ORNL-3429
UC-34 – Physics
TID-4500 (20th ed.)

A FORTRAN PROGRAM FOR CALCULATING THE SCATTERING OF NUCLEONS FROM A NONLOCAL OPTICAL POTENTIAL

F. G. Perey

OAK RIDGE NATIONAL LABORATORY
operated by
UNION CARBIDE CORPORATION
for the
U.S. ATOMIC ENERGY COMMISSION

Printed in USA. Price: $1.75 Available from the
Office of Technical Services
U. S. Department of Commerce
Washington 25, D. C.

ORNL-3429

Contract No. W-7405-eng-26

Neutron Physics Division

A FORTRAN PROGRAM FOR CALCULATING THE SCATTERING OF NUCLEONS FROM A NONLOCAL OPTICAL POTENTIAL

F. G. Perey

Date Issued

JUN 26 1963

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee
Operated by
UNION CARBIDE CORPORATION
for the
U. S. ATOMIC ENERGY COMMISSION

ABSTRACT

In this report is given the listing of a FORTRAN program for calculating the scattering of nucleons from a nonlocal optical potential. The mathematical formulation of the problem is given, together with the numerical methods used in the code. The input to the program is explained and a brief functional description of each subroutine of the code is given.

Table of Contents

INTRODUCTION

This version of the FORTRAN program for calculating the elastic scattering of nucleons by a nonlocal optical potential grew out of many attempts at the numerical solution of the problem. The author regrets that many of the idiosyncrasies of the earlier versions of the code have been retained (originally, to facilitate intercomparison of the codes), but he did not think it worthwhile to rewrite a code of this magnitude to eliminate some of the clumsiness in the programming.

The physical justifications, mathematical formulation, and comparison with neutron experimental data up to 24 Mev are given elsewhere.[1] The mathematical formulation of the problem is given again in this report to facilitate understanding of the limitations of the numerical solution and of the coding. Also given are a description of the input parameters and their restrictions, a brief description of each subroutine, and the flow of logic of the calculation. The listing of each subroutine and some test cases are presented as an appendix.

MATHEMATICAL FORMULATION

The integrodifferential Schrödinger equation is

$$\left(\frac{\hbar^2}{2M}\nabla^2 + E\right)\Psi(\underline{r}) = -\left[(U_{SP} + iW_{SP})\ S(r)\ \underline{L}\cdot\underline{\sigma} + V_C(r)\right]\Psi(\underline{r}) + \int V(\underline{r},\underline{r}')\ \Psi(\underline{r}')\ d\underline{r}', \qquad (1)$$

where M is the reduced mass of the incident particle and E its energy in the center-of-mass system of coordinates.

The kernel $V(\underline{r},\underline{r}')$ is taken to be separable and is written in the form

$$V(\underline{r},\underline{r}') = U\left[\frac{1}{2}(r + r')\right] H(|\underline{r} - \underline{r}'|). \qquad (2)$$

1. F. G. Perey and B. Buck, Nuclear Phys. 32, 353 (1962).

It would have been preferable to take $\frac{1}{2}\,|\underline{r} + \underline{r}'|$ for argument of U, but this complicates very much the calculation, and the simpler choice indicated in Eq. 2 was made. Hence $V(\underline{r},\underline{r}')$ depends on the variables r, r' and cosχ, where χ is the angle between the vectors $\underline{r}$ and $\underline{r}'$.

$H(\underline{r},\underline{r}')$ is chosen to be a gaussian function:

$$H(|\underline{r} - \underline{r}'|) = \frac{\exp\left[-\left(\frac{\underline{r} - \underline{r}'}{\beta} \right)^2 \right]}{\pi^{3/2}\,\beta^3}, \tag{3}$$

which is normalized so that

$$\int H(|\underline{r} - \underline{r}'|)\, d\underline{r}' = 1. \tag{4}$$

We define

$$p = \frac{1}{2}\,(r + r'). \tag{5}$$

The form chosen for U(p) in Eq. 2 is similar to those employed in local optical model calculations:

$$- U(p) = (V_S + iW_S)\; f_S(p) + iW_D\, f_D(p), \tag{6}$$

where

$$f_S(p) = \left[1 + \exp\left(\frac{p - R_S}{a_S} \right) \right]^{-1}, \qquad R_S = r_{OS} A^{1/3}, \tag{7}$$

$$f_D(p) = 4 \exp\left(\frac{p - R_D}{a_D} \right) \left[1 + \exp\left(\frac{p - R_D}{a_D} \right) \right]^{-2}, \quad R_D = r_{OD} A^{1/3}. \tag{8}$$

Here $f_S(p)$ is a Saxon form factor, while $f_D(p)$ gives a surface imaginary term of the Saxon derivative type which has unity at its maximum value. Note that the variable is p and not r, as in local calculations.

The spin-orbit function in Eq. 1 is

$$S(r) = \left(\frac{\hbar}{2Mc}\right)^2 \frac{1}{a_S r} \frac{\exp\left(\frac{r - R_S}{a_S}\right)}{\left[1 + \exp\left(\frac{r - R_S}{a_S}\right)\right]^2}, \tag{9}$$

where M is the nucleon mass. The function S(r) is often defined with M as the pi meson mass; in this case the coefficients U_{SP} and W_{SP} in Eq. 1 are smaller by a factor of 180.59.

The coulomb potential in Eq. 1 is assumed to come from a uniform charge distribution of radius R_S and is therefore

$$V_C(r) = \frac{Ze^2}{2R_S}\left(3 - \frac{r^2}{R_S^2}\right), \quad \text{for } r \leq R_S,$$

$$V_C(r) = \frac{Ze^2}{r}, \quad \text{for } r > R_S. \tag{10}$$

The functions $\underline{\Psi}(\underline{r})$, $\underline{\Psi}(r')$, and $V(\underline{r},\underline{r}')$ in Eq. 1 are expanded in partial waves:

$$\underline{\Psi}(\underline{r}) = \sum_{j\ell m} \frac{u_{j\ell}(r)}{r} C^{j\ell s}_{m\lambda\sigma} \left\{i^{\ell} Y^{\lambda}_{\ell}(\theta,\phi)\right\} X^{\sigma}_{s}, \tag{11}$$

$$\underline{\Psi}(\underline{r}') = \sum_{j'\ell'm'} \frac{u_{j'\ell'}(r')}{r'} C^{j'\ell's'}_{m'\lambda'\sigma'} \left\{i^{\ell'} Y^{\lambda'}_{\ell'}(\theta',\phi')\right\} X^{\sigma'}_{s'}, \tag{12}$$

$$V(\underline{r},\underline{r}') = \sum_{LM} \frac{g_L(r,r')}{rr'} \left\{i^L Y^M_L(\theta,\phi)\right\} \left\{i^L Y^M_L(\theta',\phi')\right\}^*. \tag{13}$$

Substitution of Eqs. 11-13 in Eq. 1 yields the following equation for the radial functions $u_{j\ell}(r)$:

$$\frac{\hbar^2}{2M}\left[\frac{d^2}{dr^2} - \frac{\ell(\ell+1)}{r^2}\right] u_{j\ell}(r) + E\, u_{j\ell}(r)$$

$$= -\left[(U_{SP} + iW_{SP})\, S(r)\, T(j\ell) + V_C(r)\right] u_{j\ell}(r) + \int_0^\infty g_\ell(r,r')\, u_{j\ell}(r')\, dr', \tag{14}$$

where

$$T(j\ell) = j(j+1) - \ell(\ell+1) - \frac{3}{4}. \tag{15}$$

From the expansion of $V(\underline{r},\underline{r}')$ we easily obtain

$$g_\ell(r,r') = 2\pi\, rr'\, U\left[\frac{1}{2}(r + r')\right]$$

$$\times \int_{-1}^{+1} \frac{\exp\left[-\left(\frac{\underline{r} - \underline{r}'}{\beta}\right)^2\right]}{\pi^{3/2}\,\beta^3}\, P_\ell(\cos\chi)\, d(\cos\chi). \tag{16}$$

Now let

$$g_\ell(r,r') = U\left[\frac{1}{2}(r + r')\right] h_\ell(r,r'). \tag{17}$$

Then, a straightforward evaluation of the integral in Eq. 16 gives

$$h_\ell(r,r') = \frac{1}{\pi^{1/2}\,\beta}\, \exp\left[-\left(\frac{r^2 + r'^2}{\beta^2}\right)\right] K_\ell(z), \tag{18}$$

where

$$K_\ell(z) = 2\ i^\ell\ z\ j_\ell(-iz), \tag{19}$$

$$z = 2\,\frac{rr'}{\beta}\ . \tag{20}$$

In Eq. 19 $j_\ell(x)$ is a spherical Bessel function. The following expression for $K_\ell(z)$ is analytically exact for all z:

$$K_\ell(z) = Q_\ell(z)\ \exp(z) + (-)^{\ell+1}\ Q_\ell(-z)\ \exp(-z), \tag{21}$$

where $Q_\ell(z)$ is a finite polynomial in z^{-1}, given by

$$Q_\ell(z) = \sum_{S=0}^{\ell} A_S(\ell)\ (-z)^{-S},\quad A_0(\ell) = 1, \tag{22}$$

$$A_S(\ell) = \left[\ \frac{\ell(\ell+1) - S(S+1)}{2S}\ \right] A_{S-1}(\ell). \tag{23}$$

Thus we obtain the exact expression

$$h_\ell(r,r') = \frac{1}{\pi^{1/2}\ \beta}\ \left[Q_\ell\ \left(2\,\frac{rr'}{\beta^2}\right)\exp\ \left\{-\left(\frac{r - r'}{\beta}\right)^2\right\}\right.$$

$$\left. + (-)^{\ell+1}\ Q_\ell\left(-\ 2\,\frac{rr'}{\beta^2}\right)\exp\ \left\{-\left(\frac{r + r'}{\beta}\right)^2\right\}\right]. \tag{24}$$

When z is small, the exact expression (21) is not useful for numerical work because of large cancellations between the component terms. Hence for small z it is better to expand $K_\ell(z)$ as a power series in z:

$$K_\ell(z) = C(\ell)\ z^{\ell+1} \sum_{S=0}^{\infty} B_S(\ell)\ z^S, \tag{25}$$

$$C(\ell) = \frac{2}{(2\ell + 1)!!}, \quad B_0(\ell) = 1, \quad B_1(\ell) = 0, \tag{26}$$

$$B_S(\ell) = \frac{B_{S-2}(\ell)}{S(S + 2\ell + 1)} . \tag{27}$$

The expansion (25) is used until more than a preassigned number of terms are required to obtain an accuracy of one part in 10^3. Then the exact form (24) is employed.

Various methods were tried for solving the integrodifferential Eq. 14; the one which proved the fastest was an iteration procedure. In order to start the iteration and to improve on the convergence of the method, use is made of the local WKB (Wentzel, Kramers, and Brillouin) approximation to the nonlocal potential. We are going to replace the integral of Eq. 1 by an approximate local expression involving only r. The integral in Eq. 1 can be written

$$I = \int U \left[\frac{1}{2}(\underline{r} + \underline{r}')\right] \frac{\exp\left[-\left(\frac{\underline{r} - \underline{r}'}{\beta}\right)^2\right]}{\pi^{3/2}\beta^3} \Psi(\underline{r}')\, d\underline{r}'. \tag{28}$$

Note that in order to derive the WKB approximation the argument of U is $\frac{1}{2}(\underline{r} + \underline{r}')$. The approximate expression we want is of the form

$$I \cong U_L(r)\, \Psi(\underline{r}). \tag{29}$$

Let

$$\underline{r}' - \underline{r} = \beta \underline{s}, \tag{30}$$

so that $\frac{1}{2}(\underline{r} + \underline{r}') = \underline{r} + \frac{1}{2}\beta\underline{s}$. Substitution in the integral gives

$$I = \int U \left(\underline{r} + \frac{1}{2}\beta\underline{s}\right) \frac{\exp(-s^2)}{\pi^{3/2}} \Psi(\underline{r} + \beta\underline{s})\, d\underline{s}. \tag{31}$$

Using the operator form of the Taylor expansion, the integral I can be written as an operator expression:

$$I = \left\{ \int \exp\left[\beta \underline{s} \cdot \left(\frac{1}{2} \nabla_1 + \nabla_2 \right) \right] \frac{\exp(-s^2)}{\pi^{3/2}} d\underline{s} \right\} U(\underline{r}) \, \Psi(\underline{r}), \qquad (32)$$

where

$$\nabla_1 \text{ operates only on } U(\underline{r}),\ \nabla_2 \text{ operates only on } \Psi(\underline{r}). \qquad (33)$$

Treating the expression $\frac{1}{2}\nabla_1 + \nabla_2$, for the moment, as an algebraic quantity, the integral over $d\underline{s}$ is easily evaluated and the following operator form is obtained for the integral:

$$I = \exp\left[\frac{\beta^2}{4} \left(\frac{1}{2} \nabla_1 + \nabla_2 \right)^2 \right] U(\underline{r}) \, \Psi(\underline{r}). \qquad (34)$$

Making the approximation that ∇_1 operating on U can be neglected, since in the interior of the nucleus U(r) is isotropic, then

$$I \cong U(\underline{r}) \exp\left(\frac{\beta^2}{4} \nabla^2 \right) \Psi(\underline{r}). \qquad (35)$$

By substituting the form in Eq. 29 for the integral in Eq. 1 and neglecting the spin-orbit potential,

$$\nabla^2 \Psi(\underline{r}) = -\frac{2M}{\hbar^2} \left[E - V_C(r) - U_L(r) \right] \Psi(\underline{r}). \qquad (36)$$

In Eq. 35 the approximation is made that the operator ∇^2 can be replaced by its numerical value as given in Eq. 36; the desired result is then obtained:

$$U_L(r) = U(r) \exp\left[-\frac{M\beta^2}{2\hbar^2} \left\{ E - V_C(r) - U_L(r) \right\} \right]. \qquad (37)$$

In the program Eq. 37 is solved by an iteration procedure at the first mesh point and calculated at all other mesh points by integrating the differential equation obtained by the differentiation of Eq. 37.

The iteration scheme will now be explained, but to simplify the notation, let

$$E^*_{j\ell} = E - \frac{\ell(\ell + 1)}{r^2} + (U_{SP} + iW_{SP})\ S(r)\ T(j\ell) - V_C(r); \tag{38}$$

then Eq. 14 is rewritten as

$$\frac{\hbar^2}{2M}\ \frac{d^2}{dr^2}\, u_{j\ell}(r) + E^*_{j\ell}(r)\ u_{j\ell}(r) = \int_0^{R_M} g_\ell(r,r')\ u_{j\ell}(r')\ dr'. \tag{39}$$

The upper limit R_M is the radius at which the kernel function becomes negligible; that is, $g_\ell(r,r') \cong 0$ for $r,r' \geq R_M$. Thus R_M is the radius for which the solutions u(r) are matched to the free-state wave functions in order to obtain the scattering matrix elements. The details of the computation of the free-state Coulomb functions and of the S-matrix elements are very similar to those of ref 2.

When the local WKB potential U_L is subtracted from both sides of Eq. 39 and the subscripts $j\ell$ are dropped for clarity, then

$$\frac{\hbar^2}{2M}\ \frac{d^2}{dr^2}\, u(r) + \left[E^*(r) - U_L(r)\right] u(r)$$
$$= \left[\int_0^{R_M} g_\ell(r,r')\ u(r')\ dr' - U_L(r)\ u(r)\right]. \tag{40}$$

If the suffix n denotes the nth-order approximation to the correct solution, the iteration procedure is written

2. B. Buck, R. N. Maddison, and P. E. Hodgson, Phil. Mag. 5, 1181 (1960).

$$\frac{\hbar^2}{2M} \frac{d^2}{dr^2} u_o(r) + \left[E^*(r) - U_L(r)\right] u_o(r) = 0, \tag{41}$$

$$\frac{h^2}{2M} \frac{d^2}{dr^2} u_n(r) + \left[E^*(r) - U_L(r)\right] u_n(r)$$

$$= \left[\int_o^{R_M} g_\ell(r,r') \; u_{n-1}(r') \; dr' - U_L(r) \; u_{n-1}(r) \right]. \tag{42}$$

Notice that Eq. 41 for $u_o(r)$ is a homogeneous local wave equation so that the scheme is easily started. The inhomogeneous iterations, Eq. 42, now yield a set of functions $u_n(r)$ which rapidly converge to the true solution $u(r)$. The iteration is continued until the logarithmic derivative at R_M, obtained from u_n, agrees to within a preassigned percentage with that obtained from u_{n-1}.

DESCRIPTION OF INPUT TO THE CODE

Since the nonlocal matrices $h_\ell(r,r')$ (Eq. 18) are only a function of β, the nonlocality range, they need be calculated only once for all computations using the same value of β and the same integration mesh size. Therefore provision is made in the code to generate and update a "library tape" of matrices $h_\ell(r,r')$. Since, frequently, several calculations are made successively while the code is in core and since searching the library tape for the appropriate $h_\ell(r,r')$ would be time-consuming, provision is made in the code to write the matrices $h_\ell(r,r')$ on a scratch tape to be used for several computations.

The code has therefore three functions: to generate nonlocal matrices to be added to a library tape of nonlocal matrices (on logical tape 3), to transfer nonlocal matrices from the library tape to a scratch tape (logical tape 1), and to do a nonlocal optical model calculation using the nonlocal matrices on logical tape 1.

Most of the input numbers to the code are in floating-point format (the number may be anywhere in the field indicated but must have a decimal point punched), except a few specifically indicated control integers, which are in fixed-point format (no decimal point, and the number must occupy the right-most position in the field).

The three functions of the program are controlled by a Master Control Card, which must precede every calculation. The system is described below:

MASTER CONTROL CARD

Column 1

If = 1: Function I

If = 2: Function II

If = 3: Function III

Function I: Calculation of nonlocal matrices

First Card: Master control card

Second Card: Matrix parameters (format 4F10.8,3I2)

Column 1-10: h_a, step length (in fermi) of last set of matrices on tape

11-20: β_a, nonlocality range (in fermi) of last set of matrices on tape (to start a new tape $h_a = \beta_a = 0$ or blank)

21-30: h_b, step length (in fermi) of new matrices to be added to tape

31-40: β_b, nonlocality range (in fermi) of new matrices to be added to tape

41-42: N_T (fixed point), number of partial waves (a maximum of 25 partial waves may be used)

43-44: N_B (fixed point), number of terms in asymptotic expansion of $h_\ell(r,r')$; may be left blank, in which case 5 terms will be used but up to 10 terms may be used

45-46: I_p, if = 01, a list of the matrices on the tape will be printed;

if = 00 or blank, no list will be printed

Function II: Transferring nonlocal matrices from logical tape 3 to logical tape 1

First Card: Master control card

Second Card: Matrix parameters (format 2F10.8)

Column 1-10: h_a, step length (in fermi) of matrices to be transferred

11-20: β_a, nonlocality range (in fermi) of matrices to be transferred

Note: The matrices must be on logical tape 3; otherwise, a tape error will occur

Function III: Nonlocal optical model calculation

First Card: Master control card

Second Card: Experimental parameters (format 5F10.8)

Column 1-10: ELAB, laboratory energy (in MeV) of incident particle

11-20: AMT, mass of target nuclei (in AMU)

21-30: AMI, mass of incident particle (in AMU)

31-40: ZT, charge of target nuclei (in electron units)

41-50: ZI, charge of incident particle (in electron units)

Third Card: Optical potential parameters I (format 7F10.8)

Column 1-10: US, Saxon potential real-well depth (in MeV)

11-20: RS, Saxon potential radius parameter (r_{OS} in fermi)

21-30: AS, Saxon potential diffuseness parameter (a_S in fermi)

31-40: WS, Saxon potential imaginary-well depth (in MeV)

41-50: WD, Derivative Saxon potential imaginary-well depth (in MeV)

51-60: RD, Derivative Saxon potential radius parameter (r_{OD} in fermi)

61-70: AD, Derivative Saxon potential diffuseness parameter (a_D in fermi)

Fourth Card: Optical potential parameters II (format 3F10.8)

Column 1-10: VSP, real-well depth of spin-orbit potential (in MeV)

11-20: WSP, imaginary-well depth of spin-orbit potential (in MeV)

21-30: BETA, nonlocality range (in fermi)

Notes: VSP positive gives the same convention as normal shell model; BETA is not used except for output purposes; the calculation is made with whatever matrices are on tape 1

Fifth Card: Numerical parameters (format 6F10.8,I2)

Column 1-10: H, integration step length (in fermi) -- must be the same as the step length of nonlocal matrices on logical tape 1

11-20: HC, Coulomb subroutine integration step length (in fermi) (may be different from H; a value of 0.1 is satisfactory)

21-30: C, cutoff parameter in iteration scheme (percentage difference in the logarithmic derivative between two successive iterations to accept convergence; the value 0.0005 is usually sufficient)

31-40: THETA1 (θ_{min})
41-50: THETA2 ($\Delta\theta$)
51-60: THETA3 (θ_{max})

} $\sigma(\theta)$ will be computed from θ_{min} to θ_{max} in $\Delta\theta$ steps; the θ's are in degrees; not more than 100 angles are allowed; $\theta_{min} \neq 0$ for charged particles

61-62: NC (fixed point), should be greater than the number of partial waves needed for the calculation (see limitations below)

Sixth Card: Derivative control card (format I1,8X,I1,9I3); numbers are all in fixed point on this card

Column 1: ISP, if = 0 or blank, no spin-orbit potential
if = 1 for spin 1/2 projectile and with a spin-orbit potential

10: NV = number of parameters with respect to which the derivatives of the S-matrix elements are desired; if none desired, leave blank

11-13: IPA(1)
14-16: IPA(2)
17-19: IPA(3)
⋮
up to IPA(9)

(11-13 through IPA(9): the IPA's are key numbers to indicate the parameter (see below); there should be NV entries in the IPA vector)

Key numbers for the derivatives:

1 ≡ US
2 ≡ RS
3 ≡ AS
4 ≡ WS
5 ≡ WD
6 ≡ RD
7 ≡ AD
8 ≡ USP
9 ≡ WSP

Example: if the derivatives with respect to RS and WD are wanted, NV = 2
IPA(1) = 2, IPA(2) = 5.

Limitations in Values of Input Parameters and Error Returns

Because of the various computational techniques used and because of the space allotted to the various arrays in the computer, the calculation is either unreliable or cannot proceed if the input parameters are not within a certain range. To the extent that those restrictions were known to the author at the time of programming, several checks have been incorporated in the code to test the calculation, and various messages are printed out when an error condition is detected. The following limitations are known to apply to this version of the code:

1. The matching radius R_M is taken as the largest of the two following quantities:

$$r_{OS}A^{1/3} + 7a_S, \quad r_{OD}A^{1/3} + 7a_D.$$

The limitation on step length for a given matching radius is

$$H \leqq \frac{R_M}{97}.$$

If this condition is not met, the code word DATA is printed.

2. For a given step length H there is a limitation on the size of β, the nonlocality range, since at most 30 off-diagonal elements are used in the matrix $g_\ell(r,r')$ (Eq. 17). For the calculation to proceed, the criterion $g_\ell(r, r + 30H) \leqq g_0(R_M, R_M)$ must be met for every partial wave. This criterion may be changed by altering the value of GRM and GIM in subroutine NEWMAT. When the above criterion is not met, the code word MATRIX is printed, and the calculation proceeds to the next case.

3. The value of NC should be greater than $kR_M - \eta$, where η is the Coulomb parameter. A value of $kR_M + 5$ is usually satisfactory. In the Coulomb subroutine if the Wronskian of the free-state equation is not 1, the message COULOMB is printed and the value of NC should be checked.

4. When the iteration scheme has not converged after 25 iterations, the message INTEGRATION is printed, indicating a breakdown of the method. The value of the potential parameters should be modified.

5. If the value of NC is too small, the message LMAX TOO SMALL is printed, and NC should be increased, 25 being the maximum value it may have.

BRIEF DESCRIPTION OF THE FUNCTION OF EACH SUBROUTINE

The program contains a short main program to control the flow of logic and a series of subroutines to perform the calculations. With one exception the subroutines have no arguments transmitted by the calling program. The quantities of interest are stored in common. The main program reads the Master Control Card and transfers control to subroutine ADTAPE for Function I, to subroutine GETHL2 for Function II, and to subroutine INPUT4 for Function III.

Function I: Calculation of Nonlocal Matrices

The nonlocal matrices $h_\ell(r,r')$ are symmetric with the largest elements along the diagonal, the magnitude of the elements decreasing rapidly as one moves off the diagonal. Because of storage limitation, only a band 30 elements wide on one side of the diagonal is stored in the computer. The length of the diagonal is 100 elements.

ADTAPE: Sets up the library tape and handles the writing of the matrices, which are obtained by calling MATRHL.

MATRHL: Controls the packing and unpacking of the matrices and calls the subroutine HL(I,J), which computes the elements by using Eqs. 21 or 25, depending on the size of $\frac{rr'}{\beta^2}$.

Function II: Transfer of Matrices

GETHL2: Searches the logical tape 3 for the appropriate matrices and transfers them to logical tape 1.

Function III: Nonlocal Optical Model Calculation

INPUT4: Reads the input cards described in "Description of Input to the Code."

SANGLE: Sets up a vector with the angles at which the cross sections are to be calculated.

PRINPU: Prints the input parameters.

DATAPR: Converts the various input parameters to the same unit system in the center-of-mass coordinate system. The matching radius is calculated and if there is not enough space, the code word DATA is printed.

VECTOR: Computes the various potentials at each mesh point, as well as their derivatives with respect to the parameters. The local WKB approximation is also calculated here.

COULOM: Calculates the Coulomb phase shifts and the regular and irregular Coulomb functions and their derivatives at the matching radius. If the Wronskian fails to be 1, the calculation stops and the code word COULOMB is printed.

At this point a loop is entered to calculate the S-matrix elements for each partial wave. The loop terminates when the S-matrix elements have reached their asymptotic value of 1. With $\ell = 0$ the loop is entered at subroutine NEWMAT.

NEWMAT: Sets up the matrix $g_\ell(r,r') = U\left[\frac{1}{2}(r + r')\right] h_\ell(r,r')$, reading the $h_\ell(r,r')$ matrices from logical tape 1. The matrix $g_\ell(r,r')$ is stored in the computer in the same manner as the matrix $h_\ell(r,r')$. If, because of the step length used and the nonlocality range, more than 30 off-diagonal elements are required, the code word MATRIX is printed.

INTEGR: The radial equation is solved by the iteration scheme given previously. After each iteration the logarithmic derivative of the wave function is calculated and the percentage deviation from the value obtained in the previous iteration is compared with the input number C. If the percentage deviation is less than C, the solution is accepted; if it is greater than C, another iteration is done, and so on until the criterion is met. If the criterion has not been met after 25 iterations, the code word INTEGRATION is printed.

SCMATR: The logarithmic derivative of the radial wave function is matched to the logarithmic derivative of the free-state wave function to obtain the scattering matrix element.

DERIVS: If the derivatives of the scattering matrix elements are required, they are calculated in this subroutine. The method of calculation is given elsewhere.[3]

CLEAN: If spin 1/2 projectile is used, the first pass in subroutine integration is done with $j = \ell + 1/2$, and control is returned to subroutine INTEGR for the case $j = \ell - 1/2$. The S-matrix elements are checked to determine whether they have reached their asymptotic value (Re $S_\ell > 0.999$, Im $S_\ell < 0.001$). If they have not, ℓ is increased by 1 and control is returned to NEWMAT for a new partial wave. If the scattering matrix elements have reached their asymptotic value, control is transferred to subroutine SIGMAS. Each time ℓ is increased in the above loop, it is verified that enough free-state Coulomb functions have been calculated; if not, the code word LMAX TOO SMALL is printed.

SIGMAS: Calculates the differential cross sections and the polarization at the angle assigned by SANGLE.

PRSIGM: Prints the differential cross section and the S-matrix elements (next to the S-matrix elements, under the column IT the number of iterations for each partial wave is given); if required, the derivatives of the S-matrix elements are printed.

---

3. F. G. J. Perey and B. Buck, Neutron Phys. Div. Ann. Progr. Rept. Sept. 1, 1961, ORNL-3193, p 250.

ACKNOWLEDGMENTS

It is with great pleasure that I acknowledge the collaboration of B. Buck in the formulation of the problem and the investigation of the numerical methods to solve the integrodifferential Schrödinger equation. I would like also to thank G. R. Satchler for many discussions and suggestions on the scattering of nucleons from nonlocal potentials.

APPENDIX

Listing of FORTRAN Subroutines

and

Test Cases for IBM-7090 and CDC-1604

```
*TYPE(FORTRAN)
      DIMENSION  F(200)
      COMMON  F
      EQUIVALENCE  (F(61),K0),(F(29),J3),(F(51),KS),
     1(F(47),IL),(F(45),LC),(F(63),I7)
    1 READ INPUT TAPE  10,2,K0,I7
    2 FORMAT  (I1)
      IF  (K0-1)  100,100,101
  100 CALL  ADTAPE
      GO  TO  1
  101 IF  (K0-2)  3,3,4
    3 CALL  GETHL2
      GO  TO  1
    4 CALL  INPUT4
      J3=0
      CALL  SANGLE
      CALL  PRINPU
      CALL  DATAPR
      IF  (LC)  5,5,12
    5 CALL  VECTOR
      CALL  COULOM
      IF  (LC-1)  6,6,14
    6 KS=0
      IL=0
    7 CALL  NEWMAT
      IF  (LC)  8,8,16
    8 CALL  INTEGR
      IF  (LC)  9,9,18
    9 CALL  SCMATR
      CALL  DERIVS
      CALL  CLEAN
      IF  (LC)  7,8,10
   10 IF  (LC-1)  11,11,20
   11 CALL  SIGMAS
      CALL  PRSIGM
      GO  TO  1
```

```
12 WRITE  OUTPUT  TAPE  9,13
   GO  TO  1
13 FORMAT  (5H0DATA)
14 WRITE  OUTPUT  TAPE  9,15
15 FORMAT(8H0COULOMB)
   GO  TO  1
16 WRITE  OUTPUT  TAPE  9,17
17 FORMAT(7H0MATRIX)
   GO  TO  1
18 WRITE  OUTPUT  TAPE  9,19
19 FORMAT(12H0INTEGRATION)
   GO  TO  1
20 WRITE  OUTPUT  TAPE  9,21
21 FORMAT(15H0LMAX TOO SMALL)
   GO  TO  1
   END
```

```
*TYPE(FORTRAN)
      SUBROUTINE  ADTAPE
      DIMENSION  D(15000),X(300),HLMAT(100,30)
      COMMON  D
      EQUIVALENCE  (D(15),H),(D(10),BETA),(D(17),M),
     1(D(16),NB),(D(51),IL),(D(201),X(1)),(D(2111),HLMAT)
      READ  INPUT  TAPE  10,1,H1,BETA1,H,BETA,M,NB,LIST
    1 FORMAT  (4F10.8,3I2)
      REWIND  3
      IF  (H1)  22,22,21
   21 IF  (LIST)  4,4,2
    2 WRITE OUTPUT TAPE 9,3
    3 FORMAT (36H1       NON-LOCALITY MATRICES ON TAPE/
     148H0       STEP LENGTH       BETA       NO.OF MATRICES)
    4 DO  11  I=1,300
      READ TAPE 3,H2,BETA2,M2
      IF  (LIST)  7,7,5
    5 WRITE OUTPUT TAPE 9,6,H2,BETA2,M2
    6 FORMAT  (1H0,F17.8,F16.8,I9)
    7 IF  (H2-H1)  10,8,10
    8 IF  (BETA1-BETA2)  10,9,10
    9 GO  TO  12
   10 DO  11  J=1,M2
   11 READ  TAPE  3
   12 DO  13  I=1,M2
   13 READ  TAPE  3
   22 WRITE  TAPE  3,H,BETA,M
      X(1)=H
      DO  14  I=2,300
   14 X(I)=X(I-1)+H
      IL=0
      IF  (NB)  15,15,16
   15 NB=5
   16 DO  17  I6=1,M
      CALL  MATRHL
      WRITE  TAPE  3,((HLMAT(I,J),I=1,100),J=1,30)
```

```
17 IL=IL+1
   IF  (LIST)  20,20,18
18 WRITE OUTPUT TAPE 9,19,H,BETA,M
19 FØRMAT  (4H0NEW,F14.8,F16.8,I9)
20 RETURN
   END
```

```
*TYPE(FORTRAN)
      SUBROUTINE  MATRHL
      DIMENSION  D(15000),X(300),EX(300),
     1HLMAT(100,30),SMZ(900),A(30),B(30)
      COMMON  D
      EQUIVALENCE  (D(64),AL),(D(51),IL),(D(10),BETA),(D(52),LO),
     1(D(16),NB),(D(53),FAC),(D(54),IZ1),(D(55),IZ2),
     2(D(59),J1),(D(58),IS),(D(60),HLR),(D(201),X(1)),
     3(D(1811),EX(1)),(D( 2111),HLMAT),(D(11561),SMZ(1)),
     4(D(12461),A(1)),(D(12491),B(1))
      AL=IL
      IF  (IL)  1,1,4
    1 B1=1./(1.7724539*BETA)
      DO  2  I=1,300
    2 EX(I)=0.
      DO  3  I=1,300
      B2=(X(I)/BETA)**2
      IF  (B2-80.)  3,3,4
    3 EX(I)=B1*EXPF(-B2)
    4 DO  5  I=1,100
      DO  5  J=1,30
    5 HLMAT(I,J)=0.
      DO  6  I=1,900
    6 SMZ(I)=0.
      DO  7  I=1,30
      A(I)=0.
    7 B(I)=0.
      DO  8  I=1,NB
      AI=I
    8 B(I)=1./((2.*AI)*(2.*AI+1.+2.*AL))
      IF  (IL)  100,100,101
  100 LO=1
      GO  TO  102
  101 LO=2
```

```
      DO  9  I=1,IL
      AI=I
    9 A(I)=(AL*(AL+1.)-AI*(AI-1.))/(2.*AI)
  102 FAC=2.
      BI=2.*AL+1.
   10 FAC=FAC/BI
      BI=BI-2.
      IF  (BI)  11,11,10
   11 IZ1=900
      IZ2=2500
      DO  14  I=1,100
      DO  13  J=I,100
      J1=J-I+1
      IF  (J1-30)  12,12,14
   12 IS=I+J
      CALL  HL(I,J)
      HLMAT(I,J1)=HLR
   13 CONTINUE
   14 CONTINUE
      RETURN
      END
```

```
*TYPE(FORTRAN)
      SUBROUTINE  HL(I,J)
      DIMENSION  D(200),X(750),FR(300),FI(560),EX(300),
     1GR(150,30),GI(150,30),IUP(150),ILO(150),JLO(150),
     2SMZ(900),A(30),B(30)
      COMMON  D,X,FR,FI,EX,GR,GI,IUP,ILO,JLO,SMZ,
     1A,B
      EQUIVALENCE  (D(10),BETA),(D(16),NB),(D(51),IL),
     1(D(52),LU),(D(53),FAC),(D(54),IZ1),(D(55),IZ2),
     2(D(58),IS),(D(59),J1),(D(60),HLR)
      IS=IS
      J4=J1-1
      IZ=I*J
      IF  (IZ-IZ1)  1,7,7
    1 IF  (SMZ(IZ))  2,2,6
    2 Z=2.*X(I)*X(J)/(BETA**2)
      Z2=Z**2
      BA=1.
      BB=1.
      DO  4  K=1,NB
      BB=BB*B(K)*Z2
      IF  (BA*.0001-BB)  4,4,3
    3 BA=BA+BB
      GO  TO  5
    4 BA=BA+BB
      IZ1=IZ
      GO  TO  8
    5 SMZ(IZ)=EXPF(-Z)*FAC*(Z**(IL+1))*BA
    6 IF  (J1-1)  21,21,22
   21 HLR=SMZ(IZ)/(1.7724539*BETA)
      RETURN
   22 HLR=EX(J4)*SMZ(IZ)
      RETURN
    7 IF  (IZ-IZ2)  8,17,17
    8 Z=-2.*X(I)*X(J)/(BETA**2)
      BA=1.
      BB=1.
      GO  TO  (11,9),LU
```

```
 9 DO   10   K=1,IL
   BB=BB*A(K)/Z
10 BA=BA+BB
11 IF   (J1-1)   23,23,24
23 HLR=BA/(1.7724539*RETA)
   GO   TO   25
24 HLR=EX(J4)*BA
25 Z=-Z
   BA=1.
   BB=1.
   GO   TO   (14,12),L0
12 DO   13   K=1,IL
   BB=BB*A(K)/Z
13 BA=BA+BB
14 BA=BA*EX(IS)
   IF   (HLR*.0001-BA)   16,16,15
15 IZ2=IZ
16 IW=IL/2
   IF   (2*IW-IL)   29,28,28
28 HLR=HLR-BA
   RETURN
29 HLR=HLR+BA
   RETURN
17 Z=-2.*X(I)*X(J)/(BFTA**2)
   BA=1.
   BB=1.
   GO   TO   (20,18),L0
18 DO   19   K=1,IL
   BB=BB*A(K)/Z
19 BA=BA+BB
20 IF   (J1-1)   26,26,27
26 HLR=BA/(1.7724539*RETA)
   RETURN
27 HLR=BA*EX(J4)
   RETURN
   END
```

```
*TYPE(FORTRAN)
      SUBROUTINE  GETHL2
      DIMENSION  F(12262),HLMAT(100,30)
      EQUIVALENCE  (F(32),BETA),(F(17),H),(F(12261),HLMAT)
      COMMON  F
      REWIND  3
      REWIND  1
      READ  INPUT  TAPE  10,6,H,BETA
    6 FORMAT  (2F10.8)
      DO  3  I=1,300
      READ TAPE 3,HI,BETAI,MI
      IF  (HI-H)  2,1,2
    1 IF  (BETAI-BETA)  2,4,2
    2 DO  3  J=1,MI
    3 READ  TAPE  3
    4 DO  5  I6=1,MI
      READ  TAPE  3,((HLMAT(I,J),I=1,100),J=1,30)
    5 WRITE  TAPE  1,((HLMAT(I,J),I=1,100),J=1,30)
      REWIND  1
      RETURN
      END
```

```
*TYPE(FORTRAN)
      SUBROUTINE  INPUT4
      DIMENSION  F(15265),IPA(10),DSRU(9,25),
     1DSIU(9,25),DSRD(9,25),DSID(9,25)
      COMMON  F
      EQUIVALENCE  (F(21),E),(F(22),AMT),(F(23),AMI),
     1(F(24),ZT),(F(25),ZI),(F(1),U),(F(2),RR),(F(3),AR),
     2(F(4),WS),(F(5),WD),(F(6),RI),(F(7),AI),(F(8),USP),
     3(F(9),WSP),(F(32),BE),(F(17),H),(F(18),HC),(F(55),TI),
     4(F(26),A1),(F(27),A2),(F(28),A3),(F(19),LM),(F(56),NV),
     5(F(20),ISP),(F(57),NDS),(F(62),IPR),(F(152A1),IPA),
     6(F(4661),DSRU),(F(4886),DSIU),(F(5111),DSRD),(F(5336),
     7DSID)
      READ INPUT TAPE 10,1,E,AMT,AMI,ZT,ZI,U,RR,
     1AR,WS,WD,RI,AI,USP,WSP,BE,H,HC,TI,A1,A2,A3,
     2LM,ISP,NV,(IPA(I),I=1,9)
    1 FORMAT  (5F10.8/7F10.8/3F10.8/6F10.8,I2/I1,8X,
     1I1,9I3)
      NDS=NV
      IPR=NV
      DO  2  I=1,9
      DO  2  J=1,25
      DSRU(I,J)=0.
      DSRD(I,J)=0.
      DSIU(I,J)=0.
    2 DSID(I,J)=0.
      RETURN
      END
```

```
*TYPE(FORTRAN)
      SURROUTINE  SANGLE
      DIMENSION  D(6000),ANG(100),ANGEXP(100)
      COMMON  D
      EQUIVALENCE  (D(26),A1),(D(27),A2),(D(28),A3),(D(5561),ANG),
     1(D(29),J3),(D(30),KM),(D(31),KMEXP),(D(5961),ANGEXP)
      IF  (J3-1)  1,4,4
    1 ANG(1)=A1
      DO  3  I=2,100
      ANG(I)=ANG(I-1)+A2
      IF  (ANG(I)-A3)  3,2,2
    2 KM=I
      GO  TO  6
    3 CONTINUE
      GO  TO  6
    4 KM=KMEXP
      DO  5  I=1,KM
    5 ANG(I)=ANGEXP(I)
    6 RETURN
      END
```

```
*TYPE(FORTRAN)
      SUBROUTINE PRINPU
      DIMENSION  F(150)
      COMMON  F
      EQUIVALENCE  (F(21),E),(F(1),UR),(F(4),WS),(F(2),RS),(F(3),AS),
     1(F(5),WD),(F(6),RD),(F(7),AD),(F(8),USP),(F(9),WSP),(F(32),BETA
     2),(F(23),AI),(F(22),AT),(F(25),ZI),(F(24),ZT),(F(17),H),(F(18),HC)
     3,(F(19),LM),(F(61),KO)
      IF  (KO-2)  1,1,3
    1 WRITE OUTPUT TAPE 9,2
    2 FORMAT(44H1      NON-LOCAL OPTICAL POTENTIAL CALCULATION)
      GO  TO  5
    3 WRITE OUTPUT TAPE 9,4
    4 FORMAT(39H1      SEARCH NON-LOCAL OPTICAL POTENTIAL)
    5 WRITE OUTPUT TAPE 9,6,E,UR,USP,H,AT,WS,WD,WSP,HC
    6 FORMAT(15H       PARAMETERS/17H       EXPERIMENTAL,5X,11HTHEORETICAL,
     155X,9HNUMERICAL/12H      ELAB  =,F7.3,8X,7HUS    =,F7.3,30X,
     27HUSP    =,F8.2,7X,4HH  =,F7.3/12H      MT    =,F7.3,8X,7HWS    =,
     3F7.3,8X,7HWD    =,F7.3,8X,7HWSP   =,F8.2,7X,4HHC =,F7.3)
      WRITE OUTPUT TAPE 9,7,ZT,RS,RD,BETA,LM,AI,AS,AD,ZI
    7 FORMAT(12H      ZT    =,F7.3,8X,7HRS     =,F7.3,8X,7HRD     =,F7.3,
     18X,7HBETA  =,F8.2,7X,4HLM =,I4/12H      MI    =,F7.3,
     28X,7HAS    =,F7.3,8X,7HAD    =,F7.3/12H      ZI    =,F7.3)
      RETURN
      END
```

```
*TYPE(FORTRAN)
      SUBROUTINE  DATAPR
      DIMENSION  F(150),X(200)
      COMMON  F,X
      EQUIVALENCE  (F(23),AMI),(F(22),AMT),(F(33),B),
     1(F(11),RR),(F(2),R0R),(F(14),RI),(F(6),R0I),(F(35),AMSP),
     2(F(36),AMSS),(F(37),ES),(F(21),ELAB),(F(39),ETA),(F(25),ZI),
     3(F(24),ZT),(F(40),EKS),(F(41),EK),(F(10),U),(F(1),UL),
     4(F(12),WS),(F(4),WSL),(F(13),WD),(F(5),WDL),(F(15),VSP),
     5(F(8),VSPL),(F(16),WSP),(F(9),WSPL),(F(3),AR),(F(7),AI),
     6(F(42),RMM),(F(17),H),(F(43),N),(F(44),NN),(F(45),LC)
      IF  (AMI-4.)  1,1,2
    1 B=AMT**(1./3.)
      GO  TO  3
    2 B=AMT**(1./3.)+AMI**(1./3.)
    3 RR=R0R*B
      RI=R0I*B
      AMSP=AMI*AMT/(AMI+AMT)
      ES=ELAB*AMT/(AMI+AMT)
      ETA=0.157454*ZI*ZT*SQRTF(AMSP/ES)
      AMSS=0.04783258*AMSP
      EKS=AMSS*ES
      EK=SQRTF(EKS)
      U=UL*AMSS
      WS=WSL*AMSS
      WD=WDL*AMSS
      WSP=WSPL*AMSS*.011057
      VSP=VSPL*AMSS*.011057
      RRM=RR+7.*AR
      RIM=RI+7.*AI
      IF  (RRM-RIM)  4,4,5
    4 RMM=RIM
      GO  TO  6
    5 RMM=RRM
    6 N=RMM/H
      IF  (N-97)  8,8,7
    7 LC=1
      GO  TO  10
```

```
    8 LC=0
      AWN=N
      RMM=AWN*H
      NN=2*N
      X(1)=H
      DO 9 I=2,200
    9 X(I)=X(I-1)+H
   10 RETURN
      END
```

```
*TYPE(FORTRAN)
      SUBROUTINE  VECTOR
      DIMENSION  F(4000),BT(200),X(200).AT(200),FRV(200),
     1FR(200),FRR(200),FRA(200),FI(200),FISR(200),FISA(200),
     2VRSP(100),VSPV(100),VSPR(100),VSPA(100),WRSP(100).WSPR(100),
     3WSPA(100),VC(200),VCR(200).FIDW(200),FIDR(200),
     4FIDA(200),UL(200),WL(200)
      COMMON  F
      EQUIVALENCE  (F(3711),BT),(F(151),X),(F(3911),AT),
     1(F(1711),FRV),(F(1511),FR),(F(1911),FRR),(F(2111),FRA),
     2(F(2311),FI),(F(2511).FISR),(F(2711).FISA),(F(811),VRSP),
     3(F(911),VSPV),(F(1011),VSPR),(F(1111),VSPA),(F(1211),WRSP),
     4(F(1311),WSPR),(F(1411),WSPA),(F(611),VC),(F(15871),VCR),
     5(F(2911),FIDW),(F(3111),FIDR),(F(3311),FIDA),(F(63),I7),(F(40),E),
     6(F(43),N),(F(44),NN),(F(11),RR),(F(3),AR),(F(10),V),(F(33),B),
     7(F(12),WS),(F(20),ISP),(F(15),VSP),(F(16),WSP),(F(39),ETA),
     8(F(41),EK),(F(17),H),(F(13),WD),(F(14),RI),(F(7),AI),
     9(F(32),BET),(F(15471),UL),(F(15671),WL)
      N=N
      NN=NN
      B1=2.*RR
      B2=2.*AR
      DO  4  I=1,NN
      BT(I)=(X(I)-B1)/B2
      IF  (BT(I)-86.)  2,1,1
    1 AT(I)=1.0E37
      GO  TO  3
    2 AT(I)=EXPF(BT(I))
    3 FRV(I)=1./(1.+AT(I))
      FR(I)=FRV(I)*V
      FRR(I)=FRV(I)*FR(I)*AT(I)*B/AR
      FRA(I)=FRR(I)*BT(I)/B
      FI(I)=FRV(I)*WS
      FISR(I)=FRR(I)*WS/V
    4 FISA(I)=FRA(I)*WS/V
      IF  (ISP-1)  18,5,5
```

```
 18 DØ  19  I=1,100
    VSPV(I)=0.
    VRSP(I)=0.
    VSPR(I)=0.
    VSPA(I)=0.
    WRSP(I)=0.
    WSPR(I)=0.
 19 WSPA(I)=0.
    GØ  TØ  7
  5 DØ  6  I=1,N
    VSPV(I)=AT(2*I)*FRV(2*I)*FRV(2*I)/(AR*X(I))
    VRSP(I)=VSPV(I)*VSP
    VSPR(I)=VRSP(I)*B*FRV(2*I)*(AT(2*I)-1.)/AR
    VSPA(I)=VSPR(I)*BT(2*I)/B-VRSP(I)/AR
    WRSP(I)=VRSP(I)*WSP/VSP
    WSPR(I)=VSPR(I)*WSP/VSP
  6 WSPA(I)=VSPA(I)*WSP/VSP
  7 IF  (ETA)  20,20,8
 20 DØ  21  I=1,200
    VC=0.
 21 VCR=0.
    GØ  TØ  11
  8 B1=ETA*EK/RR
    B2=3.*B*B1/RR
    I1=2.*RR/H
    DØ  9  I=1,I1
    B3=.25*(X(I)/RR)**2
    VC(I)=B1*(3.-B3)
  9 VCR(I)=B2*(B3-1.)
    B1=ETA*EK*4.
    I1=I1+1
    N3=2*(N+3)
    DØ  10  I=I1,N3
    VC(I)=B1/X(I)
 10 VCR(I)=0.
 11 IF  (WD)  22,22,12
```

```
22 DØ   23   I=1,200
   FIDW(I)=0.
   FIDR(I)=0.
23 FIDA(I)=0.
   GØ   TØ   17
12 B1=2.*RI
   B2=2.*AI
   DØ   16   I=1,NN
   BT(I)=(X(I)-B1)/B2
   IF   (BT(I)-88.)   14,13,13
13 AT(I)=1.0E37
   GØ   TØ   15
14 AT(I)=EXPF(BT(I))
15 FIDW(I)=4.*AT(I)/((AT(I)+1.)**2)
   FI(I)=FI(I)+FIDW(I)*WD
   FIDR(I)=FIDW(I)*WD*B*(AT(I)-1.)/(AI     *(AT(I)+1.))
16 FIDA(I)=FIDR(I)*BT(I)/B
17 IF   (I7-1)   25,31,31
25 BE=(BET/2.)**2
   UL(I)=FR(I)/2.
   WL(I)=FI(I)/2.
26 EX=EXPF(-BE*(E+UL(I)-VC(I)))
   CAS=CØSF(BE*WL(I))
   SAN=SINF(BE*WL(I))
   FRN=EX*(FR(I)*CAS+FI(I)*SAN)
   FIN=EX*(FI(I)*CAS-FR(I)*SAN)
   IF   (ABSF((FRN-UL(I))/UL(I))-.00005)   27,27,28
27 IF   (ABSF((FIN-WL(I))/WL(I))-.00005)   29,29,28
28 UL(I)=FRN
   WL(I)=FIN
   GØ   TØ   26
29 DØ   30   I=1,NN
   EX=EXPF(-BE*(E+UL(I)-VC(I)))
   DI=BE*WL(I)
   CAS=CØSF(DI)
   SAN=SINF(DI)
   DUN=FR(I+1)-FR(I)
```

```
      DWN=FI(I+1)-FI(I)
      ANR=EX*(DUN*CAS+DWN*SAN)+UL(I)*BE*(VC(I+1)-VC(I))
      ANI=EX*(DWN*CAS-DUN*SAN)+WL(I)*BE*(VC(I+1)-VC(I))
      DR=1.+BE*UL(I)
      D2=DR**2+DI**2
      DUL=(DR*ANR+DI*ANI)/D2
      DWL=(DR*ANI-DI*ANR)/D2
      UL(I+1)=UL(I)+DUL
   30 WL(I+1)=WL(I)+DWL
   31 RETURN
      END
```

```
*TYPE(FORTRAN)
      SUBROUTINE  COULOM
      DIMENSION  AB(350),SIGMA(51),F(56),FP(51),
     1G(51),GP(51)
      COMMON  AB,SIGMA,F,FP,G,GP
      EQUIVALENCE  (AB(19),LMAX),(AB(18),H),
     1(AB(39),HETA),(AB(40),AKS),(AB(41),AK),
     2(AB(45),J),(AB(42),RM),(AB(46),RA)
      J=0
   23 HETAS=HETA**2
      RA=RM
      N=0
      AL=50.
      L=51
      C1=AL+1.
      PHI=SQRTF(C1**2+HETAS)
      CHI=ATANF(HETA/C1)
      SIGMA(L)=(AL+.5)*CHI+HETA*LOGF(PHI)-HETA-SINF(CHI)/(12.*PHI)+
     1SINF(3.*CHI)/(360.*PHI**3.)-SINF(5.*CHI)/(1260.*PHI**5.)+
     2SINF(7.*CHI)/(1680.*PHI**7.)-SINF(9.*CHI)/(1188.*PHI**9.)
      SIGMA(L-1)=SIGMA(L)-ATANF(HETA/AL)
      AL=AL-1.
      L=L-1
      IF(L-2)2,1,1
    2 I=0
      R=RA+H
    9 C1=AK*R
      C2=2.*C1
      SS=1.
      S1=SS
      TS=0.
      T1=TS
      SL=0.
      S2=SL
      TL=1.-HETA/C1
```

```
      T2=TL
      AN=0.
      N1=20
    3 C3=AN+1.
      C4=C3*C2
      S4=S1
      C5=(2.*AN+1.)*HETA/C4
      C6=(HETAS-AN*C3)/C4
      S1=C5*S1-C6*T1
      T1=C5*T1+C6*S4
      S3=S2
      S2=C5*S2-C6*T2-S1/C1
      T2=C5*T2+C6*S3-T1/C1
      AN=C3
      N1=N1-1
      SS=SS+S1
      TS=TS+T1
      SL=SL+S2
      TL=TL+T2
      IF(N1)4,4,3
    4 W=ABSF(SS*TL-SL*TS-1.)
      IF(W-.0001)6,6,5
    5 RA=RA+20.*H
      N=N+10
      GØ  TØ  2
    6 THETA=C1-HETA*LØGF(C2)+SIGMA(1)
      C7=CØSF(THETA)
      C8=SINF(THETA)
      G(1)=SS*C7-TS*C8
      GP(1)=AK*(SL*C7-TL*C8)
      IF(I)8,7,8
    7 G1=G(1)
      I=1
      R=RA
      G3=G1
      GØ  TØ  9
```

```
 8 IF(RM-RA)10,11,10
10 C3=2.*HETA*AK
   C4=10.*H**2/12.
   C5=C4/10.
   G7=G(1)
   M=N
   R=RA+H
   AK1=AKS-C3/R
   R=RA
   AK2=AKS-C3/R
   AK4=AK2*G7
14 IF(M)13,13,12
12 R=R-H
   AK3=AKS-C3/R
   G2=((2.-C4*AK2)*G7-(1.+C5*AK1)*G1)/(1.+C5*AK3)
   R=R-H
   AK1=AKS-C3/R
   G1=((2.-C4*AK3)*G2-(1.+C5*AK2)*G7)/(1.+C5*AK1)
   AK4=AK4+4.*G2*AK3+2.*G1*AK1
   G7=G1
   AK2=AK1
   G1=G2
   AK1=AK3
   M=M-1
   GO  TO  14
13 G(1)=G7
   GP(1)=H*(AK4-G7*AK2)/3.+GP(1)
11 L=LMAX+4
   C5=AK*RM
   F(L+2)=0.
   F(L+1)=10.0E-36
   AL=L
15 C3=AL*2.+1.
   C4=AL+1.
   F(L)=AL/SQRTF(AL**2+HETAS)*((C3*HETA/
  1(AL*C4)+C3/C5)*F(L+1)-SQRTF(C4**2+HETAS)/
```

```
    2C4*F(L+2))
      AL=AL-1.
      L=L-1
      IF(L)16,16,15
   16 FP(1)=AK*((HETA+1./C5)*F(1)-SQRTF(HETAS+1.)*F(2))
      ALPHA=AK/(FP(1)*G(1)-F(1)*GP(1))
      FP(1)=ALPHA*FP(1)
      L1=LMAX+1
      DO  17  L=1,L1
   17 F(L)=ALPHA*F(L)
      AL=0.
      DO  18  L=1,L1
      AL=AL+1.
      C3=HETA/AL+AL/C5
      C4=SQRTF(AL**2+HETAS)/AL
      FP(L+1)=AK*(C4*F(L)-C3*F(L+1))
      G(L+1)=C3*G(L)/C4-GP(L)/(C4*AK)
      GP(L+1)=AK*(C4*G(L)-C3*G(L+1))
      W=ABSF((FP(L)*G(L)-F(L)*GP(L))/AK-1.)
      IF(W-.0001)18,18,20
   18 CONTINUE
      GO  TO  21
   20 G1=G3
      J=J+1
      IF(J-2)6,21,21
   21 RETURN
      END
```

```
*TYPE(FORTRAN)
      SUBROUTINE  NEWMAT
      DIMENSION  F(100),HLMAT(100,30),FI(200),
     1GR(100,30),GI(100,30),IUP(100),ILO(100),FR(200)
      COMMON  F
      EQUIVALENCE  (F(43),N),(F(44),NN),(F(49),AL),
     1(F(47),IL),(F(45),IC),(F(1226),HLMAT),(F(6261),GR),
     2(F(9261),GI),(F(3511),IUP),(F(3611),ILO),(F(1511),FR),
     3(F(2311),FI)
      NN=NN
      N=N
      AL=IL
      IF  (AL)  1,1,2
    1 REWIND  1
      LO=1
      GO  TO  26
    2 LO=2
   26 READ  TAPE  1,((HLMAT(I,J),I=1,100),J=1,30)
    3 DO  4  I=1,100
      DO  4  J=1,30
      GR(I,J)=0.
    4 GI(I,J)=0.
      DO  5  I=1,100
      IUP(I)=0
    5 ILO(I)=0
      GO  TO  (6,7),LO
    6 GRM=FR(NN)*HLMAT(N,1)
      GIM=FI(NN)*HLMAT(N,1)
    7 DO  8  I=1,10
      DO  8  J=I,10
      J1=J-I+1
      IS=J+I
      GR(I,J1)=FR(IS)*HLMAT(I,J1)
    8 GI(I,J1)=FI(IS)*HLMAT(I,J1)
      DO  14  I=1,11
      DO  13  J=11,N
      J1=J-I+1
      IF  (J1-30)  10,10,9
```

```
 9 LC=J1
   GØ  TØ  25
10 IS=J+I
   GR(I,J1)=FR(IS)*HLMAT(I,J1)
   GI(I,J1)=FI(IS)*HLMAT(I,J1)
   IF  (GR(I,J1)-GRM)  11,13,13
11 IF  (GI(I,J1)-GIM)  12,13,13
12 IUP(I)=J
   GØ  TØ  14
13 CØNTINUE
14 CØNTINUE
   DØ  20  I=12,N
   DØ  19  J=I,N
   J1=J-I+1
   IF  (J1-30)  16,16,15
15 LC=J1
   GØ  TØ  25
16 IS=I+J
   GR(I,J1)=FR(IS)*HLMAT(I,J1)
   GI(I,J1)=FI(IS)*HLMAT(I,J1)
   IF  (GR(I,J1)-GRM)  17,19,19
17 IF  (GI(I,J1)-GIM)  18,19,19
18 IUP(I)=J
   GØ  TØ  20
19 CØNTINUE
20 CØNTINUE
   K0=IUP(1)
   DØ  21  I=1,K0
21 ILØ(I)=1
   DØ  24  I=2,N
   K1=IUP(I)-IUP(I-1)
   IF  (K1)  24,24,22
22 K2=K0+1
   K3=K0+K1
   DØ  23  J=K2,K3
23 ILØ(J)=I
```

```
      K0=K3
   24 CONTINUE
      LC=0
   25 RETURN
      END
```

```
*TYPE(FØRTRAN)
      SUBRØUTINE  INTEGR
      DIMENSIØN  F(9265),UR(100),UI(100),FRH(100),
     1FIH(100),PR(100),PI(100),QRØ(100),QIØ(100),ILA(100),
     2IUP(100),GR(100,30),GI(100,30),X(200),VC(100),IU(25),
     3ID(25),VRSP(100),WRSP(100),
     4FRHL(100),FIHL(100),UL(200),WL(200)
      CØMMØN  F
      EQUIVALENCE  (F(48),IL1),(F(49),AL),(F(47),IL),
     1(F(43),N),(F(50),NI),(F(17),H),(F(51),KSP),(F(40),EK2),
     2(F(52),DERR),(F(53),DERI).(F(54),DERN),(F(55),TIAF),
     3(F(45),LC),(F(4311),UR),(F(4411),UI),(F(3711),FRH),
     4(F(3811),FIH),(F(3911),PR),(F(4011),PI),(F(4111).QRØ),
     5(F(4211),QIØ),(F(3611),ILA),(F(6261),GR),(F(9261),GI),
     6(F(3511),IUP),(F(151),X),(F(611),VC),(F(811),VRSP),
     7(F(1211),WRSP).(F(4511),IU),(F(4536),ID)
     8,(F(15271),FRHL),(F(15371),FIHL),(F(15471),UL),
     9(F(15671),WL),(F(63),I7)
      IL1=IL+1
      AL=IL
      N=N
      NI=0
      N1=N+1
      N2=N+2
      N3=N+3
      N4=N-1
      N5=N-2
      N6=N-3
      H2=H**2
      H3=H2*H
      IF  (KSP)  1,1,3
    1 DØ  2  I=1,100
      UR(I)=0.
    2 UI(I)=0.
    3 DØ  4  I=1,100
      FRH(I)=0.
```

```
      FIH(I)=0.
      PR(I)=0.
      PI(I)=0.
      QRØ(I)=0.
    4 QIØ(I)=0.
      IF  (I7-1)  41,40,40
   40 DØ  7  I=1,N
      K1=ILØ(I)
      DØ  5  J=K1,I
      J1=I-J+1
      QRØ(I)=QRØ(I)+GR(J,J1)
    5 QIØ(I)=QIØ(I)+GI(J,J1)
      K2=IUP(I)
      K1=I+1
      DØ  6  J=K1,K2
      J1=J-I+1
      QRØ(I)=QRØ(I)+GR(I,J1)
    6 QIØ(I)=QIØ(I)+GI(I,J1)
      QRØ(I)=QRØ(I)*H3
    7 QIØ(I)=QIØ(I)*H3
      GØ  TØ  43
   41 DØ  42  I=1,N
      QRØ(I)=UL(2*I)*H2
   42 QIØ(I)=WL(2*I)*H2
   43 IF  (KSP)  8,8,11
    8 DØ  9  I=1,N
      FRH(I)=QRØ(I)+H2*(EK2-AL*(AL+1.)/(X(I)**2)-VC(2*I)
     1+AL*VRSP(I))
    9 FIH(I)=QIØ(I)+H2*AL*WRSP(I)
      DØ  10  I=N1,N3
   10 FRH(I)=H2*(EK2-AL*(AL+1.)/(X(I)**2)-VC(2*I))
      GØ  TØ  14
   11 DØ  12  I=1,N
      FRH(I)=QRØ(I)+H2*(EK2-AL*(AL+1.)/(X(I)**2)-VC(2*I)
     1-(AL+1.)*VRSP(I))
   12 FIH(I)=QIØ(I)-H2*(AL+1.)*WRSP(I)
      DØ  13  I=N1,N3
```

```
13 FRH(I)=H2*(EK2-AL*(AL+1.)/(X(I)**2)-VC(2*I))
   IF  (17-I)  14,19,19
14 UR(1)=H**(IL+1)
   UR(2)=(2.-FRH(1))*UR(1)
   UI(2)=-FIH(1)*UR(1)
   DO  15  I=2,4
   BLA=2.-FRH(I)
   UR(I+1)=BLA*UR(I)+FIH(I)*UI(I)-UR(I-1)
15 UI(I+1)=BLA*UI(I)-FIH(I)*UR(I)-UI(I-1)
   DO  18  I=5,N2
   B3R=12.+FRH(I+1)
   B1R=24.-10.*FRH(I)
   B1I=-10.*FIH(I)
   B2R=12.+FRH(I-1)
   CR=B1R*UR(I)-B1I*UI(I)-B2R*UR(I-1)+FIH(I-1)*UI(I-1)
   CI=B1I*UR(I)+B1R*UI(I)-FIH(I-1)*UR(I-1)-B2R*UI(I-1)
   B3N=B3R**2+FIH(I+1)**2
   UR(I+1)=(CR*B3R+CI*FIH(I+1))/B3N
18 UI(I+1)=(CI*B3R-CR*FIH(I+1))/B3N
19 DERNM=1000000.
20 URP=(UR(N3)-UR(N6)+9.*(UR(N5)-UR(N2))+45.*(UR(N1)
  1-UR(N4)))/(60.*H)
   UIP=(UI(N3)-UI(N6)+9.*(UI(N5)-UI(N2))+45.*(UI(N1)
  1-UI(N4)))/(60.*H)
   U2=UR(N)**2+UI(N)**2
   DERR=(URP*UR(N)+UIP*UI(N))/U2
   DERI=(UIP*UR(N)-URP*UI(N))/U2
   DERN=DERR**2+DERI**2
   IF  (ABSF((DERN-DERNM)/DERN)-TIOF)  30,21,21
21 IF  (NI-25)  22,22,31
22 NI=NI+1
   DERNM=DERN
   DO  25  I=1,N
   PR(I)=0.
   PI(I)=0.
   K1=ILO(I)
   K3=I+1
```

```
      DØ  23  J=K1,I
      J1=K3-J
      PR(I)=PR(I)+GR(J,J1)*UR(J)-GI(J,J1)*UI(J)
   23 PI(I)=PI(I)+GI(J,J1)*UR(J)+GR(J,J1)*UI(J)
      K2=IUP(I)
      K1=I-1
      DØ  24  J=K3,K2
      J1=J+K1
      PR(I)=PR(I)+GR(I,J1)*UR(J)-GI(I,J1)*UI(J)
   24 PI(I)=PI(I)+GI(I,J1)*UR(J)+GR(I,J1)*UI(J)
      PR(I)=H3*PR(I)-QRØ(I)*UR(I)+QIØ(I)*UI(I)
   25 PI(I)=H3*PI(I)-QIØ(I)*UR(I)-QRØ(I)*UI(I)
      UR(2)=(2.-FRH(1))*UR(1)+PR(1)
      UI(2)=-FIH(1)*UR(1)+PI(1)
      DØ  26  I=2,4
      BLA=2.-FRH(I)
      UR(I+1)=BLA*UR(I)+FIH(I)*UI(I)-UR(I-1)+PR(I)
   26 UI(I+1)=BLA*UI(I)-FIH(I)*UR(I)-UI(I-1)+PI(I)
      DØ  29  I=5,N2
      B3R=12.+FRH(I+1)
      B1R=24.-10.*FRH(I)
      B1I=-10.*FIH(I)
      B2R=12.+FRH(I-1)
      CR=B1R*UR(I)-B1I*UI(I)-B2R*UR(I-1)+FIH(I-1)
     1*UI(I-1)-PR(I+1)-10.*PR(I)-PR(I-1)
      CI=B1I*UR(I)+B1R*UI(I)-FIH(I-1)*UR(I-1)-B2R
     1*UI(I-1)-PI(I+1)-10.*PI(I)-PI(I-1)
      B3N=B3R**2+FIH(I+1)**2
      UR(I+1)=(CR*B3R+CI*FIH(I+1))/B3N
   29 UI(I+1)=(CI*B3R-CR*FIH(I+1))/B3N
      GØ  TØ  20
   30 LC=0
      IF  (KSP)  32,32,33
   32 IU(IL1)=N1
      ID(IL1)=0
      GØ  TØ  34
```

```
33 ID(ILI)=NI
34 RETURN
31 LC=1
   GO TO 34
   END
```

```
*TYPE(FØRTRAN)
      SUBRØUTINE  SCMAIR
      DIMENSIØN  F(4700),FC(56),FCP(51),GC(51),
     1GCP(51),SRU(25),SRD(25),SIU(25),SID(25)
      CØMMØN  F
      EQUIVALENCE  (F(48),ILI),(F(51),KSP),(F(52),DERR),
     1(F(53),DERI),(F(402),FC),(F(458),FCP),(F(509),GC),
     2(F(560),GCP),(F(4561),SRU),(F(4586),SIU),
     3(F(4611),SRD),(F(4636),SID)
      ILI=ILI
      SNR=FCP(ILI)-FC(ILI)*DERR+GC(ILI)*DERI
      SNI=GCP(ILI)-FC(ILI)*DERI-GC(ILI)*DERR
      SDR=FC(ILI)*DERR+GC(ILI)*DERI-FCP(ILI)
      SDI=GCP(ILI)+FC(ILI)*DERI-GC(ILI)*DERR
      SNN=SDR**2+SDI**2
      SRR=(SNR*SDR+SNI*SDI)/SNN
      SRI=(SNI*SDR-SNR*SDI)/SNN
      IF  (KSP)  1,1,2
    1 SRU(ILI)=SRR
      SIU(ILI)=SRI
      GØ  TØ  3
    2 SRD(ILI)=SRR
      SID(ILI)=SRI
    3 RETURN
      END
```

```
*TYPE(FORTRAN)
      SUBROUTINE  DERIVS
      DIMENSION  F(12265),SRU(25),SIU(25),SRD(25),
     1SID(25),FC(56),GC(51),UR(100),UI(100),IPA(10),
     2FRH(200),          QRO(100),QIO(100),ILO(100),
     3HLMAT(100,30),IUP(100),FRR(200),PR(200),FIDW(200),
     4FISR(200),FRA(200),FISA(200),FRV(200),FIDR(200),
     5FIDA(200),VCR(200),VSPR(100),WSPR(100),VSPA(100),
     6WSPA(100),VSPV(100),DSRU(9,25),DSIU(9,25),DSRD(9,25),
     7DSID(9,25)
      COMMON  F
      EQUIVALENCE  (F(44),NN),(F(43),N),(F(48),ILI),(F(51),KSP),(F(17),H
     1),(F(41),EK),(F(20),ISP),(F(49),AL),(F(56),NV),          (F(101
     21),VSPR),(F(1311),WSPR),(F(1111),VSPA),(F(1411),WSPA),(F(911),VSPV
     3),(F(4661),DSRU),(F(4886),DSIU),(F(5111),DSRD),(F(5336),DSID),(F(4
     4561),SRU),(F(4586),SIU),(F(4611),SRD),(F(4436),SID),(F(402),FC),(F
     5(509),GC),(F(4311),UR),(F(4411),UI),(F(15261),IPA),(F(3711),FRH),(
     6F(1711),FRV),(F(4111),QRO),(F(4211),QIO),(F(3611),ILO),(F(12261),H
     7LMAT),(F(3511),IUP),(F(1911),FRR),(F(3911),PR),(F(2511),FISR),(F(2
     81111),FRA),(F(2711),FISA),(F(2911),FIDW),(F(3111).FIDR),(F(3311),FI
     9DA),(F(57),NDS),(F(15871),VCR)
      IF  (NDS)  51,51,52
   52 NN=NN
      NN=NN
      N=N
      ILI=ILI
      IF  (KSP)  100,100,101
  100 SOUR=SRU(ILI)
      SOUI=SIU(ILI)
      GO  TO  102
  101 SOUR=SRD(ILI)
      SOUI=SID(ILI)
  102 TAR=FC(ILI)*(1.+SOUR)+SOUI*GC(ILI)
      TAI=GC(ILI)*(1.-SOUR)+SOUI*FC(ILI)
      URN=UR(N)**2+UI(N)**2
      CONR=(TAR*UR(N)+TAI*UI(N))/URN
      CONI=(TAI*UR(N)-TAR*UI(N))/URN
```

```
      TAR=(-CONR*CONI*H)/EK
      TAI=(CONR**2-CONI**2)*H/(2.*EK)
      IF  (ISP)  103,103,104
  103 TU=0.
      GO  TO  107
  104 IF  (KSP)  105,105,106
  105 TU=AL
      GO  TO  107
  106 TU=-(AL+1.)
  107 DO  50  K=1,NV
      KI0=IPA(K)
      IF  (KI0-1)  1,1,6
    1 DO  2  J=1,NN
    2 FRH(J)=FRV(J)*.04783258
      DO  5  I=1,N
      QRO(I)=0.
      QIO(I)=0.
      K1=ILO(I)
      K3=I+1
      DO  3  J=K1,I
      J1=K3-J
      K4=I+J
      TUTR=FRH(K4)*HLMAT(J,J1)
      QRO(I)=QRO(I)+TUTR*UR(J)
    3 QIO(I)=QIO(I)+TUTR*UI(J)
      K2=IUP(I)
      K1=1-I
      DO  4  J=K3,K2
      J1=J+K1
      K4=J+I
      TUTR=FRH(K4)*HLMAT(I,J1)
      QRO(I)=QRO(I)+TUTR*UR(J)
    4 QIO(I)=QIO(I)+TUTR*UI(J)
      QRO(I)=QRO(I)*H
    5 QIO(I)=QIO(I)*H
      GO  TO  34
```

```
  6 IF   (KIØ-2)   7,7,9
  7 DØ   8   J=1,NN
    FRH(J)=FRR(J)
  8 PR(J)=FISR(J)
    GØ   TØ   12
  9 IF   (KIØ-3)   10,10,16
 10 DØ   11   J=1,NN
    FRH(J)=FRA(J)
 11 PR(J)=FISA(J)
 12 DØ   15   I=1,N
    QRØ(I)=0.
    QIØ(I)=0.
    K1=ILØ(I)
    K3=I+1
    DØ   13   J=K1,I
    J1=K3-J
    K4=I+J
    TUTR=FRH(K4)*HLMAT(J,J1)
    TUTI=PR(K4)*HLMAT(J,J1)
    QRØ(I)=QRØ(I)+TUTR*UR(J)-TUTI*UI(J)
 13 QIØ(I)=QIØ(I)+TUTR*UI(J)+TUTI*UR(J)
    K2=IUP(I)
    K1=1-I
    DØ   14   J=K3,K2
    J1=J+K1
    K4=J+I
    TUTR=FRH(K4)*HLMAT(I,J1)
    TUTI=PR(K4)*HLMAT(I,J1)
    QRØ(I)=QRØ(I)+TUTR*UR(J)-TUTI*UI(J)
 14 QIØ(I)=QIØ(I)+TUTR*UI(J)+TUTI*UR(J)
    QRØ(I)=QRØ(I)*H
 15 QIØ(I)=QIØ(I)*H
    GØ   TØ   34
 16 IF   (KIØ-4)   17,17,19
 17 DØ   18   J=1,NN
 18 PR(J)=FRV(J)*.04783258
    GØ   TØ   28
```

```
19 IF  (KIU-5)  20,20,22
20 DO  21  J=1,NN
21 PR(J)=FIDW(J)*.04783258
   GO  TO  28
22 IF  (KIU-6)  23,23,25
23 DO  24  J=1,NN
24 PR(J)=FIDR(J)
   GO  TO  28
25 IF  (KIU-7)  26,26,32
26 DO  27  J=1,NN
27 PR(J)=FIDA(J)
28 DO  31  I=1,N
   QRO(I)=0.
   QIO(I)=0.
   K1=ILO(I)
   K3=I+1
   DO  29  J=K1,I
   J1=K3-J
   K4=I+J
   TUTI=PR(K4)*HLMAT(J,J1)
   QRO(I)=QRO(I)-UI(J)*TUTI
29 QIO(I)=QIO(I)+UR(J)*TUTI
   K2=IUP(I)
   K1=1-I
   DO  30  J=K3,K2
   J1=J+K1
   K4=I+J
   TUTI=PR(K4)*HLMAT(I,J1)
   QRO(I)=QRO(I)-TUTI*UI(J)
30 QIO(I)=QIO(I)+TUTI*UR(J)
   QRO(I)=QRO(I)*H
31 QIO(I)=QIO(I)*H
   GO  TO  34
32 DO  33  J=1,N
   QRO(J)=0.
33 QIO(J)=0.
```

```
34 IF   (KIU-1)   46,46,35
35 IF   (KIU-2)   36,36,38
36 DO   37   I=1,N
   TUTR=-VCR(2*I)+TU*VSPR(I)
   TUTI=TU*WSPR(I)
   QRA(I)=QRO(I)+TUTR*UR(I)-TUTI*UI(I)
37 QIA(I)=QIO(I)+TUTR*UI(I)+TUTI*UR(I)
   GO   TO   46
38 IF   (KIU-3)   39,39,41
39 DO   40   I=1,N
   TUTR=TU*VSPA(I)
   TUTI=TU*WSPA(I)
   QRA(I)=QRO(I)+TUTR*UR(I)-TUTI*UI(I)
40 QIA(I)=QIO(I)+TUTR*UI(I)+TUTI*UR(I)
   GO   TO   46
41 IF   (KIU-8)   46,42,44
42 DO   43   I=1,N
   QRA(I)=TU*VSPV(I)*UR(I)*.04783258*.011057
43 QIA(I)=TU*VSPV(I)*UI(I)*.04783258*.011057
   GO   TO   46
44 DO   45   I=1,N
   QRA(I)=-TU*VSPV(I)*UI(I)*.04783258*.011057
45 QIA(I)=TU*VSPV(I)*UR(I)*.04783258*.011057
46 TUTR=0.
   TUTI=0.
   DO   47   I=1,N
   TUTR=TUTR+QRO(I)*UR(I)-QIO(I)*UI(I)
47 TUTI=TUTI+QRO(I)*UI(I)+QIA(I)*UR(I)
   IF   (KSP)   48,48,49
48 DSRU(KIU,ILI)=TAR*TUTR-TAI*TUTI
   DSIU(KIU,ILI)=TAR*TUTI+TAI*TUTR
   GO   TO   50
49 DSRD(KIU,ILI)=TAR*TUTR-TAI*TUTI
   DSID(KIU,ILI)=TAR*TUTI+TAI*TUTR
50 CONTINUE
51 RETURN
   END
```

```
*TYPE(FORTRAN)
      SUBROUTINE CLEAN
      DIMENSION  SRD(25),SRU(25),SIU(25),SID(25),
     1DSRD(9,25),DSID(9,25),DSRU(9,25),DSIU(9,25),
     2F(5400)
      COMMON  F
      EQUIVALENCE  (F(48),ILI),(F(20),ISP),(F(51),KSP),(F(47),IL),
     1(F(45),LC),(F(19),ILM),(F(4611),SRD),(F(4561),SRU),
     2(F(4586),SIU),(F(4636),SID),(F(5111),DSRD),(F(5336),DSID),
     3(F(4661),DSRU),(F(4886),DSIU)
      ILI=ILI
      IF  (ISP)  1,1,3
    1 SRD(ILI)=SRU(ILI)
      SID(ILI)=SIU(ILI)
      DO  2  I=1,9
      DSRD(I,ILI)=DSRU(I,ILI)
    2 DSID(I,ILI)=DSIU(I,ILI)
      GO  TO  9
    3 IF  (ILI-1)  4,4,6
    4 SRD(1)=SRU(1)
      SID(1)=SIU(1)
      DO  5  I=1,9
      DSRD(I,1)=DSRU(I,1)
    5 DSID(I,1)=DSIU(I,1)
      GO  TO  14
    6 IF  (KSP)  7,7,8
    7 KSP=1
      LC=0
      RETURN
    8 KSP=0
    9 IF  (ABSF(SRD(ILI))-0.999)  14,10,10
   10 IF  (ABSF(SRU(ILI))-0.999)  14,11,11
   11 IF  (ABSF(SID(ILI))-0.001)  12,12,14
   12 IF  (ABSF(SIU(ILI))-0.001)  13,13,14
   13 LC=1
      RETURN
```

```
14 IL=IL+1
   IF  (IL-ILM)  16,16,15
15 LC=2
   RETURN
16 LC=-1
   RETURN
   END
```

```
*TYPE(FØRTRAN)
      SUBRØUTINE  SIGMAS
      DIMENSIØN   F(5900),CAS(25),SAN(25),PL(25),PLM(25),
     1SIGC(25),SRU(25),SIU(25),SRD(25),SID(25),ANG(100),
     2SIG(100),PØ(100),SIGCØU(100)
      CØMMØN  F
      EQUIVALENCE  (F(30),KM),(F(34),ILM),(F(47),IL),(F(58),SIGA),
     1(F(40),AKS),(F(39),ETA),(F(59),SIGE),(F(60),SIGT),(F(41),AK),
     2(F(351),SIGC),(F(4561),SRU),(F(4586),SIU),(F(4611),SRD),
     3(F(4636),SID),(F(5561),ANG),(F(5661),SIG),(F(5761),PØ),
     4(F(5861),SIGCØU)
      KM=KM
      ILM=IL+1
      B1=1.
      B2=0.
      SIGA=0.
      DØ  1  I=1,ILM
      SIGA=SIGA+B1*(1.-SRU(I)**2-SIU(I)**2)+B2*(1.-
     1SRD(I)**2-SID(I)**2)
      B1=B1+1.
    1 B2=B2+1.
      SIGA=SIGA*31.415927/AKS
      IF  (ETA)  2,2,5
    2 B1=1.
      B2=0.
      SIGE=0.
      DØ  3  I=1,ILM
      SIGE=SIGE+B1*((1.-SRU(I))**2+SIU(I)**2)+
     1B2*((1.-SRD(I))**2+SID(I)**2)
      B1=B1+1.
    3 B2=B2+1.
      SIGE=SIGE*31.415927/AKS
      B1=1.
      B2=0.
      SIGT=0.
      DØ  4  I=1,ILM
```

```
      SIGT=SIGT+B1*(1.-SRU(I))+B2*(1.-SRD(I))
      B1=B1+1.
    4 B2=B2+1.
      SIGT=SIGT*62.831854/AKS
    5 DØ  6  I=1,ILM
      CAS(I)=CØSF(2.*SIGC(I))
    6 SAN(I)=SINF(2.*SIGC(I))
      B1=2.*AK
      B2=2.*SIGC(1)
      B3=2.*ETA
      DØ  12  I=1,KM
      PL(1)=1.
      PL(2)=CØSF(ANG(I)*0.0174532925)
      PLM(1)=0.
      PLM(2)=SINF(ANG(I)*0.0174532925)
      C1=3.
      C2=1.
      C3=2.
      DØ  7  J=3,ILM
      PL(J)=(C1*PL(2)*PL(J-1)-C2*PL(J-2))/C3
      PLM(J)=PL(2)*PLM(J-1)+C3*PLM(2)*PL(J-1)
      C1=C1+2.
      C2=C2+1.
    7 C3=C3+1.
      C1=0.
      SUMAR=0.
      SUMAI=0.
      SUMBR=0.
      SUMBI=0.
      DØ  8  J=1,ILM
      SRM=(C1+1.)*SRU(J)+C1*SRD(J)-2.*C1-1.
      SIM=(C1+1.)*SIU(J)+C1*SID(J)
      SUMAR=SUMAR+(SRM*SAN(J)+SIM*CAS(J))*PL(J)
      SUMAI=SUMAI-(SRM*CAS(J)-SIM*SAN(J))*PL(J)
      SRM=SRU(J)-SRD(J)
      SIM=SIU(J)-SID(J)
```

```
      SUMBR=SUMBR+(SRM*SAN(J)+SIM*CAS(J))*PLM(J)
      SUMBI=SUMBI-(SRM*CAS(J)-SIM*SAN(J))*PLM(J)
    8 CI=CI+1.
      FCR=0.
      FCI=0.
      IF  (ETA)  9,10,9
    9 CI=SINF(ANG(I)*0.0087266463)
      C2=B2-B3*LØGF(CI)
      FCR=-ETA*CØSF(C2)/(CI**2)
      FCI=-ETA*SINF(C2)/(CI**2)
   10 SUMAR=SUMAR+FCR
      SUMAI=SUMAI+FCI
      SIG(I)=(SUMAR**2+SUMAI**2+SUMBR**2+SUMBI**2)*10.
     1/(BI**2)
      PØ(I)=20.*(SUMAI*SUMBR-SUMBI*SUMAR)/(SIG(I)*
     1(BI**2))
      IF  (ETA)  11,12,11
   11 SIGCØU(I)=SIG(I)*(BI**2)*.1/(FCR**2+FCI**2)
   12 CØNTINUE
      RETURN
      END
```

```
*TYPE(FORTRAN)
      SUBROUTINE  PRSIGM
      DIMENSION  F(5865),ANG(100),SIG(100),PO(100),ID(25),IU(25),
     1SIGC(100),SRU(25),SIU(25),SRD(25),SID(25),DSRU(9,25),
     2DSIU(9,25),DSRD(9,25),DSID(9,25)
      COMMON  F
      EQUIVALENCE  (F(39),ETA),(F(60),SIGT),(F(59),SIGE),(F(58),SIGA),
     1(F(30),KM),(F(34),LM),(F(62),IPR),(F(5561),ANG),(F(5661),SIG),
     2(F(5761),PO),(F(5861),SIGC),(F(4561),SRU),(F(4586),SIU),
     3(F(4611),SRD),(F(4636),SID),(F(4661),DSRU),(F(4886),DSIU),
     4(F(5111),DSRD),(F(5336),DSID),(F(4511),IU),(F(4536),ID)
      IF  (ETA)  1,1,15
    1 DO  2  I=1,KM
    2 SIGC(I)=0.
      GO  TO  3
   15 SIGT=0.
      SIGE=0.
    3 WRITE OUTPUT TAPE 9,4,SIGA,SIGE,SIGT
    4 FORMAT(12H0    RESULTS/19H0    SIG.REACTION =,
     10PE14.6,5X,13HSIG.ELASTIC =,0PE14.6,5X,
     211HSIG.TOTAL =,0PE14.6/85X,9HS  MATRIX/
     36X,5HTHETA,8X,5HSIGE.,10X,4HPOL.,9X,8HSIG/SIGC,
     411X,1HL,4X,3HSR+,6X,3HSI+,4X,2HIT,6X,3HSR-,6X,
     53HSI-,4X,2HIT)
       IX=0
      DO  5  I=1,LM
      WRITE OUTPUT TAPE 9,6,ANG(I),SIG(I),PO(I),SIGC(I),
     1IX,SRU(I),SIU(I),IU(I),SRD(I),SID(I),ID(I)
    5 IX=IX+1
    6 FORMAT(5X,0PF6.2,4X,0PE12.4,4X,0PE12.4,4X,0PE12.4,
     19X,I2,2X,0PF7.4,2X,0PF7.4,2X,I2,4X,0PF7.4,2X,0PF7.4,
     22X,I2)
       IX=IX+1
      DO  7  I=IX,KM
    7 WRITE OUTPUT TAPE 9,8,ANG(I),SIG(I),PO(I),SIGC(I)
```

```
 8 FØRMAT(5X,0PF6.2,4X,0PE12.4,4X,0PE12.4,4X,0PE12.4)
   IF  (IPR-1)  13,9,9
 9 WRITE ØUTPUT TAPE 9,10
10 FØRMAT(1H1,30X,
  141HDERIVATIVES ØF SCATTERING MATRIX ELEMENTS/
  27X,1HL,6X,2HUS,10X,2HRS,10X,2HAS,10X,2HWS,10X,2HWD,
  310X,2HRD,10X,2HAD,10X,3HUSP,9X,3HWSP)
   IX=0
   DØ  12  I=1,LM
   WRITE ØUTPUT TAPE 9,11,IX,(DSRU(J,I),J=1,9),
  1(DSIU(J1,I),J1=1,9),(DSRD(J2,I),J2=1,9),
  2(DSID(J3,I),J3=1,9)
11 FØRMAT(6HU SR+ ,I3,9(0PE12.3)/2X,7HSI+    ,
  19(0PE12.3)/2X,7HSR-    ,9(0PE12.3)/2X,7HSI-    ,
  29(0PE12.3))
12 IX=IX+1
13 RETURN
   END
```

TEST CASE FOR IBM-7090

SEARCH NON-LOCAL OPTICAL POTENTIAL
PARAMETERS

EXPERIMENTAL		THEORETICAL			
ELAB	# 4.100	US	# 71.000		
MT	# 56.000	WS	# -0.	WD	# 15.000
ZT	# -0.	RS	# 1.220	RD	# 1.220
MI	# 1.000	AS	# 0.650	AD	# 0.470
ZI	# -0.				

RESULTS

SIG.REACTION # 0.157058E 04 SIG.ELASTIC # 0.207966E 04

THETA	SIGE.	POL.	SIG/SIGC
0.	0.1740E 04	0.	0.
5.00	0.1706E 04	0.2637E-02	0.
10.00	0.1609E 04	0.4911E-02	0.
15.00	0.1458E 04	0.6420E-02	0.
20.00	0.1267E 04	0.6683E-02	0.
25.00	0.1054E 04	0.5075E-02	0.
30.00	0.8373E 03	0.7307E-03	0.
35.00	0.6315E 03	-0.7619E-02	0.
40.00	0.4489E 03	-0.2196E-01	0.
45.00	0.2972E 03	-0.4561E-01	0.
50.00	0.1797E 03	-0.8463E-01	0.
55.00	0.9581E 02	-0.1513E-00	0.
60.00	0.4220E 02	-0.2730E-00	0.
65.00	0.1358E 02	-0.4958E-00	0.
70.00	0.3970E 01	-0.2167E-00	0.
75.00	0.7396E 01	0.7078E 00	0.
80.00	0.1849E 02	0.5770E 00	0.
85.00	0.3278E 02	0.4472E-00	0.
90.00	0.4682E 02	0.3554E-00	0.
95.00	0.5817E 02	0.2806E-00	0.
100.00	0.6534E 02	0.2104E-00	0.
105.00	0.6765E 02	0.1375E-00	0.
110.00	0.6517E 02	0.5636E-01	0.
115.00	0.5854E 02	-0.3798E-01	0.
120.00	0.4890E 02	-0.1510E-00	0.
125.00	0.3768E 02	-0.2886E-00	0.
130.00	0.2645E 02	-0.4548E-00	0.
135.00	0.1672E 02	-0.6336E 00	0.
140.00	0.9704E 01	-0.7024E 00	0.
145.00	0.6164E 01	-0.2332E-00	0.
150.00	0.6284E 01	0.7039E 00	0.
155.00	0.9634E 01	0.9861E 00	0.
160.00	0.1525E 02	0.8296E 00	0.
165.00	0.2180E 02	0.6022E 00	0.
170.00	0.2784E 02	0.3869E-00	0.
175.00	0.3206E 02	0.1888E-00	0.
180.00	0.3357E 02	0.6393E-07	0.

TEST CASE FOR IBM-7090 (CONTINUED)

NUMERICAL

USP	#	1300.00	H	#	0.150
WSP	#	-0.	HC	#	0.100
BETA	#	0.85	LM	#	20

SIG.TOTAL # 0.365024E 04

S MATRIX

L	SR+	SI+	IT	SR-	SI-	IT
0	0.2740	-0.1583	6	0.2740	-0.1583	0
1	-0.2011	-0.6271	5	-0.3219	-0.5925	5
2	0.0039	-0.4952	6	-0.0053	-0.2687	6
3	0.7897	0.1612	4	0.8948	0.0196	6
4	0.9694	-0.0048	4	0.9492	0.0038	5
5	0.9957	0.0049	4	0.9984	0.0026	4
6	0.9999	0.0003	4	0.9999	0.0001	3

TEST CASE FOR CDC-1604

SEARCH NØN-LØCAL ØPTICAL PØTENTIAL
PARAMETERS

EXPERIMENTAL		THEØRETICAL			
ELAR =	4.100	US =	71.000		
MT =	56.000	WS =	.000	WD =	15.000
ZT =	.000	RS =	1.220	RD =	1.220
MI =	1.000	AS =	.650	AD =	.470
ZI =	.000				

RESULTS
SIG.REACTIØN = 1.569143E 003   SIG.ELASTIC = 2.081708E 003

THETA	SIGE.	PØL.	SIG/SIGC
.00	1.7415E 003	.0000E 000	.0000E 000
5.00	1.7079E 003	2.6111E-003	.0000E 000
10.00	1.6104E 003	4.8573E-003	.0000E 000
15.00	1.4590E 003	6.3370E-003	.0000E 000
20.00	1.2682E 003	6.5670E-003	.0000E 000
25.00	1.0554E 003	4.9215E-003	.0000E 000
30.00	8.3823E 002	5.3469E-004	.0000E 000
35.00	6.3218E 002	-7.8648E-003	.0000E 000
40.00	4.4936E 002	-2.2263E-002	.0000E 000
45.00	2.9747E 002	-4.5988E-002	.0000E 000
50.00	1.7982E 002	-8.5118E-002	.0000E 000
55.00	9.5863E 001	-1.5197E-001	.0000E 000
60.00	4.2179E 001	-2.7416E-001	.0000E 000
65.00	1.3535E 001	-4.9876E-001	.0000E 000
70.00	3.9236E 000	-2.1806E-001	.0000E 000
75.00	7.3713E 000	7.1254E-001	.0000E 000
80.00	1.8495E 001	5.7799E-001	.0000E 000
85.00	3.2818E 001	4.4734E-001	.0000E 000
90.00	4.6883E 001	3.5523E-001	.0000E 000
95.00	5.8254E 001	2.8025E-001	.0000E 000
100.00	6.5429E 001	2.0998E-001	.0000E 000
105.00	6.7743E 001	1.3698E-001	.0000E 000
110.00	6.5251E 001	5.5782E-002	.0000E 000
115.00	5.8611E 001	-3.8626E-002	.0000E 000
120.00	4.8952E 001	-1.5170E-001	.0000E 000
125.00	3.7714E 001	-2.8950E-001	.0000E 000
130.00	2.6474E 001	-4.5603E-001	.0000E 000
135.00	1.6730E 001	-6.3565E-001	.0000E 000
140.00	9.7040E 000	-7.0620E-001	.0000E 000
145.00	6.1564E 000	-2.3916E-001	.0000E 000
150.00	6.2690E 000	7.0049E-001	.0000E 000
155.00	9.6139E 000	9.8514E-001	.0000E 000
160.00	1.5226E 001	8.2922E-001	.0000E 000
165.00	2.1774E 001	6.0198E-001	.0000E 000
170.00	2.7808E 001	3.8674E-001	.0000E 000
175.00	3.2030E 001	1.8871E-001	.0000E 000
180.00	3.3543E 001	7.8020E-009	.0000E 000

TEST CASE FOR CDC-1604 (CONTINUED)

NUMERICAL

USP	=	1300.00	H	=	.150
WSP	=	.00	HC	=	.100
BETA	=	.85	LM	=	20

SIG.TOTAL = 3.6508851E 003

S MATRIX

L	SR+	SI+	IT	SR-	SI-	IT
0	.2744	-.1584	6	.2744	-.1584	0
1	-.2026	-.6268	5	-.3233	-.5920	5
2	.0037	-.4962	3	-.0054	-.2693	6
3	.7902	.1603	4	.8950	.0187	6
4	.9694	-.0050	4	.9493	.0036	5
5	.9957	.0049	4	.9984	.0025	4
6	.9999	.0003	4	.9999	.0001	4

ORNL-3429
UC-34 — Physics
TID-4500 (20th ed.)

INTERNAL DISTRIBUTION

1. Biology Library
2-3. Central Research Library
4. Reactor Division Library
5-6. ORNL — Y-12 Technical Library Document Reference Section
7-56. Laboratory Records Department
57. Laboratory Records, ORNL R.C.
58. Nancy Alexander
59. J. B. Ball
60. R. Bender
61. R. E. Biggers
62. E. P. Blizard
63. A. C. Downing
64. C. B. Fulmer
65. C. D. Goodman
66. F. F. Haywood
67. W. H. Jordan
68. W. E. Kinney
69. C. E. Larson
70. F. K. McGowan
71. F. C. Maienschein
72. C. D. Moak
73. C. W. Nestor
74. S. K. Penny
75-84. F. G. Perey
85. A. M. Perry
86. C. A. Preskitt
87. M. J. Skinner
88. C. D. Susano
89. J. A. Swartout
90. D. K. Trubey
91. A. M. Weinberg
92. C. D. Zerby
93. R. A. Charpie (consultant)
94. P. F. Gast (consultant)
95. R. F. Taschek (consultant)
96. T. J. Thompson (consultant)

EXTERNAL DISTRIBUTION

97-98. University of California, Los Angeles (1 copy each to M. A. Melkanoff and D. S. Saxon)
99-100. Argonne National Laboratory (1 copy each to P. A. Moldauer and A. Elwin)
101. R. S. Caswell, National Bureau of Standards, Washington
102. L. Cranberg, Los Alamos Scientific Laboratory
103-104. Brookhaven National Laboratory (1 copy each to E. Averbach and C. E. Porter)
105. M. H. Kalos, United Nuclear Corporation, White Plains, New York
106. Research and Development Division, AEC, ORO
107-743. Given distribution as shown in TID-4500 (20th ed.) under Physics category (75 copies — OTS)