LBL-1418 c.2

A COMPUTER PROGRAM TO CALCULATE LOW ENERGY ELECTRON DIFFRACTION INTENSITIES

Marlo Martin

December 1972

Prepared for the U.S. Atomic Energy Commission under Contract W-7405-ENG-48

TWO-WEEK LOAN COPY

This is a Library Circulating Copy which may be borrowed for two weeks. For a personal retention copy, call Tech. Info. Division, Ext. 5545

LBL-1418 c.2

DISCLAIMER

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

A COMPUTER PROGRAM TO CALCULATE LOW ENERGY ELECTRON DIFFRACTION INTENSITIES

Marlo Martin

Inorganic Materials Research Division, Lawrence Berkeley Laboratory
and Department of Chemistry; University of California,
Berkeley, California

Introduction

This report consists of a detailed writeup of a multiple scattering Low Energy Electron Diffraction program. The first section exhibits the essential features of the formalism used in the computation, and references are made to the numbered equations in a later section where the program is described in detail. This report is not intended as a substitute for a proper derivation of the formulas used. The reader is referred especially to references 1, 2 and 3 for this purpose. Section II presents the commonly used models used to account for the "inner potential" and the "inelastic damping" of the scattered electron. Section III is concerned with the presentation of the formalism for introducing temperature dependence into the multiple scattering problem. In Appendix A, the computer program is described taking each subroutine separately. Appendix B is included which reproduces a listing of the program plus sample input and output formats.

I. The t-matrix method for LEED calculations.

The LEED calculation and the computer program which is the subject of this report is based on the t-matrix approach to the electron-crystal scattering problem. The method to be followed was first set forth by Beeby[1] who derived the multiple scattering, or t-matrix method, but did not include the inelastic processes which are responsible for the severe attenuation of the reflected elastic electron intensities. This attenuation of the elastically scattered electrons was introduced phenomenologically by Duke and Tucker[2] who related an electron-electron mean free path for inelastic collisions, λ_{ee}, to the complex electron self energy of an electron moving through a crystal. It is the presence of this damping factor in an exponential which rapidly attenuates the elastic electron beams and makes multiple scattering summations converge rapidly. We employ the notation used by Laramore and Duke[3] in our description of the t-matrix method and the computer program we have constructed.

The differential cross section of an elastically reflected LEED beam is proportional to the observed intensity.

$$\left.\frac{d^2\sigma}{dEd\Omega}\right|_{\underset{\sim}{k}_i \to \underset{\sim}{k}_f} = \frac{m^2}{(2\pi\hbar^2)^2}\,\delta(E_f-E_i)\;\left|I(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E)\right|^2 . \qquad (1)$$

The differential cross section is taken with respect to the energy of the incident (and emerging) beam and the solid angle $d\Omega$. The delta function restricts this description to elastic scattering processes for which the incident and emerging propagation vectors at points far outside the

crystal surface are $\underset{\sim}{k}_i$ and $\underset{\sim}{k}_f$, respectively. The mass of an electron is denoted by m. In Figure 1 the propagation vectors and their projections onto the surface are shown.

The quantity $I(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E)$ is obtained by employing the two-dimensional periodicity of the surface, resulting in the expression

$$I(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E) = \frac{(2\pi)^2}{A}\sum_{\lambda}\sum_{\underset{\sim}{g}} e^{-i\underset{\sim}{g}\cdot\underset{\sim}{a}_\lambda}\,\delta(\underset{\sim}{k}_{f_\parallel}-\underset{\sim}{k}_{i_\parallel}-\underset{\sim}{g})\; e^{-i(\underset{\sim}{k}_{f_\perp}-\underset{\sim}{k}_{i_\perp})d_\lambda}\cdot T_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E) \qquad (2)$$

The normalization factor $(2\pi)^2/A$ (where A is the area of the two dimensional unit cell of the surface layer) and the summation over the two dimensional reciprocal lattice vectors $\underset{\sim}{g}$ results from performing a summation over the periodic lattice positions in the surface layer. This also gives rise to the delta function which restricts the difference between the components of the wavevectors $\underset{\sim}{k}_i$ and $\underset{\sim}{k}_f$ parallel to the surface ($\underset{\sim}{k}_{i_\parallel}$ and $\underset{\sim}{k}_{f_\parallel}$) to be a reciprocal lattice vector $\underset{\sim}{g}$. In other words, this delta function, which is obtained from imposing the condition of an infinitely extended perfect two dimensional lattice, describes the experimentally observed pattern of discrete spots observed on the LEED screen. The two dimensional vector $\underset{\sim}{a}_\lambda$ is the shift of the origin position of the λth plane parallel to the surface from an arbitrary zero position (often taken to be the origin position of the surface plane). We choose a coordinate system in which the $(\hat{x},\hat{y})$ directions are in a plane parallel to the surface and $\hat{z}$ is the inward

normal to the surface. The remaining exponential term in eq. (2) accounts for the phase difference between the normal components of the incident and emerging beams, $k_{f_\perp}$ and $k_{i_\perp}$, due to the perpendicular distance between the origin and the λth plane, d_λ.

The factor $T(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E)$ is the t-matrix for the λth layer. That is to say, it is the matrix element for all orders of scattering by the λth layer, in the presence of all the other layers, between incident and final plane wave states of energy E denoted by the asymptotic wave vectors $\underset{\sim}{k}_i$ and $\underset{\sim}{k}_f$. This factor can be conveniently expressed in terms of two intermediate quantities whose physical significance can readily be appreciated. We introduce $\tau_\lambda(\underset{\sim}{k}_2,\underset{\sim}{k}_1;E)$ as the t-matrix for scattering an initial plane wave $\underset{\sim}{k}_1$ into a final plane wave $\underset{\sim}{k}_2$ by a single atomic layer parallel to the surface. This includes the effects of all possible multiple scattering events within the λth plane and ignores all scatterings between different planes. The second factor $G^{\lambda\lambda'}(\underset{\sim}{k},\underset{\sim}{k}_i;E)$ is the propagator for scattering between planes λ' and λ. It contains the phase and amplitude factor for the sum of all spherically scattered waves measured at the origin of the λth plane due to all the ion core locations in plane λ'. Detailed expressions for τ_λ and $G^{\lambda\lambda'}$ will be presented.

$$T_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E) = \tau_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E) + \sum_{\lambda'\neq\lambda}\sum_{\underset{\sim}{k}} \tau_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k};E)\, G^{\lambda\lambda'}(\underset{\sim}{k},\underset{\sim}{k}_i;E) \cdot T_{\lambda'}(\underset{\sim}{k},\underset{\sim}{k}_i;E) \qquad (3)$$

The first term in this expression contains the effect of scattering of the incident beam, $\underset{\sim}{k}_i$, into the final beam, $\underset{\sim}{k}_f$, by means of all single and multiple scatterings within layer λ. The second term represents the effects of electrons scattered from all planes $\lambda' \neq \lambda$ summed over all propagation directions $\underset{\sim}{k}$ and impinging on plane λ, and include all subsequent multiple scatterings within plane λ. When the inelastic damping factor is sufficiently large and the t-matrices, τ_λ, are sufficiently weak, this expression can be expanded in a perturbation series by always expressing the final $T_{\lambda'}$-factor in terms of the right hand side of equation (3). It will be seen later, when we work with a particular matrix representation of this equation, that the inversion of a rather large matrix will yield the complete multiple scattering result for T_λ.

The layer t-matrix $\tau_\lambda(\underset{\sim}{k}_2,\underset{\sim}{k}_1;E)$ appearing in eq. (3) can finally be related to the scattering of electrons from individual ion cores of the crystal. We write the t-matrix for ion-core scattering from plane wave states $\underset{\sim}{k}_1$ to $\underset{\sim}{k}_2$ as $t_\lambda(\underset{\sim}{k}_2,\underset{\sim}{k}_1;E)$. We further restrict all ions within the plane labelled by λ to be of the same material. The resultant simplification in the calculation causes no unnecessary limitation in generality since additional planes, λ' (now referred to as sub-planes), made up of different types of ions can coexist even at the same perpendicular distance from the surface, i.e., $d_{\lambda'} = d_\lambda$. The expression for τ_λ can be written in a manner similar to eq. (3).

$$\tau_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E) = t_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E) + \sum_{\underset{\sim}{k}} t_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k};E)\, G^{sp}(\underset{\sim}{k},\underset{\sim}{k}_i;E) \cdot \tau_\lambda(\underset{\sim}{k},\underset{\sim}{k}_i;E) \qquad (4)$$

The structure of this equation is similar to that of eq. (3) with G^{sp} being interpreted as the sub-plane propagator which contains the phase and amplitude factor for the sum of all spherically scattered waves impinging on an ion-core due to scattering from all other ion cores in the same sub-plane. A schematic diagram of the relationship between these quantities is furnished in Figure 3.

A perturbation method can be applied to solve eq. (4) as long as relatively weak scattering materials are used.[4] A more general technique involves a matrix inversion, derived by Beeby,[1] in which all quantities appearing in eq. (4) are written in an appropriate matrix representation (to be specified below). Using a simplified notation, we write formally,

$$\tau_\lambda = t_\lambda + t_\lambda G^{sp} t_\lambda + t_\lambda G^{sp} t_\lambda G^{sp} t_\lambda + \ldots \tag{5a}$$

$$= t_\lambda (I + (G^{sp} t_\lambda) + (G^{sp} T_\lambda)^2 + (G^{sp} t_\lambda)^3 + \ldots)$$

This series can be summed to yield the exact expression,

$$\tau_\lambda = t_\lambda [I - G^{sp} t_\lambda]^{-1} \tag{5b}$$

In order to proceed further we must specify the matrix representation to be used with all the foregoing equations. It is convenient to use an angular momentum representation in which these quantities are expressed in terms of summations involving spherical harmonics. In what follows, we write the conventional complex spherical harmonic $Y_{\ell m}(\theta,\phi) \equiv Y_L(\underset{\sim}{k})$, where $\underset{\sim}{k}$ is a vector whose angular components are θ and ϕ. The subscript L is written to replace the combined (ℓ,m) subscripts. Values of L

corresponding to various values of ℓ and m are listed explicitly in the writeup for the spherical harmonic subroutine (YLM) appearing later in this report.

The t-matrix $t_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E)$ can be expanded in a series of Legendre polynomials $P_\ell(\cos\theta)$, where θ is the angle between $\underset{\sim}{k}_f$ and $\underset{\sim}{k}_i$. When the ion-core potential is spherically symmetrical there is no functional dependence of t_λ on the azimuthal angle ϕ. Due to this lack of dependence on the quantum number m, designation of ℓ or L alone suffices as a label for such quantities.

$$t_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E) = \sum_{\ell=0}^{\infty} t_\lambda^\ell(E)\, P_\ell(\cos\theta) \tag{6}$$

The Legendre polynomials can be expressed in terms of the spherical harmonics whose angular arguments are the vectors $\underset{\sim}{k}_f$ and $\underset{\sim}{k}_i$ by the formula,[5]

$$\frac{(2\ell+1)}{4\pi} P_\ell(\cos\theta) = \sum_{m=-\ell}^{+\ell} Y_{\ell m}^*(\underset{\sim}{k}_f)\, Y_{\ell m}(\underset{\sim}{k}_i) \tag{7}$$

The (ℓ,m) indices will be combined as indicated in the previous paragraph and the expansion can be rewritten as,

$$t_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E) = \sum_L t_\lambda^L(E)\, Y_L^*(\underset{\sim}{k}_f)\, Y_L(\underset{\sim}{k}_i) \tag{8a}$$

For the quantities $\tau_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E)$ and $T_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E)$ one can no longer assume spherical symmetry and the expansion in terms of Legendre

polynomials is not possible. However, it can be shown that any such function which depends on $\underset{\sim}{k}_i$ and $\underset{\sim}{k}_f$ (and not simply $\underset{\sim}{k}_f - \underset{\sim}{k}_i$) can be expanded in terms of a <u>double sum</u> over the spherical harmonics,[1,3]

$$\tau_\lambda(\underset{\sim}{k}_f, \underset{\sim}{k}_i; E) = \sum_L \sum_{L'} \tau_\lambda^{LL'}(E)\, Y_L^*(\underset{\sim}{k}_f)\, Y_{L'}(\underset{\sim}{k}_i) \tag{8b}$$

$$T_\lambda(\underset{\sim}{k}_f, \underset{\sim}{k}_i; E) = \sum_L \sum_{L'} T_\lambda^{LL'}(E)\, Y_L^*(\underset{\sim}{k}_f)\, Y_{L'}(\underset{\sim}{k}_i) \tag{8c}$$

Substitution of equation (8c) into equation (2) allows us to write

$$I(\underset{\sim}{k}_f, \underset{\sim}{k}_i; E) = \left| \sum_L \sum_{L'} Y_L^*(\underset{\sim}{k}_f) Y_{L'}(\underset{\sim}{k}_i) \sum_\lambda e^{-i(k_{f_\perp} - k_{i_\perp})d_\lambda}\, T_\lambda^{LL'}(E) \cdot \frac{(2\pi)^2}{A} \sum_{\underset{\sim}{g}} (\underset{\sim}{k}_{f_\parallel} - \underset{\sim}{k}_{i_\parallel} - \underset{\sim}{g})\, e^{-i\underset{\sim}{g}\cdot\underset{\sim}{a}_\lambda} \right|^2 \tag{9}$$

A similar substitution in equations (3) and (4) allows us to factor out the spherical hermonics and to work directly with the matrices $\tau_\lambda^{LL'}(E)$ and $T_\lambda^{LL'}(E)$. For a given maximum ℓ-value of ℓ_{max} the maximum value $L_{max} = (\ell_{max} + 1)^2$ become $(L_{max} \times L_{max})$ square matrices. Thus, the complexity and cost of the computer calculation will vary approximately as $(\ell_{max} + 1)^4$. The value of ℓ_{max} is set, in a particular problem, by the requirement that the diagonal matrix element $t_\lambda^L(E)$ for L corresponding to $(\ell_{max} + 1)$ be negligible. The t-matrix elements for the ion core scattering problem can be directly related to the phase shifts for the scattering amplitude $f(\theta)$,[6]

$$f(\theta) = \frac{1}{k} \sum_{\ell} (2\ell + 1)\, P_{\ell}(\cos\theta) \left[\frac{e^{i2\delta_{\ell}} - 1}{2i} \right], \tag{10}$$

by the equation

$$f(\underset{\sim}{k}_f, \underset{\sim}{k}_i) = -\frac{m}{2\pi\hbar^2}\, t(\underset{\sim}{k}_f, \underset{\sim}{k}_i) \quad . \tag{11}$$

Equating (10) and (11) and using the relationship of eq. (7), the ℓ-component of the expansion (eq. 8a) becomes,

$$t_{\lambda}^{\ell}(E) = i\, \frac{4\pi^2\hbar^2}{mk} \left(e^{i2\delta_{\ell}(E)} - 1 \right), \tag{12}$$

where $\delta_{\ell}(E)$ is the energy dependent phase shift that must be calculated separately from an assumed form for the ion-core potential. For the purposes of the computer program being described here, the $\delta_{\ell}(E)$ are supplied as necessary input parameters, and the reader is referred to the literature for various approximations which find frequent use in LEED calculations.[2,7,8]

We now write explicitly the matrix forms of equations (3) and (4). These functions can be written so that the independent variable is k, which is the absolute value of $\underset{\sim}{k}_i$ and $\underset{\sim}{k}_f$, and is related to the real part of the renormalized energy of the electron in the crystal by $E = \frac{\hbar^2}{2m}\, k^2$.

$$T_\lambda^{LL'}(k) = \tau_\lambda^{LL'}(k) + \sum_{\lambda'\neq\lambda} \sum_{L'',L'''} \tau_\lambda^{LL''}(k) G_{L''L'''}^{\lambda\lambda'}(\underset{\sim}{k}_i)\, T_{\lambda'}^{L'''L'}(k) \quad (13)$$

$$\tau_\lambda^{LL'}(k) = t_\lambda^{LL'}(k) + \sum_{L'',L'''} t_\lambda^{LL''}(k)\, G_{L''L'''}^{sp}(\underset{\sim}{k}_i)\, \tau_\lambda^{L'''L'}(k) \quad (14)$$

There are a number of manipulations necessary in order to transform equations (3) and (4) into (13) and (14), and these are detailed in the paper by Laramore and Duke.[3] The simplicity of the last two equations and their straightforward substitution in eq. (9) are offset by the complexity of the matrix forms of the propagators $G_{LL'}^{sp}(\underset{\sim}{k}_i)$ and $G_{LL'}^{\lambda\lambda'}(\underset{\sim}{k}_i)$. Explicitly they are,

$$G_{LL'}^{sp}(\underset{\sim}{k}_i) = \sum_{P\neq 0} e^{-\underset{\sim}{k}_i\cdot\underset{\sim}{P}} \sum_{L''} \left\{ \frac{(-i)^{\ell''}}{2\pi^2} \left[\frac{-m\pi i k}{\hbar^2}\right] h_{\ell''}^{(1)}(kP) \cdot Y_{L''}(\underset{\sim}{P})\, I(L,L',L'') \right\} \quad (15)$$

and

$$G_{LL'}^{\lambda\lambda'}(\underset{\sim}{k}_i) = \sum_{\underset{\sim}{P}} e^{i\underset{\sim}{k}_i[\underset{\sim}{P}+\underset{\sim}{d}_{\lambda'}-\underset{\sim}{d}_\lambda]} \sum_{L''} \left\{ \frac{(-i)^{\ell''}}{2\pi^2} \left[\frac{-m\pi i k}{\hbar^2}\right] h_{\ell''}^{(1)}(k|\underset{\sim}{P}+\underset{\sim}{d}_{\lambda'}-\underset{\sim}{d}_\lambda|) \cdot Y_{L''}(\underset{\sim}{P}+\underset{\sim}{d}_{\lambda'}-\underset{\sim}{d}_\lambda)\, I(L,L',L'') \right\} \quad (16)$$

A derivation of these results is presented in reference 3.

The vector $\underset{\sim}{P}$ is the distance from the origin in a plane parallel to the surface to any of the other lattice points in the same plane.

The summation in eq. (15) extends over all such lattice points except the one at the origin. In eq. (16), the summation includes $\underset{\sim}{P} = 0$ since in that case the difference between the origin positions of planes λ and λ' must be non-zero. When $\underset{\sim}{d}_\lambda$ is written as a vector it denotes the three dimensional vector from the origin chosen for the problem to the origin of the λth subplane, $\underset{\sim}{d}_\lambda = (\underset{\sim}{a}_\lambda, d_\lambda)$, where $\underset{\sim}{a}_\lambda$ is the displacement parallel to the crystal surface and d_λ is in the z-direction (inward normal to the surface). The spherical hankel function of the first kind $(h^{(1)}_{\ell''}(kP))$ describes the phase and amplitude of an outgoing spherical wave of energy $E = \frac{\hbar^2 k^2}{2m}$. The quantity $I(L,L',L'')$ is the angular integral of a product of three spherical harmonics,

$$I(L,L',L'') = \int d\Omega \; Y_L(\Omega) \; Y^*_{L'}(\Omega) \; Y^*_{L''}(\Omega)$$

$$= [(2\ell'+1)(2\ell''+1)/4\pi(2\ell+1)]^{\frac{1}{2}} \cdot$$

$$C(\ell',\ell'',\ell;\; m',m'',m)\; C(\ell',\ell'',\ell;0,0,0). \quad (17)$$

The present section reproduces the results of the t-matrix formalism which are to be incorporated into a computer program. There are two further points to be discussed in the next two sections before we can proceed to the details of the actual program. These involve the inner potential and inelastic damping factor, and the inclusion of temperature dependence in the formalism for the reflected beam intensities.

II. The inner potential and inelastic damping.

The propagation of an electron inside a metal differs from the propagation of an electron in a vacuum. There are three principle mechanisms operating on an electron moving through a metallic crystal. The first is a surface dipole layer due to the fact that metallic electron wavefunctions extend a short distance outside the surface layer of ion cores. This gives rise to the work function which presents a barrier (typically 2 to 4 eV in height) that must be overcome by an electron if it is to escape the metal surface. Conversely, an external electron entering the surface will experience an increase in its kinetic energy upon crossing the dipole layer region. This simplified model is sufficient for use with LEED calculations, though more detailed theoretical treatments are available.[9]

The second mechanism that alters the propagation of an electron inside a metallic crystal is the presence of positively charged ion cores. The third mechanism is due to the presence of valence and conduction electrons, which tend to screen the attractive potentials due to the ion cores. The screening is a dynamic effect and is dependent on the local free electron density and the momentum of the incident electron. The behavior of an electron moving through a uniform electron gas (the positive charge is assumed to be distributed uniformly in order to maintain charge neutrality) is well understood and yields a model which is useful for the understanding of the more complex problem at hand.[10]

Electron states at the fermi surface of a metal are infintely long lived. Electrons injected with higher energies form what are known as quasiparticle states, which possess finite lifetimes.[11] These quasiparticles may be thought of as a "bare" electron surrounded by a "cloud" of electrons which interact with it, and the dynamical behavior of such an entity is considerably more complex than that of a single free electron. The finite lifetime of a quasiparticle results when part of its energy is given up in the creation of an electron-hole pair, or by plasmon excitations. Since we are normally concerned with elastically diffracted electrons in LEED analysis, the effect of a finite lifetime is to diminish the elastic beam intensity as the beam travels through a metal. The quasiparticle energy can be expressed as the energy of a free electron whose wavevector outside the electron gas is $\underset{\sim}{k}$ plus a self-energy correction $\Sigma(\underset{\sim}{k})$, which is a complex quantity.

Figure 4 shows the relationship of the real parts of the various energies considered. ε is the kinetic energy of the electron far outside the crystal and ϕ is the work function. The real part of the electron self energy correction, $\Sigma_1(\underset{\sim}{k},\varepsilon)$, is a negative quantity as measured from the Fermi level. The real part of the quasiparticle energy, $E(\underset{\sim}{k},\varepsilon)$, is the effective "free particle" energy of the quasiparticle in the solid.

$$E(\underset{\sim}{k},\varepsilon) = \varepsilon - \Sigma_1(\underset{\sim}{k},\varepsilon) \tag{18}$$

The imaginary part of the quasiparticle energy $\Gamma(\underset{\sim}{k},\varepsilon)$ is simply the negative of the imaginary part of the self energy correction, $\Sigma_2(\underset{\sim}{k},\varepsilon)$.

$$\Gamma(\underset{\sim}{k},\varepsilon) = -\Sigma_2(\underset{\sim}{k},\varepsilon) \tag{19}$$

Finally, the complex propagation vector inside the electron gas is obtained from

$$[k_1(\varepsilon) + ik_2(\varepsilon)]^2 = \frac{2m}{\hbar^2}[E(\underset{\sim}{k},\varepsilon) + i\Gamma(\underset{\sim}{k},\varepsilon)] \tag{20}$$

Lundqvist has performed calculations[10] to determine $\Sigma(\underset{\sim}{k},\varepsilon)$ for a uniform electron gas of different densities. The real part, $\Sigma_1(\underset{\sim}{k},\varepsilon)$, is a negative quantity and can be identified with the inner potential correction familiar in LEED studies if we write $V_o(\varepsilon) = -\Sigma_1(\underset{\sim}{k},\varepsilon)$. The imaginary part, $\Sigma_2(\underset{\sim}{k},\varepsilon)$ is a positive quantity that goes to zero at the fermi energy, and attains a nearly constant value for energies greater than twice this value. The inner potential $V_o(\varepsilon)$ is nearly constant below an energy equal to twice the fermi energy, and diminishes at higher energies, approaching an asymptotic 1/k behavior.

It should be stressed that the foregoing discussion concerns a simplified model of the metallic crystal in which it is approximated as a uniform electron gas. While the concepts are useful, the validity of the detailed form of $\Sigma(\underset{\sim}{k},\varepsilon)$ for the uniform electron gas is questionable when applied to a real metal. The positive charge in a real metal is localized at the ion cores instead of being uniformly distributed, and the excitation of core electrons presents an additional inelastic loss mechanism. In the computer program under discussion here, we have made provisions for the calculation of $k_1(\varepsilon) + ik_2(\varepsilon)$

from the Lundqvist model,[10] and also from an optical model potential.

The optical model potential treats the real and imaginary parts of the quasiparticle energy as constants, independent of the incident electron energy ε over the LEED range, in which case

$$E(\underset{\sim}{k},\varepsilon) = \varepsilon + V_o \tag{21a}$$

$$\Gamma(\underset{\sim}{k},\varepsilon) = -\Sigma_2 \tag{21b}$$

Comparisons between theoretical calculations and experimental LEED profiles have not yet been able to resolve which of these treatments is most accurate. In Figure 5 we plot the (00) beam of the aluminum (100) surface using both approximations to allow direct comparison between the two approaches.

III. Temperature dependence of LEED calculations.

Laramore and Duke[3,12] have investigated the effects of lattice vibrations on the elastic LEED beam intensities. The result of their work is that temperature effects can be quite simply accounted for by introducing a quantity

$$b_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E) = \exp(-W_\lambda(\underset{\sim}{k}_f-\underset{\sim}{k}_i))\ t_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E) \tag{22}$$

where $W_\lambda(\underset{\sim}{k}_f-\underset{\sim}{k}_i)$ is the Debye-Waller factor of the lattice. If the expression for b_λ is inserted for t_λ in equations (4) and (5), the effects of thermal lattice motions on the LEED beams are properly taken into consideration.

The partial wave decomposition for b_λ is similar to that for t_λ of eq. (6a), except that b_λ need no longer possess the same spherical symmetry of t_λ and the ion-core potential.

$$b_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E) = \sum_L\sum_{L'} b_\lambda^{LL'}(k)\ Y_L^*(\underset{\sim}{k}_f)\ Y_{L'}(\underset{\sim}{k}_i) \tag{23}$$

In the model we use here, W_λ is taken to be a function possessing spherical symmetry; i.e., the Debye-Waller factor is independent of direction in the crystal.

$$W_\lambda(\underset{\sim}{k}_f-\underset{\sim}{k}_i) = W_\lambda(|\underset{\sim}{k}_f-\underset{\sim}{k}_i|) \tag{24}$$

In such a case, b_λ and t_λ are both of the same symmetry and the matrix $b_\lambda^{LL'}$ is diagonal, as is $t_\lambda^{LL'}$. A further simplification is achieved if we approximate the phonon spectrum by a Debye model.[3]

$$W_\lambda(\underset{\sim}{k}_f-\underset{\sim}{k}_i) = \frac{3h^2(\underset{\sim}{k}_f-\underset{\sim}{k}_i)^2}{2M\ k_B\ \Theta_D^\lambda}\left[\frac{1}{4}+\left(\frac{T}{\Theta_D^\lambda}\right)^2\right]\int_0^{\Theta_D^\lambda/T}\frac{x}{e^x-1}\,dx \qquad (25)$$

In this equation, M is the ionic mass, k_B is the Boltzmann constant, T is the absolute temperature of the crystal and Θ_D^λ is the effective Debye temperature of the λth subplane of the crystal. The propagation vectors $\underset{\sim}{k}_i$ and $\underset{\sim}{k}_f$ refer to the incident and final waves scattered by a given lattice site, and do not represent the intial and final wave-vectors for scattering from the entire crystal. Thus, $\underset{\sim}{k}_i$ and $\underset{\sim}{k}_f$ are different for each order of multiple scattering and the Debye-Waller factor cannot be factored out of the final expression as in the case of single or kinematic scattering.

The expression for W of eq. (25) can be written

$$W_\lambda(\underset{\sim}{k}_f-\underset{\sim}{k}_i) = (\underset{\sim}{k}_f-\underset{\sim}{k}_i)^2\ \bar{W}_\lambda$$

Since only elastic collision are considered, $|\underset{\sim}{k}_i| = |\underset{\sim}{k}_f| = k$ and

$$W_\lambda(\underset{\sim}{k}_f-\underset{\sim}{k}_i) = 2k^2\bar{W}_\lambda\ (1-\cos\theta),$$

where θ is the angle between the vectors $\underset{\sim}{k}_i$ and $\underset{\sim}{k}_f$. Using this expression for W_λ in Eq. (22) and expanding t_λ in spherical harmonics, one obtains

$$b_\lambda(\underset{\sim}{k}_1,\underset{\sim}{k}_2) = e^{-2k^2\bar{W}_\lambda}\sum_{\ell,m} t_\lambda^\ell(k)\ e^{2k^2\bar{W}_\lambda\cos\theta}\,Y^*_{\ell m}(\underset{\sim}{k}_2)\ Y_{\ell m}(\underset{\sim}{k}_1) \qquad (26)$$

The exponential term inside the summation can be expanded in a series of Legendre polynomials or of spherical harmonics,

$$e^{2k^2\bar{W}_\lambda \cos\theta} = \sum_{\ell'} A_{\ell'} P_{\ell'}(\cos\theta) = \sum_{\ell'} \frac{4\pi}{2\ell'+1} A_{\ell'} \cdot \sum_{m'=-\ell'}^{\ell'} Y^*_{\ell' m'}(\underset{\sim}{k}_2)\, Y_{\ell' m'}(\underset{\sim}{k}_1), \quad (27a)$$

where

$$A_{\ell'} = \left(\frac{2\ell'+1}{2}\right) \int_{-1}^{1} e^{2k^2\bar{W}_\lambda x} P_{\ell'}(x)\, dx \quad (27b)$$

Substitution of eq. (27a) into (26) yields,

$$b_\lambda(\underset{\sim}{k}_1, \underset{\sim}{k}_2) = e^{-2k^2\bar{W}_\lambda} \sum_{\ell} t^{\ell}_{\lambda}(k) \sum_{\ell'} \frac{4\pi}{2\ell'+1} A_{\ell'} \cdot \sum_{m,m'} Y^*_{\ell m}(\underset{\sim}{k}_2)\, Y_{\ell' m'}(\underset{\sim}{k}_1)\, Y_{\ell m}(\underset{\sim}{k}_1)\, Y^*_{\ell' m'}(\underset{\sim}{k}_2) \quad (28)$$

The summation over m and m' is performed using standard formulas for such products[13] and the double summation can be expressed as

$$\sum_{\ell''} \sum_{m''=-\ell''}^{+\ell''} Y^*_{\ell'' m''}(\underset{\sim}{k}_2)\, Y_{\ell'' m''}(\underset{\sim}{k}_1)\, \delta(\ell, \ell', \ell'') \quad (29)$$

where the delta function is unity for any combination of ℓ''-values which satisfy the combination rule for vectors $\ell'' = |\ell - \ell'|, \ldots \ell + \ell'$, and is zero when ℓ'' is outside these bounds. The expression for b_λ can then be simplified to

$$b_\lambda(\underset{\sim}{k}_1,\underset{\sim}{k}_2) = \sum_{\ell'',m''} \left\{ e^{-2k^2\bar{W}_\lambda} \sum_{\ell=0} \sum_{\ell'=|\ell''-\ell|}^{\ell''+\ell} \left[t_\lambda^\ell(k) \left(\frac{4\pi}{2\ell'+1}\right) A_{\ell'} \right] \right\} \cdot$$

$$Y^*_{\ell''m''}(\underset{\sim}{k}_2)\ Y_{\ell''m''}(\underset{\sim}{k}_1) \qquad (30)$$

The quantity inside the braces thus represents the matrix element of b_λ corresponding to ℓ'' and m'', and these are the values b_λ^L calculated by subroutine (FILLB) to be described shortly.

References

1. J. L. Beeby, J. Phys. (Proc. Phys. Sco.) [2], 1, 82 (1968).
2. C. B. Duke and C. W. Tucker, Jr., Surface Sci. 15, 231 (1969).
3. G. E. Laramore and C. B. Duke, Phys. Rev. B2, 4783 (1970).
4. J. B. Pendry, J. Phys. (Sol. St. Phys.) 4, 3095 (1971).
5. See, for example, J. D. Jackson, "Classical Electrodynamics" (p. 67, 68) Wiley, New York (1962).
6. L. I. Schiff, "Quantum Mechanics", third edition, p. 321, McGraw-Hill, New York (1968).
7. S. Y. Tong and T. N. Rhodin, Phys. Rev. Lett. 26, 711 (1971).
8. D. W. Jepsen, P. M. Marcus and F. Jona, Phys. Rev. Lett. 26, 1365 (1971); Phys. Rev. B5, 3933 (1972).
9. N. D. Lang and W. Kohn, Phys. Rev. B3, 1215 (1971).
10. B. I. Lundqvist, Phys. Stat. Sol. 32, 273 (1969).
11. D. Pines and P. Nozières, "The Theory of Quantum Liquids", Vol. 1, Benjamin, New York (1966).
12. C. B. Duke and G. E. Laramore, Phys. Rev. B2, 4765 (1970).
13. A. R. Edmonds, "Angular Momentum in Quantum Mechanics", Princeton (1957).

Figure Captions

Fig. 1 The incident electron beam $\underset{\sim}{k}_i$ and one of the diffracted beams $\underset{\sim}{k}_f$, and their projections onto the surface $\underset{\sim}{k}_{i_{\parallel}}$ and $\underset{\sim}{k}_{f_{\parallel}}$ are shown. The angles θ and ϕ are used to describe the orientation of the incident beam.

Fig. 2 The white circles represent the real space lattice positions for the surface layer of a crystal, with the central ion-core chosen as the origin. The black circles show the lattice positions of the λth layer which are shifted with respect to the origin by a vector $\underset{\sim}{a}_\lambda$ which is parallel to the surface layer.

Fig. 3 Schematic view of the relationships between the t-matrices; (a) t_λ for a single ion-core; (b) τ_λ for the multiple scattering of $\underset{\sim}{k}_i$ between all ion-cores in the λth subplane. The subplane propagator G^{sp} accounts for the phase and amplitude of the scattered wave between the ion-core encounters; (c) T_λ for the multiple scattering between subplanes (τ_λ) including all possible scattering paths ending on layer λ. The interplane propagator $G^{\lambda\lambda'}$ accounts for the phase and amplitude of the scattered wave between subplane encounters.

Fig. 4 Relationships between the real parts of the energies for quasiparticle motion are shown. ε is the kinetic energy of an electron travelling in the vacuum, and $E(\underset{\sim}{k},\varepsilon)$ is the quasiparticle energy inside the electron gas, where k is the propagation vector in the vacuum. The metallic work function ϕ and the real part of the renormalized electron energy inside the metal, $\Sigma_1(\underset{\sim}{k},\varepsilon)$, are measured with respect to the Fermi level ε_F.

Fig. 5 The (00) beam of the aluminum (100) surface is plotted using two approximations for the electron self energy. The solid curve uses the Lundqvist model of $\Sigma(E)$ for a free electron gas. The dashed curve uses the optical model potential with the real part of the self energy, $\Sigma_1 = -11$ eV.

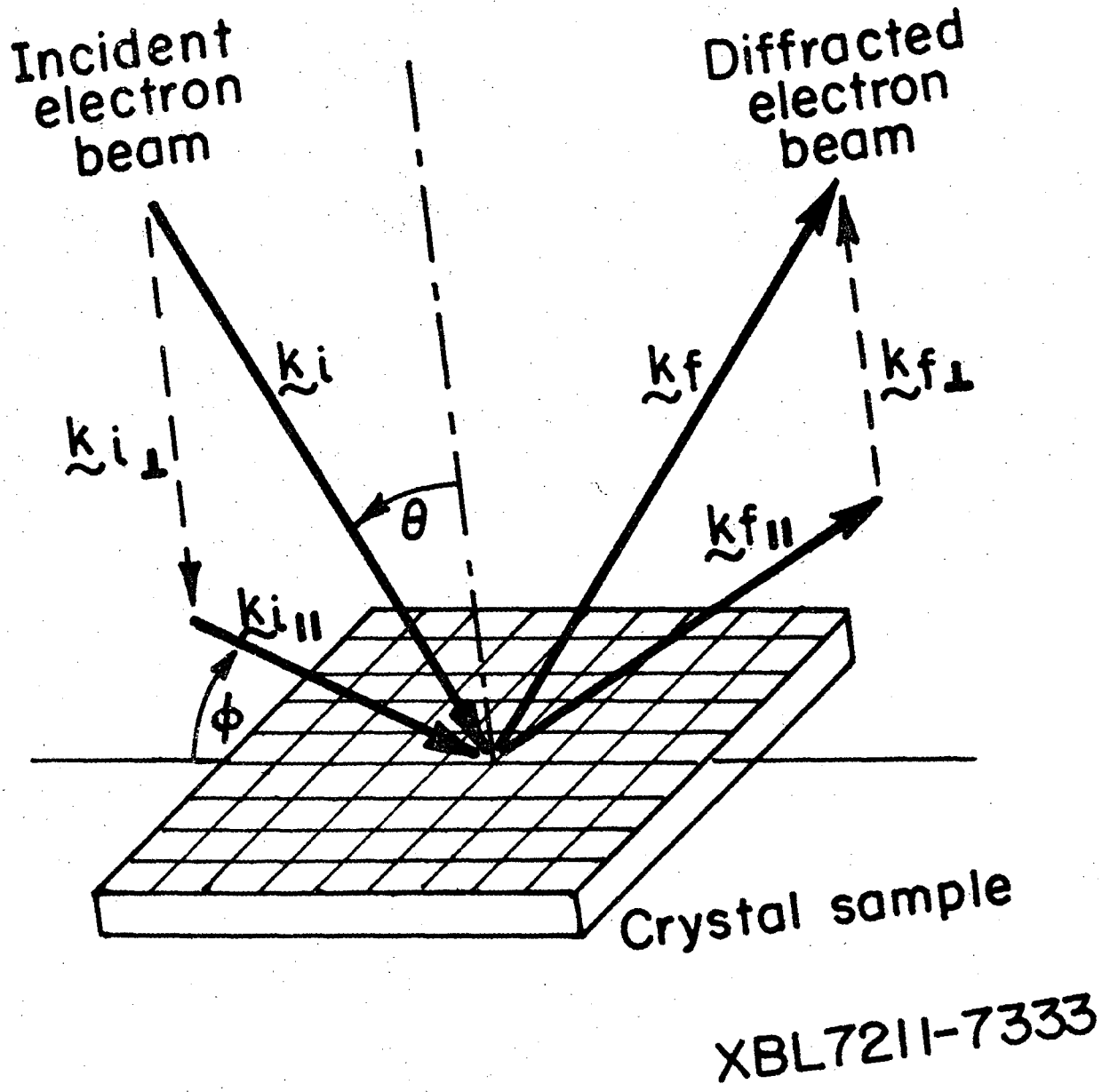
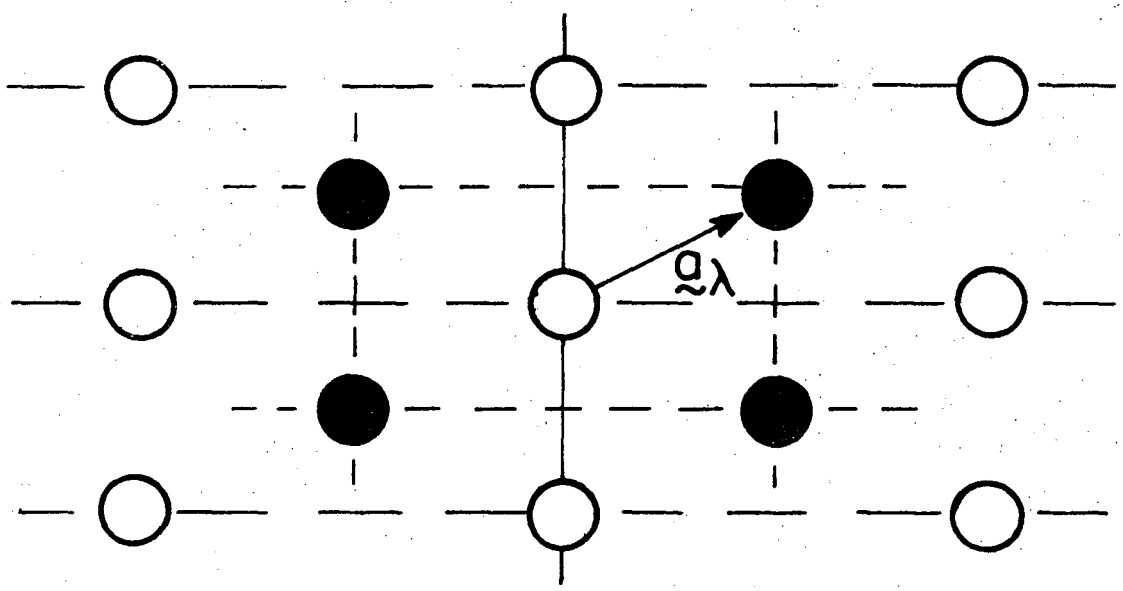

Incident electron beam
Diffracted electron beam
$\underset{\sim}{k}_i$
$\underset{\sim}{k}_f$
$\underset{\sim}{k}_{f\perp}$
$\underset{\sim}{k}_{i\perp}$
θ
$\underset{\sim}{k}_{f\parallel}$
$\underset{\sim}{k}_{i\parallel}$
ϕ
Crystal sample
XBL7211-7333

Fig. 1



XBL 7211-7334

Fig. 2

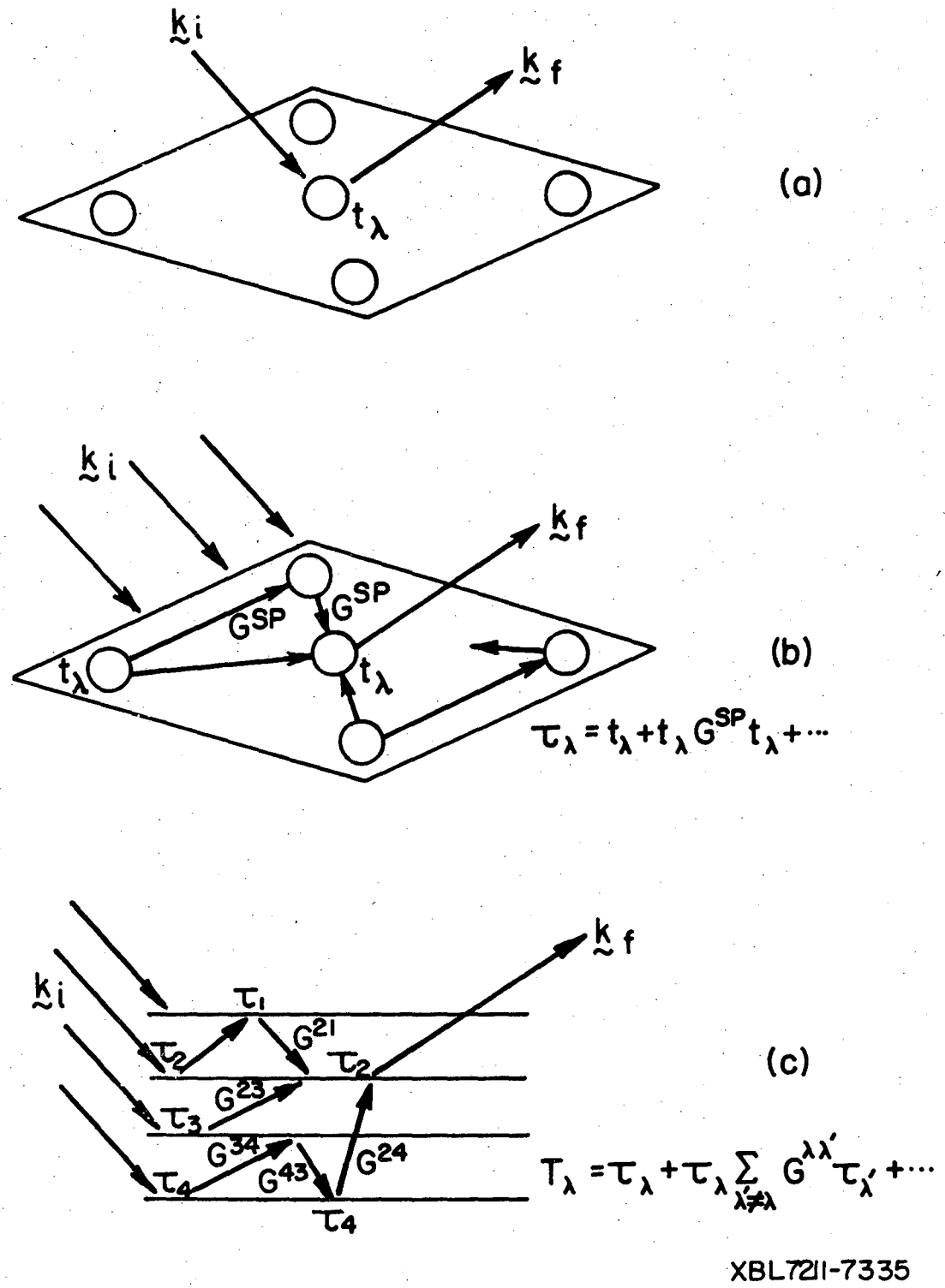

$\underset{\sim}{k}_i$
$\underset{\sim}{k}_f$
t_λ
(a)
$\underset{\sim}{k}_i$
$\underset{\sim}{k}_f$
G^{SP}
G^{SP}
t_λ
t_λ
(b)
$\tau_\lambda = t_\lambda + t_\lambda G^{SP} t_\lambda + \cdots$
$\underset{\sim}{k}_f$
$\underset{\sim}{k}_i$
τ_1
G^{21}
τ_2
τ_2
G^{23}
τ_3
G^{34}
G^{43}
G^{24}
τ_4
τ_4
(c)
$T_\lambda = \tau_\lambda + \tau_\lambda \sum_{\lambda' \neq \lambda} G^{\lambda\lambda'} \tau_{\lambda'} + \cdots$
XBL7211-7335

Fig. 3

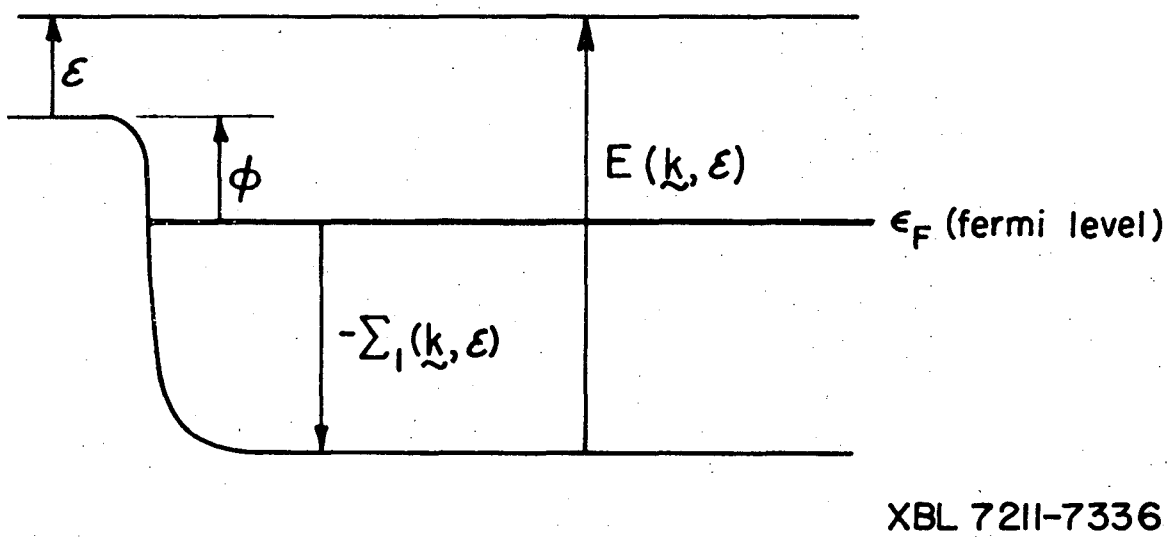


Fig. 4

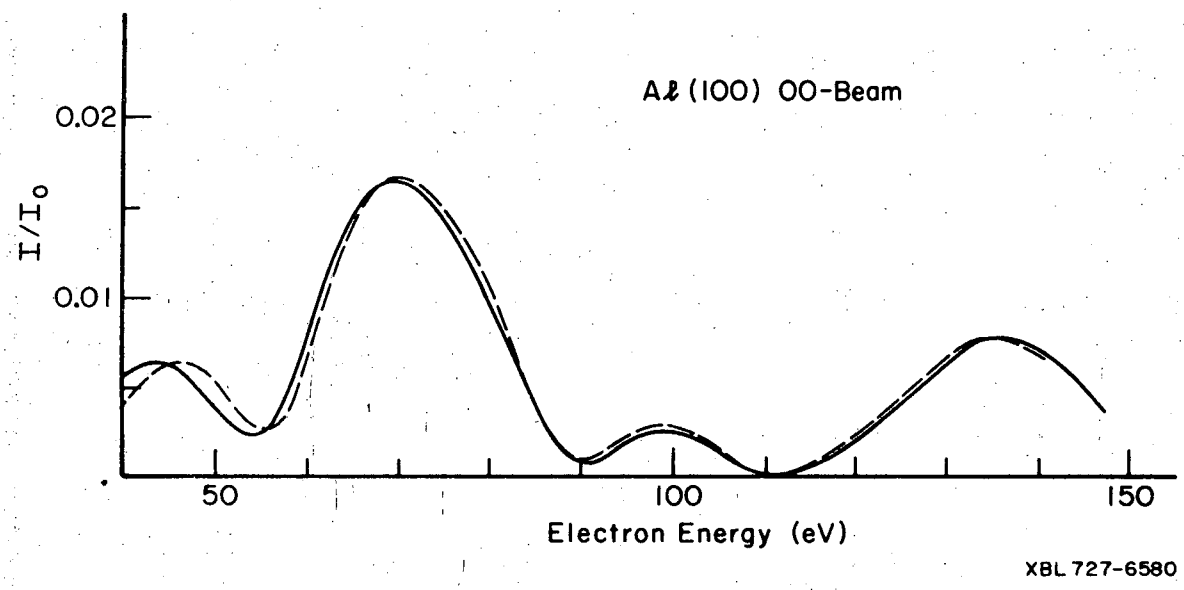
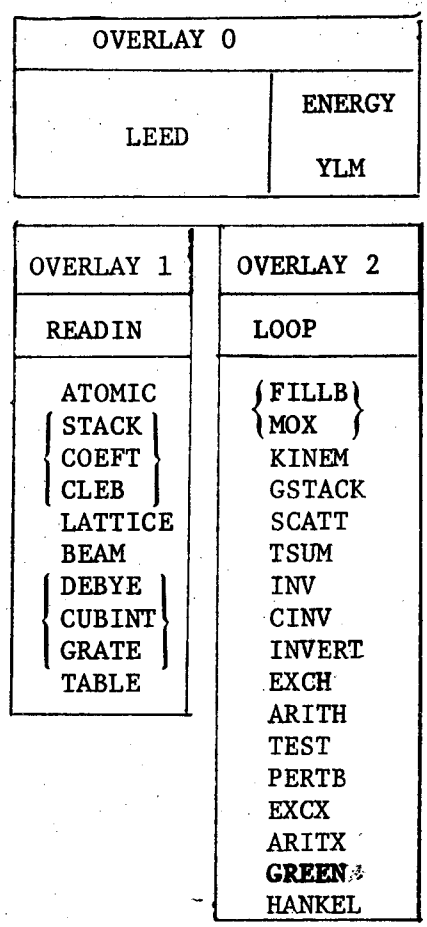

Aℓ (100) 00-Beam
0.02
0.01
I/I₀
50
100
150
Electron Energy (eV)
XBL 727-6580

Fig. 5

Appendix A. Organization of the Computer Program

The LEED program which is the subject of this report has been written in Fortran IV for use on a CDC 7600 computer. Use has been made of the extended core facility of this computer and provisions will have to be made for this is a different machine is to be used. If this is to be done, care must be taken in the handling of all variables appearing in the LARGE statements.

The program is written using overlays to minimize necessary core requirements. The 0-level overlay contains a driver program (LEED) and two subroutines that are used in both overlays that are called by LEED. In addition, it contains a number of COMMON block definitions through which most of the subroutines are connected. The first overlay called consists of a program (READIN) which causes input data cards to be read and several arrays to be initialized. The second overlay (LOOP), also called by LEED, causes the multiple scattering problem to be solved at a sequence of incident electron energies as specified in the input data.


OVERLAY 0
LEED
ENERGY
YLM
OVERLAY 1
READIN
ATOMIC
STACK
COEFT
CLEB
LATTICE
BEAM
DEBYE
CUBINT
GRATE
TABLE
OVERLAY 2
LOOP
FILLB
MOX
KINEM
GSTACK
SCATT
TSUM
INV
CINV
INVERT
EXCH
ARITH
TEST
PERTB
EXCX
ARITX
GREEN
HANKEL

OVERLAY (0)

LEED

This is the driver program that controls the calling of the two overlays and causes punched card output to be produced. All common blocks through which data are passed between the overlays and subroutines are defined here.

ENERGY

This subroutine is furnished with the incident electron energy (NRG) in electron volts. It then computes the real and imaginary parts of the quasiparticle energy $E(\varepsilon)$ and $\Gamma(\varepsilon)$ as defined in equations 17 or 19 (ESIG and SIG2). Also calculated are the real and imaginary components of the quasiparticle propagation vector $k_1(\varepsilon)$ and $k_2(\varepsilon)$ (XK and DAMP), and the complex renormalization constant ZNORM. The quantity NRGY is an index that is used to label the atomic t_λ^L-values as a function of the incident energy.

The subroutine is not completely general. The data in arrays SG1, SG2, ZRL, and ZIM are taken from the work of Lundqvist[10] and refer to an electron gas whose density is that of the conduction electrons in aluminum, as does the fermi energy (EFERMI).

Provisions are also made for calculating $E(\varepsilon)$ and $\Gamma(\varepsilon)$ by using an optical model potential (eq. 19) in which the inner potential V_o and the imaginary part of the self energy $i\Sigma_2$ are supplied as constant values in the input data cards (POTL and GAMA). An integer variable

read into the program causes the Lundqvist model to be calculated if SIGMA = 0 and the optical model potential is calculated if SIGMA = 1.

YLM

The spherical harmonics $Y_{\ell m}(\theta,\phi)$ are computed for $m = -\ell,-\ell+1, \ldots,+\ell$ for $\ell = 0,1,\ldots,$LMAX. The resulting values are stored in a complex array Y. The input angles (θ,ϕ) are given by specifying the three components of a vector $\underset{\sim}{k}$, of arbitrary magnitude, having the same angular directions. If CONJ equals -1, the complex conjugate of the entire array is taken, $Y^*_{\ell m}(\theta,\phi)$.

The linear array $Y_L(\underset{\sim}{k}) = Y_{\ell m}(\underset{\sim}{k})$ is stored using the algorithm $L = \ell^2+\ell+1+m$, or ...

L	ℓ	m
1	0	0
2	1	-1
3	1	0
4	1	1
5	2	-2
6	2	-1
7	2	0
8	2	1
9	2	2
10	3	-3
11	3	-2
	.	
	.	
	.	

OVERLAY (1)

READIN

This program reads in the following quantities:

DATE ... alphanumeric data specifying date of calculation.

MULT = 0 for kinematic scattering

= 1 for multiple scattering

MAXR ... maximum value (in Angstroms) of radial distance between ion cores to be included in the propagator summations 15 and 16.

TIME = 0 ... computer time intervals for each of the major segments of the calculation are not printed.

= 1 ... computer time intervals are printed on the output.

SIGMA = 0 ... specifies Lundqvist model for electron self-energy.

= 1 ... specifies optical model potential for electron self-energy.

= 2 ... specifies a constant elastic electron damping length.

POTL ... inner potential in electron volts (a negative number)

... ignored if SIGMA = 0

GAMA ... ignored if SIGMA = 0

... imaginary part of self-energy (electron volts) if SIGMA = 1

... damping length (angstroms) if SIGMA = 2

NDIF ... the number of iterations to be carried out in the perturbation expansion of eq. 13.

The following subroutines, each of which reads in and organizes further data, are read in by READIN:

ATOMIC
STACK
LATTICE
BEAM
DEBYE

The subroutine continues by calling ENERGY to ascertain if NLOW is within the range of energy increments for which the $t_\lambda^L(k)$ are supplied, and stops the program if it is too large. The energy and intensity arrays E, EN, and S are initialized to zero and the direction of the incident $\underset{\sim}{k}$-vector is calculated and stored in V1. Then, YLM is called to store the spherical harmonic array for the incident beam in YIN.

Subroutine TABLE is now called and control is returned to LEED. At this point all arrays are prepared for one or a series of LEED calculations at various energies.

ATOMIC

The angular momentum matrix elements, $t_\lambda^L(k)$, defined in eq. 12 and used in the summation of eq. 8a, are stored as a function of the incident electron energy. These values are read in at 2 eV intervals and stacked in the array T(I,J,K). The index K specifies the angular momentum quantum number ℓ ~ ($K=\ell+1$), and due to the spherical symmetry assumed for the ion core scattering centers, there is no azimuthal quantum number dependence (independent of m). The J-index labels the energy

intervals, and the I-dimension of the array allows several different ion-core types to be read in to the calculation. Presently a maximum of 2 types is permitted.

This subroutine first reads ITYPE, the number of ion core types to be included in the calculations. ITYPE must presently equal 1 or 2. The maximum value of the angular momentum quantum number, ℓ, is read in as LMAX. This establishes the dimensions of all arrays in the program.

For each ion-core type, an identifying TITLE card, the atomic weight of the ion-core (ATWT) and the following parameters are read in:

LTOP = total number of ion-core t-values read in at each energy (LTOP = ℓ+1). Note that LTOP > LMAX, otherwise the program will subsequently set LMAX = LTOP - 1.

LOWEN = lowest energy value (in eV) for which a member of the array $t_\lambda^\ell(k)$ is read in.

NNRG = total number of energy values for which array member $t_\lambda^\ell(k)$ are read in.

Finally the array T(I,J,K) is read in and the subroutine returns to READIN.

STACK

The expression I(L,L',L") is calculated by function COEFT and stored here in real array C. For each ℓ-value there are 2ℓ+1 values of the azimuthal quantum number m. Thus, the corresponding maximum value $L_{max} = (\ell_{max} + 1)^2$. If the quantity I(L,L',L") were to be

stored as a 3-dimensional array, its total storage requirements would be $(\ell_{max}+1)^2\cdot(\ell_{max}+1)^2\cdot(2\ell_{max}+1)^2$. For a typical value $\ell_{max} = 4$, this amounts to storage of 50625 numbers, many of which are zero! We reduce the size of the array C to a tractable value by storing only the non-zero values in C (NUM).

The L"-values for which I(L,L',L") are non-zero are recorded in array LL (NUM). For given values of L and L' (eqs. 15 and 16), the number of non-zero values in the summation over L" is stored as KX(ISTEP). ISTEP ranges from 1 to $(\ell_{max}+1)^4$ as the one dimensional index corresponding to the two angular momentum indices L and L'. Specifically, we have

$$\text{ISTEP} = (L-1)\cdot\text{LMAX3} + L'$$

$$\text{where LMAX3} = (2\ell_{max}-1)^2$$

$$\text{and} \quad L = \ell^2 + \ell + m + 1$$

$$L' = \ell'^2 + \ell' + m' + 1.$$

The three arrays KX, LL and C are used in subroutine GREEN for the summations in the subplane and interplane propagator expressions (eqs. 15 and 16).

COEFT

This function delivers the expression I(L,L',L") of equation 17, which is stored by STACK in array C. It employs a library function CLEB to calculate the Clebsch-Gordan coefficients.

LATTICE

The purpose of this routine is to read in the lattice geometry and to calculate the two dimensional reciprocal lattice vectors parallel to

the surface. The following parameters are read:

TYPE ... specifies the type of crystal structure.

Allowable input in A3 format is,

BCC - for body centered cubic lattices

FCC - for face centered cubic lattices

HCP - for hexagonal close packed lattices

GR1 - for graphite

FACE ... can be 100, 110 or 111 for BCC or FCC. Unnecessary input for hexagonal surfaces which can presently only be used for the 0001 face.

ALAT ... lattice constant a_o in Angstroms.

HEX ... Distance between hexagonal stacked planes for a hexagonal structure. This quantity is ignored if the type is BCC or FCC.

LAYER ... total number of crystal layers parallel to the surface over which the calculation is to be performed. Since each such layer contains ion cores of a single type, LAYER more precisely specifies the number of subplanes.

The following corrections are read in layer by layer, starting with the outermost layer: KT, ITP, MM, SHIFT. These values cause the geometry of a given layer to be altered and the origin of the subplane to be shifted. When the program encounters a blank card, the remaining layers are left unaltered from the bulk layer geometry. If the surface is simply a truncation of the bulk structure, the first card should be blank.

KT ... denotes the ion-core type of the subplane and is stored in KTYP.

ITP ... denotes the subplane geometry type, stored in ILYR.

MM ... an array of dimension 4, these quantities specify the primitive lattice translations in a given subplane in terms of the primitive lattice vectors parallel to the surface, $\underset{\sim}{A}$ and $\underset{\sim}{B}$.

$$\underset{\sim}{A}' = M(1) \cdot \underset{\sim}{A} + M(2) \cdot \underset{\sim}{B}$$

$$\underset{\sim}{B}' = M(3) \cdot \underset{\sim}{A} + M(4) \cdot \underset{\sim}{B} \quad .$$

SHIFT(I) ... A 3-member array which expresses the deviation of the subplane. The x and y components of the shifts are specified as a decimal fraction of the lattice constant a_o (ALAT). The z-component (I=3) is specified as a decimal fraction of the interplane distance. The coordinate system is such that z is zero at the surface and increases in the direction away from the surface into the bulk. A positive value for the z-component (SHIFT(3) indicates an expansion over the bulk interlayer separation, and a negative value is a contraction.

The subroutine continues by filling the arrays A(I,J) and B(I,J). These are the two dimensional primitive lattice translations within the Ith subplane. The index J specifies the x and y components for J=1 and 2 respectively. The location of the origin of each subplane is stored as a three dimensional vector in D(I,J). Again, I specifies the subplane and J=1,2,3 indicates the x,y or z component of the vector. The (x,y) components of D(I,J) are the vectors $\underset{\sim}{a}_\lambda$ appearing in eq. 2, and the z-component is d_λ from the same equation. In eq. 16, the three dimensional vector has been referred to as $\underset{\sim}{d}_\lambda = (\underset{\sim}{a}_\lambda, d_\lambda)$.

Finally the reciprocal lattice vectors are calculated for a two dimensional mesh whose primitive translations, A(I,J) and B(I,J), have been stored earlier in this routine. Details of the surface geometry are printed on the output before a return is made to READIN.

BEAM

The angular coordinates of the incident beam are read in degrees, θ = ITHT and ϕ = IPHI. The parameters specifying the energy values over which the calculation is to be performed are read:

NLOW = low incident electron energy value in eV.

NNUM = total number of energies to be calculated.

NDELTA = energy increment (in eV) between successive calculations.

This subroutine reads ITP and NN, which are instructions about which reciprocal lattice vectors, $\underset{\sim}{g}$, are to be used in the computation of

of various LEED beams.

ITP ... indicates the type of subplane geometry (which must have previously been defined in the input to subroutine LATTICE).

NN ... an array composed of six pairs of integers. These integer pairs are the beam indices in standard notation ... i.e., (00), (01), (21), etc. Each of the pairs listed will appear in the calculation. Each set of indices refers to the reciprocal lattice vectors $\underset{\sim}{g}$ of the subplane geometry given by the corresponding parameter ITP.

The two dimensional arrays VXX(I,J), where I=1,2 are the x and y components, stores the parallel components of the k-vector $\underset{\sim}{k}_{\parallel}$ (eq. 2). For the index J=1,2,3,... components are stored for all beams specified in the input instructions.

DEBYE

This subroutine performs the operations indicated in eq. 25 and calculates the coefficient of the factor $(\underset{\sim}{k}_f-\underset{\sim}{k}_i)^2$. The quantity stored is W(I), where I is the subplane index and

$$W_I(\underset{\sim}{k}_f-\underset{\sim}{k}_i) = W(I)\cdot(\underset{\sim}{k}_f-\underset{\sim}{k}_i)^2.$$

The lattice temperature (TEMP) and the bulk Debye temperature (BULK) are read in (degrees Kelvin). The integral

$$\int_0^{\Theta_D^\lambda/T} \frac{x}{e^x-1}\, dx$$

is performed numerically by calling subroutine CUBINT, which is a standard library routine.

TABLE

The summations that appear in eqs. 15 and 16 over the planar lattice vector $\underset{\sim}{P}$ are essential in order to correctly account for the scattering from each ion-core. The values $\underset{\sim}{P}$ assumes are determined by the lattice structure and are calculated in this subroutine. The vector quantities $\underset{\sim}{P}$ and $\underset{\sim}{P}+\underset{\sim}{d}_\lambda, -\underset{\sim}{d}_\lambda$ are necessary as arguments for the spherical harmonics $Y_{L''}(\underset{\sim}{P})$ and $Y_{L''}(\underset{\sim}{P}+\underset{\sim}{d}_\lambda, -\underset{\sim}{d}_\lambda)$. These values are stored serially in the array RVEC, where the x, y and z components of $\underset{\sim}{P}$ and $\underset{\sim}{P}+\underset{\sim}{d}_\lambda, -\underset{\sim}{d}_\lambda$ occur in that order. Since the direction of the incident beam $\underset{\sim}{k}_i$ is fixed during a calculation over a number of incident energies ($|k| = \sqrt{\frac{2m}{\hbar^2} E}$), we store the scalar product $\underset{\sim}{k}_i^o \cdot \underset{\sim}{P}$ and $\underset{\sim}{k}_i^o \cdot (\underset{\sim}{P}+\underset{\sim}{d}_\lambda, -\underset{\sim}{d}_\lambda)$ in array XMAG.* Finally, the magnitudes $|\underset{\sim}{P}|$ and $|\underset{\sim}{P}+\underset{\sim}{d}_\lambda, -\underset{\sim}{d}_\lambda|$ are stored in RMAG for use as an argument to the hankel function $h_{\ell''}^{(1)}(k|\underset{\sim}{P}+\underset{\sim}{d}_\lambda, -\underset{\sim}{d}_\lambda|)$. The array dimensions required by the program are printed and compared with the actual storage allotted in each case for arrays C, LL, KX, RMAG, XMAG, RVEC and G.

---

*$\underset{\sim}{k}_i^o$ is the value of the incident wave vector calculated for the lowest energy specified, NLOW.

OVERLAY (2)

LOOP

This overlay performs a series of intensity calculations over a number of incident electron energies. The number of energy values is supplied by NNUM. The lowest energy calculated is NLOW and the increment between energies is NDELTA. The variable NRG is set equal to the energy value being calculated in a given pass through the loop.

Subroutine ENERGY is called and calculates the real and imaginary parts of the renormalized electron energy and propagation vector. It also supplies NRGY, which is the index of the array $t_\ell(E)$ corresponding to the value E = NRGY. From this the incident electron propagation vector V1 is computed and stored. The three components depend on the incident beam angles and the real part of the propagation vector inside the crystal. The spherical harmonics corresponding to the incident direction are computed by YLM and stored in YIN.

The reciprocal lattice vectors VXX have been calculated previously (subroutine BEAM) for a specified set of beam indices. These are added to the x,y-components of the incident beam V1 to yield the outgoing propagation vector components V2, V2 = $\underset{\sim}{k}_{i_{\parallel}} + \underset{\sim}{g}$. The third component of V2 (in the perpendicular direction) is calculated from the condition that the scattering be elastic. The outgoing wavevector V2 defines a direction which is used as input to subroutine YLM to calculate the outgoing spherical harmonics YOUT = $Y_L^*(\underset{\sim}{k}_f)$.

Subroutine FILLB is called to compute the array BBB = b_λ^ℓ as defined by eq. 30. This corrects the ion-core t-matrices (t_λ^ℓ) for finite temperature effects. Subroutine GREEN(1) is called to compute the subplane propagator $G_{sp}^{LL'}$, which is stored in a linear array (G) in extended core memory of the CDC 7600.

PERTB performs a perturbation series expansion using the subplane propagator and the quantities b_λ^ℓ in a manner similar to that shown in eq. 5a.

$$\tau_\lambda = b_\lambda + b_\lambda G^{sp} b_\lambda + b_\lambda G^{sp} b_\lambda G^{sp} b_\lambda + \ldots$$

The same quantity (τ_λ) is then computed by an exact matrix inversion method (eq. 5b) by calling INV. The matrix inversion result is always employed, but both results are printed on the output. This is a check on the convergence of the iterative expansion method. If the two values for τ_λ so calculated are in serious disagreement it is an indication that the iterative method of solving for $T_\lambda^{LL'}$ (eq. 13) may be non-convergent, and that a complete matrix inversion treatment of the problem may be necessary.

The values $T_\lambda^{LL'}$ are calculated in the single scattering (kinematic) approximation by KINEM. Using these values, subroutine TSUM calculates the differential scattering cross section for each specified beam using equations 1 and 2.

If a multiple scattering computation is requested (MULT > 1), all interplane propagators are calculated by GREEN and stored linearly in array G.

The second order diffraction computation is performed by subroutine GSTACK, after which TSUM calculates the cross section for each beam. Subsequent iterations are made (to the diffraction limit specified by NDIF) by subroutines SCATT and TSUM.

If the parameter TIME $\neq$ 0, central processor times in seconds are stored and printed at the end of each LOOP calculation. This indicates the amount of time spent in performing each of the above steps.

At the end of the series of energy loop calculations the final results for the scattering cross sections of each beam, S, as a function of the incident beam energy, E, are printed as output.

Control is then passed back to the main program LEED which punches the same output if requested.

FILLB

The array BBB corresponding to the temperature renormalized ion-core t-matrices, $b_\lambda = e^{-W_\lambda(\underset{\sim}{k}_f-\underset{\sim}{k}_i)}t_\lambda$, is computed in this subroutine. Each of the diagonal matrix elements b_λ^L can be written as a linear combination of the diagonal unrenormalized matrix elements t_λ^L,

$$b_\lambda^\ell = e^{-\beta} \sum_{\ell''=0}^{\infty} \sum_{\ell'=|\ell-\ell''|}^{\ell+\ell''} B_{\ell'}\, t_\lambda^{\ell''}$$

The first call to this subroutine computes an array CP of the coefficients of the Legendre polynomials. Using these values the coefficients $e^{-\beta}B_{\ell'}$ are computed and stored in array BCF. Subroutine MOX performs the summation and delivers the result b_λ^ℓ. Since this value is (2m+1)-fold

degenerate (depends only on the ℓ quantum number and not on m), the final array BBB is filled including the degeneracies.

KINEM, GSTACK and SCATT

These three subroutines are used with TSUM to calculate values of $T_\lambda(\underset{\sim}{k}_f,\underset{\sim}{k}_i;E)$ for each beam. KINEM is used to obtain T_λ in the kinematic scattering approximation whereby this quantity is simply replaced by the ion-core t-matrix renormalized for lattice motion,

$$T_\lambda = \sum_{LL'} b_\lambda^{LL'} \delta_{LL'} Y_L^*(\underset{\sim}{k}_f) Y_{L'}(\underset{\sim}{k}_i)$$

GSTACK is called to compute $T_\lambda^{LL'}$ in the approximation of double diffraction. The quantities $Q_\lambda^{L''L'}$ and $S_\lambda^{L''L'}$ are stored linearly in a single complex array G, which resides in extended core memory of the CDC 7600 computer.

$$Q_\lambda^{L''L'} = \delta^{L''L'} + S_\lambda^{(2)L''L'}$$

where $$S_\lambda^{(2)L''L'} = \sum_{\lambda'\neq\lambda} \sum_{L'''} G_{\lambda\lambda'}^{L''L'''} \tau_{\lambda'}^{L'''L'}$$

The arrays $Q_\lambda^{L''L'}$ are passed to TSUM where T_λ is obtained from

$$T_\lambda = \sum_{L,L'} \sum_{L''} \tau_\lambda^{LL''} Q_\lambda^{L''L'} Y_L^*(\underset{\sim}{k}_f) Y_{L'}(\underset{\sim}{k}_i)$$

SCATT adds one more term to the perturbation expansion for $Q_\lambda^{L''L'}$. The previous final term is $S_\lambda^{L''L'}$ and the additional term becomes

$$S_\lambda^{(3)L''L'} = \sum_{L,L'''} \sum_{\mu\neq\lambda} G_{\lambda\mu}^{L''L'''} \tau_\mu^{L'''L} S_\mu^{(2)LL'} .$$

The process is repeated for each iteration (i) and one obtains $S_\lambda^{(i+1)L''L'}$ from each previous $S_\lambda^{(i)L''L'}$. The matrices delivered to TSUM are obtained from the summation

$$Q_\lambda^{(i)L''L'} = \delta^{L''L'} + S_\lambda^{(2)L''L'} + S_\lambda^{(3)L''L'} + \cdots + S_\lambda^{(i)L''L'} .$$

<u>TSUM</u> (ITR, FTR, JINDX)

If the interation ITR = 1, this subroutine uses the values $T_\lambda(\underset{\sim}{k}_f, \underset{\sim}{k}_i; E)$ for each beam obtained from the kinematic approximation of <u>KINEM</u>. The scattering cross section is then calculated for each beam by means of equations 1 and 2.

If ITR > 1, the quantities T_λ are recomputed from the intermediate arrays $Q_\lambda^{L''L'}$ supplied by subroutines <u>GSTACK</u> and <u>SCATT</u>.

$$T_\lambda = \sum_{L,L'} \sum_{L''} \tau_\lambda^{LL''} Q_\lambda^{L''L'} Y_L^*(\underset{\sim}{k}_f) Y_{L'}(\underset{\sim}{k}_i)$$

These values are then used to recompute the beam scattering cross sections from equations 1 and 2.

<u>INV</u>

This subroutine performs the matrix inversion for the subplane scattering matrix (see equation 5b).

$$\tau_\lambda^{LL'} = b_\lambda^L (I - G_{sp} b_\lambda)^{-1}_{LL'}$$

Array $G_{sp}^{LL'}$ must appear in the first $(\ell_{max}+1)^4$ locations of the complex array G (stored in extended core memory). This subroutine computes the matrix in parentheses and stacks it rowwise in the singly dimensioned array Q (also in extended core memory).

Subroutine CINV is called to perform the actual inversion of the matrix Q. The inverted matrix is then delivered in array Q, this time stacked columnwise. The matrix elements $\tau_\lambda^{LL'}$ are obtained by multiplying b_λ^L by Q. Subroutine TEST is called to print out results.

TEST

This is an auxiliary subroutine used to calculate and print out values for τ_λ. This can be called by INV and PERTB so that the matrix inversion method and the iterative method of obtaining τ_λ can be directy compared.

CINV

A complex square matrix with $(\ell_{max}+1)^2$ elements on a side is supplied by the real array Q in the extended core memory of the CDC 7600 computer. Real and imaginary arrays are constructued, and the complex matrix is inverted. All computations are performed directly from the extended core storage locations using the following subroutines, EXCH, INVERT, ARITH.

The resulting inverted array is stored columnwise in array Q. The original array, Q, to be inverted, must be stored rowwise in the singly dimensioned array.

EXCH(N1,N2)

The real matrix whose first word address is Q(N2+1) is transposed and stored at Q(N1+1). N1 may equal N2.

INVERT(N1)

A real matrix inversion is performed on the matrix stored at Q(N1+1). The inverted matrix is returned in the same location.

ARITH(M1,OP,M2,M3,SIGN)

OP may equal 1 indicating a matrix addition, or 2, indicating a matrix multiplication, between the two real matrices stored at Q(M1+1) and Q(M2+1). The result is delivered in location Q(M3+1).

SIGN can equal +1.0 or -1.0 and can be used to change the sign of the product matrix when a multiplication is performed.

Care must be taken when OP=2 is called for that the matrix at Q(M1+1) is stored rowwise and that stored at Q(M2+1) is columnwise. The resultant matrix at Q(M3+1) is delivered columnwise.

PERTB

This subroutine performs a perturbation expansion (see eq. 5a) to obtain the matrix $\tau_{\lambda}^{LL'}$. The same matrix is calculated by a matrix inversion process (INV). This subroutine may be removed from the program along with the Fortran instruction that calls it in program LOOP. It is retained solely for purposes of checking the convergence

of the iterative expansion as an approximation to the true matrix inversion method.

Use is made of subroutines EXCX, which transposes a matrix stored in complex array G, and ARITX which performs matrix additons and multiplications. EXCX and ARITX are the counterparts of EXCH and ARITH that operate on the large core array G (instead of Q). Up to 20 iterations are performed, with the computation terminating when the convergence of successive iterations is within 10^{-4}.

GREEN (INDEX,IBEGIN)

This subroutine is called repeatedly by LOOP to calculate the complex matrices $G^{sp}_{LL'}$ and $G^{\lambda\lambda'}_{LL'}$. When INDEX=1, the subplane propagator $G^{sp}_{LL'}$ is computed from eq. 15 and stored in the first $(\ell_{max}+1)^4$ locations of array G (stored in extended core memory). When INDEX > 1, the array $G^{\lambda\lambda'}_{LL'}$ is stored beginning at G(IBEGIN+1) and is calculated from eq. 16.

The positions of all ion-cores have previously been calculated from the lattice geometry and stored in arrays RMAG and RVEC (See subroutine TABLE). The spherical harmonics $Y_{L''}(\underset{\sim}{P} + \underset{\sim}{d}_{\lambda'} - \underset{\sim}{d}_{\lambda})$ are computed and stored in array YINT. At each value for the incident energy, each lattice point must be summed over to obtain its contribution to the subplane or interplane propagators. Due to the necessity of computing a complete set of spherical harmonics and the spherical hankel function $h^{(1)}_{\ell''}(k|\underset{\sim}{P} + \underset{\sim}{d}_{\lambda'} - \underset{\sim}{d}_{\lambda}|)$ at each such point,

the major portion of the computing time for a given problem is spent within this subroutine and auxiliary routines called by it.

HANKEL

The spherical hankel functions $h^{(1)}(K|\underset{\sim}{P}+\underset{\sim}{d}_\lambda,-\underset{\sim}{d}_\lambda|)$ needed to evaluate the expression in eq. (16) are calculated for use in subroutine GREEN. They are delivered by the complex array H.

```
      PROGRAM LEED(INPUT,OUTPUT,PUNCH)                                  LEED.3
C                                                                       LEED.4
      COMMON/BUFF/LCG,NBC,KKK                                           LEED.5
      COMMON/BLOCKA/A(10,2),B(10,2),D(10,3),RA(10,2),RB(10,2),ALAT,SPACEA.2
     2,BASIS(2),NBASIS                                                  A.3
      COMMON/BLOCKB/T(2,8C,7),ILYR(10),LSYM(10)                         B.2
      COMPLEX T                                                         B.3
      COMMON/BLOCKC/V1(3),V2(3,12),V3(3,12),VX,VY,VZ,SQK(12),GGK(12),   C.2
     2  YIN(49),YOUT(49,12),XKZERO,AREA2                                C.3
      COMPLEX YIN,YOUT                                                  C.4
      COMMON/BLOCKD/NLOW,NNUM,NDELTA,NRG,NRGY,LAYER,NUMMAX,NCIF,IST(6,5)D.2
     2  ,LR                                                             D.3
      COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE                                               E.3
      COMMON/BLOCKF/ITHT,IPHI,THTT,DIST,CSTH(12),VXX(2,12),KNUM,JM(12), F.2
     2  JN(12),IG(12),KG(12)                                            F.3
      COMMON/BLOCKG/ESIG,SIG2,XK,DAMP,ZNORM,PCTL,GAMA,SIGMA,LOWEN,NNRG  G.2
     2  ,LAMBDA                                                         G.3
      REAL LAMBDA                                                       G.4
      INTEGER SIGMA                                                     G.5
      COMPLEX ZNORM                                                     G.6
      COMMON/BLOCKH/KX(2401),C(7532),LL(7532),YINT(169)                 H.2
      COMPLEX YINT                                                      H.3
      COMMON/BLOCKJ/MAXX(51),RMAG( 9000),XMAG( 9000),MAXR               J.2
      COMMON/BLOCKK/BBB(36,2),TAU(36,36,2),TT(10,12),KTYP(10)           K.2
      COMPLEX  BBB,TAU,TT                                               K.3
      COMMON/BLOCKL/ARG,H(169),XD(9),PATH1(10),PATH2(10,12)             L.2
      COMPLEX  ARG,H                                                    L.3
      COMMON/BLOCKM/RATIO,XCON,KSTAR                                    M.2
      COMPLEX KSTAR,XCON                                                M.3
      COMMON/BLOCKN/E(51),EN(51),S(12,51)                               N.2
      COMMON/BLOCKX/TIME,ITEM,MULT,ISCALE,IMAX,ITYPE,LSUM,ILOOP,IPWR    X.2
      INTEGER TIME                                                      X.3
      COMMON/BLKDEH/W(10),ATWT(10),TEMP,SURF,BULK,CIN(10),COUT(10,12)   DEH.2
      INTEGER HAH(9)                                                    LEED.21
      COMMON/CONST/PI,PI4,PI42,CON1,CON2,CON3,CON4,CONSTT,IM,CNST       CONST.2
      COMPLEX IM                                                        CONST.3
      LARGE C.QM(1296),RVEC(30000),C.G(49000)                           LARGE.2
C                                                                       DIM.2
C     DIMENSIONS OF KX,C,AND LL.                                        DIM.3
      NM3=2401 $ NM1=7532 $ NM2=7532                                    DIM.4
C     DIMENSIONS OF RMAG AND XMAG                                       DIM.5
 5    NM4=9000 $ NM5=9000                                               DIM.6
C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.              DIM.7
 6    NM6=30000 $ NM7=49000                                             DIM.8
C                                                                       DIM.9
C                                                                       LEED.25
C     PROGRAM HANDLES MAXIMUM OF 10 LAYERS                              LEED.26
C     LMAX=5                                                            LEED.27
C                                                                       LEED.28
C                                                                       LEED.29
10    LX=36                                                             LEED.30
11    ISCALE=1                                                          LEED.31
12    IMAX=51                                                           LEED.32
13    IM=CMPLX(0.0,1.0)                                                 LEED.33
16    PI=3.141592654                                                    LEED.34
17    PI4=4*PI                                                          LEED.35
21    PI42=PI4**2                                                       LEED.36
```

```
 23          CCN1=2.*PI/SCRT(150.0)                                          LEED.37
 27          CCN2=PI/150.0                                                   LEED.38
 30          CCN3=PI/180.0                                                   LEED.39
 32          CCN4=CCN2**2                                                    LEED.40
 33          CCNSTT=CCN1**4*PI4**3*PI                                        LEED.41
 36          CNST=PI42*CCN1**4                                               LEED.42
 40          ITEM=0                                                          LEED.43
 41          READ 810, JPUNCH,HAP                                            LEED.44
       C                                                                     LEED.45
 51      100 CONTINUE                                                        LEED.46
       C                                                                     LEED.47
       C     OVERLAY(1,0)  READS IN PARAMETERS AND INITIALIZES ARRAYS *****  LEED.48
       C                                                                     LEED.49
 51          CALL OVERLAY(2LT1,1,0,6HKECALL)                                 LEED.50
 54          IF(ITEM.EQ.2)  STOP                                             LEED.51
 60          IF(ITEM.NE.5)  GO TO 110                                        LEED.52
 62          PRINT 800                                                       LEED.53
 65          STOP                                                            LEED.54
       C                                                                     LEED.55
 70      110 CONTINUE                                                        LEED.56
       C                                                                     LEED.57
       C     OVERLAY(2,0)  PERFORMS ENERGY LOOP CALCULATIONS *****           LEED.58
       C                                                                     LEED.59
 70          CALL OVERLAY(2LT1,2,0,6HKECALL)                                 LEED.60
       C                                                                     LEED.61
       C     PUNCH OUTPUT  (LEED CURVES)                                     LEED.62
       C                                                                     LEED.63
 73          IF(JPUNCH.EQ.0)  GO TO 120                                      LEED.64
 74          PUNCH 812, HAP                                                  LEED.65
102          DO 115  I=1,KNUM                                                LEED.66
104          PUNCH 802, IQ(I),JM(I),JN(I),LMAX,LAYER,ITHT,IPHI,NNUM          LEED.67
132          PUNCH 801, (E(J),S(I,J), J=1,NNUM )                             LEED.68
153      115 CONTINUE                                                        LEED.69
156      120 CONTINUE                                                        LEED.70
       C                                                                     LEED.71
       C                                                                     LEED.72
          801 FORMAT( 4(F7.2,E13.5) )                                        LEED.73
          802 FORMAT( 2X*TYPE=*I2,* BEAM(*2I3,*) L=*I2,* LYR=*I2,* THETA=*I3, LEED.74
             2 * PHI=*I3,*  NO. OF POINTS =*  I3 )                           LEED.75
          810 FORMAT( 6X I2, 9A8 )                                           LEED.76
          812 FORMAT( 9A8 )                                                  LEED.77
       C                                                                     LEED.78
          800 FORMAT( // 10X*$$$$$ RUN ABORTED $$$$$  ARRAY DIMENSIONS EXCEEDED.LEED.79
             2  * )                                                          LEED.80
       C                                                                     LEED.81
156          END                                                             LEED.82
```

```
PROGRAM LENGTH INCLUDING I/O BUFFERS
001776

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
100    -  000052    110    -  000071    120    -  000157    800    -  000237
801    -  000214    802    -  000317    810    -  000232    812    -  000235
```

```
      SUBROUTINE ENERGY(NRG,NRGY)                                       ENERGY.2
      COMMON/BLOCKG/ESIG,SIG2,XK,DAMP,ZNORM,PCTL,GAMA,SIGMA,LCHEN,NNRG  G.2
     2  ,LAMBDA                                                         G.3
      REAL LAMBDA                                                       G.4
      INTEGER SIGMA                                                     G.5
      COMPLEX ZNORM                                                     G.6
C                                                                       ENERGY.4
      COMPLEX Y,Z                                                       ENERGY.5
      DIMENSION  SG1(13),SG2(13),ZRL(13),ZIM(13)                        ENERGY.6
      DATA SG1/.83,.85,.85,.75,.66,.58,.53,.48,.44,.41,.38,.36,.34/     ENERGY.7
      DATA SG2/.04,.15,.33,.40,.43,.44,.44,.44,.43,.41,.40,.39,.37/     ENERGY.8
      DATA ZRL/.0,.70,.82,.86,.90,.92,.93,.94,.95,.96,.96,.96,.97/      ENERGY.9
      DATA ZIM/.0,.35,.30,.26,.23,.21,.19,.17,.16,.14,.135,.13,.12/     ENERGY.10
      PI=3.14159265E4                                                   ENERGY.11
      CCN1=2*PI/SQRT(150.)                                              ENERGY.12
      CONVRG=1.0E-04                                                    ENERGY.13
C                                                                       ENERGY.14
      EFERMI=11.9                                                       ENERGY.15
C                                                                       ENERGY.16
C                                                                       ENERGY.17
      IF(SIGMA.EQ.1)  GO TO 160                                         ENERGY.18
      IF(SIGMA.EQ.2)  GO TO 170                                         ENERGY.19
      XKK=0.0                                                           ENERGY.20
      DPP=0.0                                                           ENERGY.21
      XX=SQRT(NRG/EFERMI)                                               ENERGY.22
C                                                                       ENERGY.23
  100 CONTINUE                                                          ENERGY.24
      I=1                                                               ENERGY.25
      X=1.50                                                            ENERGY.26
  110 CONTINUE                                                          ENERGY.27
      IF(X.GE.4.5)  GO TO 145                                           ENERGY.28
      IF(XX.LE.X)  GO TO 120                                            ENERGY.29
      X=X+0.25                                                          ENERGY.30
      I=I+1                                                             ENERGY.31
      GO TO 110                                                         ENERGY.32
  120 CONTINUE                                                          ENERGY.33
C                                                                       ENERGY.34
      D=X-XX                                                            ENERGY.35
      IF(X.LE.1.50)  I=2                                                ENERGY.36
      SD1=SG1(I)-SG1(I-1)                                               ENERGY.37
      SD2=SG2(I)-SG2(I-1)                                               ENERGY.38
      Z1=ZRL(I)-ZRL(I-1)                                                ENERGY.39
      Z2=ZIM(I)-ZIM(I-1)                                                ENERGY.40
      D1=4*D*SD1                                                        ENERGY.41
      D2=4*D*SD2                                                        ENERGY.42
      D3=4*D*Z1                                                         ENERGY.43
      D4=4*D*Z2                                                         ENERGY.44
      IF(X.LE.1.50)  I=1                                                ENERGY.45
      EEEE=(SG1(I)-D1)*EFERMI+NRG                                       ENERGY.46
      SSSS=(SG2(I)-D2)*EFERMI                                           ENERGY.47
      Y=CMPLX(EEEE,SSSS)                                                ENERGY.48
      Z=CCN1*CSQRT(Y)                                                   ENERGY.49
      XY=REAL(Z)                                                        ENERGY.50
      DCCC=AIMAG(Z)                                                     ENERGY.51
      IF(ABS(XY-XKK).LE.CONVRG)  GO TO 140                              ENERGY.52
  130 CONTINUE                                                          ENERGY.53
      DPP=DCCC                                                          ENERGY.54
      XKK=XY                                                            ENERGY.55
```

```
136         XX=SQRT(EEEE/EFERMI)                                    ENERGY.56
143         GO TO 100                                               ENERGY.57
      C                                                             ENERGY.58
145     140 CONTINUE                                                ENERGY.59
145         IF(ABS(DDDD-DPP).LE.CONVRG)  GO TO 150                  ENERGY.60
152         GO TO 130                                               ENERGY.61
152     145 CONTINUE                                                ENERGY.62
152         EEEE=0.0                                                ENERGY.63
      C                                                             ENERGY.64
153     150 CONTINUE                                                ENERGY.65
154         ZR= ZRL(I)-C3                                           ENERGY.66
156         ZI=-(ZIM(I)-C4)                                         ENERGY.67
160         ZNCRM=CMPLX(ZR,ZI)                                      ENERGY.68
      C                                                             ENERGY.69
      C     END OF LUNDQVIST METHOD                                 ENERGY.70
      C                                                             ENERGY.71
163         GO TO 180                                               ENERGY.72
      C                                                             ENERGY.73
      C     OPTICAL MODEL POTENTIAL                                 ENERGY.74
      C                                                             ENERGY.75
163     160 CONTINUE                                                ENERGY.76
      C                                                             ENERGY.77
163         EEEE=NRG-POTL                                           ENERGY.78
165         SSSS=GAMA                                               ENERGY.79
167         Y=CMPLX(EEEE,SSSS)                                      ENERGY.80
171         Z=CON1*CSQRT(Y)                                         ENERGY.81
202         XY=REAL(Z)                                              ENERGY.82
202         DDDD=AIMAG(Z)                                           ENERGY.83
204         ZNCRM=CMPLX(1.0,0.0)                                    ENERGY.84
207         GO TO 180                                               ENERGY.85
      C                                                             ENERGY.86
      C     SIGMA=2, CONSTANT DAMPING LENGTH  (LAMBDA).             ENERGY.87
      C                                                             ENERGY.88
211     170 CONTINUE                                                ENERGY.89
211         EEEE=NRG-POTL                                           ENERGY.90
213         AK1=CON1**2*EEEE                                        ENERGY.91
216         AK2=CON1*SQRT(EEEE)/LAMBDA                              ENERGY.92
221         Y=CMPLX(AK1,AK2)                                        ENERGY.93
224         Z=CSQRT(Y)                                              ENERGY.94
227         XY=REAL(Z)                                              ENERGY.95
230         DDDD=AIMAG(Z)                                           ENERGY.96
231         ZNCRM=CMPLX(1.0,0.0)                                    ENERGY.97
234         PRINT 900, LAMBDA                                       ENERGY.98
        900 FORMAT( / 10X*LAMBDA=* F10.5 )                          ENERGY.99
245         GO TO 180                                               ENERGY.100
      C                                                             ENERGY.101
      C     DETERMINE ENERGY ARRAY INDEX (NRGY)                     ENERGY.102
      C                                                             ENERGY.103
245     180 CONTINUE                                                ENERGY.104
246         N=EEEE/2                                                ENERGY.105
250         N=2*N                                                   ENERGY.106
252         IF((EEEE-N).GT.1.0)  N=N+2                              ENERGY.107
257         NRGY=(N-LOWEN)/2+1                                      ENERGY.108
262         IF(NRGY.LT.1)  NRGY=0                                   ENERGY.109
264         IF(NRGY.GT.NNRG)  NRGY=0                                ENERGY.110
267         ESIG=EEEE                                               ENERGY.111
270         SIG2=SSSS                                               ENERGY.112
272         XK=XY                                                   ENERGY.113
```

```
273          DAMP=DDDD                                                    ENERGY.114
       C                                                                  ENERGY.115
       C                                                                  ENERGY.116
275          RETURN                                                       ENERGY.117
276          END                                                          ENERGY.118
```

```
SUBPROGRAM LENGTH
000501

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
100     -  000032    110     -  000035    120     -  000046    130     -  000134
140     -  000146    145     -  000153    150     -  000154    160     -  000164
170     -  000212    180     -  000246    900     -  000315

BLOCK NAMES AND LENGTHS
BLOCKG -  000014

VARIABLE ASSIGNMENTS
AK1     -  000476    AK2     -  000477    CONVRG  -  000450    CON1    -  000447
D       -  000457    DAMP    -  000003S01 DDDD    -  000473    DPP     -  000453
D1      -  000464    D2      -  000465    D3      -  000466    D4      -  000467
EEEE    -  000470    EFERMI  -  000451    ESIG    -  000000S01 GAMA    -  000007S01
I       -  000455    LAMBDA  -  000013S01 LOWEN   -  000011S01 N       -  000500
NNRG    -  000012S01 PI      -  000446    POTL    -  000006S01 SD1     -  000460
SD2     -  000461    SG1     -  000362    SG2     -  000377    SIGMA   -  000010S01
SIG2    -  000001S01 SSSS    -  000471    X       -  000456    XK      -  000002S01
XKK     -  000452    XX      -  000454    XY      -  000472    Y       -  000356
Z       -  000360    ZI      -  000475    ZIM     -  000431    ZNORM   -  000004S01
ZR      -  000474    ZRL     -  000414    Z1      -  000462    Z2      -  000463

START OF CONSTANTS
000301

START OF TEMPORARIES
000322

START OF INDIRECTS
000344

UNUSED COMPILER SPACE
123600
```

```
      SUBROUTINE YLM(VX,VY,VZ,Y,LMAX,CONJ)                              YLM.2
C                                                                       YLM.3
C     LMAX MUST NOT EXCEED 10 UNLESS DIMENSIONS ARE CHANGED***          YLM.4
C                                                                       YLM.5
      DIMENSION  P(11),VECTOR(3)                                        YLM.6
      DOUBLE CF1,CF2,FAC(31),FACT                                       YLM.7
      COMPLEX X,Y(1),XX,BOP                                             YLM.8
      INTEGER  CONJ                                                     YLM.9
      DATA FAC/   1.0000000000000000000000000D+00,                      YLM.10
     * 1.0000000000000000000000000D+00, 2.0000000000000000000000000D+00,  YLM.11
     * 6.0000000000000000000000000D+00, 2.4000000000000000000000000D+01,  YLM.12
     * 1.2000000000000000000000000D+02, 7.2000000000000000000000000D+02,  YLM.13
     * 5.0400000000000000000000000D+03, 4.0320000000000000000000000D+04,  YLM.14
     * 3.6288000000000000000000000D+05, 3.6288000000000000000000000D+06,  YLM.15
     * 3.9916800000000000000000000D+07, 4.7900160000000000000000000D+08,  YLM.16
     * 6.2270208000000000000000000D+09, 8.7178291200000000000000000D+10,  YLM.17
     * 1.3076743680000000000000000D+12, 2.0922789888000000000000000D+13,  YLM.18
     * 3.5568742809600000000000000D+14, 6.4023737057280000000000000D+15,  YLM.19
     * 1.2164510040883200000000000D+17, 2.4329020081766400000000000D+18,  YLM.20
     * 5.1090942171709440000000000D+19, 1.1240007277776076800000000D+21,  YLM.21
     * 2.5852016738884976640000000D+22, 6.2044840173323943936000000D+23,  YLM.22
     * 1.5511210043330985984000000D+25, 4.0329146112660563558400000D+26,  YLM.23
     * 1.0888869450419352160768000D+28, 3.0488834461171386050150400D+29,  YLM.24
     * 8.8417619937397019545436160D+30, 2.6525285981219105863630850D+32/  YLM.25
      PI=3.141592654                                                    YLM.26
11    PI4=4*PI                                                          YLM.27
C                                                                       YLM.28
C                                                                       YLM.29
12    VECTOR(1)=VX                                                      YLM.30
14    VECTOR(2)=VY                                                      YLM.31
15    VECTOR(3)=VZ                                                      YLM.32
C                                                                       YLM.33
16    IF(LMAX.GT.15)  GO TO 500                                         YLM.34
21    IF(LMAX.GT.0)  GO TO 50                                           YLM.35
23    Y(1)=1./SQRT(PI4)                                                 YLM.36
22    RETURN                                                            YLM.37
33 50 CONTINUE                                                          YLM.38
33    LMAX1=LMAX+1                                                      YLM.39
34    LMX=LMAX-1                                                        YLM.40
35    IND=1                                                             YLM.41
C                                                                       YLM.42
C                                                                       YLM.43
36    R=VECTOR(1)**2+VECTOR(2)**2                                       YLM.44
41    RHO=SQRT(R)                                                       YLM.45
43    RR=SQRT(R+VECTOR(3)**2)                                           YLM.46
50    IF( RR.EQ.0.0)  GO TO 400                                         YLM.47
57    STH=RHO/RR                                                        YLM.48
60    CTH=VECTOR(3)/RR                                                  YLM.49
61    IF(RHO.EQ.0.0)  GO TO 100                                         YLM.50
64    SPH=VECTOR(2)/RHO                                                 YLM.51
65    CPH=VECTOR(1)/RHO                                                 YLM.52
67    GO TO 110                                                         YLM.53
67 100 CONTINUE                                                         YLM.54
67    SPH=0.0                                                           YLM.55
70    CPH=0.0                                                           YLM.56
C                                                                       YLM.57
C     ONLY M=0 SPHERICAL HARMONICS ARE NON-ZERO ***                     YLM.58
C                                                                       YLM.59
```

```
 70          INC=0                                                  YLM.60
 71          LX2=LMAX1**2                                           YLM.61
 73          DO 105  I=1,LX2                                        YLM.62
101          Y(I)=(0.0,0.0)                                         YLM.63
102      105 CONTINUE                                               YLM.64
      C                                                             YLM.65
110      110 CONTINUE                                               YLM.66
110          X=CMPLX(CPH,SPH)                                       YLM.67
      C                                                             YLM.68
      C      COMPUTE M=0 VALUES FOR  (P)  AND  (Y) ***              YLM.69
      C                                                             YLM.70
113          P(1)=1.0                                               YLM.71
114          Y(1)=1./SQRT(PI4)                                      YLM.72
124          P(2)=CTH                                               YLM.73
126          Y(3)=SQRT(3.0/PI4)*P(2)                                YLM.74
      C                                                             YLM.75
137          DO 120  L=2,LMAX                                       YLM.76
141          L2=L-1                                                 YLM.77
142          L1=L+1                                                 YLM.78
143          LL=L*(L+1)+1                                           YLM.79
146          CFT=SQRT( (2*L+1)/PI4 )                                YLM.80
161          P(L1)=( (2*L2+1)*CTH*P(L)-L2*P(L2) )/L                 YLM.81
173          Y(LL)=CFT*P(L1)                                        YLM.82
176      120 CONTINUE                                               YLM.83
200          IF(INC.EQ.0)  GO TO 140                                YLM.84
      C                                                             YLM.85
      C      COMPUTE  M.NE.0  VALUES OF (Y) ***                     YLM.86
      C                                                             YLM.87
201          XX=(1.0,0.0)                                           YLM.88
204          DO 130  M=1,LMAX                                       YLM.89
205          XX=XX*X                                                YLM.90
213          SIGN=(-1.0)**M                                         YLM.91
216          NN=(M+1)**2                                            YLM.92
220          NM=NN-2*M                                              YLM.93
222          M2=2*M+1                                               YLM.94
223          CF1=DSQRT(M2*FAC(M2)/PI4)                              YLM.95
252          CF2=SIGN*FAC(M+1)*2.0**M                               YLM.96
266          Y(NN)=CF1/CF2*STH**M*XX                                YLM.97
304          Y(NM)=SIGN*CONJG(Y(NN))                                YLM.98
315          IF(M.EQ.LMAX)  GO TO 140                               YLM.99
      C                                                             YLM.100
317          DO 130  L=M,LMX                                        YLM.101
320          L1=L+1                                                 YLM.102
321          CL1=(2*L+3)*(L-M+1)                                    YLM.103
326          CL2=(2*L+1)*(L+M+1)                                    YLM.104
333          CF1=SQRT(CL1/CL2)                                      YLM.105
340          CL1=L+M                                                YLM.106
342          CL2=L-M+1                                              YLM.107
344          CF2=SQRT(CL1/CL2)                                      YLM.108
      C                                                             YLM.109
354          L2=L*(L+1)                                             YLM.110
357          N1=L2+2*L+M+3                                          YLM.111
362          N2=L2+M+1                                              YLM.112
363          N3=L2+M                                                YLM.113
364          N4=N1-2*M                                              YLM.114
      C                                                             YLM.115
366          Y(N1)=CF1*(CTH*Y(N2)-CF2*STH*X*Y(N3) )                 YLM.116
422          Y(N4)=SIGN*CONJG(Y(N1))                                YLM.117
```

```
      C                                                                 YLM.118
437     130 CONTINUE                                                    YLM.119
444     140 CONTINUE                                                    YLM.120
      C                                                                 YLM.121
444         IF(CONJ.NE.(-1))  RETURN                                    YLM.122
446         LXX2=LMAX1**2                                               YLM.123
450         DO 150  I=1,LXX2                                            YLM.124
456         Y(I)=CONJG(Y(I))                                            YLM.125
460     150 CONTINUE                                                    YLM.126
461         RETURN                                                      YLM.127
      C                                                                 YLM.128
462     400 CONTINUE                                                    YLM.129
462         PRINT 800, VECTOR                                           YLM.130
474     500 CONTINUE                                                    YLM.131
474         PRINT 802, LMAX                                             YLM.132
      C                                                                 YLM.133
      C                                                                 YLM.134
        800 FORMAT( /// 10X 39HSPHERICAL HARMONIC ROUTINE *** VECTOR=(  3E16.8YLM.135
           *  , 1H)   /// )                                             YLM.136
        802 FORMAT( /// 10X 45HSUBROUTINE.YLM *** L-VALUE IS TOO LARGE  (L=  YLM.137
           2  I3,*)*  )                                                 YLM.138
507         RETURN                                                      YLM.139
507         END                                                         YLM.140
```

```
SUBPROGRAM LENGTH
000773

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
50      -  000034    100     -  000070    110     -  000111    140     -  000445
400     -  000463    500     -  000475    800     -  000527    802     -  000540

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
B       -  000741    BCP     -  000732    CFT     -  000756    CF1     -  000622
CF2     -  000624    CL1     -  000764    CL2     -  000765    CPH     -  000747
CTH     -  000745    FAC     -  000626    FACT    -  000724    I       -  000751
IND     -  000740    L       -  000752    LL      -  000755    LMAX1   -  000736
LMX     -  000737    LXX2    -  000772    LX2     -  000750    L1      -  000754
L2      -  000753    M       -  000757    M2      -  000763    MM      -  000762
N1      -  000761    N1      -  000766    N2      -  000767    N3      -  000770
N4      -  000771    P       -  000604    PI      -  000734    PI4     -  000735
RHO     -  000742    PR      -  000743    SIGN    -  000760    SPH     -  000746
STH     -  000744    VECTOR  -  000617    X       -  000725    XX      -  000730

START OF CONSTANTS
000512

START OF TEMPORARIES
000551

START OF INDIRECTS
000577
```

```
      PROGRAM READIN                                                    READIN.3
C                                                                       READIN.4
      COMMON/BLOCKA/A(10,2),B(10,2),D(10,3),RA(10,2),RB(10,2),ALAT,SPACEA.2
     2,BASIS(2),NBASIS                                                  A.3
      COMMON/BLOCKB/T(2,8C,7),ILYR(10),LSYM(10)                         B.2
      COMPLEX T                                                         B.3
      COMMON/BLOCKC/V1(3),V2(3,12),V3(3,12),VX,VY,VZ,SGK(12),CGK(12),   C.2
     2  YIN(49),YOUT(49,12),XKZERO,AREA2                                C.3
      COMPLEX YIN,YOUT                                                  C.4
      COMMON/BLOCKD/NLOW,NNUM,NDELTA,NRG,NRGY,LAYER,NUMMAX,NDIF,IST(6,5)D.2
     2  ,LR                                                             D.3
      COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE                                               E.3
      COMMON/BLOCKF/ITHT,IPHI,THTT,DIST,CSTH(12),VXX(2,12),KNUM,JM(12), F.2
     2  JN(12),IQ(12),KQ(12)                                            F.3
      COMMON/BLOCKG/ESIG,SIG2,XK,DAMP,ZNORM,POTL,GAMA,SIGMA,LOWEN,NNRG  G.2
     2  ,LAMBDA                                                         G.3
      REAL LAMBDA                                                       G.4
      INTEGER SIGMA                                                     G.5
      COMPLEX ZNORM                                                     G.6
      COMMON/BLOCKH/KX(2401),C(7532),LL(7532),YINT(169)                 H.2
      COMPLEX YINT                                                      H.3
      COMMON/BLOCKJ/MAXX(91),RMAG( 9000),XMAG( 9000),MAXR               J.2
      COMMON/BLOCKK/BRB(36,2),TAU(36,36,2),TT(10,12),KTYP(10)           K.2
      COMPLEX  BBR,TAU,TT                                               K.3
      COMMON/BLOCKL/ARG,F(169),XD(9),PATH1(10),PATH2(10,12)             L.2
      COMPLEX  ARG,F                                                    L.3
      COMMON/BLOCKM/RATIO,XCON,KSTAR                                    M.2
      COMPLEX KSTAR,XCON                                                M.3
      COMMON/BLOCKN/E(51),EN(51),S(12,51)                               N.2
      COMMON/BLOCKX/TIME,ITEM,MULT,ISCALE,IMAX,ITYPE,LSUM,ILOOP,IPWR    X.2
      INTEGER TIME                                                      X.3
      COMMON/BLKDER/W(10),ATWT(10),TEMP,SURF,BULK,DIN(10),DOUT(10,12)   DER.2
      COMMON/CONST/PI,PI4,PI42,CON1,CON2,CON3,CON4,CONSTT,IM,CNST       CONST.2
      COMPLEX IM                                                        CONST.3
C                                                                       READIN.21
C                                                                       READIN.22
C     READ IN INSTRUCTIONS CONTAINED IN FOLLOWING SUBROUTINES *****     READIN.23
C                                                                       READIN.24
      READ 8C8,CA,TE                                                    READIN.25
 12   READ 8CC, H1,H2,H3,H4,H5,H6,H7,H8                                 READIN.26
 36   PRINT 8C1                                                         READIN.27
 42   PRINT 8C2, H1,H2,H3,H4,H5,H6,H7,H8,CA,TE                          READIN.28
C                                                                       READIN.29
C     MULT=0 *** SINGLE SCATTERING , NO PRINTOUT                        READIN.30
C     MULT=1 *** SINGLE SCATTERING , PLOT SINGLE                        READIN.31
C     MULT=2 *** DOUBLE SCATTERING , PLOT DOUBLE                        READIN.32
C     MULT=3 *** DOUBLE SCATTERING , PLOT SINGLE+DOUBLE+COMBINED        READIN.33
C     MULT=4 *** DOUBLE SCATTERING , PLOT SINGLE+DOUBLE+COMBINED,(SAME SREADIN.34
C     MULT=5 *** DOUBLE SCATTERING , NO PRINTOUT                        READIN.35
C                                                                       READIN.36
 72   READ 8C3, MULT,MAXR,LSUM,TIME                                     READIN.37
C                                                                       READIN.38
C     READ SELF ENERGY PARAMETERS                                       READIN.39
C                                                                       READIN.40
106   READ 8C6, SIGMA,POTL,GAMA                                         READIN.41
120   IF(SIGMA.EQ.2)  LAMBDA=GAMA                                       READIN.42
C                                                                       READIN.43
```

```
      C     READ MATRIX INVERSION INSTRUCTIONS                          READIN.44
      C                                                                 READIN.45
 124          READ 804,NDIF                                             READIN.46
      C                                                                 READIN.47
      C     READ ATOMIC SCATTERING FACTORS *****                        READIN.48
      C                                                                 READIN.49
 132          CALL ATOMIC                                               READIN.50
 133          LMAX1=LMAX+1                                              READIN.51
 135          LMAX2=2*LMAX+1                                            READIN.52
 136          LMAX3=LMAX1**2                                            READIN.53
 137          LMAX4=LMAX3**2                                            READIN.54
 141          CALL STACK                                                READIN.55
 142          I=ITEM                                                    READIN.56
 143          ITEM=2                                                    READIN.57
 144          IF(ITYPE.EQ.0)  STOP                                      READIN.58
 150          ITEM=I                                                    READIN.59
 152      110 CONTINUE                                                  READIN.60
      C                                                                 READIN.61
      C     READ LATTICE PARAMETERS BY SUBPLANES *****                  READIN.62
      C                                                                 READIN.63
 152          CALL LATTICE                                              READIN.64
 153      120 CONTINUE                                                  READIN.65
      C                                                                 READIN.66
      C     READ BEAM COORDINATES,ENERGY RANGE AND DAMPING PARAMETERS ***** READIN.67
      C                                                                 READIN.68
 153          PRINT 801                                                 READIN.69
 157          PRINT 802, H1,H2,H3,H4,H5,H6,H7,H8,DA,TE                  READIN.70
 207          CALL BEAM                                                 READIN.71
 210          CALL DEBYE                                                READIN.72
 211      140 CONTINUE                                                  READIN.73
      C                                                                 READIN.74
      C     ESTABLISH LOW ENERGY (NLOW) AND K-VALUE *****               READIN.75
      C                                                                 READIN.76
 211          CALL ENERGY(NLOW,NRGY)                                    READIN.77
 213          IF(NRGY.GT.0)  GO TO 145                                  READIN.78
 216          IF(NLOW.GT.200)  STOP                                     READIN.79
 222          NLOW=NLOW+2                                               READIN.80
 224          GO TO 140                                                 READIN.81
      C                                                                 READIN.82
 224      145 CONTINUE                                                  READIN.83
 224          XKZERO=1./XK                                              READIN.84
 226          XKZ=XK                                                    READIN.85
      C                                                                 READIN.86
      C     CLEAR ENERGY AND INTENSITY ARRAYS *****                     READIN.87
      C                                                                 READIN.88
 226          NRG=NLOW-NDELTA                                           READIN.89
 231          DO 150  J=1,51                                            READIN.90
 245          EN(J)=0.0                                                 READIN.91
 246          NRG=NRG+NDELTA                                            READIN.92
 247          E(J)=NRG                                                  READIN.93
 250          S(1,J)=0.0                                                READIN.94
 251          S(2,J)=0.0                                                READIN.95
 251          S(3,J)=0.0                                                READIN.96
 252          S(4,J)=0.0                                                READIN.97
 252      150 CONTINUE                                                  READIN.98
      C                                                                 READIN.99
      C     SET INCIDENT K-VECTOR DIRECTION *****                       READIN.100
      C                                                                 READIN.101
```

```
255          THETA=ITHT*CCN3                                          READIN.102
257          PHI=IPHI*CCN2                                            READIN.103
261          ST=SIN(THETA)                                            READIN.104
262          VX=ST*CCS(PHI)                                           READIN.105
265          VY=ST*SIN(PHI)                                           READIN.106
270          VZ=CCS(THETA)                                            READIN.107
273          THTT=1./VZ                                               READIN.108
274          V1(1)=VX*XKZ                                             READIN.109
276          V1(2)=VY*XKZ                                             READIN.110
277          V1(3)=VZ*XKZ                                             READIN.111
       C                                                              READIN.112
       C     SPHERICAL HARMONICS FOR INCIDENT BEAM *****              READIN.113
       C                                                              READIN.114
300          LOX=2*LMAX                                               READIN.115
302          CALL YLM(V1(1),V1(2),V1(3),YIN,LOX,+1)                   READIN.116
306          NRG=NLCW-NDELTA                                          READIN.117
310          IF(NNUM.LE.51)  GO TO 160                                READIN.118
313          ISCALE=2                                                 READIN.119
313          IMAX=101                                                 READIN.120
315      160 CONTINUE                                                 READIN.121
315          IF(TIME.NE.0)  CALL SECOND(TIME1)                        READIN.122
320          CALL TABLE                                               READIN.123
321           IF(TIME.NE.0)  CALL SECOND(TIME2)                       READIN.124
324          IF(TIME.NE.0)  PRINT 834, TIME1,TIME2                    READIN.125
335          AREA2=CONSTT/(A(1,1)*B(1,2)-A(1,2)*B(1,1))**2            READIN.126
341          DIST=ABS(C(9,3)-C(8,3))*0.5                              READIN.127
       C                                                              READIN.128
       C                                                              READIN.129
         800 FORMAT( 8A10)                                            READIN.130
         801 FORMAT( 1H1 )                                            READIN.131
         802 FORMAT(   / 10X 6H*****   8A10, 14X 2A10, // )           READIN.132
         803 FORMAT( 4I5 )                                            READIN.133
         804 FORMAT( I5 )                                             READIN.134
         806 FORMAT( 5X I2,2F12.4 )                                   READIN.135
         808 FORMAT( 2A10 )                                           READIN.136
         834 FORMAT( / 10X*TIME FOR EXECUTION OF TABLE * F8.3,* TO *F8.3 )  READIN.137
       C                                                              READIN.138
       C                                                              READIN.139
345          END                                                      READIN.140
```

PROGRAM LENGTH INCLUDING I/O BUFFERS
000463

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

110	-	000152	120	-	000154	140	-	000212	145	-	000225
160	-	000316	800	-	000363	801	-	000365	802	-	000367
803	-	000375	804	-	000377	806	-	000401	808	-	000404
834	-	000406									

BLOCK NAMES AND LENGTHS

BLOCKA	-	000163	BLOCKB	-	004324	BLOCKC	-	002542	BLOCKD	-	000055
BLOCKE	-	000012	BLOCKF	-	000131	BLOCKG	-	000014	BLOCKH	-	042613
BLOCKJ	-	043254	BLOCKK	-	012712	BLOCKL	-	000737	BLOCKM	-	000005
BLOCKN	-	001312	BLOCKX	-	000011	BLKDEB	-	000231	CONST	-	000013

```
      SUBROUTINE ATOMIC                                                 ATOMIC.2
C                                                                       ATOMIC.3
C     ITYPE  IS NUMBER OF DIFFERENT ATOMIC TYPES *****                  ATOMIC.4
C                                                                       ATOMIC.5
      COMMON/BLOCKB/T(2,80,7),ILYR(10),LSYM(10)                         B.2
      COMPLEX T                                                         B.3
      COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE                                               E.3
      COMMON/BLOCKG/ESIG,SIG2,XK,DAMP,ZNORM,POTL,GAMA,SIGMA,LOWER,NNRG  G.2
     2  ,LAMBDA                                                         G.3
      REAL LAMBDA                                                       G.4
      INTEGER SIGMA                                                     G.5
      COMPLEX ZNORM                                                     G.6
      COMMON/BLOCKX/TIME,ITEM,MULT,ISCALE,IMAX,ITYPE,LSUM,ILOOP,IPWR    X.2
      INTEGER TIME                                                      X.3
      COMMON/BLKDER/W(10),ATWT(10),TEMP,SURF,BULK,DIN(10),DCLT(10,12)   DER.2
C                                                                       ATOMIC.11
      READ 800, ITYPE                                                   ATOMIC.12
    7 IF(ITYPE.EQ.0)  GO TO 200                                         ATOMIC.13
   10 READ 801, LMAX                                                    ATOMIC.14
C                                                                       ATOMIC.15
C                                                                       ATOMIC.16
   16 DO 100  I=1,ITYPE                                                 ATOMIC.17
   20 READ 802, H1,H2,H3,H4,H5,H6                                       ATOMIC.18
   27 READ 808, ATWT(I)                                                 ATOMIC.19
   46 READ 803, LTOP,LOWER,NNRG                                         ATOMIC.20
   60 IF(LMAX.GE.LTOP)  LMAX=LTOP-1                                     ATOMIC.21
C                                                                       ATOMIC.22
   65 DO 90  L=1,LTOP                                                   ATOMIC.23
   67 READ 804, (T(I,J,L), J=1,NNRG)                                    ATOMIC.24
  105   90 CONTINUE                                                     ATOMIC.25
  110 PRINT 806, LMAX,I,H1,H2,H3,H4,H5,H6                               ATOMIC.26
  133  100 CONTINUE                                                     ATOMIC.27
  136  200 CONTINUE                                                     ATOMIC.28
C                                                                       ATOMIC.29
C                                                                       ATOMIC.30
  800 FORMAT( I2 )                                                      ATOMIC.31
  801 FORMAT( I5 )                                                      ATOMIC.32
  802 FORMAT( 6A10 )                                                    ATOMIC.33
  803 FORMAT( 3I5 )                                                     ATOMIC.34
  804 FORMAT( 5E16.8)                                                   ATOMIC.35
  806 FORMAT( // 10X *T-FACTORS READ IN TO L=* I3, * FOR ATOMIC TYPE *  ATOMIC.36
     2  I2, *.... * 6A10 )                                              ATOMIC.37
  808 FORMAT( F10.5 )                                                   ATOMIC.38
C                                                                       ATOMIC.39
  136 RETURN                                                            ATOMIC.40
  137 END                                                               ATOMIC.41
```

```
SUBPROGRAM LENGTH
000211

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
200    -  000137    800    -  000145    801    -  000147    802    -  000151
803    -  000153    804    -  000155    806    -  000160    808    -  000172
```

```
      SUBROUTINE STACK                                                  STACK.2
C                                                                       STACK.3
C     THIS SUBROUTINE DOES INTEGRAL OVER 3 SPHERICAL HARMONICS FOR WHICHSTACK.4
C     (L1,M1) AND (L2,M2) ARE INDICES OF TWO OF THEM.                   STACK.5
C     LL(NUM) DELIVERS COMBINED INDICES FOR (L3,M3), FOR WHICH THE INTEGSTACK.6
C     IS NOT EQUAL TO ZERO.  (L3,M3) ARE INDICES OF COMPLEX CONJUGATE FUSTACK.7
C                                                                       STACK.8
C                                                                       STACK.9
      COMMON/BLOCKD/NLOW,NNUM,NDELTA,ARG,NRGY,LAYER,NUMMAX,NDIF,IST(6,6)D.2
     2  ,LR                                                             D.3
      COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE                                               E.3
      COMMON/BLOCKH/KX(2400),C(7532),LL(7532),YINT(169)                 H.2
      COMPLEX YINT                                                      H.3
C                                                                       STACK.13
      NUM=0                                                             STACK.14
2     ISTEP=0                                                           STACK.15
C                                                                       STACK.16
C     LOW  ARRAY  C(IN,OUT)  IS STACKED BACKWARDS IN ORDER              STACK.17
C     TO PERFORM MATRIX MULTIPLICATION EFFICIENTLY.                     STACK.18
C     NOTE ... INCOMING BEAM INDEX VARIES MOST RAPIDLY FOR C(IN,OUT) ***STACK.19
C                                                                       STACK.20
C     SUM OVER (LOUT) *****                                             STACK.21
C                                                                       STACK.22
3     DO 100  LL1=1,LMAX1                                               STACK.23
4     MX1=2*LL1-1                                                       STACK.24
5     K1=-LL1                                                           STACK.25
6     L3=LL1-1                                                          STACK.26
10    DO 100  MM1=1,MX1                                                 STACK.27
11    M3=MM1+K1                                                         STACK.28
C                                                                       STACK.29
C     SUM OVER (LIN) *****                                              STACK.30
C                                                                       STACK.31
12    DO 100  LL2=1,LMAX1                                               STACK.32
14    MX2=2*LL2-1                                                       STACK.33
15    K2=-LL2                                                           STACK.34
16    L2=LL2-1                                                          STACK.35
20    DO 100  MM2=1,MX2                                                 STACK.36
21    M2=MM2+K2                                                         STACK.37
C                                                                       STACK.38
C                                                                       STACK.39
22    M1=M2-M3                                                          STACK.40
24    ISTEP=ISTEP+1                                                     STACK.41
26    MM=IABS(M1)+1                                                     STACK.42
27    LOW=IABS(L2-L3)+1                                                 STACK.43
33    LHI=L2+L3+1                                                       STACK.44
34    KKK=0                                                             STACK.45
C                                                                       STACK.46
C     SUM OVER  (LAMBDA)  *****                                         STACK.47
C                                                                       STACK.48
35    DO  90  LL3=LOW,LHI,2                                             STACK.49
36    IF(LL3.LT.MM) GO TO 90                                            STACK.50
40    KKK=KKK+1                                                         STACK.51
42    NUM=NUM+1                                                         STACK.52
43    L1=LL3-1                                                          STACK.53
C                                                                       STACK.54
44    C(NUM)=COEFT(L1,L3,L2,M1,M3,M2)                                   STACK.55
52    LL(NUM)=L2*(L1+1)+M1+1                                            STACK.56
```

```
 57      90 CONTINUE                                                    STACK.57
 62         KX(ISTEP)=KKK                                               STACK.58
      C                                                                 STACK.59
 64     100 CONTINUE                                                    STACK.60
 76         NUMMAX=NUM                                                  STACK.61
      C                                                                 STACK.62
 77         PRINT 800,LMAX,ISTEP,L1,M1,L2,M2,L3,M3,NUMMAX               STACK.63
      C                                                                 STACK.64
      C                                                                 STACK.65
        800 FORMAT( / 10X*SUBROUTINE STACK FORMS INTERMEDIATE ARRAYS (LL) AND STACK.66
           2(C),*  / 10X*L=* I3,*   LMAX4=*I5,*  (L1,M1,L2,M2,L3,M3)=(*  STACK.67
           3  6I3, *)   NUMMAX=* I5 )                                   STACK.68
      C                                                                 STACK.69
125         RETURN                                                      STACK.70
126         END                                                         STACK.71
```

```
SUBPROGRAM LENGTH
000211

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
90      -  000060     800     -  000133

BLOCK NAMES AND LENGTHS
BLOCKD -  000055     BLOCKE -  000012     BLOCKH -  042613

VARIABLE ASSIGNMENTS
C       -  004541S03 IST     -  000010S01 ISTEP   -  000165    KKK     -  000206
KX      -  000000S03 K1      -  000170    K2      -  000176    LHI     -  000205
LL      -  023315S03 LL1     -  000166    LL2     -  000174    LL3     -  000207
LMAX    -  000000S02 LMAX1   -  000002S02 LOW     -  000204    L1      -  000210
L2      -  000177    L3      -  000171    MM      -  000203    MM1     -  000172
MM2     -  000200    MX1     -  000167    MX2     -  000175    M1      -  000202
M2      -  000201    M3      -  000173    NUM     -  000164    NUMMAX  -  000006S01
YINT    -  042071S03

START OF CONSTANTS
000131

START OF TEMPORARIES
000154

START OF INDIRECTS
000162

UNUSED COMPILER SPACE
124500
```

```
      FUNCTION CCEFT(L1,L2,L3,M1,M2,M3)                                 COEFT.2
C                                                                       COEFT.3
      DIMENSION  X(6),Y(6)                                              CCEFT.4
C                                                                       COEFT.5
      X(1)=L1                                                           CCEFT.6
      Y(1)=L1                                                           CCEFT.7
      X(2)=L2                                                           COEFT.8
      Y(2)=L2                                                           CCEFT.9
      X(3)=L3                                                           COEFT.10
      Y(3)=L3                                                           CCEFT.11
      X(4)=M1                                                           CCEFT.12
      X(5)=M2                                                           COEFT.13
      X(6)=M3                                                           CCEFT.14
      Y(4)=C.C                                                          COEFT.15
      Y(5)=C.C                                                          CCEFT.16
      Y(6)=C.C                                                          CCEFT.17
C                                                                       COEFT.18
      CFT=SQRT((2*L1+1)*(2*L2+1)/(12.566370*(2*L3+1)))                  COEFT.19
      CCEFT=CFT*CLEB(X)*CLEB(Y)                                         COEFT.20
C                                                                       COEFT.21
C                                                                       CCEFT.22
      RETURN                                                            COEFT.23
      END                                                               CCEFT.24
```

```
      FUNCTION CLEB(X)                                                  CLEB.2
      DIMENSION X(6)                                                    CLEB.3
      DIMENSION FACT(101)                                               CLEB.4
      DATA EPS/10.E-5/                                                  CLEB.5
      DATA(FACT(I),I=1,30)/                                             CLEB.6
     X 1.000000000000000E+00, 1.000000000000000E+00,                    CLEB.7
     X 1.414213562373410E+00, 2.449489742783710E+00,                    CLEB.8
     X 4.898979485566356E+00, 1.095445115010800E+01,                    CLEB.9
     X 2.683281573000885E+01, 7.099255737972240E+01,                    CLEB.10
     X 2.007984065683010E+02, 6.023952191048880E+02,                    CLEB.11
     X 1.904940940396762E+03, 6.317974358925950E+03,                    CLEB.12
     X 2.188610518115345E+04, 7.891147445084570E+04,                    CLEB.13
     X 2.952597012802185E+05, 1.143535950586437E+06,                    CLEB.14
     X 4.574143623458270E+06, 1.885967730626340E+07,                    CLEB.15
     X 8.001483428548570E+07, 3.487765766345650E+08,                    CLEB.16
     X 1.559776268628950E+09, 7.147792818187560E+09,                    CLEB.17
     X 3.352812008337670E+10, 1.607856235454200E+11,                    CLEB.18
     X 7.876854713229570E+11, 3.938427356614530E+12,                    CLEB.19
     X 2.008211794424290E+13, 1.043497458094200E+14,                    CLEB.20
     X 5.521469535670540E+14, 2.973510046011420E+15/                    CLEB.21
      DATA(FACT(I),I=31,60)/                                            CLEB.22
     X 1.628658527168630E+16, 9.067986906786410E+16,                    CLEB.23
     X 5.129628266011950E+17, 2.946746955339330E+18,                    CLEB.24
     X 1.718233974286170E+19, 1.016520927790880E+20,                    CLEB.25
     X 6.099125566743510E+20, 3.709553246495910E+21,                    CLEB.26
     X 2.286968774304130E+22, 1.428211541792820E+23,                    CLEB.27
     X 9.032802905216800E+23, 5.783815921430640E+24,                    CLEB.28
     X 3.748341123414030E+25, 2.457951648490470E+26,                    CLEB.29
     X 1.630420674175550E+27, 1.093719437813100E+28,                    CLEB.30
     X 7.417966136206320E+28, 5.085501366725750E+29,                    CLEB.31
     X 3.523338699633900E+30, 2.466337089757220E+31,                    CLEB.32
     X 1.743993680958760E+32, 1.245439180883170E+33,                    CLEB.33
     X 8.980998654291020E+33, 6.538259159770850E+34,                    CLEB.34
     X 4.804619624253910E+35, 3.563201278846410E+36,                    CLEB.35
     X 2.655455671111360E+37, 2.013129889117190E+38,                    CLEB.36
     X 1.533154046814450E+39, 1.177637968751370E+40/                    CLEB.37
      DATA(FACT(I),I=61,90)/                                            CLEB.38
     X 9.121944816679300E+40, 7.124465393154760E+41,                    CLEB.39
     X 5.609810447782430E+42, 4.452649004111660E+43,                    CLEB.40
     X 3.563119203269330E+44, 2.871872314708400E+45,                    CLEB.41
     X 2.333120097790110E+46, 1.909741105955410E+47,                    CLEB.42
     X 1.574812859487110E+48, 1.308137807824050E+49,                    CLEB.43
     X 1.094466130046200E+50, 9.222139602914510E+50,                    CLEB.44
     X 7.825244940320430E+51, 6.685922078091400E+52,                    CLEB.45
     X 5.731421947193550E+53, 4.980877514158200E+54,                    CLEB.46
     X 4.342228346865210E+55, 3.810289910557210E+56,                    CLEB.47
     X 3.365156932149470E+57, 2.991016957701700E+58,                    CLEB.48
     X 2.675248649262250E+59, 2.407722164236140E+60,                    CLEB.49
     X 2.180628915036680E+61, 1.986334304609700E+62,                    CLEB.50
     X 1.820506461261820E+63, 1.678423103484870E+64,                    CLEB.51
     X 1.556505555974130E+65, 1.451811729640340E+66,                    CLEB.52
     X 1.361920123400390E+67, 1.284832874752820E+68/                    CLEB.53
      DATA(FACT(I),I=91,101)/                                           CLEB.54
     X 1.218659489043040E+69, 1.162756005204140E+70,                    CLEB.55
     X 1.115276380735020E+71, 1.075533591778910E+72,                    CLEB.56
     X 1.042768505767750E+73, 1.015365017495110E+74,                    CLEB.57
     X 9.958302741108330E+74, 9.807790764431750E+75,                    CLEB.58
     X 9.709217501182770E+76, 9.660549437809040E+77,                    CLEB.59
```

```
              X 9.66C5494377984EE+7d/                                   CLEB.60
               CLEB=C.                                                  CLEB.61
      2        NA=2.*X(1)+EPS                                           CLEB.62
      5        NB=2.*X(2)+EPS                                           CLEB.63
     10        NC=2.*X(3)+EPS                                           CLEB.64
     12        ND=SIGN(ABS(2.*X(4))+EPS,X(4))                           CLEB.65
     20        NE=SIGN(ABS(2.*X(5))+EPS,X(5))                           CLEB.66
     26        NF=-SIGN(ABS(2.*X(6))+EPS,X(6))                          CLEB.67
     33        IF(ABS(X(4)+X(5)-X(6)) .GT. EPS) RETURN                  CLEB.68
     43        L3=X(2)+X(5)+EPS                                         CLEB.69
     46        L8=X(1)+X(4)+EPS                                         CLEB.70
     51        L10=X(3)-X(6)+EPS                                        CLEB.71
     54        IF(ABS(X(2)+X(5)-FLOAT(L3)) .GT. EPS) RETURN             CLEB.72
     64        IF(ABS(X(1)+X(4)-FLOAT(L8)).GT. EPS) RETURN              CLEB.73
     73        IF(ABS(X(3)-X(6)-FLOAT(L10)) .GT. EPS) RETURN            CLEB.74
    103        LC=L3+L8+L1C                                             CLEB.75
    105        IF(LC .GT. 99) GC TC 2                                   CLEB.76
    111        IF(IABS(ND) .GT. NA  .OR.                                CLEB.77
              X    IABS(NE) .GT. NB  .OR.                               CLEB.78
              X      IABS(NF) .GT. NC) RETURN                           CLEB.79
    130        L1=L0-NC                                                 CLEB.80
    132        L2=L8-NC                                                 CLEB.81
    134        L9=L3-NE                                                 CLEB.82
    136        L11=L1C-NF                                               CLEB.83
    140        L6=L0-NB                                                 CLEB.84
    142        L7=LC-NA                                                 CLEB.85
    143        L4=L6-L2                                                 CLEB.86
    145        L5=L7-L3                                                 CLEB.87
    147        L2C=NC+1                                                 CLEB.88
    151        KMIN=MAXO(0,-L4,-L5)                                     CLEB.89
    157        KMAX=MINO(L1,L2,L3)                                      CLEB.90
    165        IF(KMAX .LT. KMIN)RETURN                                 CLEB.91
    172        CC1=FACT(L1+1)*FACT(L6+1)*FACT(L7+1)*FACT(L8+1)          CLEB.92
              X     *FACT(L2+1)*FACT(L3+1)                              CLEB.93
              X*FACT(L9+1)*FACT(L10+1)*FACT(L11+1)/FACT(L0+2)          CLEB.94
              X   *FACT(L20+1)/FACT(L20)                               CLEB.95
    215        SGN=1.                                                   CLEB.96
    216        DC 1 K=KMIN,KMAX                                         CLEB.97
    230        L12=L1-K+1                                               CLEB.98
    232        L13=L2-K+1                                               CLEB.99
    234        L14=L3-K+1                                               CLEB.100
    235        L15=L4+K+1                                               CLEB.101
    236        L16=L5+K+1                                               CLEB.102
    240        DEN=FACT(K+1)*FACT(L12)*FACT(L13)*FACT(L14)*FACT(L15)*FACT(L16)  CLEB.103
    247        CLEB=CLEB+SGN*CC1/DEN**2                                 CLEB.104
    252        SGN=-SGN                                                 CLEB.105
    253      1 CONTINUE                                                 CLEB.106
    263        IF(MOD(KMIN,2) .NE. 0) CLEB=-CLEB                        CLEB.107
    267        RETURN                                                   CLEB.108
    271      2 ARFF=C.                                                  CLEB.109
    272        CLEB=C./ARFF                                             CLEB.110
    274        RETURN                                                   CLEB.111
    274        END                                                      CLEB.112
```

SUBPROGRAM LENGTH
000565

```
      SUBROUTINE LATTICE                                                LATTICE.2
C                                                                       LATTICE.3
C                                                                       LATTICE.4
      COMMON/BLOCKA/A(10,2),B(10,2),D(10,2),RA(10,2),RB(10,2),ALAT,SPACEA.2
     2,BASIS(2),NBASIS                                                  A.3
      COMMON/BLOCKB/T(2,80,7),ILYR(10),LSYM(10)                         B.2
      COMPLEX T                                                         B.3
      COMMON/BLOCKD/NLCW,NNUM,NDELTA,NRG,NRGY,LAYER,NUMMAX,NCTF,IST(6,6)D.2
     2  ,LF                                                             D.3
      COMMON/BLOCKK/BBB(36,36,2),TAU(36,36,2),TT(10,12),KTYP(10)        K.2
      COMPLEX  BBB,TAU,TT                                               K.3
      COMMON/BLOCKL/ARG,H(169),XD(9),PATH1(10),PATH2(10,12)             L.2
      COMPLEX  ARG,H                                                    L.3
      COMMON/BLOCKX/TIME,ITEM,MULT,ISCALE,IMAX,ITYPE,LSUP,ILOOP,IPWR    X.2
      INTEGER TIME                                                      X.3
      COMMON/CONST/PI,PI4,PI42,CON1,CON2,CON3,CON4,CONSTT,IM,CNST       CONST.2
      COMPLEX IM                                                        CONST.3
C                                                                       LATTICE.12
      DIMENSION  AX(2),BX(2),SHIFT(3),MM(4)                             LATTICE.13
      DIMENSION Q(27,8)                                                 LATTICE.14
      INTEGER TYPE,FACE                                                 LATTICE.15
      DATA(Q(I),I=1,27)  /.5,0.,0.,.5,.5,.5,-.5,0.,.5,0.,.5,0.,.5,0.,.5,LATTICE.16
     20.,.5,10*0.0/                                                     LATTICE.17
      DATA(Q(I),I=28,54) /.35355339,0.,0.,0.,1.,.70710678,0.,           LATTICE.18
     2 0.,.35355339,0.,.35355339,0.,35355339,0.,.35355339,0.,.35355339, LATTICE.19
     3 0.,.5,0.,.5,0.,.5,0.,.5/                                         LATTICE.20
      DATA(Q(I),I=55,81)/ .57735026,0.,0.,.35355339,.61237244,.35355339,LATTICE.21
     2 -.61237244,0.,.35355339,.70710678,0.,.35355339,.70710678,        LATTICE.22
     3 0.,.35355339,.70710678,0.,0.,.20412415,.40824830,0.,.20412415,   LATTICE.23
     4 .40824830,0.,.20412415,.40824830,0./                             LATTICE.24
      DATA(Q(I),I=82,108) / .5,0.,0.,0.,1.,1.,0.,0.,.5,0.,.5,0.,.5,0.,  LATTICE.25
     2  .5,0.,.5,0.,.5,0.,.5,0.,.5,0.,.5/                               LATTICE.26
      DATA(Q(I),I=109,135) / .70710678,0.,0.,.70710678,.5,.70710678,-.5,LATTICE.27
     2 0.,.70710678,0.,.70710678,0.,.70710678,0.,.70710678,0.,.70710678,LATTICE.28
     3 10*0.0/                                                          LATTICE.29
      DATA(Q(I),I=136,162)/.28867513,0.,0.,.70710678,1.2247449,.70710678LATTICE.30
     2,-1.2247449,10*0.0,0.,.81649658,1.63299316,0.,.81649658,1.63299316LATTICE.31
     3 ,0.,.81649658,1.63299316,0./                                     LATTICE.32
      DATA(Q(I),I=163,189)/1., 0.0, 0.0     ,.5,.866025404,.5,-.866025404LATTICE.33
     2,0.,.5,0.,.5,0.,.5,0.,.5,0.,.5,0.,.288675135,0.,.288675135,0.,    LATTICE.34
     3.288675135,0.,.288675135,0.,.288675135/                           LATTICE.35
      DATA(Q(I),I=190,216)/1.0,.866025404,.5,.866025404,1.5,1.7320508C8,LATTICE.36
     2 0.0,0.0,.866025404,0.0,.866025404,0.0,.866025404,0.0,.866025404, LATTICE.37
     3 0.0,.866025404,0.0,-.5,0.0,-.5,0.0,-.5,0.0,-.5,0.0,-.5/          LATTICE.38
C                                                                       LATTICE.39
C                                                                       LATTICE.40
C     READ IN LATTICE SPECIFICATIONS.                                   LATTICE.41
C                                                                       LATTICE.42
      READ 800, TYPE,FACE,ALAT,HEX                                      LATTICE.43
15    IF(HEX.EQ.0.0)  HEX=ALAT                                          LATTICE.44
17    READ 810, LAYER                                                   LATTICE.45
25    JJ=0                                                              LATTICE.46
26    IF(TYPE.EQ.3HFCC)  GO TO 100                                      LATTICE.47
30    IF(TYPE.EQ.3HBCC)  GO TO 110                                      LATTICE.48
32    IF(TYPE.EQ.3HFCP)  GO TO 120                                      LATTICE.49
33    IF(TYPE.EQ.3HGR1)  GO TO 130                                      LATTICE.50
C                                                                       LATTICE.51
35   99 CONTINUE                                                        LATTICE.52
```

```
 35         PRINT 820, TYPE,FACE                                          LATTICE.53
 45         STOP                                                          LATTICE.54
      C                                                                   LATTICE.55
      C     FCC CRYSTAL FACES.                                            LATTICE.56
      C                                                                   LATTICE.57
 47     100 CONTINUE                                                      LATTICE.58
 47         IF(FACE.EQ.100.OR.FACE.EQ.010.OR.FACE.EQ.001)  JJ=1           LATTICE.59
 63         IF(FACE.EQ.110.OR.FACE.EQ.101.OR.FACE.EQ.011)  JJ=2           LATTICE.60
 76         IF(FACE.EQ.111)  JJ=3                                         LATTICE.61
101         GO TO 160                                                     LATTICE.62
      C                                                                   LATTICE.63
      C     BCC CRYSTAL FACES.                                            LATTICE.64
      C                                                                   LATTICE.65
102     110 CONTINUE                                                      LATTICE.66
102         IF(FACE.EQ.100.OR.FACE.EQ.010.OR.FACE.EQ.001)  JJ=4           LATTICE.67
116         IF(FACE.EQ.110.OR.FACE.EQ.101.OR.FACE.EQ.011)  JJ=5           LATTICE.68
131         IF(FACE.EQ.111)  JJ=6                                         LATTICE.69
134         GO TO 160                                                     LATTICE.70
      C                                                                   LATTICE.71
      C     HCP CRYSTAL,  0001-FACE.                                      LATTICE.72
      C                                                                   LATTICE.73
135     120 CONTINUE                                                      LATTICE.74
135         JJ=7                                                          LATTICE.75
136         GO TO 160                                                     LATTICE.76
      C                                                                   LATTICE.77
137     130 CONTINUE                                                      LATTICE.78
137         JJ=8                                                          LATTICE.79
      C                                                                   LATTICE.80
      C                                                                   LATTICE.81
      C      TRANSFER PROPER CRYSTAL GEOMETRY FROM DATA STATEMENT.        LATTICE.82
      C                                                                   LATTICE.83
140     160 CONTINUE                                                      LATTICE.84
140         IF(JJ.EQ.0)  GO TO 99                                         LATTICE.85
152         SPACE=Q(1,JJ)                                                 LATTICE.86
153         ZSP=SPACE*HFX                                                 LATTICE.87
154         BASIS(1)=Q(2,JJ)*ALAT                                         LATTICE.88
156         BASIS(2)=Q(3,JJ)*ALAT                                         LATTICE.89
157         AX(1)=Q(4,JJ)*ALAT                                            LATTICE.90
160         AX(2)=Q(5,JJ)*ALAT                                            LATTICE.91
161         BX(1)=Q(6,JJ)*ALAT                                            LATTICE.92
162         BX(2)=Q(7,JJ)*ALAT                                            LATTICE.93
163         NBASIS=1                                                      LATTICE.94
164         IF(BASIS(1).EQ.0.0.AND.BASIS(2).EQ.0.0)  NBASIS=0             LATTICE.95
173         K=7                                                           LATTICE.96
      C                                                                   LATTICE.97
174         DO 170  J=1,2                                                 LATTICE.98
176         DO 170  I=1,10                                                LATTICE.99
214         K=K+1                                                         LATTICE.100
215         D(I,J)=Q(K,JJ)*ALAT                                           LATTICE.101
216         A(I,J)=AX(J)                                                  LATTICE.102
216         B(I,J)=BX(J)                                                  LATTICE.103
220     170 CONTINUE                                                      LATTICE.104
      C                                                                   LATTICE.105
      C                                                                   LATTICE.106
224         SP=-ZSP                                                       LATTICE.107
225         DO 172  I=1,10                                                LATTICE.108
235         ILYR(I)=1                                                     LATTICE.109
236         SP=SP+ZSP                                                     LATTICE.110
```

```
237         D(I,3)=SP                                                   LATTICE.111
237         KTYP(I)=1                                                   LATTICE.112
240         LSYM(I)=1                                                   LATTICE.113
240     172 CONTINUE                                                    LATTICE.114
      C                                                                 LATTICE.115
      C     ALTER THE LATTICE GEOMETRY LAYER BY LAYER                   LATTICE.116
      C                                                                 LATTICE.117
242         LR=0                                                        LATTICE.118
243     174 CONTINUE                                                    LATTICE.119
243         READ 812,KT,ITP,MM,SHIFT                                    LATTICE.120
257         IF(KT.EQ.0)  GO TO 176                                      LATTICE.121
265         LR=LR+1                                                     LATTICE.122
266         ILYR(LR)=ITP                                                LATTICE.123
267         KTYP(LR)=KT                                                 LATTICE.124
270         A(LR,1)=MM(1)*AX(1) + MM(2)*BX(1)                           LATTICE.125
275         A(LR,2)=MM(1)*AX(2) + MM(2)*BX(2)                           LATTICE.126
301         B(LR,1)=MM(3)*AX(1) + MM(4)*BX(1)                           LATTICE.127
306         B(LR,2)=MM(3)*AX(2) + MM(4)*BX(2)                           LATTICE.128
313         D(LR,1)=D(LR,1)+SHIFT(1)*ALAT                               LATTICE.129
316         D(LR,2)=D(LR,2)+SHIFT(2)*ALAT                               LATTICE.130
320         D(LR,3)=D(LR,3)-SHIFT(3)*ZSP                                LATTICE.131
323         IF(LR.GT.1)  XD(LR-1)=D(LR,3)-D(LR-1,3)                     LATTICE.132
330         GO TO 174                                                   LATTICE.133
      C                                                                 LATTICE.134
331     176 CONTINUE                                                    LATTICE.135
      C                                                                 LATTICE.136
331         LR1=LR+1                                                    LATTICE.137
333         IF(LR1.LT.2)  LR1=2                                         LATTICE.138
336         DO 180  I=LR1,10                                            LATTICE.139
345         D(I,3)=D(I-1,3)+ZSP                                         LATTICE.140
346         XD(I-1)=D(I,3)-D(I-1,3)                                     LATTICE.141
347     180 CONTINUE                                                    LATTICE.142
      C                                                                 LATTICE.143
      C                                                                 LATTICE.144
      C                                                                 LATTICE.145
      C                                                                 LATTICE.146
      C     CALCULATE RECIPROCAL LATTICE NET.                           LATTICE.147
      C                                                                 LATTICE.148
351         DO 190  J=1,LAYER                                           LATTICE.149
361         AREA=A(J,1)*B(J,2)-A(J,2)*B(J,1)                            LATTICE.150
363         XX=2.*PI/AREA                                               LATTICE.151
364         RA(J,1)= B(J,2)*XX                                          LATTICE.152
365         RA(J,2)=-B(J,1)*XX                                          LATTICE.153
366         RB(J,1)=-A(J,2)*XX                                          LATTICE.154
367         RB(J,2)= A(J,1)*XX                                          LATTICE.155
370     190 CONTINUE                                                    LATTICE.156
      C                                                                 LATTICE.157
372         IF(JJ.EQ.1)  PRINT 801                                      LATTICE.158
377         IF(JJ.EQ.2)  PRINT 802                                      LATTICE.159
405          IF(JJ.EQ.3)  PRINT 803                                     LATTICE.160
413          IF(JJ.EQ.4)  PRINT 804                                     LATTICE.161
421          IF(JJ.EQ.5)  PRINT 805                                     LATTICE.162
427          IF(JJ.EQ.6)  PRINT 806                                     LATTICE.163
435          IF(JJ.EQ.7)  PRINT 807                                     LATTICE.164
443         PRINT 830, ALAT,ZSP                                         LATTICE.165
453         PRINT 840, BASIS                                            LATTICE.166
      C                                                                 LATTICE.167
461         DO 200  I=1,LAYER                                           LATTICE.168
```

```
463         PRINT 850,I,KTYP(I),ILYR(I),(C(I,J),J=1,3),A(I,1),A(I,2),B(I,1),  LATTICE.169
           2  B(I,2)                                                          LATTICE.170
522     200 CONTINUE                                                          LATTICE.171
525         PRINT 858                                                         LATTICE.172
530         PRINT 860,(J,RA(J,1),RA(J,2),RB(J,1),RB(J,2), J=1,LAYER )         LATTICE.173
      C                                                                       LATTICE.174
      C                                                                       LATTICE.175
        800 FORMAT( A3,1XI3,3X 2F10.4 )                                       LATTICE.176
        801 FORMAT( //////10X 5H***** *FCC (100)-SURFACE* )                   LATTICE.177
        802 FORMAT( //////10X 5H***** *FCC (110)-SURFACE* )                   LATTICE.178
        803 FORMAT( //////10X 5H***** *FCC (111)-SURFACE* )                   LATTICE.179
        804 FORMAT( //////10X 5H***** *BCC (100)-SURFACE* )                   LATTICE.180
        805 FORMAT( //////10X 5H***** *BCC (110)-SURFACE* )                   LATTICE.181
        806 FORMAT( //////10X 5H***** *BCC (111)-SURFACE* )                   LATTICE.182
        807 FORMAT( //////10X 5H***** *HEX. CLOSE PACKED LATTICE. * )         LATTICE.183
        810 FORMAT( IF )                                                      LATTICE.184
        812 FORMAT( 6I5,3F10.4 )                                              LATTICE.185
        820 FORMAT( // 10X*UNRECOGNIZABLE CRYSTAL TYPE   * A3,1XI3 )          LATTICE.186
        830 FORMAT( // 10X*LATTICE CONSTANT IS * F9.5, * ANGSTROMS,    INTERLALATTICE.187
           2YER SPACING IN BULK IS * F9.5, * ANGSTROMS* / )                   LATTICE.188
        840 FORMAT(/ 10X*BASIS VECTOR IN SURFACE PLANE IS (* F9.5,*,* F9.5,   LATTICE.189
           2 *)* )                                                            LATTICE.190
        850 FORMAT( 10X*(* I2,*) CORE(* I2,*)  SYM(* I2,*)   C=(* 3F10.5,     LATTICE.191
           2  *)    A=(*2F10.5,*)    B=(* 2F10.5, *)* )                      LATTICE.192
        858 FORMAT( //10X*RECIPROCAL LATTICE, SURFACE NET VECTORS* / )        LATTICE.193
        860 FORMAT( 35X*(* I2,*)  RA=(*2F10.5,*)   RB=(* 2F10.5, *)* )        LATTICE.194
      C                                                                       LATTICE.195
557         RETURN                                                            LATTICE.196
560         END                                                               LATTICE.197
```

```
SUBPROGRAM LENGTH
001360

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
99      -  000036    100     -  000050    110     -  000103    120     -  000136
130     -  000140    160     -  000141    174     -  000244    176     -  000332
800     -  000600    801     -  000604    802     -  000612    803     -  000620
804     -  000626    805     -  000634    806     -  000642    807     -  000650
810     -  000657    812     -  000661    820     -  000664    830     -  000673
840     -  000707    850     -  000717    858     -  000733    860     -  000742

BLOCK NAMES AND LENGTHS
BLOCKA  -  000163    BLOCKB  -  004324    BLOCKD  -  000055    BLOCKK  -  012712
BLOCKL  -  000737    BLOCKX  -  000011    CONST   -  000013

VARIABLE ASSIGNMENTS
A       -  000000S01 ALAT    -  000156S01 AREA    -  001356    ARG     -  000000S05
AX      -  000777    B       -  000024S01 BASIS   -  000160S01 BBB     -  000000S04
BX      -  001000    C       -  000050S01 FACE    -  001343    F       -  000002S05
HEX     -  001344    I       -  001351    ILYR    -  004300S02 IM      -  000010S07
IST     -  000010S03 ITP     -  001354    J       -  001350    JJ      -  001345
K       -  001347    KT      -  001353    KTYP    -  012700S04 LAYER   -  000003S03
LR      -  000054S03 LR1     -  001355    LSYM    -  004312S02 MM      -  001006
NBASIS  -  000162S01 PATH1   -  000535S05 PATH2   -  000547S05 PI      -  000000S07
```

```
      SUBROUTINE BEAM                                                   BEAM.2
C                                                                       BEAM.3
      COMMON/BLOCKA/A(10,2),B(10,2),D(10,2),RA(10,2),RB(10,2),ALAT,SPACEA.2
     2,BASIS(2),NBASIS                                                  A.3
      COMMON/BLOCKB/T(2,80,7),ILYR(10),LSYM(10)                         B.2
      COMPLEX T                                                         B.3
      COMMON/BLOCKC/NLOW,NNUM,NDELTA,NRG,NRGY,LAYER,NUMMAX,NDIF,IST(6,4)D.2
     2  ,LR                                                             D.3
      COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE                                               E.3
      COMMON/BLOCKK/RRR(36,2),TAU(36,36,2),TT(10,12),KTYP(10)           K.2
      COMPLEX  RRR,TAU,TT                                               K.3
      COMMON/BLOCKF/ITHT,IPHI,THTT,DIST,CSTH(12),VXX(2,12),KNUM,JM(12), F.2
     2  JN(12),IC(12),KG(12)                                            F.3
      DIMENSION  NN(2,6)                                                BEAM.10
C                                                                       BEAM.11
C     READ IN BEAM COORDINATES *****                                    BEAM.12
C                                                                       BEAM.13
      READ 800, ITHT,IPHI                                               BEAM.14
C                                                                       BEAM.15
C     READ IN ENERGY RANGE AND DAMPING LENGTH AT 30 E.V. *****          BEAM.16
C                                                                       BEAM.17
 11   READ 803, NLOW,NNUM,NDELTA                                        BEAM.18
C                                                                       BEAM.19
C                                                                       BEAM.20
C                                                                       BEAM.21
C     KNUM IS TOTAL NUMBER OF BEAMS TO BE COMPUTED.                     BEAM.22
C                                                                       BEAM.23
 23   KNUM=0                                                            BEAM.24
 24 100 CONTINUE                                                        BEAM.25
 24   READ 808, ITP,NN                                                  BEAM.26
 34   IF(ITP.EQ.0)  GO TO 150                                           BEAM.27
 35   DO 110  I=1,LAYER                                                 BEAM.28
 44   K=I                                                               BEAM.29
 45   KT=KTYP(I)                                                        BEAM.30
 46   IF(ILYR(K).EQ.ITP)  GO TO 120                                     BEAM.31
 51 110 CONTINUE                                                        BEAM.32
 53   PRINT 812, ITP                                                    BEAM.33
 60   READ 808, ITP,NN                                                  BEAM.34
 70   IF(ITP.NE.0)  STOP                                                BEAM.35
 73   GO TO 150                                                         BEAM.36
C                                                                       BEAM.37
 74 120 CONTINUE                                                        BEAM.38
C                                                                       BEAM.39
C                                                                       BEAM.40
 74   DO 140  I=1,6                                                     BEAM.41
 76   IF(I.EQ.1.AND.ITP.EQ.1)  GO TO 130                                BEAM.42
104   IF(NN(1,I).EQ.0.AND.NN(2,I).EQ.0)  GO TO 140                      BEAM.43
114 130 CONTINUE                                                        BEAM.44
120   M=NN(1,I)                                                         BEAM.45
120   N=NN(2,I)                                                         BEAM.46
121   KNUM=KNUM+1                                                       BEAM.47
123   IF(KNUM.GT.12)  GO TO 140                                         BEAM.48
133   VXX(1,KNUM)=M*RA(K,1) + N*RB(K,1)                                 BEAM.49
137   VXX(2,KNUM)=M*RA(K,2) + N*RB(K,2)                                 BEAM.50
143   JM(KNUM)=M                                                        BEAM.51
144   JN(KNUM)=N                                                        BEAM.52
145   IC(KNUM)=ITP                                                      BEAM.53
```

```
146          KC(KNUM)=KT                                                    BEAM.54
150      140 CONTINUE                                                       BEAM.55
       C                                                                    BEAM.56
152          IF(KNUM.GT.12)  KNUM=12                                        BEAM.57
156          GO TO 100                                                      BEAM.58
157      150 CONTINUE                                                       BEAM.59
       C                                                                    BEAM.60
       C                                                                    BEAM.61
157          NHIGH=NLOW+(NNUM-1)*NDELTA                                     BEAM.62
164          PRINT 804, ITHT,IPHI                                           BEAM.63
173          PRINT 806, NLOW,NHIGH                                          BEAM.64
203          PRINT 810,LMAX                                                 BEAM.65
       C                                                                    BEAM.66
       C                                                                    BEAM.67
         800 FORMAT( 2I5 )                                                  BEAM.68
         802 FORMAT( 3I5 )                                                  BEAM.69
         804 FORMAT( // 10X*INCIDENT BEAM ANGLES ARE THETA=* I4,*  PHI=* I4,  BEAM.70
            2  / 42X 13H*************  )                                    BEAM.71
         806 FORMAT( // 10X *INCIDENT ELECTRON ENERGY FROM * I3, * TO * I3,  BEAM.72
            2  * E.V. * )                                                   BEAM.73
         808 FORMAT( I5,6(1X 2I2) )                                         BEAM.74
         810 FORMAT( / 10X *MAXIMUM L-VALUE IS *  I2 )                      BEAM.75
         812 FORMAT( /// 10X*ION CORE TYPE * I2,* ENCOUNTERED IN SUBROUTINE BEABEAM.76
            2M, BUT PHASE SHIFTS NOT SUPPLIED.* / )                         BEAM.77
       C                                                                    BEAM.78
211          RETURN                                                         BEAM.79
212          END                                                            BEAM.80
```

```
SUBPROGRAM LENGTH
000342

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
100     -  000025    110     -  000052    120     -  000075    130     -  000115
140     -  000151    150     -  000160    800     -  000221    802     -  000223
804     -  000225    806     -  000237    808     -  000250    810     -  000253
812     -  000260

BLOCK NAMES AND LENGTHS
BLOCKA -  000163    BLOCKB -  004324    BLOCKD -  000055    BLOCKE -  000012
BLOCKK -  012712    BLOCKF -  000131

VARIABLE ASSIGNMENTS
A       -  000000S01 B       -  000024S01 BASIS   -  000160S01 BBB     -  000000S05
CSTH    -  000004S06 C       -  000050S01 I       -  000334    ILYR    -  004300S02
IPHI    -  000001S06 IQ      -  000101S06 IST     -  000010S03 ITHT    -  000000S06
ITP     -  000333    JM      -  000051S06 JN      -  000065S06 K       -  000335
KNUM    -  000050S06 KO      -  000110S06 KT      -  000336    KTYP    -  012700S05
LAYER   -  000005S03 LMAX    -  000000S04 LSYM    -  004312S02 M       -  000337
N       -  000340    NDELTA  -  000002S03 NHIGH   -  000341    NLOW    -  000000S03
NN      -  000317    NNUM    -  000001S03 RA      -  000106S01 RB      -  000132S01
T       -  000000S02 TAU     -  000220S05 TT      -  012320S05 VXX     -  000020S06

START OF CONSTANTS
000215
```

```
      SUBROUTINE DEBYE                                                  DEBYE.2
C                                                                       DEBYE.3
      COMMON/BLOCKB/T(2,80,7),ILYR(10),LSYM(10)                         B.2
      COMPLEX T                                                         B.3
      COMMON/BLOCKC/NLOW,NNUM,NDELTA,NRG,NRGY,LAYER,NUMMAX,NDIF,IST(6,6)D.2
     2  ,LR                                                             D.3
      COMMON/BLKDEB/W(10),ATWT(10),TEMP,SURF,BULK,CIN(10),DCLT(10,12)   DEB.2
      DIMENSION  X(200),Y(200)                                          DEBYE.7
C                                                                       DEBYE.8
      CONST=72.04073                                                    DEBYE.9
  2   NUMBER=200                                                        DEBYE.10
C                                                                       DEBYE.11
  4   READ 800, TEMP                                                    DEBYE.12
 11   READ 802, SURF,BULK                                               DEBYE.13
C                                                                       DEBYE.14
 21   ENTRY DEBYE1                                                      DEBYE.15
C                                                                       DEBYE.16
 26   IF(TEMP.LT.1.0)  GO TO 150                                        DEBYE.17
 32   SRF=SURF/TEMP                                                     DEBYE.18
 33   BLK=BULK/TEMP                                                     DEBYE.19
 35   LMN=0                                                             DEBYE.20
 35   DEL=SRF/NUMBER                                                    DEBYE.21
 40 100 CONTINUE                                                        DEBYE.22
C                                                                       DEBYE.23
 40   X(1)=0.0                                                          DEBYE.24
 41   Y(1)=1.0                                                          DEBYE.25
 42   LMN=LMN+1                                                         DEBYE.26
 44   DO 110  I=2,NUMBER                                                DEBYE.27
 46   X(I)=X(I-1)+DEL                                                   DEBYE.28
 51   Y(I)=X(I)/(EXP(X(I))-1.0 )                                        DEBYE.29
 56 110 CONTINUE                                                        DEBYE.30
C                                                                       DEBYE.31
 60   CALL CUBINT(X,Y,NUMBER,ANS)                                       DEBYE.32
C                                                                       DEBYE.33
 63   IF(LMN.GE.2)  GO TO 120                                           DEBYE.34
 66   FSURF=CONST*(0.25+ANS/SRF)/SURF                                   DEBYE.35
 72   DEL=BLK/NUMBER                                                    DEBYE.36
 75   GO TO 100                                                         DEBYE.37
C                                                                       DEBYE.38
 75 120 CONTINUE                                                        DEBYE.39
 75   FBULK=CONST*(0.25+ANS/BLK)/BULK                                   DEBYE.40
C                                                                       DEBYE.41
C     CALCULATE DEBYE-WALLER CORRECTION FOR EACH LAYER  (W(I)) *****    DEBYE.42
C                                                                       DEBYE.43
101   W(1)=FSURF/ATWT(1)                                                DEBYE.44
C                                                                       DEBYE.45
104   DO 130  I=2,LAYER                                                 DEBYE.46
C                                                                       DEBYE.47
C     FOR SINGLE ATOMIC TYPE... NN=1.                                   DEBYE.48
C                                                                       DEBYE.49
113   NN=1                                                              DEBYE.50
114   W(I)=FBULK/ATWT(NN)                                               DEBYE.51
114 130 CONTINUE                                                        DEBYE.52
116 140 CONTINUE                                                        DEBYE.53
C                                                                       DEBYE.54
116   PRINT 804, TEMP,SURF,BULK                                         DEBYE.55
130   RETURN                                                            DEBYE.56
131 150 CONTINUE                                                        DEBYE.57
```

```
131   DO 160  I=1,10                                                    DEBYE.58
136   W(I)=0.0                                                          DEBYE.59
137 160 CONTINUE                                                        DEBYE.60
140   GO TO 140                                                         DEBYE.61
C                                                                       DEBYE.62
C                                                                       DEBYE.63
    800 FORMAT( F10.5 )                                                 DEBYE.64
    802 FORMAT( 2F10.5 )                                                DEBYE.65
    804 FORMAT( // 10X *LATTICE TEMPERATURE=* F5.0, /10X*DEBYE TEMPERATURDEBYE.66
       2S =*F5.0, * AND* F5.0 )                                         DEBYE.67
C                                                                       DEBYE.68
140   END                                                               DEBYE.69
```

```
                SUBROUTINE CUBINT( X, Y, N, ANS)                                CUBINT.2
                DIMENSION X(500), Y(500)                                        CUBINT.3
                IGC = 1                                                         CUBINT.4
     11         CC1 = (Y(2)-Y(1))/(X(2)-X(1))                                   CUBINT.5
     15         C12 = (Y(3)-Y(2))/(X(3)-X(2))                                   CUBINT.6
     21         DC2 = (C12-CC1)/(X(3)-X(1))                                     CUBINT.7
     24         N1 = N-3                                                        CUBINT.8
     26         DO1 I=1,N1                                                      CUBINT.9
     30         C23 = (Y(I+3)-Y(I+2))/(X(I+3)-X(I+2))                           CUBINT.10
     36         C13 = (C23-C12)/(X(I+3)-X(I+1))                                 CUBINT.11
     42         DC3 = (C13-DC2)/(X(I+3)-X(I))                                   CUBINT.12
     45         A = Y(I)-X(I)*(CC1-X(I+1)*(DD2-X(I+2)*DC3))                     CUBINT.13
     53         B = CC1-DC2*(X(I)+X(I+1))+DD2*(X(I)*X(I+1)+X(I+2)*X(I)+X(I+2)*X(I+CUBINT.14
               1))                                                              CUBINT.15
     63         C = DC2-DC3*(X(I)+X(I+1)+X(I+2))                                CUBINT.16
     70         D = DC3                                                         CUBINT.17
     70         IF( N .EQ. 4) GO TO 6                                           CUBINT.18
     72         GO TO (2, 3, 4), IGC                                            CUBINT.19
    101    2    CALL GRATE(A, B, C, D, 0.0, 0.0, X(1), X(3), ANS)               CUBINT.20
    120         IGC = 2                                                         CUBINT.21
    120         IF( I .EQ. N-4) IGC = 3                                         CUBINT.22
    124         GO TO 5                                                         CUBINT.23
    125    3    CALL GRATE(A, B, C, D, 0.0, 0.0, X(I+1), X(I+3), SLP)           CUBINT.24
    145         ANS = ANS+SLP                                                   CUBINT.25
    146         IF( I .EQ. N-4) IGC = 3                                         CUBINT.26
    152         GO TO 5                                                         CUBINT.27
    153    4    CALL GRATE(A, B, C, D, 0.0, 0.0, X(I+1), X(I+3), SLP)           CUBINT.28
    173         ANS = ANS+SLP                                                   CUBINT.29
    174    5    CONTINUE                                                        CUBINT.30
    174         DD1 = C12                                                       CUBINT.31
    175         C12 = C23                                                       CUBINT.32
    177         DC2 = C13                                                       CUBINT.33
    200    1    CONTINUE                                                        CUBINT.34
    203         RETURN                                                          CUBINT.35
    203    6    CALL GRATE( A, B, C, D, 0.0, 0.0, X(1), X(4), ANS)              CUBINT.36
    222         RETURN                                                          CUBINT.37
    222         END                                                             CUBINT.38
```

```
SUBPROGRAM LENGTH
000314

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
Z       -  000102    3       -  000126    4       -  000154    5       -  000175
6       -  000204

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
A       -  000307    B       -  000310    C       -  000311    D       -  000312
DD1     -  000277    DD2     -  000301    DD3     -  000306    C12     -  000300
C13     -  000305    C23     -  000304    I       -  000309    IGC     -  000276
N1      -  000302    SLP     -  000313

START OF CONSTANTS


000229

START OF TEMPORARIES
000236

START OF INDIRECTS
000266

UNUSED COMPILER SPACE
124300
```

```
                SUBROUTINE GRATE( A, B, C, D, E, F, X1, X2, ANS)                CUBINT.39
                DIMENSION G(2), Z(2)                                            CUBINT.40
                Z(1) = X1                                                       CUBINT.41
     14         Z(2) = X2                                                       CUBINT.42
     15         DO1 I=1,2                                                       CUBINT.43
     21         X1 = A*Z(I)                                                     CUBINT.44
     23         A2 = B*Z(I)**2)/2.0                                             CUBINT.45
     25         A3 = C*Z(I)**3)/3.0                                             CUBINT.46
     27         A4 = D*Z(I)**4)/4.0                                             CUBINT.47
     32         A5 = E*Z(I)**5)/5.0                                             CUBINT.48
     34         A6 = F*Z(I)**6)/6.0                                             CUBINT.49
     37    1    G(I) = A1+A2+A3+A4+A5+A6                                        CUBINT.50
     52         ANS = G(2)-G(1)                                                 CUBINT.51
     55         RETURN                                                          CUBINT.52
     55         END                                                             CUBINT.53
```

```
      SUBROUTINE TABLE                                                  TABLE.2
C                                                                       TABLE.3
C     A TABLE OF SPHERICAL HARMONICS (STORE) IS STORED IN THE LARGE ARRATABLE.4
C     FOR EACH LATTICE POINT CONSISTENT WITH THE GEOMETRY .             TABLE.5
C     THE PARAMETER (LAYER) DETERMINES ELECTRON PATH LENGTH CUTOFF *****TABLE.6
C                                                                       TABLE.7
      COMMON/BLOCKA/A(10,2),B(10,2),D(10,3),RA(10,2),RB(10,2),ALAT,SPACEA.2
     2,BASIS(2),NBASIS                                                  A.3
      COMMON/BLOCKC/V1(3),V2(3,12),V3(3,12),VX,VY,VZ,SQK(12),CCK(12),   C.2
     2  YIN(49),YOUT(49,12),XKZERO,AREA2                                C.3
      COMPLEX YIN,YOUT                                                  C.4
      COMMON/BLOCKD/NLOW,NNUM,NDELTA,NPG,NRGY,LAYER,NUMMAX,NDIF,IST(6,6)D.2
     2  ,LR                                                             D.3
      COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE                                               E.3
      COMMON/BLOCKH/KX(2401),C(7532),LL(7532),YINT(169)                 H.2
      COMPLEX YINT                                                      H.3
      COMMON/BLOCKJ/MAXX(51),RMAG( 9000),XMAG( 9000),MAXR               J.2
      COMMON/BLOCKX/TIME,ITEM,MULT,ISCALE,IMAX,ITYPE,LSUM,ILOOP,IPWR    X.2
      INTEGER TIME                                                      X.3
      LARGE C.QM(1296),RVEC(30000),C.G(49000)                           LARGE.2
C                                                                       DIM.2
C     DIMENSIONS OF KX,C,AND LL.                                        DIM.3
      NM3=2401 $ NM1=7532 $ NM2=7532                                    DIM.4
C     DIMENSIONS OF RMAG AND XMAG                                       DIM.5
 4    NM4=9000 $ NM5=9000                                               DIM.6
C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.              DIM.7
 5    NM6=30000 $ NM7=49000                                             DIM.8
C                                                                       DIM.9
C                                                                       TABLE.17
C                                                                       TABLE.18
10    LINT=LMAX2-1                                                      TABLE.19
11    MMAX=200                                                          TABLE.20
12    RMAX=2.*D(LAYER,3)+ 0.05                                          TABLE.21
16    IF(MAXR.NE.0)  RMAX=MAXR                                          TABLE.22
20    LYR=LAYER                                                         TABLE.23
21    IF(MULT.LE.1)  LYR=1                                              TABLE.24
25    PRINT 800, RMAX                                                   TABLE.25
C                                                                       TABLE.26
33    LCC=0                                                             TABLE.27
34    INDEX=0                                                           TABLE.28
34    LCCC=0                                                            TABLE.29
C                                                                       TABLE.30
C     INDEX=(NU1-1)*LAYER+NU2                                           TABLE.31
C     SUMMATION IS FASTEST OVER LAYER OF INITIAL SCATTERING *****       TABLE.32
C                                                                       TABLE.33
35    DO 200  NU2=1,LYR                                                 TABLE.34
37    DO 200  NU1=1,LYR                                                 TABLE.35
40    IF(NU1.EQ.NU2.AND.NU1.NE.1)  GO TO 200                            TABLE.36
47    INDEX=INDEX+1                                                     TABLE.37
  920 FORMAT( 5X*LAYERS *I3,*,*I3,*   INDEX=* I4 )                      TABLE.38
C                                                                       TABLE.39
52    A1=A(NU1,1)                                                       TABLE.40
54    A2=A(NU1,2)                                                       TABLE.41
56    B1=B(NU1,1)                                                       TABLE.42
57    B2=B(NU1,2)                                                       TABLE.43
61    RR1=D(NU1,1)-D(NU2,1)                                             TABLE.44
63    RR2=D(NU1,2)-D(NU2,2)                                             TABLE.45
```

```
 65         R3 =D(NU1,3)-D(NU2,3)                                          TABLE.46
      C                                                                    TABLE.47
 67         RSQ3=R3**2                                                     TABLE.48
 70         RX=RMAX-D(NU1,3)-D(NU2,3)                                      TABLE.49
      C                                                                    TABLE.50
      C     LSUM=2  CAUSES SUMMATION OVER CYLINDRICAL VOLUME.              TABLE.51
      C     LSUM=1  CAUSES SUMMATION OVER HEMISPHERICAL VOLUME.            TABLE.52
      C                                                                    TABLE.53
 72         IF(LSUM.EQ.2)  RX=RMAX                                         TABLE.54
      C                                                                    TABLE.55
 75         RX2=RX**2                                                      TABLE.56
      C                                                                    TABLE.57
      C     RX IS THE MAXIMUM PATH LENGTH FOR SUMMATION IN LAYER (NU2) *****  TABLE.58
      C     (NU2)-LAYER SUMMATION FOLLOWS *****                            TABLE.59
      C                                                                    TABLE.60
 77         DO 180  M1=1,MMAX                                              TABLE.61
100         M=M1-1                                                         TABLE.62
101         ICOM=0                                                         TABLE.63
      C                                                                    TABLE.64
102     100 CONTINUE                                                       TABLE.65
      C                                                                    TABLE.66
      C     SUM OVER (N.GE.ZERO) *****                                     TABLE.67
      C                                                                    TABLE.68
102         RBS=0.0                                                        TABLE.69
103         DO 120  N1=1,MMAX                                              TABLE.70
105         N=N1-1                                                         TABLE.71
107         R1=M*A1+N*B1+RP1                                               TABLE.72
114         R2=M*A2+N*B2+RP2                                               TABLE.73
121         RSQ=R1**2+R2**2+RSQ3                                           TABLE.74
124         IPASS=0                                                        TABLE.75
124         IF(NBASIS.EQ.0)  GO TO 105                                     TABLE.76
126         R4=R1+BASIS(1)                                                 TABLE.77
127         R5=R2+BASIS(2)                                                 TABLE.78
131         RBS=R4**2+R5**2+RSQ3                                           TABLE.79
135         IPASS=1                                                        TABLE.80
136     105 CONTINUE                                                       TABLE.81
      C                                                                    TABLE.82
136         IF(N.EQ.0)  RMM=RSQ                                            TABLE.83
141         IF(N.EQ.0.AND.NBASIS.NE.0)  GO TO 102                          TABLE.84
147         IF(RSQ.LT.0.01)  GO TO 120                                     TABLE.85
151     102 CONTINUE                                                       TABLE.86
151         IF(RSQ.LE.RX2)  GO TO 110                                      TABLE.87
154         IF(NBASIS.EQ.0)  GO TO 130                                     TABLE.88
155         IF(RBS.LE.RX2)  IPASS=2                                        TABLE.89
160         IF(IPASS.EQ.2)  GO TO 110                                      TABLE.90
162         IF(RSQ.LE.RMM) GO TO 120                                       TABLE.91
164         GO TO 130                                                      TABLE.92
      C                                                                    TABLE.93
165     110 CONTINUE                                                       TABLE.94
165         IF(IPASS.EQ.1.AND.RSQ.LT.0.01)  GO TO 119                      TABLE.95
176         IF(IPASS.EQ.2.AND.RBS.LT.0.01)  GO TO 120                      TABLE.96
206         IF(IPASS.EQ.2.AND.RBS.GT.RX2)  GO TO 120                       TABLE.97
215         LCC=LCC+1                                                      TABLE.98
216         RMAG(LCC)=SQRT(RSQ)                                            TABLE.99
222         IF(IPASS.EQ.2)  RMAG(LCC)=SQRT(RBS)                            TABLE.100
230         XMAG(LCC)=V1(1)*R1+V1(2)*R2+V1(3)*R3                           TABLE.101
236         IF(IPASS.EQ.2)  XMAG(LCC)=V1(1)*R4+V1(2)*R5+V1(3)*R3           TABLE.102
246         LCCC=LCCC+1                                                    TABLE.103
```

```
250         RVEC(LCCC)=R1                                                   TABLE.104
252         IF(IPASS.EQ.2)  RVEC(LOCO)=R4                                   TABLE.105
256         LCCC=LCCO+1                                                     TABLE.106
257         RVEC(LCCO)=R2                                                   TABLE.107
262         IF(IPASS.EQ.2)  RVEC(LOCO)=R5                                   TABLE.108
266         LCCC=LCCO+1                                                     TABLE.109
267         RVEC(LCCO)=R3                                                   TABLE.110
271         LCL=LCCC-2                                                      TABLE.111
        910 FORMAT( 10X*(* 2I4,*)   X=* F9.4,*   Y=* F9.4,*   Z=* F9.4 )    TABLE.112
273         ICOM=1                                                          TABLE.113
274     119 CONTINUE                                                        TABLE.114
274         IPASS=IPASS+1                                                   TABLE.115
276         IF(IPASS.EQ.2)  GO TO 110                                       TABLE.116
277     120 CONTINUE                                                        TABLE.117
      C                                                                     TABLE.118
302     130 CONTINUE                                                        TABLE.119
      C                                                                     TABLE.120
      C     SUM OVER (N.LT.ZERO) *****                                      TABLE.121
      C                                                                     TABLE.122
302         DO 150  N1=1,MMAX                                               TABLE.123
304         N=-N1                                                           TABLE.124
305         R1=M*A1+N*B1+RR1                                                TABLE.125
312         R2=M*A2+N*B2+RR2                                                TABLE.126
317         RSQ=R1**2+R2**2+RSQ3                                            TABLE.127
322         IPASS=0                                                         TABLE.128
323         IF(NBASIS.EQ.0)  GO TO 135                                      TABLE.129
325         R4=R1+BASIS(1)                                                  TABLE.130
326         R5=R2+BASIS(2)                                                  TABLE.131
330         RBS=R4**2+R5**2+RSQ3                                            TABLE.132
334         IPASS=1                                                         TABLE.133
335         GO TO 136                                                       TABLE.134
336     135 CONTINUE                                                        TABLE.135
336         IF(RSQ.LT.0.01)  GO TO 150                                      TABLE.136
341     136 CONTINUE                                                        TABLE.137
341         IF(RSQ.LE.RX2)  GO TO 140                                       TABLE.138
344         IF(NBASIS.EQ.0)  GO TO 160                                      TABLE.139
345         IF(RBS.LE.RX2)  IPASS=2                                         TABLE.140
350         IF(IPASS.EQ.2)  GO TO 140                                       TABLE.141
352         IF(RSQ.LE.RMM)  GO TO 150                                       TABLE.142
354         GO TO 160                                                       TABLE.143
      C                                                                     TABLE.144
355     140 CONTINUE                                                        TABLE.145
355         IF(IPASS.EQ.2.AND.RBS.GT.RX2)  GO TO 150                        TABLE.146
366         LOC=LOC+1                                                       TABLE.147
367         RMAG(LOC)=SQRT(RSQ)                                             TABLE.148
373         XMAG(LOC)=V1(1)*R1+V1(2)*R2+V1(3)*R3                            TABLE.149
401         IF(IPASS.EQ.2)  RMAG(LOC)=SQRT(RBS)                             TABLE.150
407         IF(IPASS.EQ.2)  XMAG(LOC)=V1(1)*R4+V1(2)*R5+V1(3)*R3            TABLE.151
417         LCCC=LCCO+1                                                     TABLE.152
421         RVEC(LCCO)=R1                                                   TABLE.153
423         IF(IPASS.EQ.2)  RVEC(LOCO)=R4                                   TABLE.154
427         LCCC=LCCO+1                                                     TABLE.155
430         RVEC(LCCO)=R2                                                   TABLE.156
433         IF(IPASS.EQ.2)  RVEC(LOCO)=R5                                   TABLE.157
437         LCCC=LCCO+1                                                     TABLE.158
440         RVEC(LCCO)=R3                                                   TABLE.159
442         LCL=LCCC-2                                                      TABLE.160
443         ICOM=1                                                          TABLE.161
```

```
445   149 CONTINUE                                                        TABLE.162
445       IPASS=IPASS+1                                                   TABLE.163
447       IF(IPASS.EQ.2)  GO TO 140                                       TABLE.164
450   150 CONTINUE                                                        TABLE.165
    C                                                                     TABLE.166
453   160 CONTINUE                                                        TABLE.167
    C                                                                     TABLE.168
452       IF(M.LE.0)  GO TO 170                                           TABLE.169
    C                                                                     TABLE.170
    C     INCLUDE NEGATIVE M-VALUES IN THE SUMMATION *****                TABLE.171
    C                                                                     TABLE.172
455       M=-M                                                            TABLE.173
455       GO TO 100                                                       TABLE.174
    C                                                                     TABLE.175
456   170 CONTINUE                                                        TABLE.176
456       IF(ICOM.EQ.0)  GO TO 190                                        TABLE.177
457   180 CONTINUE                                                        TABLE.178
    C                                                                     TABLE.179
    C     SUMMATION OVER LAYER (NU2) IS COMPLETED *****                   TABLE.180
    C                                                                     TABLE.181
462   190 CONTINUE                                                        TABLE.182
462       MAXX(INDEX)=LCC                                                 TABLE.183
464       NUX=NU1                                                         TABLE.184
465       NUY=NU2                                                         TABLE.185
467       PRINT 810,NUX,NUY,INDEX,MAXX(INDEX)                             TABLE.186
504   200 CONTINUE                                                        TABLE.187
511       LC=3*LCC                                                        TABLE.188
512       NL=LAYER*(LAYER+1)*LMAX4                                        TABLE.189
516       PRINT 820, NUMMAX,NM1,NUMMAX,NM2,LMAX4,NM3,LCC,NM4,LCC,NM5,LC,NM6,TABLE.190
         2 NL,NM7                                                         TABLE.191
556       PRINT 830                                                       TABLE.192
      500 FORMAT( 2CX*X=* F9.4,*   Y=*F9.4,*   Z=* F9.4 )                 TABLE.193
562       IF(NUMMAX.GT.NM1.OR.NUMMAX.GT.NM2)  ITEM=5                      TABLE.194
573       IF(LCC.GT.NM4.OR.LCC.GT.NM5)  ITEM=5                            TABLE.195
604       IF(LMAX4.GT.NM3.OR.LC.GT.NM6)  ITEM=5                           TABLE.196
    C                                                                     TABLE.197
    C                                                                     TABLE.198
      800 FORMAT( // 10X*SUBROUTINE TABLE SETS PROPAGATOR CUTOFF LENGTH AT RTABLE.199
         2MAX=* F8.3 )                                                    TABLE.200
      810 FORMAT( 10X*FINAL LAYERS OF SUMMATION ARE (* I2,*,*I2,*)   MAXX(* TABLE.201
         2 I3,*)=* I5,* = TOTAL NUMBER OF ATOMIC SITES WITHIN CUTOFF. * )  TABLE.202
      820 FORMAT( / 10X*ARRAY DIMENSIONS REQUIRED (AND SUPPLIED) BY PROGRAM*TABLE.203
         2 /27X*          C-* I6, *  (* I5, *)*                           TABLE.204
         3 /27X*         LL-* I6, *  (* I5, *)*                           TABLE.205
         4 /27X*         KX-* I6, *  (* I6, *)*                           TABLE.206
         5 /27X*       RMAG-* I6, *  (* I6, *)*                           TABLE.207
         6 /27X*       XMAG-* I6, *  (* I6, *)*                           TABLE.208
         7 /27X*       RVEC-* I6, *  (* I6, *)*                           TABLE.209
         8 /27X*G(COMPLEX)-* I6, *  (* I6, *)*   )                        TABLE.210
      830 FORMAT( ///// )                                                 TABLE.211
    C                                                                     TABLE.212
616       RETURN                                                          TABLE.213
617       END                                                             TABLE.214
```

SUBPROGRAM LENGTH
001072

```
      PROGRAM LCCP                                                      LCCP.3
C                                                                       LCCP.4
      COMMON/BLCCKA/A(10,2),B(10,2),D(10,3),RA(10,2),RB(10,2),ALAT,SPACEA.2
     2,BASIS(2),NBASIS                                                  A.3
      COMMON/BLCCKB/T(2,80,7),ILYR(10),LSYM(10)                         B.2
      COMPLEX T                                                         B.3
      COMMON/BLCCKC/V1(3),V2(3,12),V3(3,12),VX,VY,VZ,SGK(12),CCK(12),   C.2
     2  YIN(49),YOUT(49,12),XKZERO,AREA2                                C.3
      COMPLEX YIN,YOUT                                                  C.4
      COMMON/BLCCKD/NLCW,NNUM,NDELTA,NRG,NRGY,LAYER,NUMMAX,NDIF,IST(6,6)D.2
     2  ,LR                                                             D.3
      COMMON/BLCCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE                                               E.3
      COMMON/BLCCKF/ITHT,IPHI,THTT,DIST,CSTH(12),VXX(2,12),KNUM,JM(12), F.2
     2  JN(12),IC(12),KC(12)                                            F.3
      COMMON/BLCCKG/ESIG,SIG2,XK,DAMP,ZNORM,PCTL,GAMA,SIGMA,LCWEN,NNRG  G.2
     2  ,LAMBDA                                                         G.3
      REAL LAMBDA                                                       G.4
      INTEGER SIGMA                                                     G.5
      COMPLEX ZNORM                                                     G.6
      COMMON/BLCCKH/KX(2401),C(7532),LL(7532),YINT(169)                 H.2
      COMPLEX YINT                                                      H.3
      COMMON/BLCCKJ/MAXX(51),RMAG( 9000),XMAG( 9000),MAXR               J.2
      COMMON/BLCCKK/BBB(36,2),TAU(36,36,2),TT(10,12),KTYP(10)           K.2
      COMPLEX  BBB,TAU,TT                                               K.3
      COMMON/BLCCKL/ARG,H(169),XD(9),PATH1(10),PATH2(10,12)             L.2
      COMPLEX  ARG,H                                                    L.3
      COMMON/BLCCKM/RATIO,XCON,KSTAR                                    M.2
      COMPLEX KSTAR,XCON                                                M.3
      COMMON/BLCCKN/E(51),EN(51),S(12,51)                               N.2
      COMMON/BLCCKO/PX,PY,PZ                                            Q.2
      COMMON/BLCCKX/TIME,ITEM,MULT,ISCALE,IMAX,ITYPE,LSUM,ILCCP,IPWR    X.2
      INTEGER TIME                                                      X.3
      COMMON/BLKDEB/W(10),ATWT(10),TEMP,SURF,BULK,CIN(10),DCLT(10,12)   DEB.2
      COMMON/CONST/PI,PI4,PI42,CON1,CON2,CON3,CON4,CONSTT,IM,CNST       CONST.2
      COMPLEX IM                                                        CONST.3
      DIMENSION TIM(20),JB(12)                                          LCCP.23
      LARGE C.QM(1296),RVEC(30000),C.G(49000)                           LARGE.2
C                                                                       DIM.2
C     DIMENSIONS OF KX,C,AND LL.                                        DIM.3
      NM3=2401 $ NM1=7532 $ NM2=7532                                    DIM.4
C     DIMENSIONS OF RMAG AND XMAG                                       DIM.5
 5    NM4=9000 $ NM5=9000                                               DIM.6
C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.              DIM.7
 6    NM6=30000 $ NM7=49000                                             DIM.8
C                                                                       DIM.9
C                                                                       LCCP.26
13    IONE=LAYER*(LAYER-1)*LMAX4                                        LCCP.27
16    ITWO=IONE+LAYER*LMAX4                                             LCCP.28
17    ITHREE=ITWO+LAYER*LMAX4                                           LCCP.29
20    LCX=2*LMAX                                                        LOOP.30
C                                                                       LOOP.31
C     SWEEP OVER ENERGY RANGE AND CALCULATE INTENSITIES *****           LCCP.32
C                                                                       LCCP.33
22    ILCCP=1                                                           LCCP.34
23    DO 150  J=1,NNUM                                                  LOOP.35
25     IF(TIME.NE.0)  CALL SECOND(TIME3)                                LCCP.36
27    NRG=NRG+NDELTA                                                    LCCP.37
```

```
 31         CALL ENERGY(NRG,NRGY)                                          LCCP.38
 33         IF(NRGY.EQ.0)  GO TO 130                                       LCCP.39
 34         EN(J)=ESIG                                                     LCCP.40
 36         RATIO=XK*XKZERO                                                LCCP.41
 40         KSTAR=CMPLX(XK,DAMP)                                           LCCP.42
 42         XCON=-IM*CON2*PI42*KSTAR                                       LCCP.43
 52         V1(1)=XK*VX                                                    LCCP.44
 53         V1(2)=XK*VY                                                    LCCP.45
 55         V1(3)=XK*VZ                                                    LCCP.46
 57         FTR=AREA2/V1(3)**2                                             LCCP.47
      C                                                                    LCCP.48
 61         PRINT 815                                                      LCCP.49
 64         PRINT 830                                                      LCCP.50
 70         DP=1./DAMP                                                     LCCP.51
 72         PRINT 820 , NRG,NRGY                                           LCCP.52
102         PRINT 822, ESIG,SIG2                                           LCCP.53
112         PRINT 824, XK,DAMP                                             LCCP.54
122         PRINT 826, ZNORM                                               LCCP.55
130         PRINT 828, DP                                                  LCCP.56
      C                                                                    LCCP.57
      C     DAMP IS THE DAMPING FACTOR ... MULTIPLY BY PATH LENGTH *****   LCCP.58
      C     IF  V2  IS A PROPAGATING BEAM, CALCULATE ITS Z-COMPONENT *****LCCP.59
      C                                                                    LCCP.60
136         DO 115  JBEAM=1,KNUM                                           LCCP.61
151         JB(JBEAM)=1                                                    LCCP.62
151         V2(1,JBEAM)=V1(1)+VXX(1,JBEAM)                                 LCCP.63
153         V2(2,JBEAM)=V1(2)+VXX(2,JBEAM)                                 LCCP.64
155         SQK(JBEAM)=V2(1,JBEAM)**2+V2(2,JBEAM)**2                       LCCP.65
157         QQK(JBEAM)=SQRT(SQK(JBEAM))                                    LCCP.66
163         IF(QQK(JBEAM).GE.XK)  JB(JBEAM)=0                              LCCP.67
170         IF(JB(JBEAM).EQ.0)  GO TO 112                                  LCCP.68
172         XKP=SQRT(XK**2-SQK(JBEAM))                                     LCCP.69
200         V2(3,JBEAM)=-XKP                                               LCCP.70
200         CSTH(JBEAM)=XK/XKP                                             LCCP.71
      C                                                                    LCCP.72
      C     OBTAIN INCIDENT AND OUTGOING DAMPING ARRAYS *****              LCCP.73
      C                                                                    LCCP.74
202         DO 100  II=1,LAYER                                             LCCP.75
204         IF(JBEAM.EQ.1)  PATH1(II)=(D(II,3)+DIST)*THTT                  LCCP.76
211         IF(JBEAM.EQ.1)  DIN(II)=EXP(-PATH1(II)*DAMP)                   LCCP.77
221         PATH2(II,JBEAM)=(D(II,3)+DIST)*CSTH(JBEAM)                     LCCP.78
227         DOUT(II,JBEAM)=EXP(-PATH2(II,JBEAM)*DAMP)                      LCCP.79
236     100 CONTINUE                                                       LCCP.80
      C                                                                    LCCP.81
      C     CSTH(JBEAM) IS INVERSE OF COSINE FOR OUTGOING BEAM *****       LCCP.82
      C     COMPUTE SPHERICAL HARMONICS FOR OUTGOING BEAMS *****           LCCP.83
      C                                                                    LCCP.84
240         CALL YLM(V2(1,JBEAM),V2(2,JBEAM),V2(3,JBEAM),YINT,LCX,-1)      LCCP.85
      C                                                                    LCCP.86
247         DO 105  L=1,LMAX3                                              LCCP.87
260         YOUT(L,JBEAM)=YINT(L)                                          LCCP.88
262     105 CONTINUE                                                       LCCP.89
      C                                                                    LCCP.90
264         DO 110  L=1,3                                                  LCCP.91
273         V3(L,JBEAM)=V1(L)-V2(L,JBEAM)                                  LCCP.92
274     110 CONTINUE                                                       LCCP.93
276         GO TO 115                                                      LCCP.94
      C                                                                    LCCP.95
```

```
276     112 CONTINUE                                                        LOOP.96
303         V2(3,JBEAM)=0.0                                                 LOOP.97
303         V3(1,JBEAM)=0.0                                                 LOOP.98
304         V3(2,JBEAM)=0.0                                                 LOOP.99
304         V3(3,JBEAM)=0.0                                                 LOOP.100
305         CSTH(JBEAM)=0.0                                                 LOOP.101
306         DO 113  L=1,LAYER                                               LOOP.102
315         DELT(L,JBEAM)=0.0                                               LOOP.103
316         PATH2(L,JBEAM)=0.0                                              LOOP.104
316     113 CONTINUE                                                        LOOP.105
317         DO 114  L=1,LMAX3                                               LOOP.106
327         YELT(L,JBEAM)=(0.0,0.0)                                         LOOP.107
331     114 CONTINUE                                                        LOOP.108
      C                                                                     LOOP.109
332     115 CONTINUE                                                        LOOP.110
335         PRINT 812                                                       LOOP.111
340         PRINT 813, V1                                                   LOOP.112
346         DO 118  JBEAM=1,KNUM                                            LOOP.113
350         PRINT 819,IG(JBEAM),JM(JBEAM),JN(JBEAM),(V2(I,JBEAM),I=1,3),    LOOP.114
           2  (V3(I,JBEAM), I=1,3)                                          LOOP.115
374     118 CONTINUE                                                        LOOP.116
      C                                                                     LOOP.117
      C                                                                     LOOP.118
377         CALL FILLB                                                      LOOP.119
400         IF(MULT.LE.1)  GO TO 120                                        LOOP.120
403          IF(TIME.NE.0)  CALL SECOND(TIME4)                              LOOP.121
405         CALL GREEN(1)                                                   LOOP.122
407         CALL PERTB                                                      LOOP.123
410          IF(TIME.NE.0)  CALL SECOND(TIME5)                              LOOP.124
412         CALL INV                                                        LOOP.125
414         IF(TIME.NE.0)  CALL SECOND(TIME6)                               LOOP.126
417     120 CONTINUE                                                        LOOP.127
      C                                                                     LOOP.128
417         LM=1                                                            LOOP.129
420         CALL KINEM                                                      LOOP.130
421         CALL TSLM(LM,FTR,J)                                             LOOP.131
      C                                                                     LOOP.132
424         IF(MULT.LE.1)  GO TO 128                                        LOOP.133
      C                                                                     LOOP.134
      C     COMPUTE INTERPLANE PROPAGATORS. STACK HORIZONTALLY IN G(INDEX). LOOP.135
      C                                                                     LOOP.136
427         INDEX=1                                                         LOOP.137
427         IBEGIN=-LMAX4                                                   LOOP.138
431         DO 125  I2=1,LAYER                                              LOOP.139
433         DO 125  I1=1,LAYER                                              LOOP.140
434         IF(I1.EQ.I2)  GO TO 125                                         LOOP.141
436         IBEGIN=IBEGIN+LMAX4                                             LOOP.142
437         INDEX=INDEX+1                                                   LOOP.143
441         CALL GREEN(INDEX,IBEGIN)                                        LOOP.144
443     125 CONTINUE                                                        LOOP.145
450         IF(TIME.NE.0)  CALL SECOND(TIME7)                               LOOP.146
      C                                                                     LOOP.147
      C     GSTACK SETS UP MATRICES FOR DOUBLE DIFFRACTION.                 LOOP.148
      C                                                                     LOOP.149
452         LM=2                                                            LOOP.150
453         CALL GSTACK                                                     LOOP.151
454         CALL TSLM(LM,FTR,J)                                             LOOP.152
457         IF(TIME.NE.0)  CALL SECOND(TIME8)                               LOOP.153
```

```
      C                                                                 LCCP.154
      C     ITERATE TO ORDER  NDIF.                                     LCCP.155
      C                                                                 LCCP.156
462         KLM=0                                                       LCCP.157
463         NDF=NDIF-2                                                  LCCP.158
465         IF(NDF.LE.0)  GO TO 128                                     LCCP.159
466         DO 127  I=1,NDF                                             LCCP.160
470         CALL SCATT                                                  LCCP.161
471         CALL TSL4(I+2,FTP,J)                                        LCCP.162
476         IF(TIME.NE.0)  CALL SECOND(TIM(I))                          LCCP.163
501         KLM=I                                                       LCCP.164
503     127 CONTINUE                                                    LCCP.165
      C                                                                 LCCP.166
505     128 CONTINUE                                                    LCCP.167
505         IF(TIME.EQ.0)  GO TO 130                                    LCCP.168
506         PRINT 832, TIME3,TIME4,TIME5,TIME6,TIME7,TIME8              LCCP.169
526         IF(KLM.EQ.0)  GO TO 130                                     LCCP.170
527         PRINT 834,(TIM(IJK),IJK=1,KLM)                              LCCP.171
536     130 CONTINUE                                                    LCCP.172
536     140 CONTINUE                                                    LCCP.173
536         PRINT 816                                                   LCCP.174
542     150 CONTINUE                                                    LCCP.175
      C                                                                 LCCP.176
      C                                                                 LCCP.177
545         PRINT 806                                                   LCCP.178
550         DO 150  I=1,KNUM                                            LCCP.179
552         PRINT 804,IG(I),JM(I),JN(I)                                 LCCP.180
566         PRINT 802,( E(J),S(I,J), J=1,KNUM )                         LCCP.181
607     150 CONTINUE                                                    LCCP.182
      C                                                                 LCCP.183
      C                                                                 LCCP.184
        802 FORMAT( / 10X 4(*E=* F4.0,*--* E16.8,5X) )                  LCCP.185
        804 FORMAT( ///// 10X*TYPE(*I2,*)  BEAM(*I2,*,*I2,*)   ENERGY-VS-INTENLCCP.186
           2SITY* / 10X 76*******  )                                    LCCP.187
        806 FORMAT( I5, 6(1X 2I2) )                                     LCCP.188
        812 FORMAT( / )                                                 LCCP.189
        814 FORMAT( / 10X*ENERGY(*I2,*)=* F4.0,* E.V.   SINGLE=*E14.6,*   DOUBLCCP.190
           2LE=*E14.6,*      TRIPLE=*E14.6, /40X 14H  ************  10X  LCCP.191
           3  14H  ************ 12X 14H  ************  / )              LCCP.192
        816 FORMAT( 2X 13(10H* * * * *   ) )                            LCCP.193
        818 FORMAT( / 10X*PROPAGATION VECTORS IN CRYSTAL    V1=* 3F9.5 )  LCCP.194
        819 FORMAT( 19X*TYPE(*I2,*)  BEAM(*I2,*,* I2,*)    V2=* 3F9.5,  LCCP.195
           2  *     V3=* 3F9.5 )                                        LCCP.196
        820 FORMAT( / 10X*INCIDENT BEAM ENERGY IS * I3,* E.V.   $$$$$  NRGY=* LCCP.197
           2  I4 )                                                      LCCP.198
612     822 FORMAT( 10X*RENORMALIZED QUASIPARTICLE ENERGY                E=(*LCCP.199
           2 F9.4,*,* F9.4, *) E.V. * )                                 LCCP.200
        824 FORMAT( 10X*COMPLEX K-VECTOR                              KSTAR=(*LCCP.201
           2  F8.5,*,* F8.5,*) INVERSE ANGSTROMS * )                    LCCP.202
        826 FORMAT( 10X*CLASIPARTICLE RENORMALIZATION CONSTANT        ZNORM=(*LCCP.203
           2  F8.5, *,* F8.5, *)* )                                     LCCP.204
        828 FORMAT( 10X*INELASTIC DAMPING LENGTH                     LAMBDA=* LCCP.205
           2 F8.4, * ANGSTROMS * )                                      LCCP.206
        830 FORMAT( 1H1 )                                               LCCP.207
        832 FORMAT( / 10X*TIME REQUIREMENTS FOR EXECUTION OF ENERGY LOOP*  LCCP.208
           2  / 12X*BEGIN LOOP AT      * F8.3,                          LCCP.209
           3  / 12X*TO END OF FILLB    * F8.3,                          LCCP.210
           4  / 12X*TO END OF SLBPL    * F8.3,                          LCCP.211
           7  / 12X*TO END OF INV      *  F8.3,                         LCCP.212
           5  / 12X*TO END OF GREEN    * F8.3,                          LCCP.213
           6  / 12X*DOUBLE DIFFRACTION* F8.3 )                          LCCP.214
        834 FORMAT( 12X*    NEXT ITERATION* F8.3 )                      LCCP.215
      C                                                                 LCCP.216
      C                                                                 LCCP.217
612         END                                                         LCCP.218
```

```
      SUBROUTINE FILLB                                                  FILLB.2
C                                                                       FILLB.3
C     FILLB  FILLS THE B-ARRAYS, WHICH ARE OBTAINED FROM THE ATOMIC     FILLB.4
C     T-ARRAYS AND THE DEBYE WALLER FACTORS.  PROVISION IS MADE FOR     FILLB.5
C     DIFFERENT SURFACE AND BULK DEBYE-WALLER FACTORS.                  FILLB.6
C     B AND T-ARRAYS ARE BOTH DIAGONAL, IMPLYING SPHERICAL SYMMETRY*****FILLB.7
C                                                                       FILLB.8
C                                                                       FILLB.9
      COMMON/BLOCKB/T(2,30,7),ILYR(10),LSYM(10)                         B.2
      COMPLEX T                                                         B.3
      COMMON/BLOCKC/NLOW,NNUM,NDELTA,NRG,NEGY,LAYER,NLMMAX,NDIF,IST(6,6)C.2
     2  ,LR                                                             D.3
      COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,ICMP,ITWO,ITHREE                                               E.3
      COMMON/BLOCKG/SIG1,SIG2,XK,DAMP,ZNORM,POTL,GAMA,SIGMA,LOWER,NNRG  G.2
     2  ,LAMBDA                                                         G.3
      REAL LAMBDA                                                       G.4
      INTEGER SIGMA                                                     G.5
      COMPLEX ZNORM                                                     G.6
      COMMON/BLOCKK/BBB(36,2),TAU(36,36,2),TT(10,12),KTYP(10)           K.2
      COMPLEX  BBB,TAU,TT                                               K.3
      COMMON/BLOCKX/TIME,ITEM,MULT,ISCALE,IMAX,ITYPE,LSUM,ILOOP,IPWR    X.2
      INTEGER TIME                                                      X.3
      COMMON/BLKDEB/W(10),ATWT(10),TEMP,SURF,BULK,DIN(10),DCLT(10,12)   DEB.2
      DIMENSION  CP(21,21),BCF(21),XINT(20)                             FILLB.17
      COMPLEX TX(11),BOG                                                FILLB.18
C                                                                       FILLB.19
C                                                                       FILLB.20
      LDM=21                                                            FILLB.21
 2    SMALL=1.0E-10                                                     FILLB.22
C                                                                       FILLB.23
C     CALCULATE ARRAY  CP  IF ILOOP=1 *****                             FILLB.24
C                                                                       FILLB.25
 3    IF(ILOOP.NE.1)  GO TO 120                                         FILLB.26
 6    LLL=LDM-2                                                         FILLB.27
 7    LDD=LDM**2                                                        FILLB.28
11    DO 100  L=1,LDD                                                   FILLB.29
16    CP(L)=0.0                                                         FILLB.30
17  100 CONTINUE                                                        FILLB.31
C                                                                       FILLB.32
20    CP(1,1)=1.0                                                       FILLB.33
21    CP(2,2)=1.0                                                       FILLB.34
C                                                                       FILLB.35
22    DO 110  L=1,LLL                                                   FILLB.36
27    LL1=L+1                                                           FILLB.37
27    LL2=L+2                                                           FILLB.38
31    F1=1./LL1                                                         FILLB.39
32    CP(1,LL2)=-L*F1*CP(1,L)                                           FILLB.40
C                                                                       FILLB.41
40    DO 110  N=2,LL2                                                   FILLB.42
60    CP(N,LL2)=F1*((2*L+1)*CP(N-1,LL1)-L*CP(N,L))                      FILLB.43
65  110 CONTINUE                                                        FILLB.44
71    ILOOP=1                                                           FILLB.45
C                                                                       FILLB.46
72    DO 115  L=1,LDM                                                   FILLB.47
C                                                                       FILLB.48
73    SUM=0.0                                                           FILLB.49
74    DO 112  N=1,L                                                     FILLB.50
```

```
103          SUM=SUM+CP(N,L)                                                FILLB.51
104      112 CONTINUE                                                       FILLB.52
106      115 CONTINUE                                                       FILLB.53
      C                                                                     FILLB.54
110      120 CONTINUE                                                       FILLB.55
      C                                                                     FILLB.56
110          DO 220  II=1,LAYER                                             FILLB.57
112          KT=KTYP(II)                                                    FILLB.58
113          IF(II.EQ.1)  GO TO 124                                         FILLB.59
115          III=II-1                                                       FILLB.60
      C                                                                     FILLB.61
      C      IF ION CORE TYPE (KT) HAS PREVIOUSLY BEEN CALCULATED,          FILLB.62
      C      GO TO THE END OF THE LOOP.                                     FILLB.63
      C                                                                     FILLB.64
116          DO 122  J=1,III                                                FILLB.65
123          IF(KT.EQ.KTYP(J))  GO TO 220                                   FILLB.66
124      122 CONTINUE                                                       FILLB.67
126      124 CONTINUE                                                       FILLB.68
      C                                                                     FILLB.69
126          DO 130  L=1,LTOP                                               FILLB.70
140          TX(L)=T(KT,NRGY,L)                                             FILLB.71
142      130 CONTINUE                                                       FILLB.72
144      140 CONTINUE                                                       FILLB.73
      C                                                                     FILLB.74
      C      SET DEBYE-WALLER FACTOR ... BETA.  STACK EXPONENTIAL COEFFICIENTS*FILLB.75
      C                                                                     FILLB.76
144          BETA=2.*W(II)*XK**2                                            FILLB.77
      C                                                                     FILLB.78
147          XINT(1)=1.0                                                    FILLB.79
151          DO 150  I=1,19                                                 FILLB.80
153          XINT(I+1)=XINT(I)*BETA/I                                       FILLB.81
156          IF(ABS(XINT(I+1)).GT.SMALL)  GO TO 150                         FILLB.82
163          LARGE=I+1                                                      FILLB.83
164          GO TO 160                                                      FILLB.84
164      150 CONTINUE                                                       FILLB.85
166      160 CONTINUE                                                       FILLB.86
      C                                                                     FILLB.87
      C                                                                     FILLB.88
      C      COMPUTE FOURIER COEFFICIENTS  BCF *****                        FILLB.89
      C                                                                     FILLB.90
166          XFCN=EXP(-BETA)                                                FILLB.91
172          DO 190  L=1,LTOP                                               FILLB.92
173          TOTAL=0.0                                                      FILLB.93
173          ISIGN=-1                                                       FILLB.94
      C                                                                     FILLB.95
175          DO 180  N=1,L                                                  FILLB.96
176          ISIGN=-ISIGN                                                   FILLB.97
177          IF(ABS(CP(N,L)).LE.0.00001)  GO TO 180                         FILLB.98
205          LSTART=2                                                       FILLB.99
206          IF(ISIGN.GT.0)  LSTART=1                                       FILLB.100
211          SUM=0.0                                                        FILLB.101
      C                                                                     FILLB.102
212          DO 170  M=LSTART,LARGE,2                                       FILLB.103
223          SUM=SUM+XINT(M)/(M+N-1)                                        FILLB.104
227      170 CONTINUE                                                       FILLB.105
      C                                                                     FILLB.106
230          TOTAL=TOTAL+CP(N,L)*SUM                                        FILLB.107
235      180 CONTINUE                                                       FILLB.108
```

```
      C
240         LBCF=L                                                    FILLB.109
240         BCF(L)=TOTAL*XPCN                                         FILLB.110
243         IF(ABS(TOTAL).LE.SMALL)  GO TO 200                        FILLB.111
246       190 CONTINUE                                                FILLB.112
250       200 CONTINUE                                                FILLB.113
      C                                                               FILLB.114
      C     LBCF IS THE MAXIMUM INDEX FOR BCF.GE.SMALL *****          FILLB.115
      C                                                               FILLB.116
250         LLLL=0                                                    FILLB.117
      C                                                               FILLB.118
251         DO 210  L=1,LTOP                                          FILLB.119
253         M=2*L-1                                                   FILLB.120
255         CALL MCX(L,BCF,TX,LTOP,LBCF,BCG)                          FILLB.121
      C                                                               FILLB.122
261         DO 210  N=1,M                                             FILLB.123
266         LLLL=LLLL+1                                               FILLB.124
267         FBB(LLLL,KT)=BCG                                          FILLB.125
275       210 CONTINUE                                                FILLB.126
      C                                                               FILLB.127
300         PRINT 900, II,KT                                          FILLB.128
          900 FORMAT( / 10X*FILLB    LAYER* I3,*   KTYP=* I3 )        FILLB.129
310       220 CONTINUE                                                FILLB.130
      C                                                               FILLB.131
      C                                                               FILLB.132
312         RETURN                                                    FILLB.133
313         END                                                       FILLB.134
                                                                      FILLB.135
                                                                      FILLB.136
```

```
      SUBROUTINE MCX(LL,BCF,TX,LTCP,LECF,X)                             FILLB.137
C                                                                       FILLB.138
      DIMENSION  BCF(1)                                                 FILLB.139
      COMPLEX X,Y,Z,TX(1)                                               FILLB.140
C                                                                       FILLB.141
      X=(0.0,0.0)                                                       FILLB.142
12    DO 100  L1=1,LTCP                                                 FILLB.143
14    Y=TX(L1)                                                          FILLB.144
17    LCH=IABS(LL-L1)+1                                                 FILLB.145
22    LMI=LL+L1-1                                                       FILLB.146
22    IF(LMI.GT.LECF)  LMI=LECF                                         FILLB.147
26    IF(LMI.LT.LCH)  GO TO 110                                         FILLB.148
C                                                                       FILLB.149
31    DO 100  L2=LCH,LMI                                                FILLB.150
36    Z=Y*BCF(L2)                                                       FILLB.151
42    X=X+Z                                                             FILLB.152
46  100 CONTINUE                                                        FILLB.153
52  110 CONTINUE                                                        FILLB.154
C                                                                       FILLB.155
C                                                                       FILLB.156
C                                                                       FILLB.157
58    RETURN                                                            FILLB.158
59    END                                                               FILLB.159
```

```
SUBPROGRAM LENGTH
000100

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
110     -  000053

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
LMI     -  000103    LCH     -  000102    L1      -  000101    L2      -  000104
Y       -  000075    Z       -  000077

START OF CONSTANTS
000060

START OF TEMPORARIES
000060

START OF INDIRECTS
000071

UNUSED COMPILER SPACE
124700
```

```
      SUBROUTINE KINEP                                                  KINEP.2
C                                                                       KINEP.3
      COMMON/BLOCK0/V1(3),V2(3,12),V3(3,12),VX,VY,VZ,SOR(12),CCK(12),   C.2
     2  YIN(45),YOUT(45,12),XXZEND,AREA2                                C.3
      COMPLEX YIN,YOUT                                                  C.4
      COMMON/BLOCK1/NLCH,NNUM,NDELTA,ANG,NPGV,LAYER,NUMMAX,NCEP,IST(6,5)D.2
     2  ,LP                                                             D.3
      COMMON/BLOCK6/LMAX,LTCP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,ICAS,ITWO,ITHREE                                               E.3
      COMMON/BLOCK7/THET,IPHI,THTT,CIST,CSTH(12),VXX(2,12),KNUP,JM(12), F.2
     2  JN(12),IC(12),KJ(12)                                            F.3
      COMMON/BLOCK8/BPR(30,2),TAU(36,36,2),TT(10,12),KTYP(10)           H.2
      COMPLEX  BEE,TAU,TT                                               H.3
      COMMON/BLOCK9/W(10),ATWT(10),TEMP,SURF,BULK,DIN(10),DCLT(10,12)   DEP.2
      COMPLEX X                                                         KINEP.10
C                                                                       KINEP.11
C                                                                       KINEP.12
      DO 100  JBEAM=1,NNUP                                              KINEP.13
      DO 100  I=1,LAYER                                                 KINEP.14
      KJ=KTYP(I)                                                        KINEP.15
C                                                                       KINEP.16
6     X=(0.0,0.0)                                                       KINEP.17
C                                                                       KINEP.18
11    DO 100  L=1,LMAX2                                                 KINEP.19
24    X=X+BPR(L,KJ)*YOUT(L,JBEAM)*YIN(L)                                KINEP.20
37  100 CONTINUE                                                        KINEP.21
C                                                                       KINEP.22
46    TT(I,JBEAM)=A*DIN(I)*DOUT(I,JBEAM)                                KINEP.23
52  110 CONTINUE                                                        KINEP.24
C                                                                       KINEP.25
C                                                                       KINEP.26
57    RETURN                                                            KINEP.27
60    END                                                               KINEP.28
```

```
      SUBROUTINE GSTACK                                                 GSTACK.2
C                                                                       GSTACK.3
      COMMON/BLOCKD/NLOW,NNUM,NDELTA,NRG,NPGY,LAYER,NUMMAX,NDIF,IST(6,6)D.2
     2  ,LR                                                            D.3
      COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE                                              E.3
      COMMON/BLOCKK/BBB(36,2),TAU(36,36,2),TT(10,12),KTYP(10)           K.2
      COMPLEX  BBB,TAU,TT                                               K.3
      COMMON/BLKDEB/W(10),ATWT(10),TEMP,SURF,BULK,DIN(10),DOUT(10,12)   DEB.2
      COMPLEX Z,ZERO,UNIT                                               GSTACK.8
      LARGE C.QM(1296),RVEC(30000),C.G(49000)                           LARGE.2
C                                                                       DIM.2
C     DIMENSIONS OF KX,C,AND LL.                                        DIM.3
      NM3=2401 $ NM1=7532 $ NM2=7532                                    DIM.4
C     DIMENSIONS OF RMAG AND XMAG                                       DIM.5
      NM4=9000 $ NM5=9000                                               DIM.6
C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.              DIM.7
      NM6=30000 $ NM7=49000                                             DIM.8
C                                                                       DIM.9
C                                                                       GSTACK.11
C     MULTIPLY  G*TAU  AND STORE IN G(IONE). DAMPING NOT INCLUDED.      GSTACK.12
C                                                                       GSTACK.13
      UNIT=CMPLX(1.0,0.0)                                               GSTACK.14
      ZERO=(0.0,0.0)                                                    GSTACK.15
      IBEGIN=-LMAX4                                                     GSTACK.16
C                                                                       GSTACK.17
      DO 130  I2=1,LAYER                                                GSTACK.18
      DO 130  I1=1,LAYER                                                GSTACK.19
      IF(I1.EQ.I2)  GO TO 130                                           GSTACK.20
      KK=KTYP(I1)                                                       GSTACK.21
      IBEGIN=IBEGIN+LMAX4                                               GSTACK.22
C                                                                       GSTACK.23
      L2=IONE                                                           GSTACK.24
      LS=IBEGIN-LMAX3                                                   GSTACK.25
      DO 110  LOUT=1,LMAX3                                              GSTACK.26
      LS=LS+LMAX3                                                       GSTACK.27
      DO 110  LIN=1,LMAX3                                               GSTACK.28
      LG=LS                                                             GSTACK.29
      Z=ZERO                                                            GSTACK.30
C                                                                       GSTACK.31
      DO 100  L=1,LMAX3                                                 GSTACK.32
      LG=LG+1                                                           GSTACK.33
      Z=Z+G(LG)*TAU(L,LIN,KK)                                           GSTACK.34
  100 CONTINUE                                                          GSTACK.35
      L2=L2+1                                                           GSTACK.36
      G(L2)=Z                                                           GSTACK.37
  110 CONTINUE                                                          GSTACK.38
C                                                                       GSTACK.39
C     STACK  G(IONE)  BACK IN  G(1).                                    GSTACK.40
C                                                                       GSTACK.41
      DO 120  I=1,LMAX4                                                 GSTACK.42
      G(IBEGIN+I)=G(IONE+I)                                             GSTACK.43
  120 CONTINUE                                                          GSTACK.44
  130 CONTINUE                                                          GSTACK.45
C                                                                       GSTACK.46
C     INITIALIZE  G(ITWO)=0.                                            GSTACK.47
C                                                                       GSTACK.48
      JMAX=LAYER*LMAX4                                                  GSTACK.49
```

```
125          II=ITWC                                                    GSTACK.50
127          DO 140  J=1,JMAX                                           GSTACK.51
133          II=II+1                                                    GSTACK.52
134          G(II)=ZERO                                                 GSTACK.53
140      140 CONTINUE                                                   GSTACK.54
141          LR=LAYER-1                                                 GSTACK.55
142          IB=-LMAX4                                                  GSTACK.56
144          IA=ICNE-LMAX4                                              GSTACK.57
145          ID=ITWC-LMAX4                                              GSTACK.58
       C                                                                GSTACK.59
       C     SUM THE G(IBETA) OVER INCIDENT LAYER (I1) TO OBTAIN G(IDELTA)  GSTACK.60
       C       VALUES.  G(IDELTA) INCLUDES DAMPING OF INCIDENT WAVE.    GSTACK.61
       C       G(IDELTA) IS STACKED HORIZONTALLY.                       GSTACK.62
       C                                                                GSTACK.63
147          DO 180  I2=1,LAYER                                         GSTACK.64
151          ID=ID+LMAX4                                                GSTACK.65
       C                                                                GSTACK.66
152          DO 160  I1=1,LAYER                                         GSTACK.67
154          IF(I1.EQ.I2)  GO TO 160                                    GSTACK.68
157          F=DIN(I1)                                                  GSTACK.69
160          IB=IB+LMAX4                                                GSTACK.70
       C                                                                GSTACK.71
162          DO 150  L=1,LMAX4                                          GSTACK.72
173          G(ID+L)=G(ID+L)+G(IB+L)*F                                  GSTACK.73
202      150 CONTINUE                                                   GSTACK.74
203      160 CONTINUE                                                   GSTACK.75
       C                                                                GSTACK.76
       C     TRANSPOSE  G(IDELTA)  AND STORE IN  G(IALPHA).             GSTACK.77
       C     G(IALPHA)  IS STACKED VERTICALLY.                          GSTACK.78
       C                                                                GSTACK.79
206          IA=IA+LMAX4                                                GSTACK.80
207          CALL EXCX(IA+1,ID+1)                                       GSTACK.81
       C                                                                GSTACK.82
       C     ADD UNIT MATRIX TO  G(IALPHA)  AND STORE VERTICALLY IN G(IDELTA). GSTACK.83
       C                                                                GSTACK.84
214          F=DIN(I2)                                                  GSTACK.85
216          IDL=ID                                                     GSTACK.86
220          IAL=IA                                                     GSTACK.87
222          DO 170   L=1,LMAX3                                         GSTACK.88
223          DO 170  LP=1,LMAX3                                         GSTACK.89
234          IDL=IDL+1                                                  GSTACK.90
235          IAL=IAL+1                                                  GSTACK.91
236          G(IDL)=G(IAL)                                              GSTACK.92
243          IF(L.EQ.LP)  G(IDL)=G(IDL)+UNIT*F                          GSTACK.93
253      170 CONTINUE                                                   GSTACK.94
257      180 CONTINUE                                                   GSTACK.95
       C                                                                GSTACK.96
       C                                                                GSTACK.97
261          RETURN                                                     GSTACK.98
262          END                                                        GSTACK.99
```

SUBPROGRAM LENGTH
000346

FUNCTION ASSIGNMENTS

```
      SUBROUTINE SCATT                                                  SCATT.2
      COMMON/BLOCKD/NLOW,NNUM,NDELTA,NRG,NRGY,LAYER,NUMMAX,NDIF,IST(6,6)D.2
     2  ,LR                                                             D.3
      COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE                                               E.3
      COMPLEX  ZERO                                                     SCATT.5
      LARGE C.QM(1296),RVEC(30000),C.G(49000)                           LARGE.2
C                                                                       DIM.2
C     DIMENSIONS OF KX,C,AND LL.                                        DIM.3
      NM3=2401 $ NM1=7532 $ NM2=7532                                    DIM.4
C     DIMENSIONS OF RMAG AND XMAG                                       DIM.5
 4    NM4=9000 $ NM5=9000                                               DIM.6
C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.              DIM.7
 5    NM6=30000 $ NM7=49000                                             DIM.8
C                                                                       DIM.9
C                                                                       SCATT.8
10    ZERO=(0.0,0.0)                                                    SCATT.9
C                                                                       SCATT.10
C     INITIALIZE WORKING SPACE  G(IGAMMA)=0.                            SCATT.11
C                                                                       SCATT.12
12    IT1=ITHREE+1                                                      SCATT.13
14    IT2=IT1+LAYER*LMAX4                                               SCATT.14
17    DO 100  I=IT1,IT2                                                 SCATT.15
25    G(I)=ZERO                                                         SCATT.16
30  100 CONTINUE                                                        SCATT.17
C                                                                       SCATT.18
C     INCREASE ORDER OF PERTURBATION EXPANSION FOR T-MATRIX.            SCATT.19
C     G(IGAMMA)=G(IBETA)*G(IALPHA).  G(IGAMMA) IS STACKED HORIZONTALLY. SCATT.20
C                                                                       SCATT.21
31    IBETA=-LMAX4                                                      SCATT.22
32    IGAMMA=-LMAX4+ITHREE                                              SCATT.23
34    DO 110  I2=1,LAYER                                                SCATT.24
36    IGAMMA=IGAMMA+LMAX4                                               SCATT.25
37    DO 110  I1=1,LAYER                                                SCATT.26
C                                                                       SCATT.27
41    IF(I1.EQ.I2)  GO TO 110                                           SCATT.28
42    IBETA=IBETA+LMAX4                                                 SCATT.29
44    IALPHA=IONE+(I1-1)*LMAX4                                          SCATT.30
50    CALL ARITX(IBETA+1,2,IALPHA+1,IGAMMA+1,1,1)                       SCATT.31
60  110 CONTINUE                                                        SCATT.32
C                                                                       SCATT.33
C     TRANSFER  G(IGAMMA)  TO  G(IALPHA)...STACK VERTICALLY.            SCATT.34
C     ADD CONTRIBUTION FROM  G(IALPHA) TO G(IDELTA).                    SCATT.35
C                                                                       SCATT.36
65    DO 120  I2=1,LAYER                                                SCATT.37
66    II=(I2-1)*LMAX4+1                                                 SCATT.38
71    IGAM=ITHREE+II                                                    SCATT.39
72    IALPH=IONE+II                                                     SCATT.40
74    IDEL=ITWO+II                                                      SCATT.41
C                                                                       SCATT.42
76    CALL EXCX(IALPH,IGAM)                                             SCATT.43
100   CALL ARITX(IDEL,1,IALPH,IDEL,0,0)                                 SCATT.44
104 120 CONTINUE                                                        SCATT.45
C                                                                       SCATT.46
C     G(IDELTA)  IS THE ITERATED PART OF THE SCATTERING MATRIX.         SCATT.47
C       IT IS TO BE MULTIPLIED BY TAU*DCUT*G(IDELTA) *** IN TSUM.       SCATT.48
C                                                                       SCATT.49
C                                                                       SCATT.50


107   RETURN                                                            SCATT.51
107   END                                                               SCATT.52
```

```
      SUBROUTINE TSUM(ITR,FTR,JINDX)                                    TSUM.2
C                                                                       TSUM.3
      COMMON/BLCCKA/A(10,2),B(10,2),D(10,3),RA(10,2),RB(10,2),ALAT,SPACEA.2
     2,BASIS(2),NPASIS                                                  A.3
      COMMON/BLCCKC/V1(3),V2(3,12),V3(3,12),VX,VY,VZ,SQK(12),CCK(12),   C.3
     2  YIN(49),YOUT(49,12),XKZERO,AREA2                                C.3
      COMPLEX YIN,YOUT                                                  C.4
      COMMON/BLCCKD/NLCK,KNUM,NDELTA,NPG,NRGY,LAYER,NUMMAX,NCIF,IST(5,6)D.2
     2  ,LR                                                             D.3
      COMMON/BLCCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,ICNE,ITWO,ITHFEE                                               E.3
      COMMON/BLCCKF/ITHT,IPHI,THTT,DIST,CSTH(12),VXX(2,12),KALM,JM(12), F.2
     2  JN(12),IG(12),KQ(12)                                            F.3
      COMMON/BLCCKK/BBB(36,2),TAU(36,36,2),TT(10,12),KTYP(10)           K.2
      COMPLEX  BPE,TAU,TT                                               K.3
      COMMON/BLCCKN/E(51),EN(51),S(12,51)                               N.2
      COMMON/BLKDEP/W(10),ATWT(10),TEMP,SURF,BULK,DIN(10),DOUT(10,12)   DEB.2
      COMPLEX  Z,ZERO,CPX,TERM,SM1,SM(12)                               TSUM.12
      DIMENSION  NBEAM(4)                                               TSUM.13
      DATA(NBEAM(I),I=1,4)/00,01,10,11/                                 TSUM.14
      LARGE C.JM(1296),RVEC(30000),C.G(49000)                           LARGE.2
C                                                                       DIM.2
C     DIMENSIONS OF KX,C,AND LL.                                        DIM.3
      NM3=2401 $ NM1=7532 $ NM2=7532                                    DIM.4
C     DIMENSIONS OF RMAG AND XMAG                                       DIM.5
      NM4=9000 $ NM5=9000                                               DIM.6
C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.              DIM.7
      NM6=30000 $ NM7=49000                                             DIM.8
C                                                                       DIM.9
      ZERO=(0.0,0.0)                                                    TSUM.17
      IF(ITR.EQ.1)  GO TO 125                                           TSUM.18
      IBEGIN=ITWO-LMAX4                                                 TSUM.19
C                                                                       TSUM.20
      DO 120  I2=1,LAYER                                                TSUM.21
      IBEGIN=IBEGIN+LMAX4                                               TSUM.22
      KK=KTYP(I2)                                                       TSUM.23
C                                                                       TSUM.24
      DO 90  I=1,KNUM                                                   TSUM.25
      SM(I)=ZERO                                                        TSUM.26
   90 CONTINUE                                                          TSUM.27
C                                                                       TSUM.28
      DO 110  LOUT=1,LMAX3                                              TSUM.29
      IG=IBEGIN                                                         TSUM.30
      DO 110  LIN=1,LMAX3                                               TSUM.31
      TERM=ZERO                                                         TSUM.32
C                                                                       TSUM.33
      DO 100  L=1,LMAX3                                                 TSUM.34
      IG=IG+1                                                           TSUM.35
      TERM=TERM+TAU(LOUT,L,KK)*G(IG)                                    TSUM.36
  100 CONTINUE                                                          TSUM.37
C                                                                       TSUM.38
      CPX=TERM*YIN(LIN)                                                 TSUM.39
      DO 110  I=1,KNUM                                                  TSUM.40
      SM(I)=SM(I)+CPX*YOUT(LOUT,I)                                      TSUM.41
  110 CONTINUE                                                          TSUM.42
C                                                                       TSUM.43
      DO 120  I=1,KNUM                                                  TSUM.44
      TT(I2,I)=SM(I)*DOUT(I2,I)                                         TSUM.45
```

```
154     120 CONTINUE                                                    TSUM.46
      C                                                                 TSUM.47
      C     SUM  TT-VALUES OVER SURFACE LAYERS.                         TSUM.48
      C                                                                 TSUM.49
162     125 CONTINUE                                                    TSUM.50
162         GASP=0.0                                                    TSUM.51
163         PRINT 800, ITR                                              TSUM.52
171         DO 140  JBEAM=1,KNUM                                        TSUM.53
175         SM1=ZERO                                                    TSUM.54
177         PRINT 806,JM(JBEAM),JN(JBEAM)                               TSUM.55
211         DO 130  I2=1,LAYER                                          TSUM.56
215         Y=ZERO                                                      TSUM.57
217         DO 128  IND=1,3                                             TSUM.58
230         Y=Y+V3(IND,JBEAM)*D(I2,IND)                                 TSUM.59
232     128 CONTINUE                                                    TSUM.60
234         Z=CMPLX(COS(Y),SIN(Y))                                      TSUM.61
246         TERM=Z*TT(I2,JBEAM)                                         TSUM.62
255         SM1=SM1+TERM                                                TSUM.63
261         TMG=CABS(TT(I2,JBEAM))                                      TSUM.64
265         IF(TMG.GT.2.0)  GASP=100.0                                  TSUM.65
274         PRINT 802, I ,TMG,TT(I2,JBEAM),TERM,SM1                     TSUM.66
317     130 CONTINUE                                                    TSUM.67
      C                                                                 TSUM.68
322         S(JBEAM,JINDX)=FTR*CABS(SM1)**2                             TSUM.69
331         SLOG=ALOG10(S(JBEAM,JINDX))                                 TSUM.70
336         PRINT 804,E(JINDX),S(JBEAM,JINDX),SLOG                      TSUM.71
361     140 CONTINUE                                                    TSUM.72
364         IF(GASP.LT.0.0001)  RETURN                                  TSUM.73
367         PRINT 808                                                   TSUM.74
372         STOP                                                        TSUM.75
      C                                                                 TSUM.76
      C                                                                 TSUM.77
        800 FORMAT( /// 10X*ITERATION OF T-MATRIX TO ORDER* I3 / )      TSUM.78
        802 FORMAT( 10X*LAYER=*I3,*  T=/*F11.6,*/  *2F10.6,*  TERM=*2F10.6,  TSUM.79
           2 *  SUM=*2F10.6 )                                           TSUM.80
        804 FORMAT( /10X*ENERGY=* F4.0,* E.V.   INTENSITY=* E14.6,*     LOGARITSUM.81
           2THM=* F10.5,/ 42X 11H***********  )                         TSUM.82
        806 FORMAT( / 3X 2I2,* BEAM* )                                  TSUM.83
        808 FORMAT( // 10X*PROGRAM STOPPED DUE TO NON-CONVERGENCE OF LAYER SCATSUM.84
           2TTERING AMPLITUDE* )                                        TSUM.85
      C                                                                 TSUM.86
377         RETURN                                                      TSUM.87
400         END                                                         TSUM.88
```

```
          SUBROUTINE INV                                                INV.2
C                                                                       INV.3
C         INVERTS MATRIX (I-BBB*GSP) AND CALCULATES TAU-MATRIX.         INV.4
C                                                                       INV.5
          COMMON/BLOCKC/V1(3),V2(3,12),V3(3,12),VX,VY,VZ,SGK(12),CCK(12), C.2
         2  YIN(49),YOUT(49,12),XKZERO,AREA2                            C.3
          COMPLEX YIN,YOUT                                              C.4
          COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX            E.2
         2  ,IONE,ITWO,ITHREE                                           E.3
          COMMON/BLOCKF/ITHT,IPHI,THTT,DIST,CSTH(12),VXX(2,12),KNUM,JM(12), F.2
         2  JN(12),IC(12),KQ(12)                                        F.3
          COMMON/BLOCKK/BBB(36,2),TAU(36,36,2),TT(10,12),KTYP(10)       K.2
          COMPLEX  BBB,TAU,TT                                           K.3
          COMMON/BLOCKX/TIME,ITEM,MULT,ISCALE,IMAX,ITYPE,LSUM,TLOOP,IPWR X.2
          INTEGER TIME                                                  X.3
          COMMON/CONST/PI,PI4,PI42,CON1,CON2,CON3,CON4,CONSTT,IM,CNST   CONST.2
          COMPLEX IM                                                    CONST.3
          COMPLEX  X,L,TERM,P(12)                                       INV.12
          DIMENSION  PX(12)                                             INV.13
          LARGE C.CM(1296),RVEC(30000),C.G(49000)                       LARGE.2
          LARGE Q(20000)                                                INV.15
C                                                                       DIM.2
C         DIMENSIONS OF KX,C,AND LL.                                    DIM.3
          NM3=2401 $ NM1=7532 $ NM2=7532                                DIM.4
C         DIMENSIONS OF RMAG AND XMAG                                   DIM.5
 4        NM4=9000 $ NM5=9000                                           DIM.6
C         DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.          DIM.7
 5        NM6=30000 $ NM7=49000                                         DIM.8
C                                                                       DIM.9
C                                                                       INV.17
10        DO 112  KK=1,ITYPE                                            INV.18
11        INDEX=0                                                       INV.19
11        IX=0                                                          INV.20
12        DO 100  I=1,LMAX3                                             INV.21
14        DO 100  J=1,LMAX3                                             INV.22
27        X=BBB(J,KK)                                                   INV.23
31        IX=IX+1                                                       INV.24
32        INDEX=INDEX+2                                                 INV.25
34        L=(0.0,0.0)                                                   INV.26
36        IF(I.EQ.J)  U=CMPLX(1.0,0.0)                                  INV.27
42        U=U-X*G(IX)                                                   INV.28
54        Q(INDEX)=AIMAG(U)                                             INV.29
60        Q(INDEX-1)=REAL(U)                                            INV.30
62      100 CONTINUE                                                    INV.31
C                                                                       INV.32
C         Q IS NOW STACKED ROWWISE  (AS IS G)                           INV.33
C         INVERT Q-MATRIX                                               INV.34
C                                                                       INV.35
66        CALL CINV(LMAX3)                                              INV.36
C                                                                       INV.37
C         OUTPUT IN ARRAY Q IS STACKED COLUMNWISE                       INV.38
C         CALCULATE TAU-ARRAY ... TAU=Q*BBB                             INV.39
C                                                                       INV.40
70        INDEX=0                                                       INV.41
71        DO 110  I=1,LMAX3                                             INV.42
73        DO 110  J=1,LMAX3                                             INV.43
110       INDEX=INDEX+2                                                 INV.44
112       TERM=CMPLX(Q(INDEX-1),Q(INDEX))                               INV.45
```

```
120       TAU(J,I,KK)=BBB(J,KK)*TERM                                    INV.46
125     110 CONTINUE                                                    INV.47
132       I=0                                                           INV.48
133       CALL TEST(I,KK)                                               INV.49
134     112 CONTINUE                                                    INV.50
C                                                                       INV.51
C                                                                       INV.52
137       RETURN                                                        INV.53
137       END                                                           INV.54
```

```
      SUBROUTINE CINV(NDIM)                                             INV.55
C                                                                       INV.56
C     COMPLEX MATRIX INVERSION ROUTINE.                                 INV.57
C     TOTAL LARGE CORE LENGTH IS(5*NDIM*NDIM), DIMENSION IPIVOT(NDIM).  INV.58
C     THE COMPLEX MATRIX Q(NDIM,NDIM) OCCUPIES THE FIRST (2*NDIM*NDIM)  INV.59
C     LOCATIONS IN EXTENDED CORE.  THE INVERTED MATRIX IS DELIVERED     INV.60
C     IN THE SAME LOCATION.  THE REMAINING LENGTH REQUIRED IN           INV.61
C     EXTENDED CORE IS FOR INTERMEDIATE WORKING ARRAYS.                 INV.62
C                                                                       INV.63
      LARGE C.QM(1296),RVEC(30000),C.G(49000)                           LARGE.2
      LARGE Q(20000)                                                    INV.65
      COMMON/MTRX/NDIM,NSQ,N1,N2,N3,N4,N5,IPIVOT(50)                    INV.66
      INTEGER PLUS,TIMES                                                INV.67
      DATA PLUS,TIMES/1,2/                                              INV.68
C                                                                       DIM.2
C     DIMENSIONS OF KX,C,AND LL.                                        DIM.3
      NM3=2401 $ NM1=7532 $ NM2=7532                                    DIM.4
C     DIMENSIONS OF RMAG AND XMAG                                       DIM.5
 5    NM4=9000 $ NM5=9000                                               DIM.6
C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.              DIM.7
 6    NM6=30000 $ NM7=49000                                             DIM.8
C                                                                       DIM.9
C                                                                       INV.70
12    NDIM=NDIM                                                         INV.71
12    NSQ=NDIM*NDIM                                                     INV.72
14    N1=0                                                              INV.73
14    N2=NSQ                                                            INV.74
15    N3=N2+NSQ                                                         INV.75
16    N4=N3+NSQ                                                         INV.76
17    N5=N4+NSQ                                                         INV.77
C                                                                       INV.78
C     STORE REAL AND IMAGINARY PARTS OF (Q) IN BLOCKS 3 AND 4.          INV.79
C                                                                       INV.80
21    DO 100  I=1,NSQ                                                   INV.81
27    J2=2*I                                                            INV.82
30    J1=J2-1                                                           INV.83
31    Q(N3+I)=Q(J1)                                                     INV.84
34    Q(N4+I)=Q(J2)                                                     INV.85
36  100 CONTINUE                                                        INV.86
C                                                                       INV.87
C     PERFORM INVERSION ON BLOCKS 3 AND 4.                              INV.88
C     AT FINAL STEP, BLOCKS 3 AND 4 ARE THE REAL AND IMAGINARY PARTS    INV.89
C     OF THE INVERTED COMPLEX MATRIX (Q).                               INV.90
C                                                                       INV.91
C                                                                       INV.92
40    CALL EXCH(N3,N3)                                                  INV.93
42    CALL EXCH(N4,N4)                                                  INV.94
44    CALL EXCH(N2,N3)                                                  INV.95
46    CALL INVERT(N2)                                                   INV.96
50    CALL ARITH(N2,TIMES,N4,N1,+1.0)                                   INV.97
54    CALL EXCH(N4,N4)                                                  INV.98
56    CALL ARITH(N4,TIMES,N1,N2,+1.0)                                   INV.99
62    CALL ARITH(N3,PLUS,N2,N3)                                         INV.100
65    CALL INVERT(N3)                                                   INV.101
67    CALL EXCH(N1,N1)                                                  INV.102
71    CALL ARITH(N1,TIMES,N3,N4,-1.0)                                   INV.103
C                                                                       INV.104
C                                                                       INV.105
```

```
C     STORE INVERTED COMPLEX ARRAY COLUMNWISE IN (Q).                   INV.106
C                                                                       INV.107
75    DO 110  I=1,NSQ                                                   INV.108
105   J2=2*I                                                            INV.109
106   J1=J2-1                                                           INV.110
107   Q(J1)=Q(N3+I)                                                     INV.111
112   Q(J2)=Q(N4+I)                                                     INV.112
114 110 CONTINUE                                                        INV.113
C                                                                       INV.114
C                                                                       INV.115
116   RETURN                                                            INV.116
116   END                                                               INV.117
                                                                        INV.118
```

```
      SUBROUTINE INVERT(MM)                                             INV.119
C                                                                       INV.120
C     REAL MATRIX INVERSION IS PERFORMED ON THE MATRIX STORED IN        INV.121
C     EXTENDED CORE BLOCK BEGINNING AT LOCATION (MM+1).                 INV.122
C     INVERTED MATRIX IS RETURNED IN SAME LOCATION.                     INV.123
C                                                                       INV.124
      LARGE C.QM(1256),RVEC(30000),C.G(49000)                           LARGE.2
      LARGE C(20000)                                                    INV.126
      COMMON/MTRX/NDIM,NSC,N1,N2,N3,N4,N5,IPIVOT(50)                    INV.127
C                                                                       DIM.2
C     DIMENSIONS OF KX,C,AND LL.                                        DIM.3
      NM3=2401 $ NM1=7532 $ NM2=7532                                    DIM.4
C     DIMENSIONS OF RMAG AND XMAG                                       DIM.5
 5    NM4=5000 $ NM5=5000                                               DIM.6
C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.              DIM.7
 6    NM6=30000 $ NM7=49000                                             DIM.8
C                                                                       DIM.9
C                                                                       INV.129
11    SMALL=1.E-15                                                      INV.130
12    KK=0                                                              INV.131
C                                                                       INV.132
C     INITIALIZE THE UNIT MATRIX STARTING AT LOCATION (N5+1)            INV.133
C                                                                       INV.134
13    DO 100  I=1,NDIM                                                  INV.135
14    DO 100  J=1,NDIM                                                  INV.136
20    KK=KK+1                                                           INV.137
21    C(N5+KK)=0.0                                                      INV.138
24    IF(I.EQ.J)  C(N5+KK)=1.0                                          INV.139
30  100 CONTINUE                                                        INV.140
C                                                                       INV.141
C     PERFORM INVERSION USING GAUSS-JORDAN METHOD.                      INV.142
C                                                                       INV.143
34    JJ=0                                                              INV.144
34    KK=0                                                              INV.145
35    DO 160  I=1,NDIM                                                  INV.146
37    PIVOT=0.0                                                         INV.147
C                                                                       INV.148
40    DO 110  J=1,NDIM                                                  INV.149
45    JJ=JJ+1                                                           INV.150
46    IF(ABS(C(MM+JJ)).LE.PIVOT)  GO TO 110                             INV.151
53    PIV=C(MM+JJ)                                                      INV.152
55    PIVOT=ABS(PIV)                                                    INV.153
56    IPIV=J                                                            INV.154
57  110 CONTINUE                                                        INV.155
60    IF(PIVOT.LE.SMALL)  GO TO 200                                     INV.156
63    PIVOT=1./PIV                                                      INV.157
C                                                                       INV.158
64    DO 120  J=1,NDIM                                                  INV.159
72    KK=KK+1                                                           INV.160
73    M1=MM+KK                                                          INV.161
74    M2=N5+KK                                                          INV.162
75    C(M1)=C(M1)*PIVOT                                                 INV.163
102   C(M2)=C(M2)*PIVOT                                                 INV.164
104 120 CONTINUE                                                        INV.165
C                                                                       INV.166
106   MN=JJ-NDIM                                                        INV.167
107   LL=0                                                              INV.168
C                                                                       INV.169
```

```
110         DO 150  K=1,NDIM                                          INV.170
112         IF(K.NE.I)  GO TO 130                                     INV.171
114         LL=LL+NDIM                                                INV.172
115         GO TO 150                                                 INV.173
      C                                                               INV.174
116     130 CONTINUE                                                  INV.175
116         AAA=Q(MM+LL+IPIV)                                         INV.176
      C                                                               INV.177
122         DO 140  J=1,NDIM                                          INV.178
134         LL=LL+1                                                   INV.179
135         M1=MM+LL                                                  INV.180
136         M2=N5+LL                                                  INV.181
137         Q(M1)=Q(M1)-AAA*Q(MM+MN+J)                                INV.182
145         Q(M2)=Q(M2)-AAA*Q(N5+MN+J)                                INV.183
151     140 CONTINUE                                                  INV.184
154     150 CONTINUE                                                  INV.185
157         IPIVCT(I)=IPIV                                            INV.186
161     160 CONTINUE                                                  INV.187
      C                                                               INV.188
      C     RESTACK BLOCK 5 IN BLOCK MM.                              INV.189
      C                                                               INV.190
163         L=0                                                       INV.191
164         DO 170  I=1,NDIM                                          INV.192
165         K=(IPIVCT(I)-1)*NDIM                                      INV.193
171         DO 170  J=1,NDIM                                          INV.194
174         L=L+1                                                     INV.195
175         K=K+1                                                     INV.196
176         Q(MM+K)=Q(N5+L)                                           INV.197
202     170 CONTINUE                                                  INV.198
      C                                                               INV.199
206         RETURN                                                    INV.200
      C                                                               INV.201
      C                                                               INV.202
207     200 CONTINUE                                                  INV.203
207         PRINT 800,I                                               INV.204
      800 FORMAT( // 10X*DETERMINANT=0    PIVOT ELEMENT ZERO FOR ROW *I3//) INV.205
216         STOP                                                      INV.206
      C                                                               INV.207
220         END                                                       INV.208
                                                                      INV.209
```

```
      SUBROUTINE EXCH(M1,M2)                                            INV.210
C                                                                       INV.211
C     EXCHANGES STORAGE FROM COLUMNWISE TO ROWWISE AND VICE-VERSA.      INV.212
C                                                                       INV.213
      LARGE C.QM(1296),RVEC(30000),C.G(49000)                           LARGE.2
      LARGE G(20000)                                                    INV.215
      COMMON/MTRX/NDIM,NSQ,N1,N2,N3,N4,N5,IPIVOT(50)                    INV.216
C                                                                       DIM.2
C     DIMENSIONS OF KX,C,AND LL.                                        DIM.3
      NM3=2401 $ NM1=7532 $ NM2=7532                                    DIM.4
C     DIMENSIONS OF RMAG AND XMAG                                       DIM.5
 6    NM4=9000 $ NM5=9000                                               DIM.6
C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.              DIM.7
10    NM6=30000 $ NM7=49000                                             DIM.8
C                                                                       DIM.9
C                                                                       INV.218
12    L=M2                                                              INV.219
13    DO 100  I=1,NDIM                                                  INV.220
15    K=I-NDIM                                                          INV.221
16    DO 100  J=1,NDIM                                                  INV.222
23    K=K+NDIM                                                          INV.223
24    L=L+1                                                             INV.224
26    Q(N5+K)=Q(L)                                                      INV.225
32  100 CONTINUE                                                        INV.226
C                                                                       INV.227
36    DO 110  I=1,NSQ                                                   INV.228
44    Q(M1+I)=Q(N5+I)                                                   INV.229
45  110 CONTINUE                                                        INV.230
C                                                                       INV.231
C                                                                       INV.232
50    RETURN                                                            INV.233
50    END                                                               INV.234
                                                                        INV.235
```

```
            SUBROUTINE ARITH(M1,OP,M2,M3,SIGN)                          INV.236
      C                                                                 INV.237
      C     PERFORMS A*OP*B=C, WHERE OP=(PLUS,TIMES).                   INV.238
      C     FOR OP=TIMES, (A) IS ROWWISE, (B) IS COLUMNWISE, AND        INV.239
      C     (C) IS DELIVERED AS A COLUMNWISE ARRAY.                     INV.240
      C                                                                 INV.241
            LARGE C.QM(1296),RVEC(30000),C.G(49000)                     LARGE.2
            LARGE G(20000)                                              INV.243
            COMMON/MTRX/NDIM,NSQ,N1,N2,N3,N4,N5,IPIVOT(50)              INV.244
            INTEGER OP                                                  INV.245
            REAL IPIVOT                                                 INV.246
      C                                                                 DIM.2
      C     DIMENSIONS OF KX,C,AND LL.                                  DIM.3
            NM3=2401 $ NM1=7532 $ NM2=7532                              DIM.4
      C     DIMENSIONS OF RMAG AND XMAG                                 DIM.5
   11       NM4=9000 $ NM5=9000                                         DIM.6
      C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.        DIM.7
   13       NM6=30000 $ NM7=49000                                       DIM.8
      C                                                                 DIM.9
      C                                                                 INV.248
   15       IF(OP.EQ.2)  GO TO 110                                      INV.249
   17       IF(OP.NE.1)  STOP                                           INV.250
      C                                                                 INV.251
      C     MATRIX ADDITION PACKAGE.                                    INV.252
      C                                                                 INV.253
   22       DO 100  I=1,NSQ                                             INV.254
   32       Q(M3+I)=Q(M1+I)+Q(M2+I)                                     INV.255
   35   100 CONTINUE                                                    INV.256
   37       RETURN                                                      INV.257
      C                                                                 INV.258
      C     MATRIX MULTIPLICATION PACKAGE.                              INV.259
      C                                                                 INV.260
   37   110 CONTINUE                                                    INV.261
   37       II=-NDIM                                                    INV.262
   41       DO 130  I=1,NDIM                                            INV.263
   42       II=II+NDIM                                                  INV.264
   43       SMALLIN(IPIVOT,Q(M2+II+1),NDIM)                             INV.265
   52       JJ=-NDIM                                                    INV.266
      C                                                                 INV.267
   53       DO 130  J=1,NDIM                                            INV.268
   54       JJ=JJ+NDIM                                                  INV.269
   55       I1=II+J                                                     INV.270
   56       X=0.0                                                       INV.271
      C                                                                 INV.272
   60       DO 120  K=1,NDIM                                            INV.273
   65       I3=JJ+K                                                     INV.274
   66       X=X+IPIVOT(K)*Q(M1+I3)                                      INV.275
   73   120 CONTINUE                                                    INV.276
   74       G(M3+I1)=SIGN*X                                             INV.277
  102   130 CONTINUE                                                    INV.278
      C                                                                 INV.279
      C                                                                 INV.280
  107       RETURN                                                      INV.281
  107       END                                                         INV.282
```

SUBPROGRAM LENGTH

```
      SUBROUTINE TEST(ITER,KK)                                          TEST.1
C                                                                       TEST.2
C     TEST AND PRINT SUMMED TAU-VALUES.                                 TEST.4
C                                                                       TEST.5
      COMMON/BLOCKC/V1(3),V2(3,12),V3(3,12),VX,VY,VZ,SQK(12),CCK(12),    C.2
     2  YIN(49),YOUT(49,12),XKZERO,AREA2                                C.3
      COMPLEX YIN,YOUT                                                  C.4
      COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE                                               E.3
      COMMON/BLOCKF/ITHT,IPHI,THTT,DIST,CSTH(12),VXX(2,12),KNUM,JM(12), F.2
     2  JN(12),IC(12),KC(12)                                            F.3
      COMMON/BLOCKK/BBB(36,2),TAU(36,36,2),TT(10,12),KTYF(10)           K.2
      COMPLEX  EBE,TAL,TT                                               K.3
      COMMON/CONST/PI,PI4,PI42,CON1,CON2,CON3,CON4,CONSTT,IM,CNST       CONST.2
      COMPLEX IM                                                        CONST.3
      COMPLEX  TERM,PP,P(12)                                            TEST.11
      DIMENSION  PX(12)                                                 TEST.12
C                                                                       TEST.13
C                                                                       TEST.14
      DO 100  I=1,KNUM                                                  TEST.15
 11   P(I)=(0.0,0.0)                                                    TEST.16
 13 100 CONTINUE                                                        TEST.17
C                                                                       TEST.18
 14   DO 110  LIN=1,LMAX3                                               TEST.19
 17   DO 110  LOUT=1,LMAX3                                              TEST.20
 20   TERM=TAU(LOUT,LIN,KK)*YIN(LIN)                                    TEST.21
 33   DO 110  J=1,KNUM                                                  TEST.22
 45   P(J)=P(J)+TERM*YOUT(LOUT,J)                                       TEST.23
 55 110 CONTINUE                                                        TEST.24
C                                                                       TEST.25
 64   DO 120  I=1,KNUM                                                  TEST.26
 65   PX(I)=CNST*CABS(P(I))**2                                          TEST.27
 74 120 CONTINUE                                                        TEST.28
C                                                                       TEST.29
 76   PRINT 800, ITER                                                   TEST.30
103   PRINT 810, (KK,IC(I),JM(I),JN(I),PX(I), I=1,KNUM)                 TEST.31
C                                                                       TEST.32
C                                                                       TEST.33
 800 FORMAT( / 10X*NUMBER OF ITERATIONS=* I4 )                          TEST.34
 810 FORMAT( 20X*CORE(*I2,*)   SYM(*I2,*)     TAU(*I2,*,* I2, *)=*      TEST.35
    2  E16.8 )                                                         TEST.36
C                                                                       TEST.37
136   RETURN                                                            TEST.38
136   END                                                               TEST.39
```

```
      SUBROUTINE PERTB                                                  PERT.2
C                                                                       PERT.3
C     PERFORMS PERTURBATION EXPANSION TO OBTAIN TAU.                    PERT.4
C                                                                       PERT.5
      COMMON/BLOCKC/V1(3),V2(3,12),V3(3,12),VX,VY,VZ,SQK(12),CCK(12),   C.2
     2  YIN(49),YOUT(49,12),XKZERO,AREA2                                C.3
      COMPLEX YIN,YOUT                                                  C.4
      COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE,                                              E.3
      COMMON/BLOCKK/BBB(36,2),TAU(36,36,2),TT(10,12),KTYP(10)           K.2
      COMPLEX  BBB,TAU,TT                                               K.3
      COMMON/BLOCKX/TIME,ITEM,MULT,ISCALE,IMAX,ITYPE,LSUM,ILOOP,IPWR    X.2
      INTEGER TIME                                                      X.3
      COMMON/CONST/PI,PI4,PI42,CON1,CON2,CON3,CON4,CONSTT,IM,CNST       CONST.2
      COMPLEX IM                                                        CONST.3
      COMPLEX P,PP                                                      PERT.11
      DIMENSION  IL(20)                                                 PERT.12
      LARGE C.JM(1296),RVEC(30000),C.G(49000)                           LARGE.2
C                                                                       DIM.2
C     DIMENSIONS OF KX,C,AND LL.                                        DIM.3
      NM3=2401 $ NM1=7532 $ NM2=7532                                    DIM.4
C     DIMENSIONS OF RMAG AND XMAG                                       DIM.5
      NM4=9000 $ NM5=9000                                               DIM.6
C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.              DIM.7
      NM6=30000 $ NM7=49000                                             DIM.8
C                                                                       DIM.9
C                                                                       PERT.15
      IPWR=20                                                           PERT.16
      SMALL=0.0001                                                      PERT.17
C                                                                       PERT.18
      DO 170  KK=1,ITYPE                                                PERT.19
      JK=LMAX4                                                          PERT.20
      KL=0                                                              PERT.21
C                                                                       PERT.22
      DO 110  J=1,LMAX3                                                 PERT.23
      PP=BBB(J,KK)                                                      PERT.24
      DO 110  K=1,LMAX3                                                 PERT.25
      JK=JK+1                                                           PERT.26
      KL=KL+1                                                           PERT.27
      G(JK)=PP*G(KL)                                                    PERT.28
  110 CONTINUE                                                          PERT.29
C                                                                       PERT.30
      DO 120  I=1,IPWR                                                  PERT.31
      IL(I)=(I+1)*LMAX4+1                                               PERT.32
  120 CONTINUE                                                          PERT.33
C                                                                       PERT.34
      CALL EXCX(IL(1),LMAX4+1)                                          PERT.35
C                                                                       PERT.36
      DO 130  I=2,IPWR                                                  PERT.37
      CALL ARITX(LMAX4+1,2,IL(I-1),IL(I),2,C)                           PERT.38
  130 CONTINUE                                                          PERT.39
C                                                                       PERT.40
C     STACK UNIT MATRIX IN  G(2).                                       PERT.41
C                                                                       PERT.42
      JK=LMAX4                                                          PERT.43
      DO 140  I=1,LMAX3                                                 PERT.44
      DO 140  J=1,LMAX3                                                 PERT.45
      JK=JK+1                                                           PERT.46
```

```
112         G(JK)=(0.C,0.0)                                          PERT.47
116         IF(I.EQ.J)  G(JK)=CMPLX(1.0,0.0)                         PERT.48
123     140 CONTINUE                                                 PERT.49
      C                                                              PERT.50
      C     PERFORM MATRIX MULTIPLICATIONS UP TO (G*T)**IPWR.        PERT.51
      C                                                              PERT.52
127         IF(KK.EQ.1)  PRINT 800                                   PERT.53
134         DO 150  I=1,IPWR                                         PERT.54
136         CALL ARITX(LMAX4+1,1,IL(I),LMAX4+1,0,C)                  PERT.55
145         JK=LMAX4                                                 PERT.56
147         DO 145  J=1,LMAX3                                        PERT.57
150         PP=BBB(J,KK)                                             PERT.58
155         DO 145  K=1,LMAX3                                        PERT.59
167         JK=JK+1                                                  PERT.60
170         TAU(K,J,KK)=G(JK)*PP                                     PERT.61
177     145 CONTINUE                                                 PERT.62
      C                                                              PERT.63
      C     TEST FOR CONVERGENCE                                     PERT.64
      C                                                              PERT.65
203         P=(0.C,C.C)                                              PERT.66
205         DO 146  LIN=1,LMAX3                                      PERT.67
207         DO 146  LOUT=1,LMAX3                                     PERT.68
222         P=P+TAU(LOUT,LIN,KK)*YIN(LIN)*YOUT(LOUT,1)               PERT.69
235     146 CONTINUE                                                 PERT.70
241         PX=CONST*CABS(P)**2                                      PERT.71
244         IF(I.LE.-)  GO TO 148                                    PERT.72
      C                                                              PERT.73
247         IF(ABS(PX-STORE).LE.SMALL)  GO TO 160                    PERT.74
253     148 CONTINUE                                                 PERT.75
253         STORE=PX                                                 PERT.76
255     150 CONTINUE                                                 PERT.77
257     160 CONTINUE                                                 PERT.78
      C                                                              PERT.79
257         CALL TEST(I,KK)                                          PERT.80
261     170 CONTINUE                                                 PERT.81
      C                                                              PERT.82
      C                                                              PERT.83
      800 FORMAT( / 10X*OBTAIN SUBPLANE PROPAGATOR BY ITERATION* )   PERT.84
264         RETURN                                                   PERT.85
264         END                                                      PERT.86
```

```
      SUBROUTINE EXCX(M1,M2)                                            EXCH.2
C                                                                       EXCH.3
C     EXCHANGES STORAGE FROM COLUMNWISE TO ROWWISE AND VICE-VERSA       EXCH.4
C     REPLACES MATRIX STARTING AT  G(M2) BY ONE STARTING AT  G(M1).     EXCH.5
C                                                                       EXCH.6
      COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE                                               E.3
      LARGE G,QM(1296),RVEC(30000),G,G(49000)                           LARGE.2
C                                                                       DIM.2
C     DIMENSIONS OF KX,C,AND LL.                                        DIM.3
      NM3=2401 $ NM1=7532 $ NM2=7532                                    DIM.4
C     DIMENSIONS OF RMAG AND XMAG                                       DIM.5
      NM4=9000 $ NM5=9000                                               DIM.6
C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.              DIM.7
      NM6=30000 $ NM7=49000                                             DIM.8
C                                                                       DIM.9
C                                                                       EXCH.10
      LG=M2-1                                                           EXCH.11
      DO 100  I=1,LMAX3                                                 EXCH.12
      K=I-LMAX3                                                         EXCH.13
      DO 100  J=1,LMAX3                                                 EXCH.14
      K=K+LMAX3                                                         EXCH.15
      LG=LG+1                                                           EXCH.16
      QM(K)=G(LG)                                                       EXCH.17
  100 CONTINUE                                                          EXCH.18
C                                                                       EXCH.19
      DO 110  I=1,LMAX4                                                 EXCH.20
      G(M1+I-1)=QM(I)                                                   EXCH.21
  110 CONTINUE                                                          EXCH.22
C                                                                       EXCH.23
C                                                                       EXCH.24
      RETURN                                                            EXCH.25
      END                                                               EXCH.26
```

```
      SUBROUTINE ARITX(M1,OP,M2,M3,OUT,ACC)                             ARITH.2
C                                                                       ARITH.3
C     MATRIX ADDITION AND MULTIPLICATION ROUTINE.                       ARITH.4
C     OP=1 FOR ADDITION.  OP=2 FOR MULTIPLICATION.                      ARITH.5
C     G(M1) MUST BE HORIZONTAL AND G(M2) MUST BE VERTICAL.              ARITH.6
C     IF OUT=1, OUTPUT IS HORIZONTAL.  IF OUT=2, OUTPUT IS VERTICAL.    ARITH.7
C     ACC=0  FOR G(M3) TO BE A SIMPLE PRODUCT.                          ARITH.8
C     IF  ACC=1,  G(M3) IS A CUMULATIVE PRODUCT.                        ARITH.9
C                                                                       ARITH.10
      COMMON/BLOCKE/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2 ,IONE,ITWO,ITHREE                                                E.3
      INTEGER OP,OUT,ACC                                                ARITH.12
      COMPLEX X,QB(45)                                                  ARITH.13
      LARGE C.2M(1296),RVEC(30000),C.G(49000)                           LARGE.2
C                                                                       DIM.2
C     DIMENSIONS OF KX,C,AND LL.                                        DIM.3
      NM3=2401 $ NM1=7532 $ NM2=7532                                    DIM.4
C     DIMENSIONS OF RMAG AND XMAG                                       DIM.5
      NM4=9000 $ NM5=9000                                               DIM.6
C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.              DIM.7
      NM6=30000 $ NM7=49000                                             DIM.8
C                                                                       DIM.9
C                                                                       ARITH.16
      LMAX5=2*LMAX3                                                     ARITH.17
      J1=M1-1                                                           ARITH.18
      J2=M2-1                                                           ARITH.19
      J3=M3-1                                                           ARITH.20
      N1=M1                                                             ARITH.21
      N2=M2                                                             ARITH.22
      N3=M3                                                             ARITH.23
      IF(OP.EQ.2)  GO TO 110                                            ARITH.24
      IF(OP.NE.1)  STOP                                                 ARITH.25
C                                                                       ARITH.26
C     MATRIX ADDITION PACKAGE.                                          ARITH.27
C                                                                       ARITH.28
      DO 100  I=1,LMAX4                                                 ARITH.29
      G(J3+I)=G(J1+I)+G(J2+I)                                           ARITH.30
  100 CONTINUE                                                          ARITH.31
      RETURN                                                            ARITH.32
C                                                                       ARITH.33
C     MATRIX MULTIPLICATION PACKAGE                                     ARITH.34
C                                                                       ARITH.35
  110 CONTINUE                                                          ARITH.36
C                                                                       ARITH.37
      IF(OUT.EQ.2)  GO TO 115                                           ARITH.38
      N2=N1                                                             ARITH.39
      J1=J2                                                             ARITH.40
      IF(OUT.NE.1)  STOP                                                ARITH.41
  115 CONTINUE                                                          ARITH.42
C                                                                       ARITH.43
      II=-LMAX3                                                         ARITH.44
      DO 130  I=1,LMAX3                                                 ARITH.45
      II=II+LMAX3                                                       ARITH.46
      SMALLIN(QB,G(N2+II),LMAX5)                                        ARITH.47
      JJ=-LMAX3                                                         ARITH.48
C                                                                       ARITH.49
      DO 130  J=1,LMAX3                                                 ARITH.50
      JJ=JJ+LMAX3                                                       ARITH.51
```

```
      I1=II+J                                                           ARITH.52
      X=(0.0,0.0)                                                       ARITH.53
C                                                                       ARITH.54
      DO 120  K=1,LMAX3                                                 ARITH.55
      I3=JJ+K                                                           ARITH.56
      X=X+QB(K)*G(J1+I3)                                                ARITH.57
  120 CONTINUE                                                          ARITH.58
      IF(ACC.EQ.0)  G(J3+I1)=X                                          ARITH.59
      IF(ACC.EQ.1)  G(J3+I1)=G(J3+I1)+X                                 ARITH.60
  130 CONTINUE                                                          ARITH.61
C                                                                       ARITH.62
C                                                                       ARITH.63
      RETURN                                                            ARITH.64
      END                                                               ARITH.65
```

```
      SUBROUTINE GREEN(INDEX,IBEGIN)                                    GREEN.2
C                                                                       GREEN.3
      COMMON/BUFF/LOG,NBC,KKK                                           GREEN.4
      COMMON/BLOCKD/NLOW,NNUM,NDELTA,NRG,NRGY,LAYER,NUMMAX,NDIF,IST(6,6)D.2
     2  ,LR                                                            D.3
      COMMON/BLOCKE/LMAX,LTUP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE                                              E.3
      COMMON/BLOCKF/KX(2401),C(7532),LL(7532),YINT(169)                 H.2
      COMPLEX YINT                                                      H.3
      COMMON/BLOCKJ/MAXX(91),RMAG( 9000),XMAG( 9000),MAXR               J.2
      COMMON/BLOCKK/BBB(36,2),TAU(36,36,2),TT(10,12),KTYP(10)           K.2
      COMPLEX  BBB,TAU,TT                                               K.3
      COMMON/BLOCKL/ARG,H(169),XD(9),PATH1(10),PATH2(10,12)             L.2
      COMPLEX  ARG,H                                                    L.3
      COMMON/BLOCKM/RATIO,XCON,KSTAR                                    M.2
      COMPLEX KSTAR,XCON                                                M.3
      COMMON/BLOCKQ/PX,PY,PZ                                            Q.2
      COMPLEX XX,TERM                                                   GREEN.14
      LARGE C.QM(1296),RVEC(30000),C.G(49000)                           LARGE.2
C                                                                       DIM.2
C     DIMENSIONS OF KX,C,AND LL.                                        DIM.3
      NM3=2401 $ NM1=7532 $ NM2=7532                                    DIM.4
C     DIMENSIONS OF RMAG AND XMAG                                       DIM.5
      NM4=9000 $ NM5=9000                                               DIM.6
C     DIMENSIONS OF RVEC AND G-ARRAY IN LARGE CORE MEMORY.              DIM.7
      NM6=30000 $ NM7=49000                                             DIM.8
C                                                                       DIM.9
C                                                                       GREEN.17
C                                                                       GREEN.18
      ID=IBEGIN                                                         GREEN.19
      IF(INDEX.EQ.1)  ID=0                                              GREEN.20
      DO 50  I=1,LMAX4                                                  GREEN.21
      IV=ID+I                                                           GREEN.22
      G(IV)=(0.0,0.0)                                                   GREEN.23
   50 CONTINUE                                                          GREEN.24
C                                                                       GREEN.25
      MIN=1                                                             GREEN.26
      IF(INDEX.GT.1)  MIN=MAXX(INDEX-1)+1                               GREEN.27
      MAX=MAXX(INDEX)                                                   GREEN.28
      IF(MIN.GT.MAX)  GO TO 130                                         GREEN.29
      LCCC=(MIN-1)*3-2                                                  GREEN.30
      LINT=LMAX2-1                                                      GREEN.31
C                                                                       GREEN.32
      DO 120  LCC=MIN,MAX                                               GREEN.33
      LCCC=LCCC+3                                                       GREEN.34
      R1=RVEC(LCCC)                                                     GREEN.35
      R2=RVEC(LCCC+1)                                                   GREEN.36
      R3=RVEC(LCCC+2)                                                   GREEN.37
      CALL YLM(R1,R2,R3,YINT,LINT,-1)                                   GREEN.38
C                                                                       GREEN.39
C                                                                       GREEN.40
      ARG=KSTAR*RMAG(LCC)                                               GREEN.41
      XX=XCON*CEXP( CMPLX(0.0,XMAG(LCC)*RATIO) )                        GREEN.42
C                                                                       GREEN.43
      CALL HANKEL                                                       GREEN.44
C                                                                       GREEN.45
      INDX=IC                                                           GREEN.46
      NCX=0                                                             GREEN.47
```

```
110         DO 110  L=1,LMAX4                                          GREEN.48
114         JJ=KX(L)                                                   GREEN.49
115         TERM=(0.0,0.0)                                             GREEN.50
120         INDX=INDX+1                                                GREEN.51
      C                                                                GREEN.52
121         DO 100  I=1,JJ                                             GREEN.53
126         NCX=NCX+1                                                  GREEN.54
127         LLL=LL(NCX)                                                GREEN.55
      C                                                                GREEN.56
131         TERM=TERM+YINT(LLL)*H(LLL)*C(NCX)                          GREEN.57
146     100 CONTINUE                                                   GREEN.58
147         G(INDX)=G(INDX)+TERM*XX                                    GREEN.59
161     110 CONTINUE                                                   GREEN.60
162         IF(NCX.GT.NUMMAX)  PRINT 800, NCX                          GREEN.61
175     120 CONTINUE                                                   GREEN.62
200     130 CONTINUE                                                   GREEN.63
200         RETURN                                                     GREEN.64
      C                                                                GREEN.65
        800 FORMAT( 20X 44H********** INDEX OF ARRAY EXCEEDS NUMMAX   (  GREEN.66
           2   I5, 13H ) **********      )                             GREEN.67
        810 FORMAT( // 10X 10H********** * END OF FILE TAPE 2   INDEX=* I4,  GREEN.68
           2 *   LOC=* I5 )                                           GREEN.69
        820 FORMAT( // 10X 10H********** * PARITY ERROR TAPE 2   INDEX=* I4,  GREEN.70
           2 *   LOC=* I5 )                                           GREEN.71
      C                                                                GREEN.72
      C                                                                GREEN.73
201         END                                                        GREEN.74
```

```
      SUBROUTINE HANKEL                                                 HANKEL.2
C                                                                       HANKEL.3
      COMMON/BLOCKF/LMAX,LTOP,LMAX1,LMAX2,LMAX3,LMAX4,LX                E.2
     2  ,IONE,ITWO,ITHREE                                               E.3
      COMMON/BLOCKL/ARG,H(169),XD(9),PATH1(10),PATH2(10,12)             L.2
      COMPLEX  ARG,H                                                    L.3
      COMMON/CONST/PI,PI4,PI42,CON1,CON2,CON3,CON4,CONSTT,IM,CNST       CONST.2
      COMPLEX IM                                                        CONST.3
      COMPLEX HH(31),IX,HHH,XPON,ZINV                                   HANKEL.7
C                                                                       HANKEL.8
C                                                                       HANKEL.9
      MAX=LMAX2-2                                                       HANKEL.10
      XPON=CEXP(IM*ARG)                                                 HANKEL.11
      ZINV=1./ARG                                                       HANKEL.12
C                                                                       HANKEL.13
      HH(1)=-IM*ZINV*XPON                                               HANKEL.14
      HH(2)=-ZINV*(1.+IM*ZINV)*XPON                                     HANKEL.15
C                                                                       HANKEL.16
      DO 100  I=1,MAX                                                   HANKEL.17
      HH(I+2)=(I+I+1)*ZINV*HH(I+1)-HH(I)                                HANKEL.18
  100 CONTINUE                                                          HANKEL.19
C                                                                       HANKEL.20
      N=0                                                               HANKEL.21
      IX=IM                                                             HANKEL.22
C                                                                       HANKEL.23
      DO 110  I=1,LMAX2                                                 HANKEL.24
      NBR=2*I-1                                                         HANKEL.25
      IX=-IX*IM                                                         HANKEL.26
      HHH=IX*HH(I)                                                      HANKEL.27
C                                                                       HANKEL.28
      DO 110  J=1,NBR                                                   HANKEL.29
      N=N+1                                                             HANKEL.30
      H(N)=HHH                                                          HANKEL.31
  110 CONTINUE                                                          HANKEL.32
C                                                                       HANKEL.33
C                                                                       HANKEL.34
      RETURN                                                            HANKEL.35
      END                                                               HANKEL.36
```

```
$$$$$$$$$$  DATA DECK LISTING  $$$$$$$$$$
PUNCH=00
  7 DECEMBER 1972
 ALUMINUM (PERTURBATION EXPANSION)  PENDRY PHASE SHIFTS *****
    5   30    1    1   MULT,MAXR,LSUM,TIME
SIGMA=00
    5      NDIF
 1        ITYPE=NUMBER OF ATOMIC TYPES CONSIDERED
    3   LMAX
 ALUMINUM *****
 26.9815        ATOMIC WEIGHT OF ALUMINUM
    6   40   71    LTOP,LOWEN AND NNRG
 5.85363640E-01 -5.98783025E-01  5.69197909E-01 -6.21556115E-01  5.51888103E-01
-6.42409151E-01  5.33666305E-01 -6.61365386E-01  5.14745739E-01 -6.78462968E-01
 4.95321422E-01 -6.93752629E-01  4.75726765E-01 -7.06972431E-01  4.55914424E-01
-7.18610708E-01  4.36000773E-01 -7.28899886E-01  4.16091926E-01 -7.37837210E-01
 3.96284099E-01 -7.45483758E-01  3.76834837E-01 -7.51674909E-01  3.57577060E-01
-7.56837265E-01  3.38567087E-01 -7.61020432E-01  3.19855652E-01 -7.64275189E-01
 3.01488133E-01 -7.66653055E-01  2.83675748E-01 -7.68059745E-01  2.66202636E-01
-7.68779901E-01  2.49093718E-01 -7.68851214E-01  2.32371042E-01 -7.68311396E-01
 2.16053903E-01 -7.67197993E-01  2.00243446E-01 -7.65520786E-01  1.84828090E-01
-7.63333140E-01  1.69818442E-01 -7.60725536E-01  1.55223344E-01 -7.57708025E-01
 1.41049953E-01 -7.54310137E-01  1.27424002E-01 -7.50518809E-01  1.14169335E-01
-7.46431859E-01  1.01288386E-01 -7.42070043E-01  8.87857016E-02 -7.37453733E-01
 7.66529780E-02 -7.32602872E-01  6.49706038E-02 -7.27554677E-01  5.36269647E-02
-7.22259004E-01  4.26209167E-02 -7.16822637E-01  3.19538185E-02 -7.11229621E-01
 2.16145571E-02 -7.05494252E-01  1.16605716E-02 -6.99658753E-01  2.00875512E-03
-6.93651318E-01 -7.34366784E-03 -6.87572898E-01 -1.63997563E-02 -6.81404171E-01
-2.51428368E-02 -6.75155533E-01 -3.36012900E-02 -6.68840630E-01 -4.17723033E-02
-6.62465236E-01 -4.96793459E-02 -6.56037507E-01 -5.73260508E-02 -6.49565373E-01
-6.47162040E-02 -6.43056536E-01 -7.18268846E-02 -6.36554601E-01 -7.87029565E-02
-6.29968471E-01 -8.53480775E-02 -6.23394220E-01 -9.17669964E-02 -6.16807745E-01
-9.79605459E-02 -6.10214756E-01 -1.03917248E-01 -6.03657304E-01 -1.09668079E-01
-5.97041483E-01 -1.15216667E-01 -5.90461884E-01 -1.20566686E-01 -5.83892948E-01
-1.25721855E-01 -5.77338966E-01 -1.30672947E-01 -5.70810401E-01 -1.35443404E-01
-5.64302110E-01 -1.40037013E-01 -5.57917572E-01 -1.44457179E-01 -5.51360147E-01
-1.48707425E-01 -5.44933067E-01 -1.52780225E-01 -5.38546271E-01 -1.56696024E-01
-5.32192770E-01 -1.60458064E-01 -5.25875157E-01 -1.64049592E-01 -5.19595926E-01
-1.67533855E-01 -5.13357469E-01 -1.70854095E-01 -5.07142682E-01 -1.74033551E-01
-5.01011960E-01 -1.77075454E-01 -4.94909202E-01 -1.79983025E-01 -4.88855810E-01
-1.82759471E-01 -4.82853691E-01  0.                              0.
-1.93098214E-01 -9.14831520E-03 -8.31401068E-02 -6.08086787E-03 -6.42315045E-02
-3.70740220E-03 -4.62866446E-02 -1.96551081E-03 -2.92350701E-02 -7.99958752E-04
-1.30043497E-02 -1.61484916E-04  2.69561566E-03 -7.07404911E-06  1.75433186E-02
-3.05460278E-04  3.15886077E-02 -1.00925475E-03  4.48744175E-02 -2.07529984E-03
 5.74475328E-02 -3.46415598E-03  6.93697163E-02 -5.14452710E-03  8.06305278E-02
-7.07659237E-03  9.12625321E-02 -9.22951614E-03  1.01296014E-01 -1.15737675E-02
 1.10759262E-01 -1.40823892E-02  1.19611699E-01 -1.67113669E-02  1.27981043E-01
-1.94644002E-02  1.35888749E-01 -2.23223742E-02  1.43355081E-01 -2.52675485E-02
 1.50399233E-01 -2.82834531E-02  1.57031461E-01 -3.13574814E-02  1.63281311E-01
-3.44422063E-02  1.69165060E-01 -3.76026553E-02  1.74698248E-01 -4.07607811E-02
 1.79859738E-01 -4.39254015E-02  1.84816164E-01 -4.71113514E-02  1.89405124E-01
-5.02706026E-02  1.93677163E-01 -5.33962907E-02  1.97646240E-01 -5.64783407E-02
 2.01325756E-01 -5.95084174E-02  2.04470181E-01 -6.23061303E-02  2.07487606E-01
-6.51181936E-02  2.10381562E-01 -6.79453896E-02  2.13165430E-01 -7.07845879E-02
 2.15812454E-01 -7.36337424E-02  2.18431100E-01 -7.65510791E-02  2.20898881E-01
-7.94454350E-02  2.23221192E-01 -8.23139297E-02  2.25453251E-01 -8.51538587E-02
 2.27450110E-01 -8.79626862E-02  2.29349353E-01 -9.07218009E-02  2.31132263E-01
-9.34531430E-02  2.32802846E-01 -9.61548372E-02  2.34344989E-01 -9.88251221E-02
 2.35922465E-01 -1.01462346E-01  2.37167775E-01 -1.04062828E-01  2.38421607E-01
```

$\ell = 0$

$\ell = 1$

$t_\lambda^\ell(E)$ for $E = 40, 42, 44, \ldots$ eV. Values are listed with Real part followed by Imaginary part for each energy.

-1.06613223E-01	2.39587030E-01	-1.09142280E-01	2.40667031E-01	-1.11638827E-01
2.41664516E-01	-1.14101766E-01	2.42575142E-01	-1.16591113E-01	2.43412672E-01
-1.18909301E-01	2.44179532E-01	-1.21265486E-01	2.44878083E-01	-1.23588882E-01
2.45510630E-01	-1.25878752E-01	2.46074677E-01	-1.28127642E-01	2.46579696E-01
-1.30345052E-01	2.47027648E-01	-1.32530416E-01	2.47450445E-01	-1.34683210E-01
2.47759955E-01	-1.36802945E-01	2.48045146E-01	-1.38884517E-01	2.48282188E-01
-1.40934489E-01	2.48472708E-01	-1.42952488E-01	2.48678296E-01	-1.44938174E-01
2.48720507E-01	-1.46891234E-01	2.48780859E-01	-1.48871385E-01	2.48800835E-01
-1.50698369E-01	2.48781885E-01	-1.52551956E-01	2.48755423E-01	-1.54371941E-01
2.48432834E-01	-1.56158141E-01	0.	0.	
-5.50368616E-01	-3.85711499E-01	-5.48328161E-01	-4.10641741E-01	-5.44099187E-01
-4.33299020E-01	-5.38119856E-01	-4.53597894E-01	-5.30781198E-01	-4.71568589E-01
-5.22429608E-01	-4.87249270E-01	-5.13330479E-01	-4.99224772E-01	-5.03895777E-01
-5.09922406E-01	-4.94255720E-01	-5.19087368E-01	-4.84555986E-01	-5.27068718E-01
-4.74808676E-01	-5.33818069E-01	-4.65323381E-01	-5.38536431E-01	-4.55985683E-01
-5.42536930E-01	-4.46825850E-01	-5.45935253E-01	-4.37870942E-01	-5.48727050E-01
-4.29144963E-01	-5.50937724E-01	-4.20778634E-01	-5.52194594E-01	-4.12635769E-01
-5.53130117E-01	-4.04721354E-01	-5.53755998E-01	-3.97039792E-01	-5.54083814E-01
-3.89594906E-01	-5.54124953E-01	-3.82480798E-01	-5.53650623E-01	-3.75567909E-01
-5.53038137E-01	-3.68853895E-01	-5.52291302E-01	-3.62336556E-01	-5.51413986E-01
-3.56013801E-01	-5.50410087E-01	-3.49910639E-01	-5.49225557E-01	-3.43985282E-01
-5.47952677E-01	-3.38235026E-01	-5.46593770E-01	-3.32657297E-01	-5.45151192E-01
-3.27249626E-01	-5.43627313E-01	-3.22033314E-01	-5.41981010E-01	-3.16970907E-01
-5.40380898E-01	-3.12059743E-01	-5.38528241E-01	-3.07297276E-01	-5.36724332E-01
-3.02681054E-01	-5.34870473E-01	-2.98228406E-01	-5.32935891E-01	-2.93907684E-01
-5.30970709E-01	-2.89716462E-01	-5.28975499E-01	-2.85662407E-01	-5.26950851E-01
-2.81713272E-01	-5.24897374E-01	-2.77915859E-01	-5.22787567E-01	-2.74229695E-01
-5.20464786E-01	-2.70652632E-01	-5.18529070E-01	-2.67182605E-01	-5.16380487E-01
-2.63817624E-01	-5.14219123E-01	-2.60573880E-01	-5.12020202E-01	-2.57422369E-01
-5.09821618E-01	-2.54361244E-01	-5.07623031E-01	-2.51388728E-01	-5.05424134E-01
-2.48503109E-01	-5.03224649E-01	-2.45718896E-01	-5.01003448E-01	-2.43010242E-01
-4.98791953E-01	-5.40375597E-01	-4.96589593E-01	-2.37873473E-01	-4.94395833E-01
-2.35322433E-01	-4.92210168E-01	-2.32914464E-01	-4.90015456E-01	-2.30568444E-01
-4.87836502E-01	-2.28282501E-01	-4.85672619E-01	-2.26055605E-01	-4.83523160E-01
-2.23886575E-01	-4.81387506E-01	-2.21784203E-01	-4.79253372E-01	-2.19732457E-01
-4.77137909E-01	-2.17730298E-01	-4.75040418E-01	-2.15776722E-01	-4.72960230E-01
-2.13870760E-01	-4.70896707E-01	-2.12011477E-01	-4.68849240E-01	-2.10197971E-01
-4.66817243E-01	-2.08429366E-01	-4.64800159E-01	-2.06724818E-01	-4.62797451E-01
-2.05023507E-01	-4.60808603E-01	0.	0.	
-1.04212514E-01	-9.34874912E-03	-1.13994733E-01	-1.14844053E-02	-1.23434929E-01
-1.38169786E-02	-1.32559472E-01	-1.63361340E-02	-1.41390434E-01	-1.90400633E-02
-1.49946336E-01	-2.19253783E-02	-1.58720331E-01	-2.51499793E-02	-1.66990184E-01
-2.84726032E-02	-1.74786300E-01	-3.18961685E-02	-1.82136114E-01	-3.54008043E-02
-1.89064528E-01	-3.89737193E-02	-1.95810877E-01	-4.27022175E-02	-2.02062656E-01
-4.64313754E-02	-2.07847770E-01	-5.01460158E-02	-2.13192191E-01	-5.38323250E-02
-2.18120165E-01	-5.74777373E-02	-2.22693541E-01	-6.10940611E-02	-2.26873400E-01
-6.46367384E-02	-2.30681289E-01	-6.80956637E-02	-2.34137490E-01	-7.14617432E-02
-2.37261112E-01	-7.47268173E-02	-2.40023050E-01	-7.78404209E-02	-2.42512093E-01
-8.08737745E-02	-2.44744338E-01	-8.37947095E-02	-2.46731511E-01	-8.66076235E-02
-2.48487008E-01	-8.93084402E-02	-2.49961304E-01	-9.18472002E-02	-2.51260374E-01
-9.42809127E-02	-2.52392917E-01	-9.66253342E-02	-2.53367188E-01	-9.88724962E-02
-2.54191031E-01	-1.01020687E-01	-2.54814288E-01	-1.03012750E-01	-2.55332226E-01
-1.04930791E-01	-2.55749697E-01	-1.06773985E-01	-2.56071331E-01	-1.08541621E-01
-2.56301546E-01	-1.10233105E-01	-2.56403706E-01	-1.11833944E-01	-2.56444299E-01
-1.13319732E-01	-2.56425938E-01	-1.14780197E-01	-2.56351131E-01	-1.16185121E-01
-2.56222287E-01	-1.17534332E-01	-2.56016126E-01	-1.18797430E-01	-2.55773958E-01
-1.20019786E-01	-2.55497195E-01	-1.21201322E-01	-2.55187201E-01	-1.22341985E-01
-2.54845296E-01	-1.23441745E-01	-2.54458134E-01	-1.24481784E-01	-2.54049259E-01
-1.25490371E-01	-2.53619472E-01	-1.26467494E-01	-2.53149554E-01	-1.27413155E-01
-2.52700263E-01	-1.28327368E-01	-2.52204486E-01	-1.29199251E-01	-2.51694914E-01
-1.30045234E-01	-2.51172044E-01	-1.30865333E-01	-2.50636364E-01	-1.31659570E-01
-2.50088351E-01	-1.32427978E-01	-2.49524239E-01	-1.33144278E-01	-2.48950935E-01
-1.33878014E-01	-2.48368778E-01	-1.34369217E-01	-2.47778106E-01	-1.35237924E-01
-2.47179248E-01	-1.35864179E-01	-2.46569692E-01	-1.36433505E-01	-2.45954073E-01
-1.37102758E-01	-2.45332632E-01	-1.37681981E-01	-2.44725610E-01	-1.38241218E-01
-2.44073242E-01	-1.38780520E-01	-2.43435759E-01	-1.39299938E-01	-2.42793388E-01

$l=2$

$l=3$

-1.51242563E-02	-1.95367911E-04	-1.73795412E-02	-2.64365301E-04	-1.97698604E-02
-3.50164799E-04	-2.22889964E-02	-4.55138643E-04	-2.49373543E-02	-5.81770230E-04
-2.76918563E-02	-7.32648642E-04	-3.06545308E-02	-9.15788878E-04	-3.36795279E-02
-1.12472355E-03	-3.67655282E-02	-1.36766242E-03	-3.99111986E-02	-1.64073936E-03
-4.31151908E-02	-1.94816060E-03	-4.64939251E-02	-2.30385975E-03	-4.98663923E-02
-2.69385469E-03	-5.32342364E-02	-3.11921449E-03	-5.65988389E-02	-3.58102443E-03
-5.99613490E-02	-4.08038395E-03	-6.34715977E-02	-4.64026557E-03	-6.69036336E-02
-5.23075764E-03	-7.02615216E-02	-5.85121592E-03	-7.35489610E-02	-6.50104273E-03
-7.67693254E-02	-7.17968194E-03	-7.99490359E-02	-7.89126645E-03	-8.30554139E-02
-8.62860346E-03	-8.60915505E-02	-9.39096091E-03	-8.90642989E-02	-1.01776450E-02
-9.19642981E-02	-1.09879969E-02	-9.49011671E-02	-1.18455119E-02	-9.77285083E-02
-1.27144781E-02	-1.00450378E-01	-1.35932233E-02	-1.03070597E-01	-1.44801627E-02
-1.05592773E-01	-1.53737934E-02	-1.08050983E-01	-1.62851447E-02	-1.10401873E-01
-1.71899873E-02	-1.12649105E-01	-1.80957151E-02	-1.14796157E-01	-1.89978113E-02
-1.16846342E-01	-1.98948436E-02	-1.18810628E-01	-2.07882791E-02	-1.20680168E-01
-2.16726784E-02	-1.22457983E-01	-2.25467668E-02	-1.24146959E-01	-2.34093445E-02
-1.25749857E-01	-2.42592822E-02	-1.27263216E-01	-2.50990135E-02	-1.28698831E-01
-2.59132553E-02	-1.30059022E-01	-2.67191157E-02	-1.31346015E-01	-2.75097564E-02
-1.32561948E-01	-2.82843906E-02	-1.33696292E-01	-2.90365449E-02	-1.34770094E-01
-2.97740615E-02	-1.35785040E-01	-3.04963668E-02	-1.36742756E-01	-3.12029224E-02
-1.37644803E-01	-3.18932227E-02	-1.38478678E-01	-3.25598198E-02	-1.39267059E-01
-3.32126729E-02	-1.40011126E-01	-3.38514318E-02	-1.40712016E-01	-3.44757675E-02
-1.41370829E-01	-3.50853709E-02	-1.41976013E-01	-3.56731868E-02	-1.42547691E-01
-3.62490515E-02	-1.43086668E-01	-3.68127558E-02	-1.43593723E-01	-3.73641025E-02
-1.44069609E-01	-3.79029059E-02	-1.44505034E-01	-3.84232569E-02	-1.44915872E-01
-3.89335704E-02	-1.45302670E-01	-3.94337199E-02	-1.45665956E-01	-3.99235862E-02
-1.46006244E-01	-4.04030566E-02	-1.46324030E-01	-4.08720253E-02	-1.46619801E-01
-4.13303927E-02	-1.46894025E-01	-4.17780657E-02	-1.47147159E-01	-4.22149569E-02
-1.47379646E-01	-4.26409851E-02	0.	0.	
-1.78426823E-03	-2.71865443E-06	-2.12560430E-03	-3.95360888E-06	-2.53248495E-03
-5.74414023E-06	-3.00103931E-03	-8.24761311E-06	-3.52786732E-03	-1.16426690E-05
-4.10996165E-03	-1.61276075E-05	-4.71837097E-03	-2.16749273E-05	-5.39084744E-03
-2.88354083E-05	-6.12445639E-03	-3.79008068E-05	-6.91685036E-03	-4.91946381E-05
-7.76472952E-03	-6.30608252E-05	-8.66102835E-03	-7.97580099E-05	-9.61225639E-03
-9.98137156E-05	-1.06164362E-02	-1.23649140E-04	-1.16777467E-02	-1.51705821E-04
-1.27765049E-02	-1.84444780E-04	-1.39750039E-02	-2.23812437E-04	-1.51960659E-02
-2.68297282E-04	-1.64393135E-02	-3.18228870E-04	-1.77043803E-02	-3.73942806E-04
-1.89909106E-02	-4.35780616E-04	-2.02589854E-02	-5.02124873E-04	-2.15683474E-02
-5.76084009E-04	-2.29179197E-02	-6.58203828E-04	-2.43066803E-02	-7.49044440E-04
-2.57336576E-02	-8.49179814E-04	-2.72544127E-02	-9.63190896E-04	-2.87823899E-02
-1.08602099E-03	-3.03176987E-02	-1.21795469E-03	-3.18604271E-02	-1.35927935E-03
-3.34106429E-02	-1.51028498E-03	-3.49998696E-02	-1.67458697E-03	-3.65804465E-02
-1.84724056E-03	-3.81528814E-02	-2.02925760E-03	-3.97176427E-02	-2.22042830E-03
-4.12751631E-02	-2.42085102E-03	-4.28544913E-02	-2.63476631E-03	-4.44126120E-02
-2.85534888E-03	-4.59502821E-02	-3.08432689E-03	-4.74682137E-02	-3.32100932E-03
-4.89570776E-02	-3.56531562E-03	-5.04705655E-02	-3.82048000E-03	-5.19443741E-02
-4.08191734E-03	-5.33893830E-02	-4.34880311E-03	-5.48064283E-02	-4.62112103E-03
-5.61963056E-02	-4.89866256E-03	-5.75761224E-02	-5.18479442E-03	-5.89218671E-02
-5.47317415E-03	-6.02344413E-02	-5.76530801E-03	-6.15147681E-02	-6.06631299E-03
-6.27634940E-02	-6.35791634E-03	-6.39914457E-02	-6.65992858E-03	-6.51844214E-02
-6.96304365E-03	-6.63432727E-02	-7.26695736E-03	-6.74688182E-02	-7.57137714E-03
-6.85618455E-02	-7.87602153E-03	-6.96275345E-02	-8.18147347E-03	-7.06599314E-02
-8.48651523E-03	-7.16597923E-02	-8.79027194E-03	-7.26278462E-02	-9.09267963E-03
-7.35647971E-02	-9.39348475E-03	-7.44717486E-02	-9.69255615E-03	-7.53487163E-02
-9.98949361E-03	-7.61963395E-02	-1.02840707E-02	-7.70182364E-02	-1.05760700E-02
-7.78060050E-02	-1.08652831E-02	-7.85692243E-02	-1.11575101E-02	-7.93254545E-02
-1.14345590E-02	-8.00152386E-02	-1.17142463E-02	-8.06901024E-02	-1.19903957E-02
-8.13575559E-02	-1.22628386E-02	0.	0.	

$\ell=4$

$\ell=5$

```
*****  ALUMINUM (PERTURBATION EXPANSION)  ENERGY PHASE SHIFTS *****                                    7 DECEMBER 1972


T-FACTORS READ IN TO L=  3 FOR ATOMIC TYPE  1....  ALUMINUM *****

SUBROUTINE STACK FORMS INTERMEDIATE ARRAYS (LL) AND (C).
L=  3   LMAX4=  256  (L1,M1,L2,M2,L3,M3)=(  4  0  3  3  3  3)   NUMMAX=  498



*****FCC (100)-SURFACE

LATTICE CONSTANT IS   4.04000 ANGSTROMS,    INTERLAYER SPACING IN BULK IS   2.02000 ANGSTROMS

BASIS VECTOR IN SURFACE PLANE IS (, 0.00000,  0.00000)
( 1) CORE( 1)  SYM( 1)   D=(   0.00000   0.00000   0.00000)    A=(   2.02000   2.02000)    B=(   2.02000  -2.02000)
( 2) CORE( 1)  SYM( 1)   C=(   2.02000   0.00000   2.02000)    A=(   2.02000   2.02000)    B=(   2.02000  -2.02000)
( 3) CORE( 1)  SYM( 1)   C=(   0.00000   0.00000   4.04000)    A=(   2.02000   2.02000)    B=(   2.02000  -2.02000)

RECIPROCAL LATTICE, SURFACE NET VECTORS

                         ( 1)  RA=(   1.55524   1.55524)   RB=(   1.55524  -1.55524)
                         ( 2)  RA=(   1.55524   1.55524)   RB=(   1.55524  -1.55524)
                         ( 3)  RA=(   1.55524   1.55524)   RB=(   1.55524  -1.55524)
```

```
*****  ALUMINUM (PERTURBATION EXPANSION)  PENDRY PHASE SHIFTS *****                                    7 DECEMBER 1972


INCIDENT BEAM ANGLES ARE THETA=   0  PHI=   0
                                 *************

INCIDENT ELECTRON ENERGY FROM  40 TO  40 E.V.

MAXIMUM L-VALUE IS  3

LATTICE TEMPERATURE=  293
DEBYE TEMPERATURES =  426 AND  426

SUBROUTINE TABLE SETS PROPAGATOR CUTOFF LENGTH AT RMAX=  30.000
FINAL LAYERS OF SUMMATION ARE ( 1, 1)   MAXX(  1)=  348 = TOTAL NUMBER OF ATOMIC SITES WITHIN CUTOFF.
FINAL LAYERS OF SUMMATION ARE ( 2, 1)   MAXX(  2)=  648 = TOTAL NUMBER OF ATOMIC SITES WITHIN CUTOFF.
FINAL LAYERS OF SUMMATION ARE ( 3, 1)   MAXX(  3)=  897 = TOTAL NUMBER OF ATOMIC SITES WITHIN CUTOFF.
FINAL LAYERS OF SUMMATION ARE ( 1, 2)   MAXX(  4)= 1197 = TOTAL NUMBER OF ATOMIC SITES WITHIN CUTOFF.
FINAL LAYERS OF SUMMATION ARE ( 3, 2)   MAXX(  5)= 1413 = TOTAL NUMBER OF ATOMIC SITES WITHIN CUTOFF.
FINAL LAYERS OF SUMMATION ARE ( 1, 3)   MAXX(  6)= 1662 = TOTAL NUMBER OF ATOMIC SITES WITHIN CUTOFF.
FINAL LAYERS OF SUMMATION ARE ( 2, 3)   MAXX(  7)= 1878 = TOTAL NUMBER OF ATOMIC SITES WITHIN CUTOFF.

ARRAY DIMENSIONS REQUIRED (AND SUPPLIED) BY PROGRAM
                     C-   496  (  7532)
                    LL-   498  (  7532)
                    KX-   256  (  2401)
                  RMAG-  1878  (  9000)
                  XMAG-  1878  (  9000)
                  RVEC-  5634  ( 30000)
           G(COMPLEX)-   3072  ( 49000)


TIME FOR EXECUTION OF TABLE     2.374 TO     2.422
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
INCIDENT BEAM ENERGY IS  40 E.V.   $$$$$  ARGY=   E
RENORMALIZED QUASIPARTICLE ENERGY                  E=(  49.8866,   4.0855) E.V.
COMPLEX K-VECTOR                              KSTAR=( 3.62659,  .14825) INVERSE ANGSTROMS
QUASIPARTICLE RENORMALIZATION CONSTANT        ZNORM=(  .82761, -.29235)
INELASTIC DAMPING LENGTH                     LAMBDA=  6.7455 ANGSTROMS

                                                   Vx        Vy        Vz
PROPAGATION VECTORS IN CRYSTAL     V1=  0.00000  0.00000  3.62659
        TYPE( 1)  BEAM( 0, 0)      V2=  0.00000  0.00000 -3.62659     V3=  0.00000  0.00000  7.25317
        TYPE( 1)  BEAM( 0, 1)      V2=  1.55524 -1.55524 -2.88350     V3= -1.55524  1.55524  6.51008
        TYPE( 1)  BEAM( 1, 0)      V2=  1.55524  1.55524 -2.88350     V3= -1.55524 -1.55524  6.51008
        TYPE( 1)  BEAM( 1, 1)      V2=  3.11049  0.00000 -1.66467     V3= -3.11049  0.00000  5.49126

FILLB    LAYER  1   KTYP=  1

OBTAIN SUBPLANE PROPAGATOR BY ITERATION

NUMBER OF ITERATIONS=   8
        CORE( 1)   SYM( 1)     TAU( 0, 0)=  2.45357843E-01
        CORE( 1)   SYM( 1)     TAU( 0, 1)=  1.09488631E-01
        CORE( 1)   SYM( 1)     TAU( 1, 0)=  1.09488631E-01
        CORE( 1)   SYM( 1)     TAU( 1, 1)=  4.34035188E-02

NUMBER OF ITERATIONS=   0
        CORE( 1)   SYM( 1)     TAU( 0, 0)=  2.45346903E-01
        CORE( 1)   SYM( 1)     TAU( 0, 1)=  1.09494116E-01
        CORE( 1)   SYM( 1)     TAU( 1, 0)=  1.09494116E-01
        CORE( 1)   SYM( 1)     TAU( 1, 1)=  4.34094746E-02


ITERATION OF T-MATRIX TO ORDER  1

0 0 BEAM
    LAYER=  1  T=/   .146542/   -.047547  -.139036  TERM=  -.047547  -.139036  SUM=  -.047547  -.139036
    LAYER=  2  T=/   .080731/   -.026123  -.076388  TERM=   .079357   .014829  SUM=   .031810  -.124207
    LAYER=  3  T=/   .044354/   -.014352  -.041968  TERM=  -.028539   .033593  SUM=   .003271  -.090254

    ENERGY=  40 E.V.   INTENSITY=  4.021236E-03     LOGARITHM=  -2.39564
                                   ***********

0 1 BEAM
    LAYER=  1  T=/   .087618/   -.030972  -.081962  TERM=  -.030972  -.081962  SUM=  -.030972  -.081962
    LAYER=  2  T=/   .044563/   -.015753  -.041686  TERM=  -.009642   .040462  SUM=  -.040815  -.038499
    LAYER=  3  T=/   .022665/   -.008012  -.021202  TERM=   .016364  -.015681  SUM=  -.024450  -.054181

    ENERGY=  40 E.V.   INTENSITY=  1.741973E-03     LOGARITHM=  -2.75896
                                   ***********

1 0 BEAM
    LAYER=  1  T=/   .087618/   -.030972  -.081962  TERM=  -.030972  -.081962  SUM=  -.030972  -.081962
    LAYER=  2  T=/   .044563/   -.015753  -.041686  TERM=  -.009642   .040462  SUM=  -.040815  -.038499
    LAYER=  3  T=/   .022665/   -.008012  -.021202  TERM=   .016364  -.015681  SUM=  -.024450  -.054181

    ENERGY=  40 E.V.   INTENSITY=  1.741973E-03     LOGARITHM=  -2.75896
                                   ***********

1 1 BEAM
```

incident beam $\underset{\sim}{k}_i$

outgoing beams, $\underset{\sim}{k}_f$

$\underset{\sim}{k}_i - \underset{\sim}{k}_f$

τ obtained by iterative expansion.

τ obtained by matrix inversion.

```
LAYER=  1  T=/    .023683/    .010649  -.021154  TERM=   .010649  -.021154  SUM=   .010649  -.021154
LAYER=  2  T=/    .009805/    .004409  -.008758  TERM=  -.008291  -.005234  SUM=   .002359  -.026388
LAYER=  3  T=/    .004059/    .001825  -.003626  TERM=  -.002489   .003207  SUM=  -.000130  -.023181

ENERGY=  40 E.V.   INTENSITY=  2.64533CE-04     LOGARITHM=  -3.57686
                               ***********


ITERATION OF T-MATRIX TO ORDER  2

0 0 BEAM
LAYER=  1  T=/    .107327/    -.045228  -.097332  TERM=  -.045228  -.097332  SUM=  -.045228  -.097332
LAYER=  2  T=/    .049186/    -.013846  -.047197  TERM=   .047903   .011159  SUM=   .002676  -.086173
LAYER=  3  T=/    .030007/    -.005638  -.029472  TERM=  -.022334   .020040  SUM=  -.019658  -.066132

ENERGY=  40 E.V.   INTENSITY=  2.346700E-03     LOGARITHM=  -2.62954
                               ***********

0 1 BEAM
LAYER=  1  T=/    .070775/    -.011252  -.069875  TERM=  -.011252  -.069875  SUM=  -.011252  -.069875
LAYER=  2  T=/    .034556/     .003247  -.034504  TERM=  -.021733   .026995  SUM=  -.032986  -.042880
LAYER=  3  T=/    .018295/     .000193  -.018294  TERM=   .016906  -.006994  SUM=  -.016080  -.049873

ENERGY=  40 E.V.   INTENSITY=  1.353755E-03     LOGARITHM=  -2.86846
                               ***********

1 0 BEAM
LAYER=  1  T=/    .070775/    -.011252  -.069875  TERM=  -.011252  -.069875  SUM=  -.011252  -.069875
LAYER=  2  T=/    .034556/     .003247  -.034504  TERM=  -.021733   .026995  SUM=  -.032986  -.042880
LAYER=  3  T=/    .018295/     .000193  -.018294  TERM=   .016906  -.006994  SUM=  -.016080  -.049873

ENERGY=  40 E.V.   INTENSITY=  1.353755E-03     LOGARITHM=  -2.86846
                               ***********

1 1 BEAM
LAYER=  1  T=/    .052964/     .040966  -.033571  TERM=   .040966  -.033571  SUM=   .040966  -.033571
LAYER=  2  T=/    .026146/     .019559  -.017351  TERM=  -.015381  -.021144  SUM=   .025585  -.054715
LAYER=  3  T=/    .008204/     .005847  -.005755  TERM=  -.006845   .004523  SUM=   .018740  -.050192

ENERGY=  40 E.V.   INTENSITY=  1.415157E-03     LOGARITHM=  -2.84920
                               ***********


ITERATION OF T-MATRIX TO ORDER  3

0 0 BEAM
LAYER=  1  T=/    .099884/    -.042094  -.090581  TERM=  -.042094  -.090581  SUM=  -.042094  -.090581
LAYER=  2  T=/    .048017/    -.010831  -.046780  TERM=   .046057   .013580  SUM=   .003963  -.077002
LAYER=  3  T=/    .033505/    -.007820  -.032580  TERM=  -.023870   .023513  SUM=  -.019907  -.053489

ENERGY=  40 E.V.   INTENSITY=  1.605898E-03     LOGARITHM=  -2.79428
                               ***********

0 1 BEAM
LAYER=  1  T=/    .064597/    -.008122  -.064084  TERM=  -.008122  -.064084  SUM=  -.008122  -.064084
LAYER=  2  T=/    .033509/     .005812  -.033408  TERM=  -.023268   .024666  SUM=  -.031390  -.039418
LAYER=  3  T=/    .020372/    -.000695  -.020360  TERM=   .018458  -.006621  SUM=  -.012932  -.048039
```

```
      ENERGY=  40 E.V.   INTENSITY=  1.220178E-03     LOGARITHM=  -2.91358
                                      ************

1 0 BEAM
      LAYER=  1  T=/    .064597/    -.008122  -.064084  TERM=  -.008122  -.064084  SUM=  -.008122  -.064084
      LAYER=  2  T=/    .033909/     .005812  -.033408  TERM=  -.023268   .024666  SUM=  -.031290  -.039418
      LAYER=  3  T=/    .020372/    -.000695  -.020360  TERM=   .018458  -.008621  SUM=  -.012932  -.048039

      ENERGY=  40 E.V.   INTENSITY=  1.220178E-03     LOGARITHM=  -2.91358
                                      ************

1 1 BEAM
      LAYER=  1  T=/    .052141/     .043455  -.028817  TERM=   .043455  -.028817  SUM=   .043455  -.028817
      LAYER=  2  T=/    .026683/     .021160  -.016256  TERM=  -.014135  -.022631  SUM=   .029319  -.051448
      LAYER=  3  T=/    .009002/     .005954  -.006752  TERM=  -.007141   .005481  SUM=   .022178  -.045967

      ENERGY=  40 E.V.   INTENSITY=  1.284203E-03     LOGARITHM=  -2.89137
                                      ************


      ITERATION OF T-MATRIX TO ORDER  4

0 0 BEAM
      LAYER=  1  T=/    .100112/    -.042810  -.090497  TERM=  -.042810  -.090497  SUM=  -.042810  -.090497
      LAYER=  2  T=/    .047065/    -.010290  -.045926  TERM=   .045048   .013630  SUM=   .002238  -.076867
      LAYER=  3  T=/    .033797/    -.007758  -.032895  TERM=  -.024171   .023623  SUM=  -.021933  -.053244

      ENERGY=  40 E.V.   INTENSITY=  1.634807E-03     LOGARITHM=  -2.78653
                                      ************

0 1 BEAM
      LAYER=  1  T=/    .064022/    -.008642  -.063436  TERM=  -.008642  -.063436  SUM=  -.008642  -.063436
      LAYER=  2  T=/    .033587/     .006755  -.032900  TERM=  -.023779   .023720  SUM=  -.032420  -.039716
      LAYER=  3  T=/    .020521/    -.000598  -.020512  TERM=   .018636  -.008591  SUM=  -.013784  -.048307

      ENERGY=  40 E.V.   INTENSITY=  1.244122E-03     LOGARITHM=  -2.90514
                                      ************

1 0 BEAM
      LAYER=  1  T=/    .064022/    -.008642  -.063436  TERM=  -.008642  -.063436  SUM=  -.008642  -.063436
      LAYER=  2  T=/    .033587/     .006755  -.032900  TERM=  -.023779   .023720  SUM=  -.032420  -.039716
      LAYER=  3  T=/    .020521/    -.000598  -.020512  TERM=   .018636  -.008591  SUM=  -.013784  -.048307

      ENERGY=  40 E.V.   INTENSITY=  1.244122E-03     LOGARITHM=  -2.90514
                                      ************

1 1 BEAM
      LAYER=  1  T=/    .051255/     .043006  -.027884  TERM=   .043006  -.027884  SUM=   .043006  -.027884
      LAYER=  2  T=/    .027468/     .022324  -.015005  TERM=  -.013773  -.023766  SUM=   .029234  -.051650
      LAYER=  3  T=/    .009116/     .006055  -.006814  TERM=  -.007252   .005523  SUM=   .021681  -.046128

      ENERGY=  40 E.V.   INTENSITY=  1.287209E-03     LOGARITHM=  -2.89035
                                      ************


      ITERATION OF T-MATRIX TO ORDER  5

0 0 BEAM
```

```
    LAYER=  1  T=/   .099846/   -.043225  -.090005  TERM=  -.043225  -.090005  SUM=  -.043225  -.090005
    LAYER=  2  T=/   .046895/   -.010398  -.045732  TERM=   .044532   .013441  SUM=   .001707  -.076564
    LAYER=  3  T=/   .033814/   -.007325  -.033011  TERM=  -.024494   .023311  SUM=  -.022787  -.053252

    ENERGY=  40 E.V.   INTENSITY=  1.654065E-03     LOGARITHM=  -2.78145
                                   ***********

0 1 BEAM
    LAYER=  1  T=/   .063421/   -.008834  -.062802  TERM=  -.008834  -.062802  SUM=  -.008834  -.062802
    LAYER=  2  T=/   .033412/    .006698  -.032734  TERM=  -.023636   .023616  SUM=  -.032470  -.039187
    LAYER=  3  T=/   .020539/   -.000219  -.020538  TERM=   .018808  -.008253  SUM=  -.013662  -.047439

    ENERGY=  40 E.V.   INTENSITY=  1.201531E-03     LOGARITHM=  -2.92027
                                   ***********

1 0 BEAM
    LAYER=  1  T=/   .063421/   -.008834  -.062802  TERM=  -.008834  -.062802  SUM=  -.008834  -.062802
    LAYER=  2  T=/   .033412/    .006698  -.032734  TERM=  -.023636   .023616  SUM=  -.032470  -.039187
    LAYER=  3  T=/   .020539/   -.000219  -.020538  TERM=   .018808  -.008253  SUM=  -.013662  -.047439

    ENERGY=  40 E.V.   INTENSITY=  1.201531E-03     LOGARITHM=  -2.92027
                                   ***********

1 1 BEAM
    LAYER=  1  T=/   .050834/    .042934  -.027217  TERM=   .042934  -.027217  SUM=   .042934  -.027217
    LAYER=  2  T=/   .027581/    .022316  -.019855  TERM=  -.013635  -.023744  SUM=   .029295  -.050961
    LAYER=  3  T=/   .009282/    .006288  -.006828  TERM=  -.007483   .005491  SUM=   .021816  -.045471

    ENERGY=  40 E.V.   INTENSITY=  1.253962E-03     LOGARITHM=  -2.90172
                                   ***********

    TIME REQUIREMENTS FOR EXECUTION OF ENERGY LOOP
      BEGIN LOOP AT          2.423
      TO END OF FILLB        2.431
      TO END OF SUBPL        4.196
      TO END OF INV          4.305
      TO END OF GREEN       10.742
      DOUBLE DIFFRACTION    10.685
          NEXT ITERATION    11.031
          NEXT ITERATION    11.176
          NEXT ITERATION    11.321
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *


    TYPE( 1)  BEAM( 0, 0)   ENERGY-VS-INTENSITY
    *******

    E=  40--  1.65406638E-03     E=


    TYPE( 1)  BEAM( 0, 1)   ENERGY-VS-INTENSITY
    *******

    E=  40--  1.20153053E-03     E=
```

```
TYPE( 1)  BEAM( 1, 0)   ENERGY-VS-INTENSITY
*******

E=  40--  1.20153053E-03      E=
```

```
TYPE( 1)  BEAM( 1, 1)   ENERGY-VS-INTENSITY
*******

E=  40--  1.25396198E-03      E=
```

LEGAL NOTICE

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

TECHNICAL INFORMATION DIVISION
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720