)=V( J )/VNORM
C
  970 CONTINUE
C
C     ROTATION OF CO-DIAGONAL VECTORS INTO MATRIX EIGENVECTORS
C
      N=NSUB
      IF (N.LE.2) RETURN
      DO 990 I=1,M
      DO 980 K=2,N
      U(K)=0.0
      DO 980 J=2,N
  980 U(K)=U(K)+R(J,K)*A(J,I)
      DO 990 J=2,N
  990 A(J,I)=U(J)
C
      RETURN
      END
```

ACKNOWLEDGMENTS

We wish to thank Miss Martha Katzin for valuable discussions and programming assistance, and the staff of the Applied Mathematics Division at Argonne National Laboratory for excellent computational service.

ARGONNE NATIONAL LAB WEST
3 4444 00007906 1

B