Stan Williamson

Table of Contents

## The Phase Shift Analysis of Electron Scattering

First we will briefly discuss the partial wave analysis of the Dirac equation with a Coulomb field. We are interested in the scattering of high energy electrons or positrons and the results apply to the scattering of these particles. The solution of the three dimensional four component Dirac equation is considerably simplified if the Coulomb potential is assumed to be spherically symmetric and this assumption is made in all of our work. A spherically symmetric potential allows us to find solutions in the form of a sum of spherical harmonics multiplied by radial wave functions. As we wish to calculate the differential scattering cross section, we need the wave function at large distances from the origin. Outside the charge distribution of the nucleus the radial equations may be solved exactly. A discussion of this solution is given in the section on the calculation of the Coulomb functions, page C-1. One needs to solve the radial Dirac equations inside the nuclear charge distribution and match these solutions with the Coulomb functions at some fitting on radius $x_o$ outside the charge distribution. The standard reference for the method used in the paper by Yennie, Ravenhall and Wilson, Phys. Rev. 95, 500.

As we are concerned with the scattering of very high energy electrons, it is possible to neglect the rest energy of the electron in comparison

with its total energy. This produces considerable simplification as it allows us to work with a two component radial wave equation rather than a four component equation.

The Dirac equation may be written

$$(\alpha \cdot pc + \beta\, mc^2 + V)\psi = E\psi \qquad (1)$$

One can choose a representation for the Dirac matrices which allows us to separate out positive and negative helicity states in the incident wave, and facilitate the neglect of the mass term. One may choose

$$\alpha_i = \rho_3\sigma_i \quad , \quad \beta = \rho .$$

See P. M. A. Dirac Principles of Quantum Mechanics, Fourth Edition, pg. 257, for the definitions of these matrices. We can now write the two component wave equations

$$(\sigma \cdot pc + V - E)\phi = -mc^2\chi$$
$$(-\sigma \cdot pc + V - E)\chi = -mc^2\phi$$

and neglecting the mass one obtains the uncoupled equations

$$(\sigma \cdot pc + V - E)\phi = 0 \qquad (2)$$

$$(-\sigma \cdot pc + V - E)\chi = 0 \qquad (3)$$

and for a spherically symmetric potential the scattering in a given direction will be the same for either helicity state so that we need only consider one of the equations, say Eq. (2).

In order to calculate the differential cross section, one needs the asymptotic form of the wave function. Taking the incident plane wave along the z-axis the asymptotic form is

$$\phi \underset{r\to\infty}{\sim} \begin{pmatrix} 1 \\ 0 \end{pmatrix} e^{ikz} + \frac{f(\theta,\phi)}{r} \begin{pmatrix} 1 \\ \tan\frac{1}{2}\theta e^{i\phi} \end{pmatrix} e^{ikr}$$

which is the sum of the incident plane wave and an outgoing spherical wave as it should be. In Eq. (4) $f(\theta, 0)$ is the scattering amplitude and is related to the differential cross section by

$$\frac{d\sigma}{d\Omega} = |f(\theta)|^2 \left(1 + \tan^2\frac{\theta}{2}\right) \quad . \tag{4}$$

We need to calculate $f(\theta)$ as one may then directly calculate

$$\frac{d\sigma}{d\Omega} \text{ vs } \theta .$$

The solution to Eq. (2) is greatly simplified by the fact that we are assuming V to be spherically symmetric. This means that $\phi$ may be expanded in terms of Legendre polynomials. This allows us to obtain the coupled first order differential equations for the radial wave function. These equations are

$$\frac{d}{dr} G_j(r) - \frac{(j + \frac{1}{2})}{r} G_j(r) + \frac{(E - V)}{\hbar c} F_j(r) = 0$$

$$\frac{d}{dr} F_j(r) + \frac{(j + \frac{1}{2})}{r} F_j(r) - \frac{(E - V)}{\hbar c} G_j(r) = 0$$

In dimensionless form we have

$$x = kr$$

$$\nu = \frac{V}{E}$$

$$k = \frac{E}{\hbar c}$$

$$n = j - \tfrac{1}{2} \quad , \qquad n = 1, 2, \ldots$$

$$\frac{d}{dx} G_n(x) - \frac{n}{x} G_n(x) + (1 - \nu) F_n(x) = 0 \qquad (5)$$

$$\frac{d}{dx} F_n(x) + \frac{n}{x} F_n(x) - (1 - \nu) G_n(x) = 0 \qquad (6)$$

where we enumerate the states by the integer n which starts at one and is conveniently defined for programming. Equations (5) and (6) are solved numerically inside the nuclear charge distribution. These solutions are matched at the fitting on radius with the Coulomb functions and the nuclear phase shift is calculated. In order to start the numerical integration it is necessary to have the solutions at the origin or some small value of x, $x_s$. The series solutions are

$$G_n(x_s) = \sum_{k=o}^{\infty} a_k x_s^{\lambda+k} \qquad (7)$$

$$F_n(x_s) = \sum b_k x_s^{\lambda+k+1} \qquad (8)$$

Substitution of (7) and (8) into (5) and (6) gives

$$\sum_{k=o}^{\infty} (\lambda + k) a_k x_s^{\lambda+k-1} - n \sum_{k=o}^{\infty} a_k x_s^{\lambda+k-1}$$

$$+ (1 - \nu(o)) \sum_{k=o}^{\infty} b_k x_s^{\lambda+k+1} - \nu'(o) \sum_{k=o}^{\infty} b_k x_s^{\lambda+k+2}$$

$$- \tfrac{1}{2} \nu''(o) \sum_{k=o}^{\infty} b_k x_s^{\lambda+k+3} + \ldots = 0 \quad ,$$

and

$$\sum_{k=o}^{\infty} (\lambda+k+1)\, b_{k_1} x_s^{\lambda+k} + n \sum_{k=o}^{\infty} b_k\, x_s^{\lambda+k} - (1 - \nu(o)) \sum_{k=o}^{\infty} a_k\, x_s^{\lambda+k}$$

$$+ \nu'(o) \sum_{k=o}^{\infty} a_k\, x_s^{\lambda+k+1} + \tfrac{1}{2}\, \nu''(o) \sum_{k=o}^{\infty} a_k\, x_s^{\lambda+k+2} + \ldots = 0 \quad ,$$

where we assume that

$$\nu(x_s) = \nu(o) + \nu'(o)\, x_s + \nu''(o)\, \frac{x_s^2}{2} + \ldots \quad .$$

The initial equation is given by the k = 0 term and is

$$\lambda = n.$$

We can now write

$$\sum_{k=1}^{\infty} \left[ k a_k + (1 - \nu(o))\, b_{k-2} - \nu'(o) b_{k-3} - \tfrac{1}{2}\, \nu''(o) b_{k-4} + \ldots \right] x^{k+n-1} = 0$$

$$\sum_{k=1}^{\infty} \left[ (2n + 1 + k) b_k - (1 - \nu(o)) a_k + \nu'(o) a_{k-1} + \tfrac{1}{2}\, \nu''(o) a_{k-2} + \ldots \right] x^{k+n} = 0$$

and we may equate each coefficient separately to zero. We take the arbitrary coefficient $a_o$ to be unity which only scales the wave function and we obtain

$$a_o = 1 \qquad b_o = \frac{1 - \nu(o)}{2n + 1}$$

$$a_1 = 0 \qquad b_1 = \frac{-\nu'(o)}{2n + 2}$$

$$a_2 = \frac{-(1 - \nu(o)) b_o}{2} \qquad b_2 = \frac{1}{2n+3} \left[ (1 - \nu(o)) a_2 - \tfrac{1}{2} \nu''(o) \right]$$

$$a_3 = \frac{1}{3} \left[ - (1 - \nu(o)) b_1 + \nu'(o) b_o \right]$$

and so on. We have found that it is sufficient to include only the terms given above in the starting series. In fact, the wave functions are quite insensitive to their starting value, especially as n increases. In order to take advantage of this fact we start the integration at an x value given by

$$x_s = n\text{Int}\left[10^{\frac{n}{11}}\right]$$

? actually

$$x_S = \text{INT}\left[n \cdot 10^{\frac{11}{n}}\right]$$

where Int stands for integer value. The starting values are given by

$$G_n(x_s) = 10^{-5}$$

$$F_n(x_s) = x_s \left[\frac{1 - \nu(x_s)}{2n + 1}\right] 10^{-5}$$

This procedure saves considerable computer time when many phase shifts are needed as in the case of very high energy electrons scattering from heavy nuclei. We have observed no degredation in the accuracy of our results due to this method. We have also found that beyond n = 5 the regular series solutions may be calculated using n = 5 in the exponents in $x_s^n$ or $x_s^{n+1}$.

Now we will briefly discuss the calculation of the nuclear phase shift. At the fitting on radius $x_o$ we match the wave functions $F_n(x_o)$ and $G_n(x_o)$ to the regular and irregular Coulomb functions which have been calculated by the Coulomb function program and used as input to this program. Then at $x = x_o$

$$G_n(x_o) = C_n G_{n,R}(x_o) + D_n G_{n,I}(x_o) \qquad (9)$$

$$F_n(x_o) = C_n F_{n,R}(x_o) + D_n F_{n,I}(x_o) \qquad (10)$$

where the subscripts R and I stand for regular or irregular functions at the origin respectively. The asymptotic form of the Coulomb functions are given by

$$G^{(x)}_{n,R} \sim \sin\left[x + \gamma \ln 2x - \tfrac{1}{2}(n-1)\pi + \eta^c_n\right]$$

$$G^{(x)}_{n,I} \sim \sin\left[x + \gamma \ln 2x - \tfrac{1}{2}(n-1)\pi + \eta^{c\,\prime}_n\right]$$

and the wave function goes like

$$G_n(x) \sim \sin\left[x + \gamma \ln 2x - \tfrac{1}{2}(n-1)\pi + \eta_n\right]$$

where $\eta_n$ is the nuclear phase shift. The Coulomb phase shifts are known and are

$$\exp(2i\eta^c_n) = \frac{(\rho_n - i\gamma)\Gamma(\rho_n - i\gamma)}{n \qquad \Gamma(\rho_n + i\gamma)} \; e^{\pi i(n - \rho_n)}$$

and

$$\exp i(\eta^{c\,\prime}_n - \eta^c_n) = \frac{1 - i\tan\pi(n - \rho_n)\coth\pi\gamma}{|1 - i\tan\pi(n - \rho_n)\coth\pi\gamma|} \; e^{-\pi i(n - \rho_n)}$$

where

$$\rho_n = [n^2 - \gamma^2]^{\frac{1}{2}}$$

$$\gamma = \frac{Ze^2}{\hbar c}$$

From this one obtains

$$\tan(\eta_n - \eta^c_n) = \frac{\sin(\eta^{\prime c}_n - \eta^c_n)}{\frac{C_n}{D_n} + \cos(\eta^{c\,\prime}_n - \eta^c_n)} = \tan\delta_n. \qquad (11)$$

The values of $\tan\delta_n$ are printed out as the phase shifts. These values are, in general, negative and monotone decreasing. We usually cut off the calculation when

$$|\tan\delta_n| \leq 10^{-10}$$

If during the calculation the time specified is exceeded these values as well as the initial conditions for $G(x_s)$ and $F(x_s)$ will be punched out so they can be used as input if the calculation is to be continued. The calculation of the Gamma functions of complex argument is described in the section on the Gamma function subroutine, page G-1.

We now are in a position to find the scattering amplitude. The series for the scattering amplitude is given by

$$f(\theta) = \frac{1}{2ik} \sum_{n=1}^{\infty} e^{2i\eta_n} \left\{ n \cdot (P_{n-1}(\cos\theta) + P_n(\cos\theta)) \right\} . \qquad (12)$$

Yennie, Ravenhall and Wilson give a method for improving the convergence of this series. First let us write the sum in (12) in a more convenient manner:

$$f(\theta) = \frac{1}{2ik} \sum_{n=1}^{\infty} [n\, e^{2i\eta_n} + (n-1) e^{2i\eta_{n-1}}] P_{n-1}(\cos\theta) . \qquad (13)$$

The next thing to note is that the nuclear phase shift is given in terms of $\tan\delta_n$ by

$$e^{2i\eta_n} = e^{2i\eta_n^c} \frac{(1 + i\tan\delta_n)^2}{1 + \tan\delta_n} , \qquad (14)$$

and even though $\tan\delta_n$ may be zero beyond some $n_{max}$ the series in (13) must be carried out to very large n. This is a characteristic of the Faxen-Holtzmark series. It is very slow in converging. For example, $n_{max}$ has never been greater than 60 even for lead at 750 MeV; however, the series in (13) must be carried out to at least 80 terms. We have, in general,

taken 80 terms in this series, but as many at 100 terms may be taken.

The Faxen-Holtzmark series is particularly slow to converge for small angles. In this case it is wise to alter the series by adding and subtracting the non-relativistic point scattering amplitude given by

$$f_{NR}(\theta) = \frac{1}{2ik} \sum_{n=1}^{\infty} (2n - 1)\, e^{2i\beta_n}\, P_{n-1}(\cos\theta)$$

$$= \frac{\gamma e^{2i\beta_l}\ e^{i\gamma \ln(\sin^2\frac{\theta}{2})}}{ak \sin^2(\frac{\theta}{2})}$$

where the non-relativistic point Coulomb phase shifts are

$$e^{2i\beta_n} = \frac{\Gamma(n - i\gamma)}{\Gamma(n + i\gamma)} \qquad n = 1, 2, \ldots .$$

We may write the alternate expression to Eq. (13) for small angles as

$$f(\theta) = \frac{\gamma e^{2i\beta_l}\ e^{i\gamma \ln(\sin^2\frac{\theta}{2})}}{2k \sin^2\frac{\theta}{2}} + \frac{1}{2ik} \sum_{n=1}^{\infty} [ne^{2i\eta_n} + (n - 1)e^{2i\eta_{n-1}} - (2n - 1) \tag{15}$$

$$e^{2i\beta_n}]\, P_{n-1}(\cos\theta) .$$

Eq. (13) is used for angles larger than some input cut of and Eq. (15) is used for the smaller angles. The cut off angle varies from case to case becoming smaller as the energy increases.

We now have only to find the sum of a series of the form

$$S = \sum_{n=1}^{\infty} a_n\, P_{n-1}(\cos\theta) .$$

Yennie, Ravenhall and Wilson have given a reduction method for improving the convergence of this sum. This method is discussed in the write up on the reduction subroutine, page H-1. Briefly the $m^{th}$ reduced series is given by

$$(1 - \cos\theta)^m S = \sum a_n^m P_{n-1}(\cos\theta)$$

where

$$a_j^{(i+1)} = a_j^{(i)} - \frac{j+1}{2j+3} a_{j+1}^{(i)} - \frac{j}{2j-1} a_{j-1}^{(i)}$$

We have found three reductions, m=3, to be sufficient. The Legendre polynomials are calculated from the recursion relation

$$P_{n+1}(\cos\theta) = (2(n-1) + 1) \cos\theta P_n(\cos\theta) - \frac{(n-1)}{n} P_{n-1}(\cos\theta))$$

$$P_1(\cos\theta) = 1,$$

$$P_2(\cos\theta) = \cos\theta.$$

The Legendre polynomials must, of course, be calculated for every $\theta$; however, the coefficients need be calculated only twice, once for the small angle expression and once for the large angle expression. Once the series is summed it is a simple matter to calculate the cross section from Eq. (4).

The last topic for discussion before we give the program specifications is the transformation from the laboratory system to the center of mass system. The experiments are, of course, done in the laboratory system.

The calculation should be done in the center of mass system. The procedure is to take the experimental incident energy, $E_L$, and momentum $k_L = \frac{E_L}{\hbar c}$ and convert this to the appropriate center of mass momentum k to be used in the calculations. The relationship is

$$k = E_L \sqrt{\frac{M}{M+2E_L}} \tag{16}$$

where M is the rest mass of the nucleus in MeV. We, in general, want the differential cross section at convenient, evenly spaced laboratory angles for easy comparison with experiment. We therefore convert the evenly spaced laboratory angles into angles in the center of mass system. The transformation equation is

$$\tan \theta_{c.m.} = \sqrt{1 + \frac{2E_L}{M}} \left\{ \frac{2 \sin(\theta/2) \cos(\theta/2)}{\cos\theta - \frac{2E_L}{M} \sin^2\frac{\theta}{2}} \right\} \tag{17}$$

The differential cross sections are calculated at center of mass angles $\theta_{c.m.}$ and must be transformed back to the laboratory system. The transformation equation is

$$\frac{d\sigma}{d\Omega}(\theta_L, E_L) = \frac{(E_1 + E_2)^2}{[M + E_L(1 - \cos\theta_L)]^2} \quad \frac{d\sigma(\theta_{cm}, E_{cm})}{d\Omega} \tag{18}$$

where

$$E_1 = E_L \sqrt{\frac{M}{M+2E_L}}$$

$$E_2 = \sqrt{\frac{M}{M+2E_L}} \cdot (M + E_L) .$$

These transformations may be turned off as might be desirable if a comparison between theoretical calculation is desired for then the nice angular steps are in $\theta_{cm}$.

Now we will discuss the default specifications made in the RAVENHALL program. We have chosen the integration parameters such that the step size is 0.02, the starting x, $x_s$ where applicable is 0.02 and the final x-value is , an input parameter. We have suppressed all of the output which was not considered desirable at least part of the time. This means that we have taken:

a) NEXC = 0 Which does not punch out the real and imaginary parts of the scattering amplitude. This was used in our work on the monopole excitation problem. If these amplitudes are desirec set NEXC = 1 and recompile.

b) NCHECK = 1 Which turns off a large amount of printing of intermediate output. If this output is desired set NCHECK = 0 and recompile.

c) NCKF = 0 Which turns off the intermediate printing of the wave functions. If this output is desired set NCKF = 1 and recompile.

Next we discuss the input specifications for the RAVENHALL program. In the standard case, i.e., where the charge distribution is being calculated and not read in, there are three input cards plus the 2N regular and irregular Coulomb function cards. There is also at least one card required by subroutine FOLD if it is called. See the write up of this subroutine on page J-1 .

CARD ONE

GAM = Atomic number of the target divided by the fine structure constant. This is usually given to six places. It must be identical to the value used in the Coulomb function program.

XO = Fitting on radius (program units). This must be identical to the fitting on radius used in the Coulomb function program.

C = Charge distribution parameter in fermis

Z = Charge distribution parameter in fermis

$P_3$ = Charge distribution parameter dimensionless

EO = Energy of the incident electron in MeV (Usually in the lab system)

ATWT = Atomic weight of the target

CARD TWO

DELB = Experimental angular resolution used in folding the cross section over this angular uncertainty

DELO = Step size in the scattering angle (Usually in the lab system)

OTEST = The angle at which we change from the small angle expression for the scattering amplitude to the large angle expression

O1 = Initial value of the scattering angle (usually in the lab system)

OFIN = The final value of the scattering angle

CONV = Phase shifts ($\tan \delta_n$) smaller than this value will not be calculated. This should be around $10^{-10}$.

TMIN = Time in minutes allowed to complete the calculation of the phase shifts. If this time is exceeded phase shifts and starting series will be punched out on cards in order that the calculation may continue. If this field is left blank a default value of 15 minutes is taken.

CARD THREE

NF = Maximum number of phase shifts to be calculated

NST = First phase shift to be calculated, normally NST = 1

NFL = Number of terms in the Faxen-Holtzmark series for the scattering amplitude. A value of 80 is acceptable.

NFOLD, NFOLD = 1 Fold the cross sections over the angular uncertainly, but do not punch them out on cards.

NFOLD = 2 Fold the cross sections over the angular uncertainty, but do punch them out on cards.

NFOLD = 3 Do not fold or punch the cross sections

NFOLD = 4 Do not fold the cross sections, but do punch then out on cards

NCM = 0 Make all the necessary transformations laboratory to center of mass and vice versa.

NCM = 1 Do not make the transformations

IC = 0 Do read in the Coulomb functions

IC = 1 Do not read in the Coulomb functions. Rather use the Coulomb functions from the previous case

NER = 0 Do not read in the phase shifts. This is the standard mode.

NER = 1 Do read in the phase shifts and initial values. The cards become part of the input stream in this case.

If IC = 0 we read in the Coulomb functions. 2NF cards in all

FRN(I) = Regular function F
GRN(I) = Regular function G
FIN(I) = Irregular function F
GRN(I) = Irregular function G

If the charge distribution is being read in the appropriate cards would appear at this point. This completes the data cards read in by the RAVENHALL Program; however, if they are used both subroutine FOLD and subroutine FIT require additional input cards. These cards would appear after the cards read in by the RAVENHALL Program. Subroutine FOLD requires at least one card and may require several more. See the write up of subroutine FOLD, page J-1, and subroutine FIT, page K-1, for the specifications for this input.

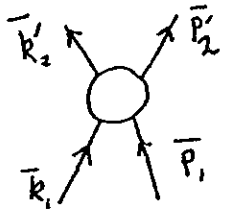
The output from the Ravenhall program consists of the initial input, the potential and charge distribution vs. the x value. The normalization integrals $I(\infty)$ and $J(\infty)$ as well as the mean square radius in fermis. The wave functions $F_n(x_o)$ and $G_n(x_o)$ are given at the fitting on radius. The phase shifts $\tan\delta_n$ are printed out and finally the center of mass cross section, real and imaginary scattering amplitude and the laboratory cross section. If subroutine FOLD is used then the folded cross section is printed and if desired, punched out on cards for subsequent use in fitting routines or in plotting the cross section. Finally, if subroutine FIT is used the results of this least squares fitting are given.

The RAVENHALL program is written in FORTRAN IV using double precision arithmetic throughout. The gamma function subroutine, DAMMA, the complex arithemtic subroutine, DOMPLX, the series reduction subroutine, RECUR, the subroutine for storing the potential, POT, the folding subroutine, FOLD, the fitting routine, FIT, the appropriate charge distribution subroutine FX, and the fourth order R.K.G. integration routine DEQ are required by the

RAVENHALL program. The execution time varies depending on the number of phase shifts calculated, the value of $x_o$, and the angular range covered. As an indication for 754 MeV electrons on lead using $x_o$ = 60 and 54 phase shifts for angles from 5 degrees to 80 degrees every 0.2 degree and including folding we used about 5 minutes of CPU time, for 248.2 MeV electrons on lead using $x_o$ = 30 and 24 phase shifts for angles from 2 degrees to 140 degress every 0.5 degrees and including folding we used about 2 minutes of CPU time.

## Breit Frame Phase Shift Analysis

In the preceding section we described the phase shift analysis for electron scattering. The kinematic efforts discussed on page A-10 were for changing from the center of mass to the laboratory frame and vice versa. It is also possible to take kinematic effects into account by doing the calculation in the Breit frame rather than the center of mass. If the scattering process is represented by




then in the lab system $\bar{k}_1$ or $\bar{p}_1$ vanish; in the center of mass system $\bar{k}_1 + \bar{p}_1$ and $\bar{k}_2' + \bar{p}_2$ vanish; and the Breit frame $\bar{k}_1 + \bar{k}_1'$ or $\bar{p}_1 + \bar{p}_1'$ vanish. For a discussion of the kinematics see Hagedorn, Relativistic Kinematics, page 65.

We will just briefly describe the procedure used to obtain the cross sections using Breit frame kinematics. We eventually want the differential cross section in the laboratory system at convenient laboratory angles. The relationship between the momentum in the Breit frame and the momentum in the lab frame is

$$k_B^2 = \frac{k^2 (1 - q^2/4kM)^2}{1 + q^2/4M^2}$$

where, $k = \frac{E_L}{\hbar c}$, $E_L$ is the incident electron energy in the lab frame, and $\theta_L$ is the lab angle and M is the mass of the target divided by $\hbar c$. Here

$$q^2 = 4kk' \sin^2 \frac{\theta_L}{2}$$

where

$$k' = \frac{k}{1 + \frac{2k}{M} \sin^2 \frac{\theta_L}{2}}$$

Unfortunately, one notices that $k_B$ is a function of the scattering angle. In view of this we calculate the phase shifts (which depend on $k_B$ through the parameters in the charge distribution) not at every $\theta_L$, but at two selected values of $k_B$, $k_{B1}$ and $k_{B2}$, and interpolate between the resulting cross sections to obtain the differential cross section in the Breit frame. We calculate the intermediate cross sections at the Breit frame angles given by

$$\sin^2 \tfrac{1}{2}\theta_B = \frac{k\,k'}{k_B} \sin^2 \tfrac{1}{2}\theta_L .$$

The two k-values $k_{B1}$ and $k_{B2}$ are

$$k_{B1} = k_L$$

and

$$k_{B2} = k_L \left(1 - \frac{k_L}{M} \sin^2 \tfrac{1}{2}\theta_M\right) \qquad (1)$$

where $\theta_M$ is an input parameter, some reasonably large angle. The phase shifts are calculated exactly as described in the previous section, A-1, only they are calculated for the two k-values given above. The interpolated differential cross section in the Breit frame is given by

$$\left(\frac{d\sigma}{d\Omega}\right)_B = \frac{k_B - k_{B2}}{k_{B1} - k_{B2}} \left(\frac{d\sigma}{d\Omega}\right)_{k_{B1}} + \frac{k_{B1} - k_B}{k_{B1} - k_{B2}} \left(\frac{d\sigma}{d\Omega}\right)_{k_{B2}}$$

The transformation of this cross section to the laboratory system is given by

$$\left(\frac{d\sigma}{d\Omega}\right)_L = \left(\frac{d\sigma}{d\Omega}\right)_B \frac{k_L^2}{k_{B2}\left(1 + \frac{2k_L \sin^2 \frac{1}{2}\theta_L}{M}\right)^2}$$

Once this cross section is found it may be folded over the experimental angular uncertainty and a fit to the experimental data may be made just as described in the preceding section.

The default specifications of this program are identical to that of the usual phase shift program RAVENHALL, and the input is the same with one exception. After CARD TWO a card giving the value of $\theta_M$ must be present. Thus the input specifications are:

CARD ONE

GAM = Atomic number of the target divided by the fine structure constant. This is usually given to six places. It must be identical to the value used in the Coulomb function program.

XO = Fitting on radius (program units). This must be identical to the fitting on radius used in the Coulomb function program.

C = Charge distribution parameter in fermis

Z = Charge distribution parameter in fermis

$P_3$ = Charge distribution parameter dimensionless

EO = Energy of the incident electron in MeV (Usually in the lab system)

ATWT = Atomic weight of the target

CARD TWO

DELB = Experimental angular resolution used in folding the cross section over this angular uncertainty

DELO = Step size in the scattering angle (Usually in the lab system)

OTEST = The angle at which we change from the small angle expression for the scattering amplitude to the large angle expression

O1 = Initial value of the scattering angle (Usually in the lab system)

OFIN = The final value of the scattering angle

CONV = Phase shifts ($\tan\delta_n$) smaller than this value will not be calculated. This should be around $10^{-10}$.

TMIN = Time in minutes allowed to complete the calculation of the phase shifts. If this time is exceeded phase shifts on the starting series will be punched out on cards in order that the calculation may continue. If this field is left blank a default value of 15 minutes it taken.

CARD THREE

THMIN = The value of $\theta_M$ appearing in the calculation of $k_{B,2}$ Eq. ( 1 ).

CARD FOUR

NF = Maximum number of phase shifts to be calculated.

NSF = First phase shift to be calculated, normally NST = 1.

NFL = Number of terms in the Faxen-Holtzmark series for the scattering amplitude. A value of 80 is acceptable.

NFOLD, NFOLD = 1 Fold the cross sections over the angular uncertainty, but do not punch them out on cards.

NFOLD = 2 Fold the cross sections over the angular uncertainty, but do punch them out on cards.

NFOLD = 3 Do not fold or punch the cross sections.

NFOLD = 4 Do not fold the cross sections, but do punch them out on cards.

NCM = 0 Make all the necessary transformations - laboratory to the Breit frame and vice versa.

NCM = 1 Do not make the transformations

IC = 0 Do read in the Coulomb functions

IC = 1 Do not read in the Coulomb functions, rather use the Coulomb functions from the previous case.

NER = 0 Do not read in the phase shifts, this is the standard mode

NER = 1 Do read in the phase shifts and initial values. The cards become part of the input stream in this case.

If IC = 0 we read in the Coulomb functions 2 NF cards in all

FRN(I) = Regular function F

GRN(I) = Regular function G

FIN(I) = Irregular function F

GRN(I) = Irregular function G

If the charge distribution is being read in, the appropriate cards would appear at this point. This completes the data cards read in by this version RAVENHALL Program; however, if they are used, both subroutine FOLD and subroutine FIT require additional input cards. These cards would appear after the cards read in by the RAVENHALL Program. Subroutine FOLD requires at least one card and may require several more. See the write up of Subroutine FOLD, page J-1, and Subroutine FIT, page K-1, for the specifications for this input.

The output from this program consists of the input parameters, the potential vs.X and the wave functions at $X_o$, and the values of $\tan\delta_n$ for both $k_{B,1}$ and $k_{B,2}$ as well as the intermediate cross sections. Then the cross sections in the Breit and Lab frames versus angle are given. Finally, any output from subroutine FOLD or FIT will appear.

This version of the RAVENHALL Program is written in FORTRAN IV using double precision arithmetic throughout. The gamma function subroutine, DAMMA, the complex arithmetic subroutine, DOMPLX, the series reduction subroutine, RECUR, the subroutine for storing the potential, POT, the folding subroutine FOLD, the fitting subroutine, FIT, the appropriate charge distribution subroutine FX, and the fourth order R.K.G. integration

routine DEQ are required by this version of the RAVENHALL program. The execution time varies depending on the number of phase shifts calculated, the value of $x_o$, and the angular range covered. As an indication of execution time the calculation of the cross section for 249.3 MeV electrons incident on $Ca^{40}$ using 20 phase shifts and a fitting on radius of 20 for angles from 2 degrees to 140 degrees in steps of 0.5 degress including folding took about 1.6 minutes of CPU time.

## Coulomb Functions for Electron Scattering

The scattering amplitude is determined by the asymptotic form of the scattered wave. This means that one must somehow determine the wave function for very large radial distances. The Dirac equations describe the electron scattered by a Coulomb potential; however, it is not possible to numerically integrate these equations to the asymptotic region. Rather, one must find the asymptotic solutions to the radial Dirac equations outside the nuclear charge distribution and then match these solutions at some fitting on radius to the solutions obtained inside the nuclear charge distribution. From this, one finds the nuclear phase shift as discussed in the section on the phase shift analysis (Page A-1 ).

In the extreme relativistic limit one has only two coupled radial equations for the radial wave function. We, of course, are assuming a spherically symmetric potential. Outside the nuclear charge distribution the Coulomb potential becomes

$$v \rightarrow \frac{-Ze^2}{xhc}$$

where we use the dimensionless form

$$x = kr,$$

$$v = \frac{V}{E},$$

and

$$k = \frac{E}{\hbar c}.$$

In these units the radial Dirac equations become

$$\frac{dGj}{dx} = + \frac{(j+\frac{1}{2})Gj}{x} - \frac{(1-v)Fj}{}$$

$$\frac{dFj}{dx} = - \frac{(j+\frac{1}{2})Fj}{x} + (1-v)Gj \quad ,$$

here j is the total quantum number for spin state $+\frac{1}{2}$, $(l+\frac{1}{2}) = j$. We wish the solution to these equations at the fitting on radius $x_o$. We first find series solutions (see Yennie, Ravenhall and Wilson, Phys. Rev. 95, 5, Pg. 500, Appendix 2).

$$G_j(x) = x^{sj} N_j \sum_{m=o}^{\infty} a_m^{(j)} x^m$$

$$F_j(x) = x^{sj} N_j \sum_{m=o}^{\infty} b_m^{(j)} x^m$$

where the indition equation is

$$s_j = \pm \rho_j = \pm \left[ (j+\tfrac{1}{2})^2 - \gamma^2 \right]^{\frac{1}{2}} .$$

The positive sign gives the solutions regular about the origin, the negative sign the irregular solutions. The normalization $N_j$ is determined by requiring the solutions to have the correct asymptotic form and is given by

$$N_j = \frac{s_j 2^{sj} \left| \Gamma (s_j + i\gamma) \right|}{\left| s_j \right| \cdot (2s_j + 1)} \left[ (j+\tfrac{1}{2}) \frac{(j+\tfrac{1}{2}+s_j)}{2} \right]^{\frac{1}{2}} e^{\frac{\pi\gamma}{2}} .$$

The coefficients in the series solutions have recussion relations given by

$$a_o^{(j)} = 1 \qquad b_o^{(j)} = \frac{\gamma}{\left[s_j + (j + \tfrac{1}{2})\right]}$$

$$m(m + 2s_j)a_m^{(j)} = - \; a_{m-1}^{(j)} \; - (s_j + m + j + \tfrac{1}{2})b_{m-1}^{(j)}$$

$$m(m + 2s_j)b_m^{(j)} = -\gamma\, b_{m-1}^{(j)} + (s_j + m - j - \tfrac{1}{2})a_{m-1}^{(j)} \; .$$

The series solutions are valid for all x; however, when they are evaluated at large values of x, such as the fitting on radius, the terms build up causing considerable cancellation before convergence and consequently a severe loss of significant figures. In order to overcome this difficulty the series is evaluated at

$$x_s = (j + \tfrac{1}{2}) \cdot f$$

where f is a constant less than one. In the calculation of the Coulomb functions f has a default value of 0.75 which has been found to be an acceptable balance between accuracy and speed. Provision has been made for a possible change in this value as will be discussed later. After the regular and irregular series solutions are calculated at $x_s$ and correctly normalized, they are taken to be the initial values in numerically solving the differential equations. Here a fourth order RKG method is used. Double precision arithmetic is employed in all of the calculations.

The calculation of the normalization factor is simplified by noting that

$$N_I^{(j)} N_R^{(j)} = -\frac{|\gamma|}{2\rho} \frac{e^{\pi\gamma} \sin \pi(j + \frac{1}{2} - \rho)}{\cosh \pi\gamma \left[\tan^2[\pi(j + \frac{1}{2} - \rho)] + \tanh^2 \pi\gamma\right]^{\frac{1}{2}}}$$

where $N_I^{(j)}$ is the normalization for the irregular Coulomb function and $N_R^{(j)}$ is the normalization for the regular Coulomb function. In this expression

$$\rho = \left[(j + \tfrac{1}{2})^2 - \gamma^2\right]^{\frac{1}{2}} ,$$

$$|\gamma| < | .$$

Because of this relationship between the regular and irregular normalization one may calculate $N_R^{(j)}$ and then find $N_I^{(j)}$ as

$$N_I^{(j)} = \frac{N_I^{(j)} N_R^{(j)}}{N_R^{(j)}} .$$

The calculation of $N_R^{(j)}$ can cause some problems as the normalization goes approximately like $\frac{2^n n!}{(2n+1)!}$, $n = j + \frac{1}{2}$, which can get very small as n increases. Further, the calculation of the factors separately can cause difficulty as they are individually very large for large n. It may be noted that the factor multiplying the series is really $x_s^{\rho}$ times $N_R^{(j)}$ which may be written

$$x_s^{\rho} N_R^{(j)} = \sqrt{\frac{n(n+\rho)}{2}} \left\{ (2x_s)^n \frac{\left|\Gamma(\rho + i\gamma)\right|}{\Gamma(2\rho + 1)} \right\} (2x_s)^{-(n-\rho)} e^{\frac{1}{2}\pi\gamma}$$

and the factor in braces may be calculated by noting that

$$(2x_s)^n \frac{|\Gamma(\rho + i\gamma)|}{\Gamma(2\rho + 1)} = \frac{|\Gamma(-n + \rho + i\gamma)|}{2\rho\,\Gamma(-2n + 2\rho)} \left\{ \frac{2x_s\sqrt{(-n + \rho)^2 + \gamma^2}}{(-2n + 2\rho)(-2n + 1 + 2\rho} \right.$$

$$\left. \frac{2x_s\sqrt{(-n + 1 + \rho)^2 + \gamma^2}}{(-2n + 2 + 2\rho)(-2n + 3 + 2\rho)} \cdots \frac{2x_s\sqrt{(-n + r + \rho)^2 + \gamma^2}}{(-2n + 2r + 2\rho)(-2n + 2r + 1 + 2\rho)} \cdots \right\}.$$

This product may be written

$$(2x_s)^n \frac{|\Gamma(\rho + i\gamma)|}{\Gamma(2\rho + 1)} = \frac{|\Gamma(-n + \rho + i\gamma)|}{2\rho\,\Gamma(-2n + 2\rho)} \prod_{r=0}^{n-1} \frac{2x_s\sqrt{(-n + \rho + r)^2 + \gamma^2}}{(-2n + 2\rho + 2r)(-2n + 2\rho + 2r + 1)}$$

The arguments of the gamma functions on the right hand side have the property that their magnitude is less than one so that the series expansion used to calculate the gamma function is convergent. The gamma functions are calculated by SUBROUTINE DAMMA described on page G-1. The method used is given in Tables of Higher Mathematical Functions, Vol. I, by H. T. Davies.

Once the properly normalized series solutions are found at $x_s$ they are used as initial values in solving, numerically, the coupled first order differential equations giving the regular and irregular solutions at the fitting on radius $x_o$. As mentioned earlier a fourth order RKG method is used. Perhaps a word of caution may be interjected at this point. One must keep in mind the fact that the numerical solution to the differential equation start at $x_s = (j + \frac{1}{2})f$ where f has been taken to be 0.75 with good results. This means that the interval size in the numerical integration

must be chosen such that one does indeed wind up at $x_0$. If the default step size, $\Delta x = 0.01$ and the default value of f, $f = 0.75$ are used, this will always be the case. If on the other hand one chose $\Delta x = .02$ the final value, for $j = \frac{1}{2}$, would be $x = 10.01$ and for $j = 3/2$, $x_s = 1.5$ and the final $x = 10.01$, etc. If the fitting on radius is taken to be 10.01 then everything is all right as long as the interval size in the RAVENHALL program is consistent with having a fitting on radius of 10.01. If the fitting on radius is not consistent between these two programs then the phase shifts will be incorrect and all is lost. An interval size of 0.01 seems to be a good choice as a balance between convenience, speed and accuracy. It is important that the Coulomb functions be very accurately determined (with $\Delta x = 0.01$ we seem to have about 10 significant figures) as the nuclear phase shift depends very critically on their values.

The program which calculates the Coulomb functions is called, appropriately, the COULOMB FUNCTION PROGRAM. If all of the default options are used there is only one input card required per case. One must provide us input the following

```
READ (5, 11) GAM, XO, NF, NCHECK
11  FORMAT (2D10.5, 2I5)
```

where

GAM = The atomic number times the fine structure constant. This is usually given to six places. Its value must be the same as the value used in the RAVENHALL program.

XO = The fitting on radius. This must be consistent with the RAVENHALL program.

NF = The number of Coulomb functions to be calculated, NF = $(j_{max} + \frac{1}{2})$.

NCHECK $\geq$ 1 Do not write any intermediate output. This is the standard mode.

NCHECK < 1 Do write the intermediate output. This is not ordinarily done.

Now unless one wishes to change the step size or the factor f multiplying $(j + \frac{1}{2})$ to get the value of x for the series solution this is the only input card required. If one wishes to change these default options a recompilation is required. To read in f one removes the C* in columns one and two on cards COUL0860 and COUL0870. This introduces a read statement and format statement into the program. An additional input card is required

```
READ (5, 400) FCTI
400 FORMAT (D10.5)
```

where

FCTI = The value of f desired. It should be less than one.

If one wishes to read in the information about the integration step size one removes the C$ in columns one and two on cards COUL0980 thru COUL1050. This introduces a read statement and format statement into the program.

Additional input cards are required,

```
READ (5, 410) IZ2, (ZV14(I), ZV15(I), NZ002(I), I = 1, IZ2)
410 FORMAT (I5/(2D10.5, I5))
```

here

IZ2 = The number of regions plus one in which integration variables will be specified. For example, if one wishes to specify that an interval size of 0.02 will be used from x-initial to x-final (the fitting on radius) then IZ2 = 2.

ZV14(I) = The initial value of the independent variable in region I. (This will be set in the program for I = 1 to be XS, ZV14(2) = XO in the example above)

ZV15(I) = The step size in the $I^{th}$ region. In the example above ZV15(I) = 0.02, ZV15(2) need not be specified - just leave the columns blank. There is only one step size in the example.

NZ00Z(I) = A frequency print control for the $I^{th}$ region. NZ002(I) = 1 causes every step in the integration to be printed (providing NCHECK is less than one). The default value is 50. In the example NZ002(2) need not be specified.

The COULOMN FUNCTION PROGRAM uses the gamma function subroutine SUBROUTINE DAMMA described on page G-1, the complex arithmetic subroutine SUBROUTINE DOMPLEX described on page F-1, and an external subroutine SUBROUTINE CFDEQF contains the differential equations. It is described on page . Finally, the 360 double precision RKG differential equations routine DEQ is used. All arithmetic is done in double precision. The program is written in FORTRAN IV.

## Mu Mesic Atom Calculations

There are two general programs used in the analysis of mesic x-ray data. In one the energy eigenvalue is found given the parameters in the nuclear charge distribution. In the other a search is made on one of the parameters in the nuclear charge distribution given the energy eigenvalue. These programs are similar in all respects and both are discussed here. Any differences between them will be noted. First we give a brief description of the calculation and then a description of the operation specifications.

There exists considerable interest in mu-mesic atoms as they provide information about the charge distribution of the nucleus. In particular, the analysis of mesic x-ray experiments gives the mean square radius of the nuclear charge distribution very well. In our treatment we assume a static, spherically symmetric nuclear potential. We do not take nuclear polarization into account, but we do include vacuum polarization. The other radiative correction, the Lamb shift, is ignored but could be included (approximately) without too many changes. We treat the negative muon in all respects just like a heavy electron ignoring any possible muon-nuclear force other than the Coulomb force. In dimensionless form the bound state Dirac equation may be written

$$\frac{d}{dx} G(x) = \frac{k}{x} G(x) + [2 + E - V(x)] F(x) \qquad (1)$$

$$\frac{d}{dx} F(x) = -\frac{k}{x} F(x) + [E + V(x)] G(x) \qquad (2)$$

where G(x) and F(x) are the radial wave functions and $k = \mp (j + \frac{1}{2})$ where $j = l \mp \frac{1}{2}$ with l the orbital quantum number and j the total angular momentum quantum number. For example, for k = 1 we have the $1S_{1/2}$ level, k = 2 the $2P_{3/2}$ level and k = -1 the $2P_{1/2}$ level. The dimensionless variable x is

$$x = \frac{r}{\lambda\!\!\!^{-}_r}$$

where $\lambda\!\!\!^{-}_r$ is the reduced Compton wave length of the muon

$$\lambda\!\!\!^{-}_r = \lambda\!\!\!^{-}_\mu /(1 + 0.1134360/A) \tag{3}$$

where A is the atomic weight and $\lambda\!\!\!^{-}_\mu = 1.86753$ f. The potential V(x) is the Coulomb potential obtained from the assumed nuclear charge density suitably corrected for vacuum polarization.

We obtain the Coulomb potential, $V_c(x)$ from the nuclear charge density so that

$$V_c(x) = \frac{\gamma'}{I(\infty)} \left[\frac{I(x)}{x} + J(\infty) - J(x)\right] \tag{4}$$

where

$$I(x) = \int_o^x \rho(x')x'^2 dx' \tag{5}$$

$$J(x) = \int_o^x \rho(x')x' dx' \tag{6}$$

with the nuclear charge density normalized by the requirement that

$$\int_o^\infty \rho(x)x^2 dx = Ze. \tag{7}$$

Here $\gamma'$ is Z times the fine structure constant. The normalization integrals $I(\infty)$ and $J(\infty)$ are calculated first just as in many of the other calculations discussed in this report.

Now let us turn our attention to the vacuum polarization correction. The interaction of the negative muon with the external potential representing the nucleus may be represented by the Feynman diagram

One might refer to Jauch & Rohrlich, Chapter 14, for Feynman rules for interaction of an electron (or muon) with an external potential. The first order vacuum polarization diagram is

where the closed fermion loop is a self energy effect, the photon line dissassociates into a virtual electron positron pair. An external potential which takes into account the polarization of the vacuum due to this electron positron pair has been given by D. G. Ravenhall and J. M. McKenley in July of 1968. The effective potential to be used in the bound state Dirac equation is

$$V(x) = V_c(x) + V_c(x)\,\frac{2}{3\pi}\left\{\ln\frac{\lambda}{C\sqrt{R_{log}^2 + x^2}} - \frac{5}{6}\right.$$

$$\left. + \frac{3\pi}{4\lambda}\sqrt{R_1^2 + x^2} + \frac{\frac{1}{6}\overline{R^2} + \frac{1}{2}R_{log}^2}{x^2 + \frac{\overline{R^2}\,R_{log}^2}{x^2}} + Q\right\}$$

$$Q = \left\{ -\frac{3}{2\lambda^2} \left(\overline{R^2} + x^2\right) + \frac{\pi}{3\lambda^3} \left[ \left(R^2\right)^{3/2} + x^3 \right] \right\}.$$

where

$V_c(x)$ = Coulomb potential

$\lambda$ = Mass of the muon/mass of the electron

$C$ = 1.781072417990198 ($e^{\gamma}$ where $\gamma$ = Eulers constant)

$\alpha$ = Fine structure constant

and

$$R_1 = I(\infty)/J(\infty)$$

$$R_{log} = e^{[K(\infty)/J(\infty)]}$$

$$\overline{R^2} = L(\infty)/I(\infty)$$

where

$$I(\infty) = \int_0^\infty \rho(x)\, x^2\, dx \tag{9}$$

$$J(\infty) = \int_0^\infty \rho(x)\, x\, dx \tag{10}$$

$$K(\infty) = \int_0^\infty \rho(x)\, x\, \ln x\, dx \tag{11}$$

$$L(\infty) = \int_0^\infty \rho(x)\, x^4\, dx \tag{12}$$

The integrals $I(\infty)$ and $J(\infty)$ are already necessary in calculating $V_c(x)$ from the charge density $\rho(x)$. The other two integrals $K(\infty)$ and $L(\infty)$ are

found at the same time. The mean square radius in fermis may also be found as

$$[\langle r^2 \rangle]^{\frac{1}{2}} = \left[\frac{L(\infty)}{I(\infty)}\right]^{\frac{1}{2}} \cdot \lambda'_r \quad . \tag{13}$$

The program is written in such a way that the vacuum polarization may be turned off, however, the default situation includes the vacuum polarization and a recompilation is necessary in order to change this situation.

Before discussing the details of the calculation there are a few other quantities which may be calculated if they are desired. In the past it was desirable to know the mean kinetic and potential energies and quantities related to the muon capture rate. The mean potential energy is given by

$$\langle PE \rangle = \frac{\int_o^\infty [F^2(x) + G^2(x)]V(x)dx}{\int_o^\infty [F^2(x) + G^2(x)]\, dx} \tag{14}$$

and the mean kinetic energy is just

$$\langle T \rangle = -[E - \langle PE \rangle]. \tag{15}$$

From the mean kinetic energy one may calculate the gyromagnetic ratio of a bound muon in the $1S_{1/2}$ state.

$$g_1 = g_o \left[1 - \frac{2}{3}\langle T \rangle\right] \tag{16}$$

where $g_o$ is the g-factor of the free muon given by

$$g_o = 2\left(1 + \frac{\alpha}{2\pi} - 0.328\,\frac{\alpha^2}{\pi^2}\right) .$$

One may also calculate the g-factor for the bound muon from two other expressions

$$g_2 = \frac{4}{3}\, g_o \frac{\int_o^\infty F(x)G(x)x\,dx}{\int_o^\infty [F^2(x) + G^2_{(x)}]dx} \qquad (17)$$

$$g_3 = 2\left\{1 - \frac{4}{3}\,\frac{\int_o^\infty F^2(x)dx}{\int_o^\infty [F^2(x) + G^2(x)]dx}\right\} . \qquad (18)$$

These quantities as well as the g-factor for a point nucleus

$$g_p = g_o\,(1 - \tfrac{1}{3}\,\alpha^2 z^2)$$

and the ratios

$$\Delta g_1 = \frac{g_o - g_1}{g_1}$$

$$\Delta g_2 = \frac{g_o - g_2}{g_2}$$

$$\Delta g_3 = \frac{g_o - g_3}{g_3}$$

may be calculated if desired.

At one time we were interested in finding the effective charge of the nucleus in connection with the muon capture rate. An expression (see Ford and Wills LAMS-2387) for $Z^4_{eff}$ is

$$Z^4_{eff} = \frac{Z}{4\alpha^3 I(\infty)} \left[ \frac{\int_0^\infty [F^2(x) + G^2(x)]\, \rho(x)dx}{\int_0^\infty [F^2(x) + G^2(x)]\, dx} \right] \tag{19}$$

We have also calculated a quantity $Z^4_{edge}$ given by

$$Z^4_{edge} = \frac{Z}{4\alpha^3 K(\infty)} \left[ \frac{\int_0^\infty [F(x)F'(x) + G(x)G'(x)]\rho(x)dx}{\int [F^2(x) + G^2(x)]dx} \right] \tag{20}$$

Recently we have not done anything with these calculations, but they may be included if desired.

Now we will discuss some of the details of our calculation. A most important feature of our calculational procedure is a technique for reducing the round off error in the integration of the Dirac equations by employing two regions of integration with independently known starting values. This is in contrast to the usual procedure in which one starts at a small value of x and integrates out to large x which may result in a large accumulation of error. In the procedure used here one performs an outward integration starting at a small value of x where F(x) and G(x) are given by their series expansions about x = o and integrates to the matching radius $x_o$. Here $x_o$ is something like the first Bohr radius

(see Fig. 1) or a value of x at which the charge distribution is small compared to its central value, i.e., $\rho(x)$ is of order $10^{-8}$ (see Fig. 1). The inward integration starts at a large value of x where F(x) and G(x) are given by their asymptotic solutions and progresses inward to $x = x_o$. In this integration the equivalent Coulomb point charge distribution is used to find $V_c(x)$ which is then corrected for vacuum polarization. The ratios of the solutions are compared at $x = x_o$ to see if the eigenvalue has been found or to see if the parameter in the charge distribution has been found. If the ratios

$$R_{in} = \frac{F_{in}(x_o)}{G_{in}(x_o)}$$

$$R_{out} = \frac{F_{out}(x_o)}{G_{out}(x_o)}$$

agree to within some convergence criteria, taken to be 1 part in $10^{-8}$ by default although this value may be changed, then the eigenvalue or parameter is considered to be found. This technique depends on the consistency of the wave functions at an intermediate x value. Hence, there is no question of testing the regularity of the solutions at large x, whereas in the usual method this must be done and it is at large x where the usual method has maximum round off error.

The values of G(x) and F(x) at $x = x_s$ where $x_s$ us the starting value for the outward integration are found from their series solutions

$$G(x) = \sum_{n=o}^{\infty} a_n x^{n+k}$$

$$k > 0 \qquad (21)$$

$$F(x) = \sum_{n=o}^{\infty} b_n x^{n+k+1}$$

and

$$G(x) = \sum_{n=o}^{\infty} a_n x^{n+|k|+1}$$

$$k < 0 \qquad (22)$$

$$F(x) = \sum_{n=o}^{\infty} b_n x^{|k|}$$

Replacing V(x) by its Taylor series expansion at small x the coefficients are given by

$$a_o = 1$$

$$b_o = \frac{E + V(o)}{2k + 1} \; a_o$$

$$a_1 = o$$

$$k > 0 \qquad (23)$$

$$b_1 = \frac{V'(o)\, a_o}{2k + 2}$$

$$a_2 = \frac{1}{2}[2 - E + V(o)]b_o$$

$$b_2 = \frac{[(E + V(o))a_2 + \frac{1}{2}V'(o)\, a_o]}{2k + 3}$$

$$a_o = \frac{2 - E - V(o)}{2|k| + 1}$$

$$b_o = 1$$

$$k < 0 \qquad (24)$$

$$a_1 = \frac{-V'(o)\, b_o}{2|k| + 1}$$

$$b_1 = 0$$

$$a_2 = \frac{[(2 - E - V(o)\, b_2 - \tfrac{1}{2} V''(o) b_o]}{2|k| + 3}$$

$$b_2 = \tfrac{1}{2}\,(E + V(o)\, a_o$$

where

$$V(o) = -\frac{\delta J(\infty)}{I(\infty)} \frac{2\alpha}{3\pi}\left\{\ln \frac{\lambda}{CR_{log}} - \frac{5}{6} + \frac{3\pi}{4\lambda} R_1 \right.$$

$$\left. - \frac{3}{2\lambda^2} \overline{R^2} + \frac{\pi}{3\lambda^3} (\overline{R^2})^{3/2} \right\}$$

$$= V_c(o)\, V_{vp}(o)$$

$$\delta = Ze^2/hc$$

$$V'(o) = 0$$

$$V''(o) = V_c''(o)\, V_{vp}(o) + V_c(o) \frac{2\alpha}{3\pi}\left\{\frac{-1}{R_{log}^2} \frac{+3\pi}{2\lambda R_1^2}\right.$$

$$\left. + \frac{2(\frac{1}{6}\overline{R^2} + \frac{1}{2} R_{log}^2)}{\overline{R^2}\ R_{log}^2} - \frac{3}{\lambda} \right\}$$

with

$$V_c''(o) = \frac{d^2 Vc}{dx^2}\bigg|_{x=o} = -\frac{2\gamma}{1(\infty)}\left[\frac{5}{6}\rho(o) + \frac{7}{12}\rho'(o).x_s + \frac{9}{40}\rho''(o)x_s^2\right].$$

Using these values from the series solutions at $x_s$ the Dirac equations are numerically integrated using the double precision fourth order RKG routine out to $x = x_o$, where the results are compared with the inward integration.

At sufficiently large values of x (say $3x_o$ for $1s_{1/2}$ states and $10x_o$ for P states) asymptotic solutions to the Dirac equations may be found. These asymptotic solutions are

$$G(x) = e^{-y}\, y^{-\lambda} \sum_{n=o}^{\infty} e_n y^{-n} \tag{25}$$

$$F(x) = e^{-y}\, y^{-\lambda} \sum_{n=o}^{\infty} f_n y^{-n} \tag{26}$$

where

$$y = x\sqrt{(2 - E)\,E}$$

$$\lambda = \gamma\left[\frac{E - 1}{\sqrt{(2 - E)E}}\right]$$

Recursion relations exist between the $f_n$ and the $e_n$ given by

$$\frac{fn}{e_n} = \frac{k - \sigma - n}{k - \sigma + n} \tag{27}$$

$$e_{n+1} = \frac{(k - \sigma + n + 1)}{2(n + 1)} \frac{(k^2 - \sigma^2 - 2\lambda n - n^2)}{(k - \sigma + n)} e_n \tag{28}$$

where

$$\sigma = \gamma\sqrt{\frac{E}{2 - E}} - \lambda \ .$$

The series are summed until succeeding terms do not change the sum by more than one part in $10^8$.

The additional calculations, such as the normalization integral, require the value of various integrals of the final wave functions over the entire range from x = o to x = ∞. These integrals are found by breaking the integral into two or even three regions when necessary. The integrals that are needed are

$$N = \int_o^\infty [F^2(x) + G^2(x)]dx \tag{29}$$

$$V = \int_o^\infty [F^2(x) + G^2(x)]V(x)\ dx \tag{30}$$

$$Z = \int_o^\infty [F^2(x) + G^2(x)]\rho(x)\ dx \tag{31}$$

$$Z_e = \int_o^\infty [F(x)F'(x) + G(x)G'(x)]\ dx \tag{32}$$

$$M = \int_o^\infty F(x)G(x)\ dx \tag{33}$$

$$H = \int_o^\infty F^2(x)\ dx \tag{34}$$

It is necessary that the normalization integral be broken into three regions first

$$N_1 = \int_{x_s}^{x_o} [F^2(x) + G^2(x)]\, dx$$

which may be written

$$\frac{dN_1}{dx} = F^2(x) + G^2(x) \ .$$

The starting value for $N_1(x_s)$ is found from the series solutions for $F(x)$ and $G(x)$ at $x_s$.

$$N_1(x_s) \simeq x_s^{2k+1} \frac{1}{2k+1} + \frac{2a_2 + b_o^2}{2k+3} x_s^2 + \ldots \qquad k > 0$$

$$N_1(x_s) \simeq x_s^{2|k|+1} \frac{1}{2|k|+1} + \frac{a_1^2 + 2b_2}{2|k|+3} x_s^2 + \ldots \qquad k < 0$$

The coefficients $a_j$ and $b_j$ are given in Eqs. (23) and (24). The second integral is

$$N_2 = - \int_{x_L}^{x_o} [F^2(x) + G^2(x)]\, dx$$

where $x_L$ is a very large value of x taken to be

$$x_L = 4x_a$$

where $x_a$ is the value at which we start the inward integration when iterating on E or a parameter in $\rho(x)$. Starting at $x_L$ assures us that the major portion of N is found from $N_1$ and $N_2$, and further that the asymptotic solutions to the Dirac equations certainly hold beyond this point. The value of $N_2(x_L)$ is

found from the asymptotic solutions for G(x) and F(x) given by Eqs. (25) and (26). The final contribution is

$$N_3 = \int_{x_L}^{\infty} [F^2(x) + G^2(x)]\, dx$$

This integral may be evaluated analytically assuming that the asymptotic solutions for F(x) and G(x) hold in this region. The value for $N_3$ is

$$N_3 = x_L^{2\eta} e^{-2\nu x_L} \left\{ \left(1 + \frac{1}{\mu^2}\right) \left[1 + \frac{2\eta}{2\nu x_L} + \frac{2\eta(2\eta+1)}{(2\nu x_L)^2} + \ldots\right] \right.$$

$$\left. + \frac{2}{\nu}\left(e_1 + \frac{f_1}{\mu^2}\right) \left[1 + \frac{2\eta - 1}{2\nu x_L} + \ldots\right] + \frac{1}{\nu x_L^2}\left(2e_2 + \frac{2f_2}{\mu} + e_1^2 + \frac{f_1^2}{\mu^2}\right) \right\}$$

where

$$\nu = \sqrt{(2 - E)\, E}$$

$$\eta = \sqrt{\nu^2 - \gamma^2}$$

$$\mu = \sqrt{\frac{2 - E}{E}}$$

and $e_j$ and $f_j$ are given in Eqs. (27) and (28). The value of $N_3$ has been found to be very much less than $N_2$ or $N_1$ and in fact it is sufficient to take

$$N_3 = \left(1 + \frac{1}{\mu^2}\right) \frac{x_L^{2\eta}}{2\nu} e^{-2\nu x_L}$$

The value of N with the various contributions properly normalized is given by

$$N = N_1(x_o) + \left[\frac{G_{out}(x_o)}{G_{in}(x_o)}\right]^2 (N_2(x_o) + N_3). \qquad (35)$$

In the other integrals, Eqs.(30) through (34), it is sufficient to consider only two regions. The starting series at $x_s$ and the asymptotic value at $x_L$ used for starting the inward integrations in the second region as summarized below.

$$V_1(x) = \int_o^{x_o} [F^2(x) + G^2(x)]\, dx$$

$$V_1(x_s) = x_s^{2k+1}\left[\frac{V(o)}{2k+1} + \frac{V'(o)}{2k+2}\, x_s + \frac{(V(o)\ (2a_2 + b_o^2))\ x_s^2}{2k+3}\right] \qquad k > 0$$

$$V_1(x_s) = x^{2|k|+1}\left[\frac{V(o)}{2|k|+1} + \frac{V'(o)}{2|k|+2}\, x_s + \frac{(V(o)\ (a_1^2 + 2b_o^2) + V''(o))\ x_s^2}{2|k|+3}\right] \qquad k < 0$$

$$V_2(x) = \int_{x_L}^{x_o} \left[F^2(x) + G^2(x)\right] dx$$

$$V_2(x_L) = -\left[F^2(x_L) + G(x_L)\right] \frac{\chi}{x_L}$$

$$V = V_1(x_s) + \left[\frac{G_{out}(x)}{G_{in}(x_o)}\right]^2 V_2(x_o)$$

$$Z_1 = \int_o^{x_o} [F^2(x) + G^2(x)]\rho(x)\, dx$$

$$Z_1(x_s) = x_s^{2k+1}\left[\frac{\rho(o)}{2k+1} + \frac{\rho'(o)}{2k+2}\, x_s + \frac{[\rho(o)\ (2a_2 + b_o^2) + \rho''(o)]\ x_s^2}{2k+3}\right], \qquad k > 0$$

$$Z_1(x_s) = x^{2|k|+1}\left[\frac{\rho(o)}{2|k|+1} + \frac{\rho'(o)}{2|k|+2} + \frac{[\rho(o)(a_1^2 + 2b_2^2) + \rho''(o)]}{2|k|+3}\right], \quad k < 0$$

as $\rho(x)$ is very close to zero beyond $x_o$ only one region is required,

$$Z_2 = \int_o^{x_o} [F(x)\, F'(x) + G(x)\, G'(x)]\, dx$$

$$Z_2(x_s) = x_s^{2k+1}\left[\frac{k}{2k+1}\,\rho(o) + \frac{[(k+1)\, b_1^2\, \rho(o) + \rho'(o)\, k]}{2k+3}\, x_s^2\right], \quad k > 0$$

$$Z_2(x_s) = x_s^{2|k|}\left[\frac{\rho(o)}{2} + \frac{k\rho'(o)x_s}{2|k|+1} + \frac{[\rho''(o)k + \rho(o)(b_2\,(k+3) + a_1^2)]}{2|k|+2}\, x_s^2\right], \quad k < 0$$

Again, only one region is required.

$$M_1(x_o) = \int_o^{x_o} F(x)G(x)\, dx$$

$$M_1(x_s) = x^{2k+3}\left[\frac{b_1}{2k+3} + \frac{b_2 + a_2 b_1}{2k+4}\, x_s + \frac{(b_2 a_2 + b_3)}{2k+5}\, x_s^2 + \ldots\right], \quad k > 0$$

$$M_1(x_s) = x^{2|k|+3}\left[\frac{a_1}{2|k|+3} + \frac{a_2 x_s}{2|k|+4} + \frac{(a_3 + b_2 a_1)}{2|k|+5}\, x_s^2 + \ldots\right], \quad k < 0$$

$$M_2(x_o) = \int_o^{x_o} F(x)G(x)\, dx$$

$$M_2(x_L) = F(x_L)G(x_L)$$

$$M = M_1(x_o) + \left[\frac{G_{out}(x_o)}{G_{in}(x_o)}\right]^2 M_2(x_o)$$

$$H_1(x_o) = \int_o^{x_o} F^2(x)\, dx$$

$$H_1(x_s) = F^2(x_s)$$

$$H_2 = \int_{x_L}^{x_o} F^2(x)\, dx$$

$$H_2(x_L) = F^2(x_L)$$

$$H = H_1(x_o) + \left[\frac{G_{out}(x_o)}{G_{in}(x_o)}\right]^2 H_2(x_o)$$

The contributions from $x_L$ to $\infty$ to integrals V, M and H are small enough to be ignored, and are not calculated.

In an actual calculation it is necessary to have an initial trial eigenvalue or charge distribution parameter in order to start the iteration. In the case of P states or higher states D, F, etc., the trial can be calculated by the program as the relativistic point eigenvalue. See Motland Sneddon, pg. 326, Eq. 104.

$$E = 1 - \left[1 + \frac{\gamma^2}{\left\{1 - j - \frac{1}{2} + \left[(j+\frac{1}{2})^2 - \gamma^2\right]^{\frac{1}{2}}\right\}^2}\right]^{-\frac{1}{2}}$$

This will usually give good results for P states and certainly for D and F states. However the $1S_{1/2}$ state is considerably affected by the finite nuclear size and therefore a better estimate should be used. In Table I we have the $1S_{1/2}$ eigenvalue in MeV for elements from Z = 10 through Z = 83. This eigenvalue was calculated using a fermi shape with

$$t = 2.4 \text{ f}$$

$$c = 1.07 \, A^{1/3} \text{ f}$$

and Z from

$$t = Z \left\{ 4 \log 3 + \ln \left[ \frac{1 + 10/9 \exp(-c/Z)}{1 + 10 \exp(-c/Z)} \right] \right\}$$

This is a shape that doesn't fit the current electron scattering data which requires at least three parameters or the available mesic x-ray data. However, in those cases where no previous calculation or data exist, it may prove to be a useful guide.

When the parameters in the charge distribution are being sought one must be guided by previous results or scale previous results to the case in mind. The trial value of the parameter depends on the shape assumed for the nuclear charge distribution, and no general rule exists. If a bad trial is chosen, one can usually see which way to go to get better agreement by looking at the programs valiant attempts to converge on something. This is also true if the eigenvalue is being sought. We have run into a few cases where the eigenvalue was not found because the wave function was crossing the axis in the vicinity of $x_o$ making it difficult for the program

to converge. In such a case one can move $x_o$ in or out (usually in) and get things to converge nicely.

Now let us examine the default specifications set in both mesic x-ray programs. It was decided to set certain variables in the program rather than make them input variables. This was done purely for convenience. Of course, any of these default specifications may be changed by recompiling the code after the desired default specifications have been changed. We have chosen to set the maximum number of iterations at 10. Almost everything seems to converge by 7 trials if it is going to converge. We have shut off all intermediate output not deemed necessary or at least useful sometimes. This means that we have set NPRINT = 0 & NPRT = 0. We require the ratios of inward and outward wave functions to match to 1 part in $10^8$ at $x_o$, i.e., CONV = $10^{-8}$. They almost always agree to 10 places when they converge. The vacuum polarization correction is made for all energy levels. We set VCON = 1. The second trial eigenvalue or parameter, P, is taken to be

$$E_2 = E_1 + 0.05E_1$$
$$P_2 = P_1 + 0.05P_1$$

We have set DX = 0.05. This seems to work alright. We have taken the integration variables for the outward integration to be

$$\Delta x = 0.01 \quad \text{for} \quad 0.01 \leq x \leq 0.20$$
$$x_s = 0.01$$
$$\Delta x = 0.10 \quad \text{for} \quad 0.20 \leq x \leq x_o$$

$x_o$ is an input parameter. In the inward integration we have

$$\Delta x = -0.10 \quad \text{for} \quad x_o \leq x \leq x_a$$

and $x_a$ is an input parameter. The inward integration step size is negative as we integrate along decreasing x values. These values for the integration parameters have been used in almost all of our previous work and seem to be a good compromise between speed and accuracy.

Now we will discuss the input requirements for the two mesic x-ray programs. In the standard operating mode only two input cards are required per case.

CARD ONE - The same for both programs

ATNO = Atomic number of the nucleus

AN = Atomic weight ($C^{12}$ scale)

C = Parameter in the charge distribution in fermis (possibly a trial value)

Z = Parameter in the charge distribution in fermis (possibly a trial value)

P3 = Parameter in the charge distribution dimensionless (possibly a trial value)

XF = The value of the matching radius (in units of $r/\chi_\mu$)

If the parameter in the charge distribution is being sought, its trial value and the values of the other two parameters are given on CARD ONE as indicated above.

CARD TWO - Program to find the eigenvalue

NREAD, NREAD = 0 Use the trial eigenvalue read in

NREAD = 1 Use the relativistic point eigenvalue as the first trial

FK = The quantum number of $k = \pm (j + \frac{1}{2})$ appearing in the Dirac equations

FP = Orbital angular momentum quantun number

FJ = Spin quantum number

E = Trial eigenvalue in MeV if NREAD = 0; otherwise may be left blank

CARD TWO - Program to find the charge distribution parameter

IP, IP = 0 Search on C

IP = 1 Search on Z

IP = 2 Search on P3 (the third parameter which is assumed to be something like the W in a parabolic fermi and dimensionless)

FK = The quantum number $k = \pm (j + \frac{1}{2})$ appearing in the Dirac equations

FP = Orbital angular momentum quantum number

FJ = Spin quantum number

E = The eigenvalue for the state in MeV.

The above two cards are all that is required in general. If one were reading in the charge distribution these cards would follow the above two. See the write up of the subroutine to read in charge distributions.

The output from the mesic x-ray programs consists of the input specifications, the initial values of the wave functions in each region and their values of the joining radius for each trial. Also at the joining radius we have the ratio of the wave functions, the potential and charge distribution for each trial. We, of course, print out the energy eigenvalue in MeV and dimensionless units and the parameter values for the parameter search for each trial.

The programs are written in FORTRAN IV using double precision arithmetic throughout. The subroutines used are FX which specifies the nuclear charge distribution, a subroutine VSUB which gives the point Coulomb potential for the inward integration, an external subroutine EDEQF which is used by both programs and which specifies the differential equations. The double precision RKG differential equations routine DEQ is used. The execution time depends on the case involved. As an indication the $1S_{y2}$, $2P_{y2}$ and $2P_{3/2}$ eigenvalues for $Ba^{138}$ were calculated in an equivalent CPU time of less than 0.6 sec values for c, z and W were found for a parabolic modified gaussian shape from the $1S_{y2}$ energy eigenvalue of $Ba^{138}$ in an equivalent CPU time of less than 0.5 sec.

TABLE I

| Element | AT. NO. | AT. WT. | c | $c/A^{1/3}$ | Z | T | $R^*$ | E(MeV) |
|---|---|---|---|---|---|---|---|---|
| Ne | 10 | 20.18 | 2.913 | 1.070 | 0.552 | 2.40 | 3.937 | 0.2772 |
| Na | 11 | 23.00 | 3.043 | 1.070 | 0.550 | 2.40 | 4.028 | 0.3346 |
| Mg | 12 | 24.32 | 3.100 | 1.070 | 0.550 | 2.40 | 4.071 | 0.3970 |
| Al | 13 | 26.99 | 3.210 | 1.070 | 0.549 | 2.40 | 4.153 | 0.4644 |
| Si | 14 | 28.10 | 3.253 | 1.070 | 0.549 | 2.40 | 4.186 | 0.5368 |
| P | 15 | 30.98 | 3.361 | 1.070 | 0.549 | 2.40 | 4.270 | 0.6136 |
| S | 16 | 32.07 | 3.400 | 1.070 | 0.548 | 2.40 | 4.298 | 0.6954 |
| Cl | 17 | 35.47 | 3.516 | 1.070 | 0.548 | 2.40 | 4.391 | 0.7811 |
| A | 18 | 39.96 | 3.658 | 1.070 | 0.547 | 2.40 | 4.502 | 0.8706 |
| K | 19 | 39.11 | 3.632 | 1.070 | 0.547 | 2.40 | 4.481 | 0.9662 |
| Ca | 20 | 40.08 | 3.662 | 1.070 | 0.547 | 2.40 | 4.506 | 1.0653 |
| Sc | 21 | 44.97 | 3.805 | 1.070 | 0.547 | 2.40 | 4.623 | 1.1663 |
| Ti | 22 | 47.90 | 3.886 | 1.070 | 0.547 | 2.40 | 4.689 | 1.2719 |
| V | 23 | 50.96 | 3.967 | 1.070 | 0.547 | 2.40 | 4.757 | 1.3809 |
| Cr | 24 | 52.01 | 3.994 | 1.070 | 0.547 | 2.40 | 4.779 | 1.4950 |
| Mn | 25 | 54.96 | 4.068 | 1.070 | 0.547 | 2.40 | 4.841 | 1.6108 |
| Fe | 26 | 55.86 | 4.090 | 1.070 | 0.547 | 2.40 | 4.860 | 1.7320 |
| Co | 27 | 58.95 | 4.164 | 1.070 | 0.547 | 2.40 | 4.922 | 1.8539 |
| Ni | 28 | 58.72 | 4.159 | 1.070 | 0.547 | 2.40 | 4.918 | 1.9830 |
| Cu | 29 | 63.57 | 4.270 | 1.070 | 0.547 | 2.40 | 5.012 | 2.1082 |
| Zn | 30 | 65.40 | 4.311 | 1.070 | 0.547 | 2.40 | 5.047 | 2.2403 |
| Ga | 31 | 69.74 | 4.404 | 1.070 | 0.546 | 2.40 | 5.124 | 2.3712 |
| Ge | 32 | 72.65 | 4.465 | 1.070 | 0.546 | 2.40 | 5.177 | 2.5064 |
| As | 33 | 74.95 | 4.511 | 1.070 | 0.546 | 2.40 | 5.217 | 2.6451 |
| Se | 34 | 79.02 | 4.592 | 1.070 | 0.546 | 2.40 | 5.287 | 2.7825 |
| Br | 35 | 79.93 | 4.609 | 1.070 | 0.546 | 2.40 | 5.302 | 2.9284 |
| Kr | 36 | 83.83 | 4.683 | 1.070 | 0.546 | 2.40 | 5.366 | 3.0700 |
| Rb | 37 | 85.50 | 4.714 | 1.070 | 0.546 | 2.40 | 5.393 | 3.2185 |
| Sr | 38 | 87.64 | 4.753 | 1.070 | 0.546 | 2.40 | 5.427 | 3.3680 |
| Y | 39 | 88.93 | 4.776 | 1.070 | 0.546 | 2.40 | 5.448 | 3.5216 |
| Zr | 40 | 91.25 | 4.817 | 1.070 | 0.546 | 2.40 | 5.484 | 3.6744 |
| Cb | 41 | 92.94 | 4.847 | 1.070 | 0.546 | 2.40 | 5.510 | 3.8307 |
| Mo | 42 | 95.91 | 4.898 | 1.070 | 0.546 | 2.40 | 5.555 | 3.9848 |
| Tc | 43 | 99.00 | 4.984 | 1.077 | 0.546 | 2.40 | 5.631 | 4.1332 |
| Ru | 44 | 101.04 | 4.984 | 1.070 | 0.546 | 2.40 | 5.631 | 4.2997 |
| Rh | 45 | 102.94 | 5.015 | 1.070 | 0.546 | 2.40 | 5.658 | 4.4615 |
| Pd | 46 | 106.39 | 5.070 | 1.070 | 0.546 | 2.40 | 5.707 | 4.6192 |
| Ag | 47 | 107.91 | 5.094 | 1.070 | 0.546 | 2.40 | 5.728 | 4.7851 |
| Cd | 48 | 112.47 | 5.165 | 1.070 | 0.546 | 2.40 | 5.792 | 4.9404 |
| In | 49 | 114.86 | 5.201 | 1.070 | 0.546 | 2.40 | 5.824 | 5.1052 |
| Sn | 50 | 118.78 | 5.260 | 1.070 | 0.546 | 2.40 | 5.876 | 5.2645 |
| Sb | 51 | 121.80 | 5.304 | 1.070 | 0.546 | 2.40 | 5.916 | 5.4284 |
| Te | 52 | 127.68 | 5.388 | 1.070 | 0.546 | 2.40 | 5.991 | 5.5807 |
| I | 53 | 126.94 | 5.378 | 1.070 | 0.546 | 2.40 | 5.982 | 5.7625 |

*Here R is the equivalent radius defined as $R = \left[\frac{5}{3} \langle r^2 \rangle\right]^{\frac{1}{2}}$.

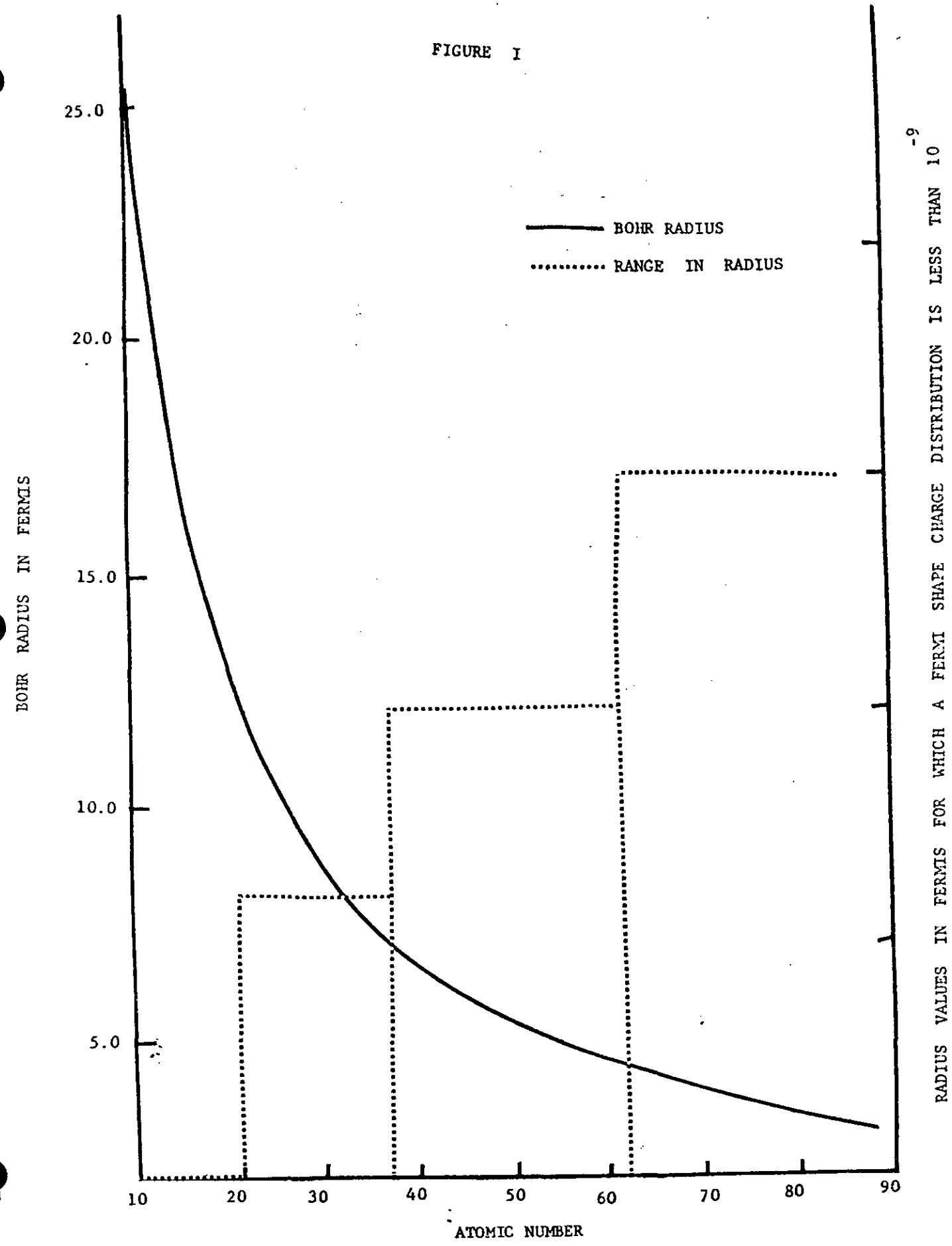
TABLE I (cont.)

| Element | AT. NO. | AT. WT. | c | $c/A^{1/3}$ | Z | T | R * | E(MeV) |
|---|---|---|---|---|---|---|---|---|
| Xe | 54 | 131.34 | 5.349 | 1.070 | 0.546 | 2.40 | 6.037 | 5.9225 |
| Cs | 55 | 132.94 | 5.461 | 1.070 | 0.546 | 2.40 | 6.057 | 6.0961 |
| Ba | 56 | 137.37 | 5.521 | 1.070 | 0.546 | 2.40 | 6.111 | 6.2568 |
| La | 57 | 138.95 | 5.542 | 1.070 | 0.546 | 2.40 | 6.130 | 6.4320 |
| Ce | 58 | 140.15 | 5.558 | 1.070 | 0.546 | 2.40 | 6.145 | 6.6101 |
| Pr | 59 | 140.95 | 5.569 | 1.070 | 0.546 | 2.40 | 6.155 | 6.7911 |
| Nd | 60 | 144.27 | 5.612 | 1.070 | 0.546 | 2.40 | 6.194 | 6.9598 |
| Pm | 61 | 145.00 | 5.621 | 1.070 | 0.546 | 2.40 | 6.202 | 7.1435 |
| Sm | 62 | 150.35 | 5.690 | 1.070 | 0.546 | 2.40 | 6.264 | 7.3011 |
| Eu | 63 | 152.01 | 5.711 | 1.070 | 0.546 | 2.40 | 6.283 | 7.4804 |
| Gd | 64 | 157.31 | 5.776 | 1.070 | 0.546 | 2.40 | 6.343 | 7.6392 |
| Tb | 65 | 158.97 | 5.796 | 1.070 | 0.546 | 2.40 | 6.361 | 7.8194 |
| Dy | 66 | 162.54 | 5.839 | 1.070 | 0.546 | 2.40 | 6.400 | 7.9885 |
| Ho | 67 | 164.98 | 5.869 | 1.070 | 0.546 | 2.40 | 6.427 | 8.1643 |
| Er | 68 | 167.31 | 5.896 | 1.070 | 0.546 | 2.40 | 6.452 | 8.3420 |
| Tm | 69 | 169.00 | 5.916 | 1.070 | 0.546 | 2.40 | 6.470 | 8.5238 |
| Yb | 70 | 173.10 | 5.963 | 1.070 | 0.546 | 2.40 | 6.513 | 8.6906 |
| Lu | 71 | 175.03 | 5.985 | 1.070 | 0.546 | 2.40 | 6.533 | 8.8716 |
| Hf | 72 | 178.55 | 6.025 | 1.070 | 0.546 | 2.40 | 6.570 | 9.0422 |
| Ta | 73 | 181.00 | 6.053 | 1.070 | 0.546 | 2.40 | 6.596 | 9.2199 |
| W | 74 | 183.78 | 6.083 | 1.070 | 0.546 | 2.40 | 6.623 | 9.3964 |
| Re | 75 | 186.27 | 6.111 | 1.070 | 0.546 | 2.40 | 6.649 | 9.5742 |
| Os | 76 | 190.39 | 6.156 | 1.070 | 0.546 | 2.40 | 6.690 | 9.7407 |
| Ir | 77 | 192.25 | 6.176 | 1.070 | 0.546 | 2.40 | 6.709 | 9.9237 |
| Pt | 78 | 194.91 | 6.204 | 1.070 | 0.546 | 2.40 | 6.735 | 10.1014 |
| Au | 79 | 197.03 | 6.226 | 1.070 | 0.546 | 2.40 | 6.755 | 10.2833 |
| Hg | 80 | 200.67 | 6.264 | 1.070 | 0.546 | 2.40 | 6.790 | 10.4536 |
| Tl | 81 | 204.45 | 6.303 | 1.070 | 0.546 | 2.40 | 6.826 | 10.6228 |
| Pb | 82 | 207.26 | 6.332 | 1.070 | 0.546 | 2.40 | 6.853 | 10.7992 |
| Bi | 83 | 209.05 | 6.350 | 1.070 | 0.546 | 2.40 | 6.869 | 10.9839 |

*Here R is the equivalent radius defined as

$$R = \left[\frac{5}{3} < r^2 >\right]^{\frac{1}{2}} .$$

FIGURE I


25.0
20.0
15.0
10.0
5.0
BOHR RADIUS IN FERMIS
BOHR RADIUS
RANGE IN RADIUS
10
20
30
40
50
60
70
80
90
ATOMIC NUMBER
RADIUS VALUES IN FERMIS FOR WHICH A FERMI SHAPE CHARGE DISTRIBUTION IS LESS THAN 10 -9

## CHARGE DISTRIBUTION SUBROUTINES

The charge distribution subroutines are set up as three parameter distributions. It is simple to extend the number of parameters by expanding the call sequence; however, this would necessitate changing the programs which call these subroutines. The simplest method is to allow the additional parameters to be either specified as assignment statements or read in. The 360/65 versions of the charge distribution routines will work in all of the 360/65 programs, the phase shift analysis, mesic x-ray programs, charge distribution programs, etc.

The charge distributions are used to calculate the Coulomb potential which appears in the Dirac equations. We write

$$V_c(x) = -\frac{1}{x}\int_o^x \rho(x')x'^2dx' - \int_x^\infty \rho(x')x'dx'$$

where $V_c(x)$ is the Coulomb potential in units appropriate calculation being done and

$$\rho(x) = \rho \quad f(x)$$

is the charge distribution. The function f(x) is calculated in the charge distribution subroutines, called SUBROUTINE FX. We normalize $\rho(x)$, in general, as follows,

$$\int_0^\infty \rho(x) x^2 dx = \frac{Ze^2}{hc} = \gamma \ .$$

For convenience define

$$I(x) = \int_0^x f(x') x'^2 dx'$$

$$J(x) = \int_0^x f(x') x' dx'$$

so that

$$\rho \, I(\infty) = \gamma \ .$$

The potential may then be written

$$V_c(x) = \frac{-\gamma}{I(\infty)} \left[ \frac{1}{x} I(x) + J(\infty) - J(x) \right] ,$$

and it is obtained by numerically integrating

$$\frac{d}{dx} I(x) = x^2 f(x)$$

$$\frac{d}{dx} J(x) = x \, f(x)$$

using a fourth order Runge-Kutta-Gill method.

Now we give a summary of the functional forms used for f(x), and a brief discussion of some special cases.

Fermi

$$f(x) = \frac{1}{1 + \exp[(x - c)/z]}$$

Modified Gaussian

$$f(x) = \frac{1}{1 + \exp[(x^2 - c^2)/z^2]}$$

Parabolic Fermi

$$f(x) = \frac{1 + W(x^2/c^2)}{1 + \exp[(x - c)/z]}$$

Parabolic Modified Gaussian

$$f(x) = \frac{1 + W(x^2/c^2)}{1 + \exp[(x^2 - c^2)/z^2]}$$

Parabolic (Fermi to the N)

$$f(x) = \frac{1 + W(x^2/c^2)}{\{1 + \exp[(x - c)/z]\}^N} \quad *$$

M. S. G.

$$f(x) = \frac{1 + W(x^2/c^2)}{1 + \exp[(x^{2.5} - c^{2.5})/z^{2.5}]}$$

Old Three Parameter

$$f(x) = \frac{1}{1 + \exp[(x - c)/z]} + \frac{W(x^2/c^2)}{\{1 + \exp[(x - c)/z]\}^2}$$

Old Three Parameter - Modified Gaussian

$$f(x) = \frac{1}{1 + \exp[(x^2 - c^2)/z^2} + \frac{W(x^2/c^2)}{\{1 + \exp[(x^2 - c^2)/z^2]\}^2}$$

*We have special programs for N = 3, N = 2, N = 1.75, N = 1.5 and N = 1.25.

Modified Parabolic Fermi, N = 1.5 $$f(x) = \frac{1 + W(x/c)^2}{1 + \exp\left[(x^{1.5} - c^{1.5})/z^{1.5}\right]}$$

Uniform $$f(x) = 1.0 \qquad x \le c$$

$$f(x) = 0.0 \qquad x > c$$

Three Parameter $$f(x) = \frac{1}{1 + \exp\left[(x^{W} - c^{W})/z^{W}\right]}$$

Heisenberg $$f(x) = \frac{1 + w(x/c)^2 + A\left[\exp\left[-(bx)^2\right] + \cos(ex)\right]/R}{1 + \exp\left[(x^2 - c^2)/z^2\right]}$$

Bethe - I $$f(x) = 1 - \tfrac{1}{2}\left\{\exp\left[-(x-c)/z\right]\right\}^2 \qquad x \le c$$

$$f(x) = \tfrac{1}{2}\exp\left[(c-x)/w\right] \qquad x > c$$

Bethe - II $$f(x) = \frac{1 - 1/3\exp\left[(x-c)/z\right]}{\left\{1 + 1/3\exp\left[(x-c)/z\right]\right\}^2} \qquad x \le c$$

$$f(x) = \tfrac{1}{2}\exp\left[(c-x)/x\right] \qquad x > c$$

Ford and Wills

$$f(x) = 1.0 - \tfrac{1}{2}\exp[(x - c)/z] \quad x \le c$$

$$= \tfrac{1}{2}\exp[-(x - c)/z] \quad x > c$$

Trapezoidal

$$f(x) = 1.0 \quad x \le (c - z)$$

$$= c + z - \frac{x}{2z} \quad x \le (c + z)$$

$$= 0.0 \quad x > (c + z)$$

Harmonic Well

$$f(x) = [1 + W(x^2/c^2)]\exp[-(x^2/c^2)]$$

I will discuss now two special cases. First, the case where the charge distribution is read in. This care presupposes the existance of f(x) vs. x on cards or tape in a form suitable to be read in. We always use double precision in our calculation of cross sections and mesic x-ray levels so f(x) must be specified as double precision. The derivatives for starting series are calculated using simple difference methods. We have found it convenient to read in two pieces of information in addition to f(x) itself, first the number of points and second the final value of x to be considered. Beyond this value f(x) is assumed zero. It may be necessary to change formats in this routine depending on the data at hand and one must remember to include all information such as the number of points and the final x value as additional input along with f(x) vs. x, to whatever program is being executed. A glance at the appropriate program tells where these cards should be included, i.e., after the input cards from the read statement

immediately preceeding the first call to SUBROUTINE FX - note that M = 1 for this call, as will be discussed later.

The second special case to be considered is really a class of many special charge distributions which have been determined from requiring that these charge distributions produce a form factor in agreement with a particular experimental form factor. In view of this it is very difficult to specify such a charge distribution. In general, a special routine will have to be written. As an example we might take the charge distribution which produces a good fit to Pb-208 cross sections at 500 and 750 MeV. It is necessary to modify the charge distribution determined by the lower energy experiments by subtracting a $\Delta\rho(r)$ given by

$$\Delta\rho(r) = \frac{4\pi Z}{2\pi^2}\ \frac{A}{3}\ \sqrt{\pi}\ \frac{1}{r^2}\left\{e^{-(b-a)^2/4}\ [\theta\cos(\phi-\theta) - \tfrac{1}{2}\, b(a-b)\operatorname{sen}(\phi-\theta)]\right.$$

$$\left. -e^{-(b+a)^2/4}\ [\ \theta\cos(\phi+\theta)\ -\tfrac{1}{2}\, b(a+b)\sin(\theta+\phi)]\right\},$$

where $o = q_o r$ — $q_o$ an input parameter

$b = \frac{2r}{z}$ — (here z is the z in the charge distribution Z = atomic number)

A = an input parameter

$\phi$ = an input parameter

a = an input parameter

Now we calculate in terms of a parameter x, and therefore we must

change units accordingly. For the phase shift analysis we need to feed in two quantities

$$k = \frac{E}{\hbar c}$$

this is printed out in the phase shift output, and the normalization integral $I(\infty)$ for the unmodified charge distribution. We then change variables so that:

$$\rho(x)_{new} = \frac{k^3 Z}{4\pi I(\infty)} \left\{ f(x)_{unmodified} - 4\pi I(\infty) \frac{1}{(2\pi)^2} \frac{A}{kz} \pi \frac{1}{x^2} \left\{ \right.\right.$$

$$e^{-(b-a)^2/4} [\theta\cos(\phi-\theta) - \tfrac{1}{2} b(a-b) \sin(\phi-\theta)] - e^{-(b+a)^2/4} [\theta \cos(\phi+\theta)$$

$$\left.\left. -\tfrac{1}{2} b(a+b) \sin(\theta+\phi) \right\} \right\} ,$$

where the overall factor $\frac{k^3 Z}{4\pi I(\infty)}$ may now be dropped and

$$\theta = \frac{(q_o)x}{k}$$

here x is in units of kr, thus c, z are specified in fermis and then converted to dimensionless units kr for the computations. In this case a parabolic modified gaussian was the unmodified charge distribution.

This case can only serve as an example. In general a new charge distribution would have to be written and the unmodified charge distribution would in general be different.

The subroutine which calculates the quantity f(x) is called SUBROUTINE FX. Its calling sequence and argument list are

CALL FX(M, FI, FJ, C, Z, DX, X, F, V, W, FO, F1, F2)

where

| | |
|---|---|
| M ≤ 1 | Calculate the initial values of the derivatives of f(x) and V(x) for the starting series. Also, if additional quantities are required as in the special cases mentioned above, the input is read in here. |
| M > 1 | Calculate f(x) for a given value of x. |
| FI = | The initial value of the integral I(x) calculated at X = DX, the starting point of the integration. |
| FJ = | The initial value of the integral J(x) calculated at X = DX, the starting point of the integration. |
| C = | A parameter in the charge distribution f(x). |
| Z = | A parameter in the charge distribution f(x) |
| DX = | The initial value of X for the integration. This is, for example, the step size in the phase shift calculation. |

X = The independent variable

F = The value of f(x)

V = The initial value of the second derivative of V(x) calculated at X = 0X.

N = A parameter in the charge distribution f(x).

F0 = The value of f(x) at X = 0

F1 = The value of the first derivative of f(x) at X = 0.

F2 = The value of the second derivative of f(x) at X = 0.

This subroutine returns to the main program the value of f(x) for a given value of x. The subroutine for calculating f(x) is used in the phase shift analysis, the mesic x-ray programs and in charge distribution and form factor programs. This routine is written in FORTRAN IV and uses double precision arithmetic.

## Complex Arithmetic Subroutine

We have a subroutine that performs standard arithmetic on double precision complex variables. The results are returned in the form of a separate, real and imaginary part. This routine was originally necessary as complex, double precision variables were not defined in that the complex and double precision attributes could not be assigned to the same variable. This is no longer a restriction in FORTRAN IV for the 360/65; however, considerable recoding would be necessary to use this current facility in FORTRAN IV and it has not been done. Perhaps in future calculations one would want to make use of this new facility.

The complex arithmetic subroutine is used extensively in the phase shift analysis of electron scattering experiments. The subroutine name is DOMPLX. Its calling sequence is

```
CALL DOMPLX(X1, Y1, X2, Y2, X3, Y3, N)
```

where the first complex variable is defined by Z1 = X1 + iY1, the second by Z2 = X2 + iY2 and the result of the complex arithmetic performed with Z1 and Z2 is given by Z3 = X3 + iY3. The integer N controls what the operation is in the equation

$$Z1 \otimes Z2 = Z3.$$

Thus

X1 = Real part of Z1

Y1 = Imaginary part of Z1

X2 = Real part of Z2

Y2 = Imaginary part of Z2

X3 = Real part of Z3

Y3 = Imaginary part of Z3

N = Integer control specifying the operation.

The variable N is defined as follows:

N = 1 The operation is addition

$$Z3 = Z1 + Z2$$

N = 2 The operation is subtraction

$$Z3 = Z1 - Z2$$

N = 3 The operation is multiplication

$$Z3 = Z1 \times Z2$$

N = 4 The operation is division

$$Z3 = \frac{Z1}{Z2}$$

N = 5 The reciprocal of the variable Z1 is found

$$Z3 = \frac{1}{Z1}$$

N = 6 The operation is to separate $e^{iY1}$ into real and imaginary parts so that

$$X3 = \cos(Y1)$$

$$Y3 = \sin(Y1)$$

N = 7 The operation is to separate $\sin(X1 + iY1)$ into its real and imaginary parts so that

$$X3 = \sin(X1)\cosh(Y1)$$

$$Y3 = \cos(X1)\sinh(Y1)$$

N = 8 The operation is to separate $\cos(X1 + iY1)$ into its real and imaginary parts so that

$$X3 = \cos(X1)\cosh(Y1)$$

$$Y3 = -\sin(X1)\sinh(Y1)$$

The appropriate values of X1, Y1, X2, Y2 and N are provided by the calling program, the subroutine calculates the values of X3 and Y3. The subroutine is written in FORTRAN IV, and uses double precision arithmetic throughout.

## GAMMA FUNCTIONS WITH COMPLEX ARGUMENTS

We wish to calculate gamma functions of the kind

$$\Gamma(\rho_n + i\gamma)$$

where

$$\rho_n = +\sqrt{n^2 + \gamma^2}$$

and $n = 1, 2, 3, \ldots n_{max}$ where $n_{max}$ may be as large as 80. In this calculation we use an expansion for $[\Gamma(z)]^{-1}$ which is rapidly convergent for $|z|<1$. The series expansion is

$$\frac{1}{\Gamma(z)} = z(z + 1)\ (1 + B_1 z + B_2 z^2 + \ldots)$$

where the $B_n$ are as follows, $B_o = 1$

| n | $B_n$ |
|---|---|
| 1 | -0.4227 8433 5098 4671 |
| 2 | -0.2330 9373 6421 7867 |
| 3 | +0.1910 9110 1387 6915 |
| 4 | -0.0245 5249 0005 4000 |
| 5 | -0.0176 4524 4550 1443 |
| 6 | +0.0080 2327 3022 2673 |
| 7 | -0.0008 0432 9775 6043 |
| 8 | -0.0003 6083 7816 2548 |
| 9 | +0.0001 4559 6142 1399 |

| n | $B_n$ | | | |
|---|---|---|---|---|
| 10 | -0.0000 | 1754 | 5859 | 7517 |
| 11 | -0.0000 | 0258 | 8995 | 0290 |
| 12 | +0.0000 | 0133 | 8501 | 5469 |
| 13 | -0.0000 | 0020 | 5474 | 3149 |
| 14 | -0.0000 | 0000 | 0159 | 5268 |
| 15 | +0.0000 | 0000 | 6275 | 6218 |
| 16 | -0.0000 | 0000 | 1273 | 6143 |
| 17 | +0.0000 | 0000 | 0092 | 3397 |
| 18 | +0.0000 | 0000 | 0012 | 0030 |
| 19 | -0.0000 | 0000 | 0004 | 2207 |
| 20 | +0.0000 | 0000 | 0000 | 5239 |
| 21 | -0.0000 | 0000 | 0000 | 0139 |
| 22 | -0.0000 | 0000 | 0000 | 0067 |

One can see that the coefficients converge rather rapidly. The reference for this expansion is Tables of Higher Mathematical Functions by H. T. Davis, Principia Press, 1933, Vol. I, Page 184.

The procedure for calculating $\Gamma(\rho_n + i\gamma)$ is as follows:

1) For a given n calculate $\rho_n$

2) Now calculate $\rho_n - n$ which is less than 1

3) Now we may write

$$\frac{1}{\Gamma(\rho_n - n + i\ )} = \left[ z(z+1) \sum_{m=0}^{22} B_m z^m \right]_{z = \rho_n - n + i\gamma}$$

4) Now invert the resulting complex number to get $\Gamma(\rho_n - n + i\gamma)$

5) We next use the property that

$$z\Gamma(z) = \Gamma(z + 1)$$

and we find

$$\Gamma(\rho_n + i\gamma) = \Gamma(\rho_n - n + i\gamma)\Big\{(\rho_n - n + i\gamma)(\rho_n - n + 1 + i\gamma)(\rho_n - n + z + i\gamma) \ldots x(\rho_n - n + (n - 1)\, i\gamma)\Big\}$$

where the product is formed as follows:

let $$z = \rho_n + i\gamma$$

$$\Gamma(z) = \Gamma(z - n)\Big\{(z - n)(z - n + 1) \ldots (z - n + n - 1)\Big\}$$

n = 1

$$\Gamma(z) = \Gamma(z - 1)\Big\{(z - 1)\Big\}$$

n = 2

$$\Gamma(z) = \Gamma(z - 2)\Big\{(z - 2)(z - 1)\Big\}$$

the number of terms in the product is just n, the calculation is conveniently done in a loop terminating at n. One must, however, be cautious in this calculation as we are essentially calculating n!. The problem of overflow is taken care of by simply scaling the gamma function, and, of course, keeping track of this scaling so that it can enter into subsequent calculations in a correct manner.

The subroutine which calculates the real and imaginary parts of $\Gamma(\rho_n + i\gamma)$ is called SUBROUTINE DAMMA, its arguments and calling sequence are given by

CALL DAMMA (A1, A2, B1, B2, N, NTEST, NFACT)

A1 = real part of the argument = $\rho_n$ - n

A2 = imaginary part of the argument

B1 = real part of $\Gamma(\rho_n + i\gamma)$ returned to the main program

B2 = imaginary part of $\Gamma(\rho_n + i\gamma)$ returned to the main program

N = the value of n

NTEST = the number of factors of $10^{+30}$ taken out during scaling

NFACT = 0 do not attempt any scaling

NFACT $\neq$ 0 do test to see if scaling is necessary. The values of A1 and A2 are destroyed. Subroutine DAMMA uses the complex arithmetic subroutine SUBROUTINE DOMPLX to do the necessary complex arithmetic. This calculation is done using double precision arithmetic. The program is written in FORTRAN IV.

Reduction of an Expansion in
Legendre Polynominals for Improvement of Convergence

Consider the following expansion

$$f(\Theta) = \sum_{\ell=0}^{\infty} A_\ell P_\ell(\cos\Theta)$$

it has been shown by Yennie, Ravenhall and Wilson (Phys. Rev., 95, P. 500 (1954)) that a series which converges more rapidly may be found by noting that

$$(1 - \cos\Theta)^m f(\Theta) = \sum A_\ell^m P_\ell(\cos\Theta)$$

where

$$A_\ell^{i+1} = A_\ell^i - \frac{\ell+1}{2\ell+3} A_{\ell+1}^i - \frac{\ell}{2\ell-1} A_{\ell-1}^i .$$

For large values of $\ell$ it turns out that

$$\left|A_\ell^{i+1}\right| = O(|A_\ell^i|/\ell^2)$$

so that the coefficients in the reduced series do decrease in magnitude more rapidly than the original series. It has been useful to carry out this reduction three times (m = 3) in the electron scattering phase shift analysis. One should note that each reduction reduces the number of terms in the series by 1, i.e., the

maximum $\ell$ value, $\ell_{max}$, upon another reduction becomes $\ell_{max}$-1. This needs to be kept in mind when successive reductions are performed.

The subroutine which calculates the reduced coefficients from the original coefficients is called SUBROUTINE RECUR. This subroutine has as its arguments and for its calling sequence:

CALL RECUR (AR, AI, BR, BI, LF, K)

AR = Array containing the real part of the original expansion coefficients

AI = Array containing the imaginary part of the original expansion coefficients

BR = Array containing the real part of the reduced coefficient

BI = Array containing the imaginary part of the reduced coefficient

LF = Number of elements in the original coefficient array

K = Control for printing the array, K ≤ 0 print the array, K > 0 as not print the array.

The maximum array size is 99.

This subroutine uses the complex arithmetic subrouting SUBROUTINE DOMPLEX. All of the calculations are done in double precision arithemtic. The subroutine is written in FORTRAN IV.

## SUBROUTINE POT

This subroutine stores the value of the potential calculated in the phase shift analysis. The correctly normalized potential for the first value of n, for the first phase shift n = 1, is stored by this routine. In subsequent calculations for higher n values this potential is inserted in the Dirac equation without being recalculated. It is not necessary to store beginning with n = 1. One may start with any n, but on the first pass through the Dirac equations the potential will be stored for use in subsequent calculations.

The calling sequence and argument list for this routine are as follows:

CALL POT(N, CD, X, DX, NST)

N = Current n value for calculation of $n^{th}$ phase shift.

CD = The value of V(x), the potential.

X = The value of the independent variable.

DX = The interval size used in the integration of the Dirac equation

NST = The n value of the first phase shift to be calculated

This routine is written in FORTRAN IV. All calculations are done in double precision. The array size for storing V(x) is set

at 6500. If the interval size is decreased below 0.02, one may have to increase this dimension size. For example, in integrating out to a fitting on radius of $X_o = 60$, one needs a maximum dimension size $I_{max}$ of 6000

$$I_{max} = X_o/(DX/2.0) + 0.51, \quad DX = 0.02$$

$$= 6000$$

On the other hand, if DX = 0.01 then

$$I_{max} = 12000$$

It is always wise to use the smallest value of $I_{max}$ possible in order to avoid storage costs. In the phase shift analysis a step size of 0.02 has been used and is assumed as the default step size.

# Gaussian Folding

$$\sigma_{MEAS}(\theta_0) = \frac{1}{\sqrt{\pi}\Delta}\int_{-\infty}^{\infty} \sigma(\theta)\, e^{-(\theta-\theta_0)^2/\Delta^2}\, d\theta$$

$$= \frac{1}{\sqrt{\pi}\Delta}\int_{-\infty}^{\infty} e^{\ln\sigma(\theta) - \frac{(\theta-\theta_0)^2}{\Delta^2}}\, d\theta$$

$$\ln\sigma(\theta) = \ln\sigma_0 + (\theta-\theta_0)\underbrace{\left.\frac{d\ln\sigma}{d\theta}\right|_{\theta_0}}_{D_1} + \frac{1}{2}(\theta-\theta_0)^2\underbrace{\left.\frac{d^2\ln\sigma}{d\theta^2}\right|_{\theta_0}}_{D_2} + \cdots$$

$$\sigma_m(\theta_0) \approx \frac{\sigma_0}{\sqrt{\pi}\Delta}\int_{-\infty}^{\infty} \exp\left[(\theta-\theta_0)D_1 + \frac{1}{2}(\theta-\theta_0)^2 D_2 - \frac{1}{\Delta^2}(\theta-\theta_0)^2\right] d\theta$$

Let $x = \theta - \theta_0$

$$\sigma_m(\theta_0) \approx \frac{\sigma_0}{\sqrt{\pi}\Delta}\int_{-\infty}^{\infty} e^{xD_1 + \frac{1}{2}x^2\left(D_2 - \frac{2}{\Delta^2}\right)}\, dx$$

Let $A = \frac{1}{2}\left(D_2 - \frac{2}{\Delta^2}\right)$

$$Ax^2 + D_1 x = A\left(x^2 + \frac{D_1}{A}x\right) = A\left[\left(x + \frac{D_1}{2A}\right)^2 - \frac{D_1^2}{4A^2}\right]$$

$$\sigma_m(\theta_0) \approx \frac{\sigma_0}{\sqrt{\pi}\Delta}\, e^{-D_1^2/4A}\int_{-\infty}^{\infty} e^{A\left(x + \frac{D_1}{2A}\right)^2} dx$$

$$2\sigma^2 = -\frac{1}{A} \qquad \sigma = \frac{1}{\sqrt{-2A}}$$

$$\frac{\sqrt{-2A}}{\sqrt{2}\sqrt{\pi}}\int_{-\infty}^{\infty} e^{A\left(x + \frac{D_1}{2A}\right)^2} dx = 1 \qquad \text{for } A < 0.$$

$$\sigma_m(\theta_0) \approx \frac{\sigma_0}{\sqrt{\pi}\Delta}\, e^{-D_1^2/4A}\left(\frac{\sqrt{\pi}}{\sqrt{\frac{1}{\Delta^2} - \frac{1}{2}D_2}}\right) = \frac{\sigma_0}{\sqrt{1 - \frac{1}{2}D_2\Delta^2}}\, e^{(D_1\Delta)^2/4\left(1 - \frac{1}{2}D_2\Delta^2\right)}$$

$$\boxed{\sigma_m(\theta_0) \approx \frac{\sigma_0}{\sqrt{1 - \frac{1}{2}D_2\Delta^2}}\, e^{\frac{(D_1\Delta)^2}{4\left(1 - \frac{1}{2}D_2\Delta^2\right)}}}$$

For finite cutoff:

$$\sigma_m(\theta_c) \approx \left(\frac{1}{N}\sigma_0\right) \int_{\theta_c-\delta}^{\theta_c+\delta} e^{(\theta-\theta_c)D_1 + \frac{1}{2}(\theta-\theta_c)^2 D_2 - \frac{1}{\Delta^2}(\theta-\theta_c)^2} d\theta$$

$$N = \int_{\theta_0-\delta}^{\theta_0+\delta} e^{-\frac{1}{\Delta^2}(\theta-\theta_0)^2} d\theta \quad \leftarrow \int_{-\delta/\Delta}^{\delta/\Delta} e^{-x^2} dx \quad \text{Let } x = \frac{\theta-\theta_0}{\Delta}$$

$$N = \Delta \int_{-\delta/\Delta}^{\delta/\Delta} e^{-x^2} dx = \Delta\sqrt{\pi}\, \text{erf}\left(\frac{\delta}{\Delta}\right)$$

$$\sigma_m(\theta_c) \approx \frac{\sigma_0}{\Delta\sqrt{\pi}\,\text{erf}(\delta/\Delta)} \int_{-\delta/\Delta}^{\delta/\Delta} e^{D_1\Delta x + \frac{1}{2}D_2\Delta^2 x^2 - x^2} dx$$

Let $s^2 = 1 - \frac{1}{2}D_2\Delta^2$:

$$-s^2x^2 + D_1\Delta x = -s^2\left[x^2 - \frac{D_1\Delta}{s^2}x\right] = -s^2\left[\left(x - \frac{D_1\Delta}{2s^2}\right)^2 - \frac{D_1^2\Delta^2}{4s^4}\right]$$

$$= -s^2\left(x - \frac{D_1\Delta}{2s^2}\right)^2 + \frac{D_1^2\Delta^2}{4s^2}$$

$$\sigma_m(\theta_c) \approx \frac{\sigma_0}{\sqrt{\pi}\,\text{erf}(\delta/\Delta)} e^{(D_1\Delta)^2/4s^2} \int_{-\delta/\Delta}^{\delta/\Delta} e^{-s^2\left(x - \frac{D_1\Delta}{2s^2}\right)^2} dx$$

Let $y = s\left(x - \frac{D_1\Delta}{2s^2}\right)$:

$$\sigma_m(\theta_c) \approx \frac{\sigma_0}{\sqrt{\pi}\, s\,\text{erf}(\delta/\Delta)} e^{(D_1\Delta/2s)^2} \int_{s\left(-\frac{\delta}{\Delta} - \frac{D_1\Delta}{2s^2}\right)}^{s\left(\frac{\delta}{\Delta} - \frac{D_1\Delta}{2s^2}\right)} e^{-y^2} dy$$

$$\sigma_m(\theta_0) \approx \frac{\sigma_0\, e^{(D_1\Delta/2s)^2}}{\sqrt{\pi}\, 2s\,\text{erf}(\delta/\Delta)} \left\{ \text{erf}\left[s\left(\frac{\delta}{\Delta} - \frac{D_1\Delta}{2s^2}\right)\right] + \text{erf}\left[+s\left(\frac{\delta}{\Delta} + \frac{D_1\Delta}{2s^2}\right)\right] \right\}$$

Folding Theoretical Cross Sections Over a Small Angular Uncertainty

The experimentally determined cross section $\sigma(\theta_e)$ is not determined at a precisely known angle $\theta_e$, but is in reality an average over a small angular region centered at $\theta_e$. If the uncertainty in the angle is given by $\Delta_e$ then the quantity measured is

$$\bar{\sigma}(\theta_e) = \frac{1}{\sqrt{\pi}\,\Delta_e} \int_{-\infty}^{+\infty} \sigma(\theta)\, e^{-(\theta_e - \theta)^2/\Delta^2 e}\, d\theta$$

where we are assuming that the effect of the experimental geometry is to cause a gaussian spread of width $\Delta_e$ in the experimental angle. We need, therefore, to fold our theoretical cross sections over the experimental uncertainty in order to get an average cross section to compare with the experimental cross section. We take advantage of the fact that $\ln\sigma(\theta)$ is a smooth function of angle to obtain an expansion for $\ln\sigma(\theta)$,

$$\ln\sigma(\theta) \simeq \ln\sigma(\theta_o) + (\theta - \theta_o)\, \frac{\sigma'(\theta)\Big|_{\theta_o}}{\sigma(\theta_o)} + \tfrac{1}{2}(\theta - \theta_o)^2\, \frac{d^2}{d\theta^2}\, \ln\sigma(\theta)\Big|_{\theta_o} + \cdots$$

where $\theta_o$ is an angle at which we have calculated the theoretical cross section. We may now write

$$\sigma(\theta) \simeq \sigma(\theta_o)\, e^{(\theta - \theta_o)D_1 + \frac{1}{2}(\theta - \theta_o)^2 D_2 + \cdots}$$

where

$$D_1 = \frac{d}{d\theta} \ln\sigma(\theta)\Big|_{\theta = \theta_o} = \frac{\sigma'(\theta_o)}{\sigma(\theta_o)}$$

$$D_2 = \frac{d^2}{d\theta^2} \ln\sigma(\theta)\Big|_{\theta = \theta_o} = \frac{\sigma''(\theta_o)}{\sigma(\theta_o)} - \frac{\sigma'(\theta_o)}{\sigma(\theta_o)}^2$$

There are two cases to be considered. First we wish to obtain the correctly folded cross sections for the angles at which we have calculated theoretical cross sections, and second we wish to obtain correctly folded cross sections at the experimental angles. If the experimental angles are included in the set angles at which theoretical cross sections have been calculated, the situation is simplified as then $\theta_e$ will correspond to some $\theta_o$. If this is not the case, then we calculate the folded cross section at this angle. Recently, the experimental angles have not been falling on a mesh point. For example, $\theta_e$ may be 52.06 degrees where, in general, we would have calculated theoretical cross sections at 52.0, 52.5, 53.0 and so on; i.e., we calculate theoretical cross sections at uniformly spaced intervals, these intervals being larger than 0.01 degrees.

Let us consider the first case. Here $\theta_e = \theta_o$ and

$$\bar{\sigma}(\theta_o) = \sigma(\theta_o) \frac{1}{\sqrt{\pi}\,\Delta_e} \int_{-\infty}^{+\infty} e^{(\theta - \theta_o)D_1 + \frac{1}{2}(\theta - \theta_o)^2 D_2 + \cdots} e^{-(\theta - \theta_o)^2/\Delta_e^2}\, d\theta$$

let $(\theta - \theta_o) = x$, then the exponent becomes

$$\text{exponent} = \frac{-x}{\Delta_e^2} + \tfrac{1}{2}x^2 D_2 + xD_1$$

$$= -\left(\frac{1}{\Delta_e^2} - \tfrac{1}{2}D_2\right)\left[x - \frac{D_1}{2\left(\frac{1}{\Delta_e^2} - \tfrac{1}{2}D_2\right)}\right]^2 + \frac{D_1^2}{4\left(\frac{1}{\Delta_e^2} - \tfrac{1}{2}D_2\right)}$$

The integral may now be done and we get

$$\sigma(\theta_o) = \sigma(\theta_o)\ \frac{1}{(1 - \tfrac{1}{2}D_2\Delta_e^2)^{\frac{1}{2}}}\ e^{\left[\frac{D_1^2\,\Delta_e^2}{4(1 - \frac{1}{2}D_2\Delta_e^2)}\right]}$$

The values of $D_1$ and $D_2$ are found using differences where we note that we have calculated theoretical cross sections at intervals in angle $\delta$, in the case then

$$D_1 = \frac{1}{2\delta}\ln\left[\frac{\sigma(\theta_o + \delta)}{\sigma(\theta_o - \delta)}\right]$$

and

$$D_2 = \frac{1}{\delta^2}\ln\left[\frac{\sigma(\theta_o + \delta)\ \sigma(\theta_o - \delta)}{\sigma(\theta_o)^2}\right]$$

In the case that $\theta_e$ is not one of the theoretical angles, we wish to first do the above calculation in order to have correctly folded theoretical cross sections at points suitable for plotting or for

other calculations. Then we must find the correctly folded cross section at the experimental angles. We wish then

$$\sigma(\theta_{exp}) = \sigma(\theta_o) \int_{-\infty}^{\infty} d\theta \exp-\left\{\frac{1}{\Delta_e^2}(\theta - \theta_{exp})^2 - D_1(\theta - \theta_o) - \tfrac{1}{2}D_2(\theta - \theta_o)^2\right\}$$

$$= \sigma(\theta_o) \frac{\exp\left[D_1(\theta_e - \theta_o) + \tfrac{1}{2}D_2(\theta_e - \theta_o)^2\right] e^{\frac{(\Delta_e D_1')^2}{4T_1}}}{\left[1 - \tfrac{1}{2}\Delta_e^2 D_2\right]^{\frac{1}{2}}},$$

where

$$D_1' = D_1 + D_2(\theta_e - \theta_o)$$

$$T_1 = 1 - \tfrac{1}{2}\Delta_e^2 D_2 .$$

The subroutine which does the folding of the theoretical cross sections over experimental angular uncertainty is called SUBROUTINE FOLD. In some cases it may be desirable only to do the first step in the procedure and provision has been made for this eventuality. It is also possible that one would want to fit the experimental data in some manner at this point rather than later and provision has been made for this fitting to be done. In addition, the calculated cross sections are punched out on cards in order that additional calculations or plotting may be done. If the approximations used in the calculation break down in a particular angular region an error message is printed out and execution continues.

The calling sequence and argument list for SUBROUTINE FOLD are as follows

CALL FOLD (DELB, IFOLD, SIG, THET, NFOLD)

where

DELB = The interval in angle for the calculated cross sections. The quantity $\delta$.

IFOLD = The number of calculated cross sections, the array size.

SIG = The calculated cross sections, maximum number 600.

THET = The array of angles corresponding to these cross sections, maximum array size 600.

NFOLD = 1 Do not punch the folded cross sections

NFOLD = 2 Do punch the folded cross sections.

In addition to these arguments passed to SUBROUTINE FOLD by the calling program, a number of quantities are read in at this time. If NFOLD is greater than 2, SUBROUTINE FOLD will not be called and these input cards may be omitted.

The input statements in SUBROUTINE FOLD are arranged as follows:

CARD ONE

```
    READ (5, 10) IFIT, NDAT,  IREAD, J, JREAD,
 10 FORMAT (5I5)
```

where

IFIT ≤ 0 Do not fit the experimental data in which case SUBROUTINE FIT will not be called.

IFIT > 0 Do fit the experimental data. SUBROUTINE FIT is called.

NDAT = The number of experimental points. If no experimental points are being read in at all then NDAT should be set to zero. If NDAT is zero then the second step in the folding at experimental angles is skipped.

IREAD > 0 Read the experimental angles, cross sections and errors. This, of course, presupposes that NDAT will not be zero

IREAD ≤ 0 Do not read the above experimental information. Rather, use the information from the previous case if necessary.

J = The number of experimental angular uncertainties. The maximum is 10

JREAD > 0 Read in the angular uncertainties and the angles at which these uncertainties change.

JREAD ≤ 0 Do not read in this information. Rather, use the information from the previous case.

Now if IREAD and JREAD are both less than or equal to zero, this completes the data. If, however, IREAD is greater than zero then we read in the NDAT experimental points. Also, if IREAD is greater than zero we read in the information on the angular uncertainties. Thus

IREAD > 0

CARDS 2 to NDAT + 1 Contain the NDAT Experimental points

READ (5, 11) (THE(I), DATA (I), COUNT (I), I = 1,NDAT)

11 FORMAT (3E10.5)

where

THE(I) - The $I^{th}$ experimental angle. The maximum array size is 100.

DATA(I) = The $I^{th}$ experimental cross section. The maximum array size is 100.

COUNT(I) = The $I^{th}$ experimental error. The maximum array size is 100.

If JREAD ≤ 0 this is all of the input. If, however, IREAD > 0 and JREAD > 0 CARDS NDAT + 2 to NDAT + 2 + 5 contain the J different angular uncertainties.

READ (5, 19) (TC(I), DELE(I), I = 1, J )

19 FORMAT (6D10.5)

where

TC(I) = The angle in degrees at which the $I^{th}$ experimental angular uncertainty becomes valid.

DELE(I) = The $I^{th}$ value of the experimental angular uncertainty of the experimental resolution, in degrees.

If IREAD $\leq$ 0 then the above cards are CARDS 2 THRU J + 1.

SUBROUTINE FOLD is written in FORTRAN IV using double precision arithmetic.

## Fitting Experimental Cross Sections

There are a number of ways of fitting the experimental data. The method used depends on the assumptions made about the experimental error. We have used several different methods of treating the experimental error, but, in general, our fitting procedure can be described as a least squares method. We usually find both relative and absolute fits to the data, and we take into account the overall error in absolute normalization of the experimental cross section. Thus we write

$$\chi^2_{rel} = \sum_{i=1}^{N} \left[ \frac{\lambda\sigma(\theta_i) - \sigma_{exp}(\theta_i)}{\sigma_{exp}(\theta_i)} \right]^2 \left[ \frac{\sigma_{exp}(\theta_i)}{\Delta(\theta_i)} \right]^2$$

where N is the total number of experimental points, $\sigma(\theta_i)$ is the theoretical cross section, $\sigma_{exp}(\theta_i)$ is the experimental cross sections, $\Delta(\theta_i)$ is the error in $\sigma_{exp}(\theta_i)$, and $\lambda$ measures the goodness of the absolute fit. In other words, to get a good absolute fit to the experimental data one must multiply $\sigma(\theta_i)$ by $\lambda$. Thus $\lambda = 1.0$ is a good fit, $\lambda = 1.05$ indicates a 5% increase is needed for a good absolute fit. If the absolute overall error in the experimental cross section is given by $\delta$ then $\lambda$ is defined by

$$\lambda = \frac{N}{\delta^2} + \sum_{i=1}^{N} \left\{ \frac{\left[\frac{\sigma_e(\theta_i)}{\Delta(\theta_i)}\right]^2 \left[\frac{\sigma(\theta_i)}{\sigma_e(\theta_j)}\right]^2}{\frac{N}{\delta^2} + \sum_{j=1}^{N} \left(\frac{\sigma(\theta_i)}{\Delta(\theta_j)}\right)^2} \right\} .$$

The Chi square for the absolute fit is given by

$$\chi^2_{ABS} = \chi^2_{REL} + \frac{N(1 - \lambda)^2}{\delta^2} .$$

It is possible that in some instances we might wish to restrict the smallness of the statistical error $\Delta(\theta)$ which is considered. This corresponds to putting a maximum on the number of counts at each point. The statistical number of counts, $N_s$, at an angle $\theta$ is

$$N_s(\theta) = \left[\frac{\sigma_{exp}(\theta)}{\Delta(\theta)}\right]^2 .$$

An arbitrary maximum may be introduced of say 500 or 200 counts. This introduces a sharp cutoff in the number of counts. A smoother way of doing this is to define an effective number of counts $N_{eff}$ such that

$$\frac{1}{N_{eff}} = \frac{1}{N_s} + \frac{1}{N_{other}}$$

where $N_{other}$ is provided by the experimentalist . In this case even if $N_s$ gets very large, $N_{eff}$ is constrained. One now uses the constrained number of counts in the expressions for $\chi^2_{rel}$ and $\lambda$ as replacing $\left[\frac{\sigma(\theta_i)}{\Delta(\theta_i)}\right]^2$ .

We have both a fitting program which takes cross sections obtained from the phase shift analysis, usually in the form of punched output, and calculates the values $\chi^2_{rel}$, $\chi^2_{abs}$, and $\lambda$. We also have this program in the form of a subroutine to be used with the phase shift analysis if the fitting is to be done at the same time the cross sections are calculated.

Let us first discuss the fitting program. This short program may be easily modified to introduce a cutoff or an effective N value. In order to introduce a cutoff value for N one only has to remove the c in column one from the cards with identification numbers FIT00154 and FIT00156 and insert the desired value for N-cutoff. The input for this program is as follows:

CARD ONE  Hollerith information about the case being considered in columns 2-72.

CARD TWO  Control integers

```
READ (5, 15) N, NJ, NREAD
15 FORMAT (3I5)
```

where  N = The number of experimental points, not to exceed 100.

NJ = The number of theoretical cross sections.

NREAD > 0  Do read in the experimental angles, cross sections and errors.

NREAD ≤ 0  Do not read in the above data. Use the values from the previous case.

If NREAD > 0 card 3 contains the value of $\delta$, the absolute error.

If NREAD ≤ 0 the previous value of $\delta$ is used.

If NREAD > 0 then

CARDS 4 thru 4 + N contain the experimental data, the information for each angle on a separate card

```
READ (5, 24) ETH(I), ESIG(I), E(I)
24 FORMAT (3E10.5)
```

where

ETH(I) = The experimental angles, no more than 100 in number.

ESIG(I) = The experimental cross sections, no more than 100 in number.

E(I) = The experimental errors, no more than 100 in number.

If NREAD ≤ 0 then

CARDS 3 thru 3 + NJ contain the theoretical cross sections and angles.

```
READ (5,30) S, T
30 FORMAT (2D15.8)
```

where S = The theoretical cross section.

T = The theoretical angle.

If NREAD > 0 then

CARDS N + 5 thru N + 5 + NJ contain the above information.

The output consists of the individual $\chi^2_{reg}$ per point on page 1, and on page 2 the input plus the values of $\chi^2_{reg}$, $\chi^2_{abs}$, and $\lambda$.

The SUBROUTINE FIT calculates the same quantities as the fitting program described above. It may be used by the phase shift program, program RAVENHALL, if desired. Its argument list and calling sequence are

Calculation of the Charge Distribution, Mean Square,
Mean Fourth and Mean Sixth Radii

It is convenient to have available the charge distribution, correctly normalized, corresponding to any of the phenomenological charge distributions which are used in the phase shift analysis or in the mu-mesic x-ray claculations. For this reason a program has been written in FORTRAN IV using double precision arithematic. The normalization integrals are obtained using a fourth order RKG method. We normalize our charge distributions such that

$$Ze^2 = \int_0^\infty \rho_n(x)\, x^2 dx \ ,$$

where $\rho_n(x)$ is the correctly normalized charge distribution. One must therefore calculate the normalization integral

$$I = \int \rho(x)\, x^2 dx$$

where $\rho(x)$ is the unnormalized charge distribution obtained from SUBROUTINE FX (see page E-1for a discussion of this subroutine and the functional forms used for the charge distributions).

Once we have the value of this integral the correctly normalized charge distribution is (in charge/fermi$^3$ the other e comes from including the electron charge)

$$\rho_n(x) = \frac{1}{I}\,\rho(x)\,Ze^2$$

where Z is the atomic number of the nucleus. As we ordinarily specify our nucleus in terms of a parameter

$$\gamma = \frac{Ze^2}{hc}$$

where $\frac{e^2}{hc} = \alpha$ the fine structure constant ($\alpha = 7.29729 \times 10^{-3}$) we write

$$\rho_n(x) = \frac{\gamma}{I}\,\frac{\rho(x)}{\alpha} \quad .$$

In addition to the normalized charge distribution, we also calculate the mean square, mean fourth and mean sixth radii where

$$< r^2 >^{\frac{1}{2}} = \left[\frac{\int \rho(x)\, x^4 dx}{\int \rho(x)\, x^2 dx}\right]^2 \quad ,$$

$$< r^4 >^{\frac{1}{4}} = \left[\frac{\int \rho(x)\, x^6 dx}{\int \rho(x)\, x^2 dx}\right]^{\frac{1}{4}} \quad ,$$

$$< r^6 >^{\frac{1}{6}} = \left[\frac{\int \rho(x)\, x^8 dx}{\int \rho(x)\, x^2 dx}\right]^{\frac{1}{6}} \quad .$$

The units are the same as the units used in the charge distribution so that if, as is usually the case, the parameters are specified

in fermis then these radii are in fermis.

The input to the charge distribution routine is quite simple. Only two cards are necessary. The first card is specified by:

CARD ONE

```
READ (5,10) GAM, C, Z, W, XS, XF, DX
10 FORMAT (7D10.5)
```

where

GAM = $\frac{Ze^2}{\hbar c}$ (for example $\gamma$ = .1459 for Ca, Z = 20)

C = Parameter in the charge distribution.

Z = Parameter in the charge distribution.

W = Parameter in the charge distribution

XS = Initial value of the independent variable x. For use in evaluating the integrals, this is usually zero.

XF = The final value of the independent variable x. This is large enough that the charge distribution is of order $10^{-8}$ at XF.

DX = The step size to be used. A step size of 0.1 is usually small enough.

The second card in a data set specifies the calculation of the normalized charge distribution and is specified by

CARD TWO

```
READ (5, 55) DX, XF
55 FORMAT (2D10.5)
```

where

DX = The intervals in radius at which the charge distribution will be calculated.

XF = The final value of the radius.

The output from this program consists of the normalization integral, the mean square, mean fourth and mean sixth radii, and a table of radius versus the normalized charge distribution. The program uses an external subroutine SUBROUTINE EDEQF which contains the differential equations. This subroutine requires no input. In addition, the differential equations routine DEQ is used, and the appropriate charge distribution provided by one of the subroutines FX.

## Calculation of the Form Factor

It is often convenient to know the form factor corresponding to a given charge distribution of the nucleus. This requires nothing more than calculating the Fourier transform of the charge distribution for this is the definition of the form factor. Thus

$$F(q) = \int \rho_N(\bar{r}) \exp(i\bar{q}\cdot\bar{r})\, d^3r$$

where $\rho_N(\bar{r})$ is the normalized density distribution of the charge. If $\rho(\bar{r})$ is assumed to be spherically symmetric then

$$F(q) = \frac{4\pi}{q} \int_o^\infty r\rho_N(r) \sin(qr)dr$$

We require that at $q = 0$ $F(q) = 1$. Thus, if N is the normalization factor

$$F(q) = \frac{4\pi}{q} \int_o^\infty rNf(r)\sin(qr)dr$$

where f(r) is the functional form of the charge distribution as described on page E-1 and provided by SUBROUTINE FX. We find then

$$F(o) = 1 = 4\pi N \int r^2 f(r)dr$$

$$N = \frac{1}{4\pi \int r^2 f(r)dr}$$

thus

$$F(q) = \frac{\int_r \sin(qr) f(r) dr}{q \int r^2 f(r) dr}$$

This is the form factor calculated. The normalization integral is found first and then the form factor at desired increments of q out to a chosen maximum value of q. As a rule of thumb one will have calculated the physically interesting form factors if one takes the maximus q value to have the same magnitude as the cut off radius in the normalization integral. In the integrations a step size of 0.1 fermis has, in general, been used with good success. One should take the integration out several mean-square radii.

Occasionally it is desirable to alter in some manner the form factor. This is usually done to obtain a form factor which is more closely in agreement with experiment. There are two ways of doing this. One may alter the form factor as it is calculated by adding the appropriate statements at the point noted in the program - in the region of card FMFT 1000, or one may punch the unaltered form factor out on cards to be used in the program to alter the form factor described on page N-1. This may be done by removing the C * from columns one and two from cards FMFT0790, FMFT1020, FMFT1030 and FMFT1055. In either case a recompilation is necessary. Of course one may make both changes in which case the altered form factor is punched out. These cards may then be used in the program to calculate the normalized charge distribution corresponding to this altered form factor.

If the charge distribution is to be calculated from an altered form factor then one must use very small intervals in q. For example, a $\Delta q$ of 0.05 or even $\Delta q = 0.1$ produces normalized charge distributions good to four significant figures when the form factor is the fourier transform of a smooth charge distribution such as a Parabolic Modified Gaussian. However, if the alteration introduces considerable oscillation into the form factor, as was the case in the analysis of the high energy lead data, then a much smaller interval must be used. This means that the calculation of F(q) is very time consuming and if possible an alternate method of finding the altered charge distribution should be found. For example, if one knows the form of the fourier transform of $\Delta F(q)$, the alteration to F(q) then one has $\Delta\rho(r)$ and may add $\Delta\rho(r)$ to the base $\Delta\rho(r)$. This may be done by hand, using the GE basic or by just inserting $\Delta\rho$ in the program to calculate the charge distribution. The appropriate statements would be after CDIS0770, for example, in the case of the alternation for lead

$$\Delta\rho = \frac{-4\pi Z}{2(\pi)^2}\ \frac{A}{z}\ \pi\ \frac{1}{r^2}\ \Big\{ e^{-(b-a)^2/4}\Big[\theta\cos(\phi-\theta)$$

$$- \tfrac{1}{2}b(a-b)\sin(o-\theta)\Big] - e^{-(b+a)^2/4}\Big[\theta\cos(\phi+\theta)$$

$$- \tfrac{1}{2}b(a+b)\ \sin(\phi+\theta)\Big]\Big\}$$

where

$$b = \frac{2r}{z}$$

$\theta = q_o r$, $q_o$ a variable

o = phase angle, a variable

A = amplitude, a variable

a = wave length, a variable

The corresponding $\Delta F(q)$ is

$$\Delta F(q) = A \sin\left[\frac{a\ (q-q_o)z}{2} + o\right]\ e^{-\left[(q-q_o)3/2\right]^2} .$$

The variables could be read in or simply set in the program. Then one would calculate $\rho(r) = \rho(r) + \Delta\rho(r)$, in this case.

If, however, the alteration to the form factor does not produce too much variation, then one may use the program to alter the form factor, punch out the altered form factor and use the program to obtain normalized charge distributions from the form factor described on page E-1 . The procedure used depends on the case under consideration and, unfortunately, a general rule is not available.

The FORM FACTOR PROGRAM is written in FORTRAN IV using double precision arithmetic throughout. This program uses the double precision differential equations routine DEQ, the charge distribution is specified by SUBROUTINE FX and an external subroutine FFDEQF specifies the equations to be solved. The input to this program is given by:

CARDS ONE AND TWO

READ (5, 10) GAM, C, Z, W, XS, XF, DX, DEL, FXF

10 FORMAT (7D10.5)

where card one contains

GAM = Atomic number times the fine structure constant (This is used for identification)

C = parameter in the charge distribution

Z = parameter in the charge distribution

W = parameter in the charge distribution

XS = initial value in the normalization integral - usually 0.0. Also used in the calculation of the Fourier transform.

XF = final value of the radius in the integrations. This is something on the order of a few mean square radii.

DX = Interval size used in the integrations.

On card two

DEL = Interval in q. The form factor is calculated starting at q = 0 in steps of DEL out to $q_{max}$ = FXF

FXF = Maximum q value.

Only two cards of input are required as long as the standard charge distributions are used. If the charge distribution is read in, see page E-5, then these cards will follow the above two cards.

## Program to Alter the Form Factor

This simple program takes a form factor, available as punched putput from the Form Factor Program, and alters it in some manner. As an example of the kind of alternation one might expect we have programmed the alteration to the form factor which gives an acceptable fit to the high energy lead cross sections. In this case the change in F(q), $\Delta F(q)$, is given by

$$\Delta F(q) = A \exp(-y^2) \sin(ay + \phi)$$

where

$y = (q - q_o)z/2$, $q_o$ = a variable value of q specified by requiring a good fit to the data.

A = amplitude, variable parameter

$\phi$ = phase angle, variable parameter

a = wavelength, variable parameter

z = parameter in the charge distribution.

Here the base form factor F(q) was found by taking the fourier transform of the parabolic modified gaussian charge distribution which gave a good fit to the small angle lead data. This procedure is described in the write up of the Form Factor Program. The values of F(q) vs. q were punched out during the calculation of F(q) and used as input.

In some cases it might be desirable to calculate the charge distribution corresponding to the altered form factor and so the values

of F(q) vs. q are punched out so that they may be used as input to the program to calculate normalized charge distributions from form factors, see page O-1. In this example this is done; however, the alteration produces so much variation in F(q) that an extremely fine mesh of q values would have to be used if reasonably accurate charge distributions are to be obtained. An alternative is to calculate the altered form factor in the charge distribution program if the analytic form of $\Delta\rho(r)$ is known.

There are, however, other kinds of alterations to F(q) which do produce altered form factors which can be used in the program to alter the form factor. Each case is different and caution must be exercised.

The program to alter the form factor has, for this example, input given by

CARD ONE

N = number of F(q) vs. q points to be read in

QO = the value of $q_o$ in $\Delta F(q)$

P = the value of a in $\Delta F(q)$

A = the value of A in $\Delta F(q)$

PHI = the value of $\phi$ in $\Delta F(q)$

Z = the value of Z in $\Delta F(q)$ and in $\rho(r)$.

CARDS 2 thru 2 + N

Q = the Q value

F = the value of the form factor

The program is written in FORTRAN IV; only standard library subroutines are used. In order to supress the punched output place a C in column one on cards AFF00374, AFF00401 and AFF00406.

## Calculation of the Charge Distribution From the Form Factor

In some cases it is desirable to have the charge distribution corresponding to a given form factor. One simply takes the fourier transform of the form factor to obtain the corresponding charge distribution. This program accepts F(q) vs. q in tabular form, available from the form factor program or from the program to alter the form factor. The first calculation is to determine the normalization integral

$$N_F = \int F(q)\, q^2 dq.$$

Then one calculates the charge distribution from

$$\rho(x) = \int \frac{q \sin(qx) F(q)\, dq}{x\, N_F}.$$

The final step is to calculate

$$N_C = \int x^2 \rho(x)\, dx$$

so that the correctly normalized charge distribution may be found. The normalized charge distribution is

$$\rho_{norm}(x) = Z \frac{\rho(x)}{N_C} = \frac{\gamma}{\alpha} \frac{\rho(x)}{N_C}$$

where

$$\gamma = Z \frac{e^2}{\hbar c}$$

$\alpha$ = fine structure constant.

As has been pointed out in the write up of the form factor program and the for factor alteration program there exist form factors which vary too rapidly to make the calculation of their corresponding charge distributions to time consuming to be, in general, very practical. In such a situation, as was the case with the analysis of the high energy lead data, an alternate method such as using an analytic form for $\Delta\rho(x)$ may be used.

This program is written in FORTRAN IV using double precision arithmetic throughout. The input is specified as follows:

CARD ONE

GAM = Z times the fine structure constant

XS = starting value for the integrations

XF = final value for the integrations

DX = step size (This must be 2 times the mesh size in F(q) vs. q).

NQ = the number of F(q) vs. q points

CARD TWO - Hollerith card identifying the case

CARDS 3 thru 3 + NQ

Q = q-value

F = value of F(q)

The program uses an external subroutine CFDEQF which specifies the differential equations. The RKG fourth order integration routine DEQ is also used. The output gives the input quantities in tables of x vs. $\rho(x)$ for both the unnormalized and normalized charge distributions. The values of $N_F$ and $N_C$ are also given.

Program to Calculate the Difference
Between Experimental and Theoretical Form Factors

In the analysis of high energy electron scattering where a large range of recoil momentum is covered it has been necessary to alter our basis charge distribution forms in order to find an acceptable fit to the high recoil momentum region without destroying the overall fit to the data. A device which has proved useful is to calculate the difference between the experimental and theoretical form factors. The theoretical form factor is obtained from the charge distribution which provides a good fit to the low recoil momentum region (the first two or so diffraction minima. The desired difference is calculated in Born approximation. We assume that

$$\left(\frac{d\sigma}{d\Omega}\right)_{exp} = \left(\frac{d\sigma}{d\Omega}\right)_{point} |F_{exp}(q)|^2$$

and

$$\left(\frac{d\sigma}{d\Omega}\right)_{theory} = \left(\frac{d\sigma}{d\Omega}\right)_{point} |F_{theory}(q)|^2$$

thus

$$F_{exp}(q) - F_{theory}(q) = \left\{1 - \left[\left(\frac{d\sigma}{d\Omega}\right)_{exp} \Big/ \left(\frac{d\sigma}{d\Omega}\right)_{theory}\right]^{\frac{1}{2}}\right\} F_{theory}(q)$$

Three experimental to theoretical cross section ratios are considered, corresponding to the central experimental point and its values at the error limits.

We then calculate the theoretical form factor defined as

$$F_{theory}(q) = \frac{\int f(x)\ x\ \sin qx\ dx}{\int f(x)\ x^2\ dx}$$

where f(x) is the charge distribution which provides a good fit to the small angle part of the data. One then calculates the difference $F_{exp}(q) - F_{theory}(q)$.

The procedure is to first calculate the normalization integral then the form factor at the q-values corresponding to the experimental angles. Here we use

$$q = 2\ k\ \sin\ (\theta_{exp}/2) \left[1 + \frac{\gamma}{kc_1}\right]$$

where $\gamma$ = Atomic number times the fine structure constant

$k = \frac{E_o}{\hbar c}$, $E_o$ is the incident electron energy in the laboratory system.

The input required is given by:

CARD ONE

```
READ (5, 30) N, NC
30 FORMAT (2I5)
```

N = The number of experimental points

NC = The number of theoretical points

CARDS 2 thru 2 + N contain the experimental data

```
READ (5, 40) ETH(I), ESIG(I), E(I)
40 FORMAT (3D10.5)
```

where

ETH(I) = Experimental angle

ESIG(I) = Experimental cross section

E(I) = Experimental error.

CARDS 3 + N and 4 + N

READ (5, 45) GAM, C, Z, W, C1, G, ALM, SX, DX, XF

45 FORMAT (7D10.5)

where

GAM = Atomic number times the fine structure constant

C = Parameter in the charge distribution

Z = Parameter in the charge distribution

W = Parameter in the charge distribution

C1 = Parameter in the differentiation of q. This is usually taken to be C.

G = Energy in MeV of the incident electron in the laboratory system

ALM = The value of $\lambda$ obtained from fitting the cross section to the experimental data, see page .

SX = The starting value of the radius in the calculation of the form factor.

DX = Step size in the integrations.

XF = Final value of the radius used in the integrations.

CARDS 5 + N thru 5 + N + NC contain the theoretical cross sections and angles

READ (5, 55) S, T

55 FORMAT (2D15.8)

S = Theoretical cross section

T = Theoretical angle.

This program is written in FORTRAN IV using double precision arithmetic throughout. The charge distribution routine SUBROUTINE FX is used along with the external subroutine EFDEQF and the differential equations routine DEQ.

## Program for Plotting Differential Cross Sections

A FORTRAN IV program for the 360/65 has been written which produces graphs of the differential cross section versus scattering angle. The cross section units are in $(\text{fermis})^3$/ steradian and the angles are in degrees. Semi-logarithmatic 5-cycle graph paper is generated and used. The graphs are produced on the Stromberg Carlson 4060 plotter which is part of the GMR system 360. In general, more than one page will be necessary in order to plot all of the points, and provision has been made for this eventuality. If more than one page is necessary, the scales on the graph paper are appropriately adjusted and the two pages may be joined to form the full curve just as one would when plotting such a curve by hand. It is also possible to plot the experimental cross sections in the form of open circles. The theoretical points are connected by a linear fill. This seems to be adequate as we ordinarily produce a rather dense set of theoretical points.

The input for this program called PLOT is specified as follows

CARD ONE

```
READ (5,5) NE, NT, NANY, NCOPY
5 FORMAT (4I5)
```

where

NE = The number of experimental points.

NT = The number of theoretical points.

NANY ≤ 0 Plot the experimental points.

NANY > 0 Do not plot the experimental points.

NCOPY = The number of copies of the graph desired.

If NANY ≤ 0 then cards 2 to 2 + NE contain the experimental cross sections and angles specified by

```
READ (5, 20) TE(I), CE(I), I = 1, NE)
20 FORMAT (2E10.5)
```

where

TE(I) = The $I^{th}$ experimental angle. A maximum number of 100 is allowed.

CE)I) = The $I^{th}$ experimental cross section. A maximum number of 100 is allowed.

Cards 3 + NE to 3 + NE + NT contain the theoretical cross sections. These are usually available as punched card output from the phase shift analysis program, program RAVENHALL. The input specified by

```
READ (5, 30)  (CC(I), TC(I), I = 1, NT)
30 FORMAT (2D15.8)
```

where

CC(I) = The $I^{th}$ theoretical cross section. A maximum of 600 is allowed.

TC(I) = The $I^{th}$ theoretical angle. A maximum of 600 is allowed.

If NANY > 0 then cards 2 thru 2 + NT contain the above theoretical cross sections; no experimental cross sections are read in.

The scales were chosen to be aesthetically pleasing and hopefully easy to read. If more than 5 cycles are used the semi-log scale becomes rather compressed. The horizontal scale was chosen to fill out the 11 by 14 page size. The degree interval is chosen as follows: If the angular range to be covered is greater than 100 degrees the major divisions are in 10-degree intervals. If the angular interval is less than 100 degrees then the angular interval is chosen to be 5 degrees.

Information about the plotting software may be obtained from the "Programmers Reference Manual for the Integrated Graphics Software Package". There are a few errors remaining in the software package. In particular, the log scale is not done in a consistent manner. This difficulty has been pointed out to the Stromberg-Carlson representative and hopefully will be corrected.

## Predicting the Parameters in the Charge Distribution For a Good Fit to Experimental Data

We wish to predict the parameters in the charge distribution which will give us a good fit to the experimental data. One starts with a theoretical cross section, $\sigma(\theta_i)$, calculated for parameters c, z and w in some three parameter distribution, which is a fairly good relative and absolute fit to the data. One then obtains the three cross sections, $\sigma_{\Delta c}(\theta_i)$, $\sigma_{\Delta z}(\theta_i)$ and $\sigma_{\Delta w}(\theta_i)$ for parameters $c + \Delta c$, z, w; c, $z + \Delta z$, w and c, z, $w + \Delta w$ respectively. From this it is possible to find how much one should change c, z and w to obtain a good fit to the experimental data. In some cases the theoretical cross sections are multiplied by the value of $\lambda$ obtained by fitting them to the experimental data. See the write up of the fitting program, page K-1, for the definition of $\lambda$. One wishes to minimize the following quantity with respect to $\alpha_c$, $\alpha_z$ and $\alpha_w$.

$$\chi^2_{min} = \sum_{i=1}^{N} \left\{ \alpha_c \left[ \sigma_{\Delta c}(\theta_i) - \sigma(\theta_i) \right] + \alpha_z \left[ \sigma_{\Delta z}(\theta_i) - \sigma(\theta_i) \right] + \alpha_w \left[ \sigma_{\Delta w}(\theta_i) - \sigma(\theta_i) \right] - \left[ \sigma_{exp}(\theta_i) - \sigma(\theta_i) \right] \right\}^2 \frac{N(\theta_i)}{\sigma^2_{exp}(\theta_i)}$$

where $\sigma_{exp}(\theta_i)$ is the experimental cross section, N the number of data points, and $N(\theta_i)$ is the number of counts obtained from the statistical error

$$N(\theta_i) = \left[\frac{\sigma_{exp}(\theta_i)}{\Delta(\theta_i)}\right]^2 ,$$

Here $\Delta(\theta_i)$ is the error in $\sigma_{exp}(\theta_i)$. One may treat $N(\theta_i)$ in several different ways depending on the assumptions made about the experimental error. For example, one may introduce an arbitrary cut off so that

$$N(\theta_i) = \left[\frac{\sigma_{exp}(\theta_i)}{\Delta\theta_i}\right]^2 \quad \text{for} \quad N(\theta_i) < N_{cut} \;,\; N(\theta_i) = N_{cut} \quad \text{for} \quad N(\theta_i) \geq N_{cut}$$

In order to find the values of $\alpha_c$, $\alpha_z$, $\alpha_w$ which produce $\chi^2_{min}$ one solves the matrix equation

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} \alpha_c \\ \alpha_z \\ \alpha_w \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

where

$$A_{kj} = \sum_{i=1}^{N} \left[\sigma_k(\theta_i) - \sigma(\theta_i)\right]\left[\sigma_j(\theta_i) - \sigma(\theta_i)\right] \frac{N(\theta_i)}{\sigma^2_{exp}(\theta_i)}$$

$$b_k = \sum_{i=1}^{N} \left[\sigma_k(\theta_i) - \sigma(\theta_i)\right]\left[\sigma_{exp}(\theta_i) - \sigma(\theta_i)\right] \frac{N(\theta_i)}{\sigma^2_{exp}(\theta_i)}$$

where

$$k = (\Delta c, \Delta z, \Delta w)$$

$$j = (\Delta c, \Delta z, \Delta w).$$

The extension to a larger number of unknowns is obvious. One solves this matrix equation using the simultaneous equation subroutine SIMEQ (see Chapter 5 of the OS 360 reference book) and obtains values for $\alpha_c$, $\alpha_z$ and $\alpha_w$. The new parameters are given by

$$c' = c + \alpha_c \Delta c$$

$$z' = z + \alpha_z \Delta z$$

$$w' = w + \alpha_w \Delta w .$$

One must be cautious, however, as large changes in the parameters may indicate a lack of linearity and a breakdown of the method. In this event, one may try several different things; obtain a better overall fit to the cross section for $\sigma(\theta_i)$, i.e., get a $\lambda$ value closer to unity and/or use smaller changes in the parameters.

The program to calculate the parameter changes necessary to give a good fit is written in FORTRAN IV. It has as input the following data

CARD ONE - Hollerith information identifying the case
CARD TWO

```
    READ (5, 10) N, NE, NR, (ALM(I), I = 1,4)

 10 FORMAT (3I5, 4E10.5)
```

where

N = The number of theoretical cross sections of each kind $\sigma(\theta_i)$, $\sigma_{\Delta k}(\theta_i)$, etc.

NE = The number of data points.

NR > 0 Do read in the experimental.

NR ≤ 0 Do not read in the data. Rather, use the data from the previous case.

ALM(I) = The values of λ corresponding to the four theoretical cross sections.

If NR > 0 CARDS 3 thru 3+ NE contain the experimental data.

```
READ (5,20) (TE(I), SE(I), E(I), I = 1, NE)
20 FORMAT (3E10.5)
```

where TE(I) = The array of experimental angles, no larger than 100.

SE(I) = The array of experimental cross sections, no larger than 100.

E(I) = The array of experimental errors, no larger than 100.

If NR ≤ 0 the above cards are not required. This makes it possible to run several sets of theoretical cross sections with the same experimental data. Next the theoretical cross sections are read in. There are four cross sections. First $\sigma(\theta_i)$, then $\sigma_{\Delta c}(\theta_i)$, $\sigma_{\Delta z}(\theta_i)$ and $\sigma_{\Delta w}(\theta_i)$, thus

CARDS 4 + NE thru 4 + NE + 4N contain theoretical cross sections and angles, in the form

```
READ (5,30) S,T
30 FORMAT (2D15.8)
```

where

S = The theoretical cross section $\sigma(\theta_i)$, or $\sigma_k(\theta_i)$.

T = The angles $\theta_i$.

The output consist of the coefficient matrix A whose elements are $A_{kj}$, the column vector $b_k$ and the values of $\alpha_c$, $\alpha_z$, $\alpha_w$. In addition, the Chi-squared of $\sigma(\theta_i)$ given by

$$\chi^2 = \sum_{i=1}^{N} \left[ \frac{\sigma(\theta_i) - \sigma_{exp}(\theta_i)}{\sigma_{exp}(\theta_i)} \right]^2 \frac{N(\theta_i)}{\sigma^2_{exp}(\theta_i)}$$

and the value of $\chi^2_{min}$. The value of $\sigma(\theta_i)$ may be multiplied by the value of $\lambda$ which results from fitting $\sigma(\theta_i)$ to the experimental data as mentioned earlier. If one now calculates a cross section using c′, z′ and w′ one would expect to obtain, upon fitting the experimental data, a $\chi^2$ quite close to $\chi^2_{min}$. If not, the procedure may have to be repeated using as the base, parameters c′, z′ and w′ , and making small changes on them. Hopefully, the process converges in three or so trials. If this is not the case, one may have to treat the errors more realistically, such as putting in a cut off value for $N(\theta_i)$. A simple cut off may be introduced by removing the C* is columns 1 and 2 from cards with identification numbers FPR00540 and FPR00550 and inserting the desired maximum value for N.

## Charge Density From the Schroedinger Equation

We wish to obtain the charge density from the correctly normalized solutions to Schroedingers equation with a Wood-Saxon potential. Particles are bound in this well below the fermi energy, $E_F$, and we find the contribution to the total charge density from each particle. We wish to solve then the Schroedinger equation in one dimension

$$\left[ -\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + E - V(x) \right] \phi_E(x) = 0$$

where we let

$$\frac{2m}{\hbar^2} E = k^2 = \frac{E}{20} f^{-2}$$

$$\frac{2m}{\hbar^2} V_o = k_o^2 = \frac{V_o}{20} f^{-2}$$

and

$$V_{ws}(x) = \frac{-V_o}{1+\exp[\frac{x-c}{z}]} .$$

The potential V(x) is

$$V(x) = V_o + V_{ws}(x) = V_o \left[ 1 - \frac{1}{1+\exp \frac{x-c}{z}} \right] ,$$

and our equation is now

$$\frac{d^2}{dx^2}\,\phi_k(x) = \left[ -k^2 + k_o\ 1 - \frac{1}{1+\exp\frac{x-c}{z}} \right] \phi_k(x) \quad .$$

We start our numerical integration at large x, $x_L$, and integrate inward to x = o. The initial values are taken to be

$$\phi_k(x_L) = 1$$

$$\phi_k'(x_L) = -K = -\sqrt{k_o^2 - k^2}$$

The wave functions are stored so that they may be normalized. The normalization is found by noting that

$$\phi_k(x) \sim A \sin(kx + \alpha)$$

$$\phi_k'(x) \sim kA \cos(kx + \alpha)$$

so that at x = o

$$\phi_k(o) = A \sin \alpha$$

$$\phi_k'(o) = k\, A \cos \alpha$$

and consequently

$$A = \left[ \phi_k^2(o) + \frac{1}{k^2}\ \phi_k'^2(o) \right]^{\frac{1}{2}} \quad .$$

So that at any x the correctly normalized wave function is

$$\phi_{k,N}(x) = \frac{\phi_k(x)}{A} \quad .$$

Now we wish to calculate the charge density. We define the fermi momentum by

$$k_F^2 = E_F \frac{2m}{\hbar^2} = \frac{E_F}{20} \text{ f}^{-2}$$

and if one had a continuum of particles below $E_F$ then the charge density would be

$$\rho(x) = \frac{1}{k_F} \int_{k=k_i}^{k_F} \left| \phi_{k,N}(x) \right|^2 dk$$

We have taken $k_i = 0.1k_F$ and calculated $\phi_{kN}(x)$ at even steps in k up to and including $k_F$. We have used 10 values of k. The wave functions oscillate and then decay exponentially for large x. The charge density has wiggles in the region of the nuclear surface and then falls off with large x. The wiggles are the Friedel oscillations.

The input specifications for this program are quite simple. Only one data card per case is required.

CARD ONE

EF = Fermi energy in MeV (less than zero)

VO = Well depth in MeV (less than zero)

DX = Step size to be used for the inward integration. We have used DX = -0.02 with good results. Note DX must be negative for inward integration.

C = Parameter in the Wood Saxon potential given in fermis

Z = Parameter in the Wood Saxon potential given in fermis.

The output from this program consists of the input parameters, the normalized wave function for each k-value and the total charge density. The program is written in FORTRAN IV using double precision arithmetic throughout. The program requires the external subroutine SCHDEQ which contains the differential equations, and the fourth order R.K.G. routine DEQ. The execution time varies depending on the case. For 10-k values with $X_L$ = 10 and DX = -0.02 about 0.43 minutes of CPU time were used.

## Interpolation of the Charge Density

In some of our calculations we have used charge densities which have been obtained from the Optical Model Calculations of Ravenhall and Mather. These charge densities are usually given as a function of radial distance in fermis at steps which are too large for use in our programs to calculate cross sections or to find information about mesic atoms. The charge densities are, however, in the correct units to be used in of charge distribution programs if things like mean square radii are desired. In the case of the phase shift programs and the mesic x-ray programs the radial units must be changed and then the charge density obtained at suitable intervals in the appropriate dimensionless variable. In order to do this one must interpolate between points in the charge distribution vs. radius array.

We here used a third order Aitkens method of successive linear interpolations. The subroutine used in called DTABX and is described in the OS/360 book. The subroutine returns double precision interpolated values of the charge distribution in convenient form for use in the RAVENHALL or mesic x-ray programs.

If the charge distribution is to be used in the RAVENHALL program then the appropriate units are

$$x = kr$$

where

$$k = \frac{E_{c.m.}}{\hbar c}$$

where $E_{c.m.}$ is the center of mass energy of the incident electron calculated from the incident laboratory frame energy $E_o$

$$E_{cm} = E_o \sqrt{\frac{M}{M+2E_o}}$$

where M is the mass of the target nucleus. In the case of the mesic x-ray calculations

$$x = r/\lambda\!\!\!^{-}_r$$

where $\lambda\!\!\!^{-}_r$ is the reduced Compton wave length of the muon given by

$$\lambda\!\!\!^{-}_r = \lambda\!\!\!^{-}_\mu \ (1 + 0.113436/A)$$

where A is the atomic weight of the nucleus and the Compton wave length $\lambda\!\!\!^{-}_\mu = 1.86753$ f. The charge density will be calculated at even equal intervals in x. In view of this a small modification to the default specifications in the mesic x-ray programs must be made. Rather than using a step size of 0.01 for x from 0.01 to 0.20 we start at X = 0.10 (ZV15(1) = 0.10) and use a step size of 0.10 throughout. This means the first interpolated value is not used (i.e., this value is not part of the punched output, thus the input to subroutine FX for reading in charge distributions begins with the charge distribution at x = 0.10). I hasten to point out that in the case of charge densities for the RAVENHALL program the first interpolated value at x = 0.01 is also deleted from the punched output so that one starts with the value at x = 0.02 as input for the FX subroutine. This is constant with the

default step size of 0.02 used in the RAVENHALL program.

The input for the interpolation routine to get charge distributions for the RAVENHALL program is,

CARD ONE

EO = Energy in the lab system in MeV

AT = Atomic weight of the target

DELX = Step size in X desired (DELX = 0.05 for constancy with default specifications in the RAVENHALL PROGRAM)

M = The number of charge density vs. radius pairs in the original table.

CARDS 2 through M + 2, The radius vs. the charge density. The format and number may vary depending on the source.

The input for the interpolation routine for the mesic x-ray program is

CARD ONE

AT = Atomic weight of the target

DELX = Step size in X desired, a good choice is DELX = 0.05 as the minimum changes in default specifications are required.

M = The number of charge density vs. radius pairs in the original table.

CARDS 2 through M + 2, the radius vs. the charge density. The format and number may vary depending on the source.

The output from both interpolation routines consist of the original data and the interpolated charge densities. The charge densities are also punched out on cards for use in subsequent calculations. The CPU time required is

around 0.26 minutes for the charge density for the RAVENHALL program and 0.11 minutes for the mesic x-ray programs. The interpolation routines use DTABX the double precision interpolation routine. Both programs are written in FORTRAN IV using double precision arithmetic throughout.

# THE OS/360 BOOK

DTABX



Refer To: Computing Service Department

Entry: DTABX

Purpose:

Given the value of an independent argument x, this subroutine performs the Kth order interpolation ($K \leq 3$) on a table of $X_i$, $Y_i$ values for the corresponding dependent argument, y.

Restrictions:

(1) The independent variables of the table must be monotone.

Method:

Aitken's method of successive linear interpolations is used. If the desired X falls outside the given table then linear extrapolation is used.

Use:

This subroutine is a double precision function with argument list as follows:

ANS = DTABX(X,Y,XIN,L)

X is the independent variable array (double precision).

Y is the dependent variable array (double precision).

XIN is the value of the independent variable for which an interpolated value of the dependent variable is desired (double precision).

L is an integer array of length 7 containing the specification parameters:

L(1) = length of independent variable array, i.e., number of pairs of data in the table.

L(2) = spacing, in terms of subscripts, between variables in the independent variable array, X.

L(3) = spacing, in terms of subscripts, between variables in the dependent variable array, Y.

L(4) = value of the subscript designating the first independent variable in the X array.

L(5) = value of the subscript designating the first dependent variable in the Y array.

L(6) = an output variable which is set equal to

1 if no extrapolation was used,
2 if extrapolation occurred.

Note: all extrapolation is linear regardless of the value of L(7).

L(7) = order of interpolation (1, 2, or 3).

Storage of Tables:

The values of the independent variables may be unevenly spaced, i.e.,

$$X_{i+1} - X_i \neq X_i - X_{i-1}$$

but must be monotonic, i.e.,

$$X_i < X_{i+1} \text{ or } X_i > X_{i+1}$$

The array format of the independent and dependent variables is intended to be general. Below are some examples of array storage and the corresponding arguments to TABX.

Examples:

1) An array A contains 40 pairs of (X,Y) data.

A(1) = X(1), A(2) = Y(1), A(3) = X(2), A(4) = Y(2)

A(5) = X(3), A(6) = Y(3), ...., A(80) = Y(40)

ANS = DTABX (A, A, X, L)

where L(1) = 40

L(2) = 2

L(3) = 2

L(4) = 1

L(5) = 2

L(7) = 1 (linear interpolation desired)

2) An array D contains the independent variable, and an array E contains the dependent variable.

D(1) = X(1), D(2) = X(2) . . . . D(k) = X(k)

E(1) = Y(1), E(2) = Y(2) . . . . D(k) = Y(k)

VAL = DTABX (D, E, X, L)

where L(1) = k

L(2) = 1

L(3) = 1

L(4) = 1

L(5) = 1

L(7) = 3 (third order interpolation desired)

3) The independent variable is contained in consecutive positions of an array S, and the dependent variable is contained in every fifth position of an array T, beginning with T(3).

S(1) = X(1), S(2) = X(2) . . . . S(N) = X(N)

T(3) = Y(1), T(8) = Y(2) . . . . T(5N-2) = Y(N)

TRY = DTABX (S, T, X, L)

where L(1) = N

L(2) = 1

L(3) = 5

L(4) = 1

L(5) = 3

L(7) = 2 (second order interpolation desired)

## Differential Equations Routine

In many of the programs discussed the fourth order R.K.G. routine has been used. This double precision routine for solving differential equations has proved to be both fast and accurate. The reference to the R.K.G. method is Gill, "A Process for the Step-by-Step Integration of Differential Equations in an Automatic Digital Computing Machine", Proceedings of the Cambridge Philosophical Society, 47, 1950. The Runge-Kutta-Gill method is used for the solution of a system of simultaneous differential equations. The differential equations must be of the first order; however, higher order equations can be solved since an nth order differential equation is equivalent to n first order differential equations.

The basic information about the variables needed for the R.K.G. process is supplied by a CALL statements of the form

```
CALL DPDEQ1 (X, Y, F, Q, N, EXTSUB)
```

where

X = An array of dimension three where

X(1) = The independent variable

X(2) = The step size used

X(3) = Final value

Y = Solution array of dimension N

F = Derivative array of dimension N

Q = Erasable array of dimension N

N = The Number of equations

EXTSUB = The name of an EXTERNAL subroutine containing the actual equations to be solved.

X, Y, F and Q must all be double precision arrays of dimension N.

Next in line one has the CALL statement

CALL DPDEQ2 (M)

where M is the frequency of desired returns to the main program, i.e., there is a return every M steps. Note that there is a return before the actual integration starts when the independent and dependent variables still have their initial values. Ordinarily the functions will have been calculated; however, the derivatives will not have been calculated (and if they are printed out will be garbage numbers). Finally, one has

CALL DPDEQ(I)

where

I = 0 The independent variable has reached its final value. One is either through or ready to continue with different parameters in the X array. If the range of integration is not an integral multiple of the step size this means

$$\left|X(3) = X(k)\right| < \left|\frac{X(2)}{2}\right| .$$

Be forewarned that errors can occur in the final value in such a case!

I = 1 The frequency return as called for in the previous CALL.

Any one of the programs using DEQ in the previous sections can serve as a guide; however, one should perhaps look at the charge distribution program as a simple example. The longer programs were written before certain simplifying improvements were available. As a simple example let us solve the equation

$$\frac{d^2}{dx^2} f = (x^2 + 2x)f$$

First one writes

$$\frac{df}{dx} = g$$

$$\frac{dg}{dx} = (x^2 + 2x)f$$

let us say that

$$f = 0 \quad \text{at} \quad x = 0$$

$$g = 0 \quad \text{at} \quad x = 0$$

and we desire values of f from 0 to 1 at intervals of 0.1. The programs would be read:

MAIN PROGRAM:

```
DOUBLE PRECISION X(3), Y(2), F(2), Q(2)
EXTERNAL TSTDEQ
COMMON/DEQ/X, Y, F
X(1) = 0.0D0 - init value of independent Var
X(2) = 0.05D0 - integration step size.
X(3) = 1.0D0 - final value of independent Var.
Y(1) = 0.0D0 - init value of Function 1
Y(2) = 0.0D0 - init value of Function 2.
   N = 2 - # of Functions
   M = 1 - # steps to take before DPDEQ return.
```

```
      CALL DPDEQ1(X, Y, F, Q, N, TSTDEQ)
      CALL DPDEQ2(M)
10    CALL DPDEQ(1)
      IF(I. NE.0) GO TO 20
      GO TO 10.
20    WRITE (6, 30) X(1), Y(1)
30    FORMAT (1HO, 'X=', F5.2, 2X,' F(x) =', D24.16)
      IF(X(1).GT.X(3)) RETURN
      GO TO 10
      END
EXTERNAL SUBROUTINE TSTDEQ
      DOUBLE PRECISION X(3), Y(2), F(2)
      COMMON/DEQ/ X, Y, F
      F(1) = Y(2)
      F(2) = (X(1) * * 2 + 2.0D0 * X(1)) * Y(1)
      RETURN
      END
```

Now I would like to list the names of the external subroutines used in the programs discussed in the previous sections, along with the corresponding calculation.

A-1 RAVENHALL PROGRAM - RAVDEQ

B-1 BREIT FRAME PROGRAM - RAVDEQ (Identical to above)

C-1 COULOMB FUNCTION PROGRAM - CFDEQF

D-1 Mesic X-RAY PROGRAMS - EDEQF

L-1 CHARGE DISTRIBUTION PROGRAM - EDEQF (Not the same as above)

M-1 FORM FACTOR PROGRAM - FFDEQF

O-1 CHARGE DISTRIBUTION FROM THE FORM FACTOR - CFDEQF

P-1 DIFFERENCE BETWEEN EXPERIMENTAL AND THEORETICAL FORM FACTORS - EFDEQF

S-1 CHARGE DENSITY FROM SCHROEDINGER EQUATION _ SCHDEQ

STEVE WILLIAMSON 9.5



# A PROCESS FOR THE STEP-BY-STEP INTEGRATION OF DIFFERENTIAL EQUATIONS IN AN AUTOMATIC DIGITAL COMPUTING MACHINE

By S. GILL

Communicated by M. V. Wilkes

*Received* 3 June 1950

*ABSTRACT.* It is advantageous in automatic computers to employ methods of integration which do not require preceding function values to be known. From a general theory given by Kutta, one such process is chosen giving fourth-order accuracy and requiring the minimum number of storage registers. It is developed into a form which gives the highest attainable accuracy and can be carried out by comparatively few instructions. The errors are studied and a simple example is given.

1. *Introduction.* Most of the methods employed in hand computing for the step-by step integration of differential equations have an essential feature in common, namely that at each step of the integration, use is made of the function-values already obtaine in preceding steps. Thus if we have arrived at the value of $y_n$ and we wish to find $y_{n+}$ these methods require a knowledge of $y_{n-1}, y_{n-2}, \ldots$, the number of values dependin upon the method and the accuracy required. Such processes have obvious advantage: The values of $y_{n-1}, y_{n-2}, \ldots$ carry information about the behaviour of $y$ in the regio of $y_n$, and this can assist in determining $y_{n+1}$; to disregard it would be wasteful. Als these processes, nearly all of which are based on difference formulae, have simp analytical forms and are easily remembered. These points weigh heavily in hand worl where every scrap of information is obtained at the cost of some mental labour, an simple formulae are definitely preferred. There are disadvantages, though not seriou ones from the hand computer's point of view. First, these processes cannot be used the beginning of a range of integration. Two or more consecutive values of $y$ are neede before the main process can operate, and to obtain these some auxiliary process must b used. Secondly, it is not a simple matter to change the size of the interval in the midd of a run. Doubling the interval is fairly easy, halving it is not too difficult, but to chan it by any other factor is quite complicated.

When high-speed automatic machines are used, these considerations assun altogether different proportions. It becomes a serious drawback to have to supply th machine with special instructions for a starting process, or for changing the size of th interval. Moreover, a process which appears simple to a hand computer may invol a considerable number of operations in an automatic machine. For example, havi found $y_{n+1}$ from, say, $y_n$, $y_{n-1}$ and $y_{n-2}$, we then wish by a repetition of the same oper tions to form $y_{n+2}$ from $y_{n+1}$, $y_n$ and $y_{n-1}$. Before this can be done we must replace $y_n$ by $y_{n-1}$, $y_{n-1}$ by $y_n$ and $y_n$ by $y_{n+1}$. These shifting operations consume a serious pr portion of the time and the instructions in a machine; the manual computer perfor them merely by moving his eyes down the page.

The number of storage registers available in any machine is limited, and if more than twenty or thirty simultaneous equations are being solved the shortage of registers may be serious. For $n$ first-order equations, the number of registers required by any process is of the form $An+B$, which if $n$ is large depends primarily on $A$. If several consecutive values, or backward differences, of each variable are to be stored at any moment, $A$ will be correspondingly large.

We are thus led to consider, for applications to automatic machines, processes which do not make use of preceding function values. The general theory of a large class of such processes has been given by Kutta (1). Kutta investigated processes of various orders of accuracy; the most attractive are those of the fourth order, in which the error in each step is of the order of $h^5$, where $h$ is the length of the interval. One of these, known as the Runge-Kutta process, is sometimes used by hand computers for starting an integration. It will be shown that the general case requires $A = 4$, but that certain cases exist which can be carried out with $A = 3$, and the choice and development of one of the latter will be described.

2. *Kutta's fourth-order processes.* Consider first a single first-order equation

$$\frac{dy}{dx} = f(x, y). \tag{1}$$

Suppose that we have arrived at the point $x = X$, $y = Y$, and we want to obtain the value of $y$ corresponding to $x = X+h$. By substitution in (1), we obtain the value of $dy/dx$ at the beginning of the interval; this gives us a first approximation to the curve, a straight line between $(X, Y)$ and $(X+h, Y+k_0)$, where

$$k_0 = hf(X, Y). \tag{2}$$

We now travel a fraction $m$ of the way along this line and perform another substitution into (1) to obtain

$$k_1 = hf(X+mh, Y+mk_0). \tag{3}$$

The estimates $k_0$ and $k_1$ are now combined to find a third point at which $f$ is calculated,

$$k_2 = hf(X+nh, Y+[n-r]k_0+rk_1), \tag{4}$$

and finally

$$k_3 = hf(X+ph, Y+[p-s-t]k_0+sk_1+tk_2). \tag{5}$$

The four estimates of $k$ are now suitably combined to give the value adopted as the increment of $y$:

$$\delta y = y(X+h)-y(X) = ak_0+bk_1+ck_2+dk_3, \tag{6}$$

where

$$a+b+c+d = 1. \tag{7}$$

By choosing the coefficients appropriately, the result may be made correct as far as the terms in $h^4$.* Kutta derived the necessary conditions and suggested several possible solutions.

Before examining these conditions, let us consider the extension to simultaneous first-order differential equations (and thus to equations of higher order). Suppose we have $n$ equations

$$\frac{dy_i}{dx} = f_i(x, y_1, y_2, \ldots, y_n) \quad (i = 1, \ldots, n). \tag{8}$$

* The process may be extended to give higher orders of accuracy, by introducing more stages into the step, i.e. more quantities in the series $k_0, k_1, k_2, k_3$. However, fifth-order accuracy requires not five but six stages, and the algebra is considerably more complicated than in the fourth-order case.

The process now takes the following form:

$$\left.\begin{aligned} k_{i0} &= hf_i(X, Y_1, Y_2, \ldots),\\ k_{i1} &= hf_i(X+mh, Y_1+mk_{10}, Y_2+mk_{20}, \ldots),\\ k_{i2} &= hf_i(X+nh, Y_1+[n-r]k_{10}+rk_{11}, Y_2+[n-r]k_{20}+rk_{21}, \ldots),\\ k_{i3} &= hf_i(X+ph, Y_1+[p-s-t]k_{10}+sk_{11}+tk_{12}, \ldots),\\ \delta y_i &= ak_{i0}+bk_{i1}+ck_{i2}+dk_{i3}, \end{aligned}\right\} \quad (i=1,\ldots,n). \tag{9}$$

It is possible to simplify the theory, and also the form of the process in an automatic machine, by writing $x = y_0$. We now have $(n+1)$ equations

$$\frac{dy_i}{dx} = f_i(y_0, y_1, \ldots, y_n) \quad (i=0,\ldots,n), \tag{10}$$

where

$$f_0 \equiv 1. \tag{11}$$

With this simplification we have in place of (9):

$$\left.\begin{aligned} k_{i0} &= hf_i(Y_0, Y_1, \ldots),\\ k_{i1} &= hf_i(Y_0+mk_{00}, Y_1+mk_{10}, \ldots),\\ k_{i2} &= hf_i(Y_0+[n-r]k_{00}+rk_{01}, Y_1+[n-r]k_{10}+rk_{11}, \ldots),\\ k_{i3} &= hf_i(Y_0+[p-s-t]k_{00}+sk_{01}+tk_{02}, \ldots),\\ \delta y_i &= ak_{i0}+bk_{i1}+ck_{i2}+dk_{i3}, \end{aligned}\right\} \quad (i=0,\ldots,n). \tag{12}$$

Now $y_i(X+h)-y_i(X)$ can be expanded as a power series in $h$, and so can $\delta y_i$ as given by (12). By equating the terms up to the fourth order in $h$, we obtain the conditions to be satisfied by the coefficients in (12). Columns 2 and 3 of Table 1 show the results of these expansions, including the terms in $h^5$ which we require later for an estimation of the error. We thus arrive at the conditions:

Table 1

| Term $\left(f_i^j = \left(\frac{\partial f_i}{\partial y_j}\right)_X\right)$ | Coefficient in $y_i(X+h)-y_i(X)$ | Coefficient in $\delta y_i$ from (12) | Coefficient in $\delta y_i$ from (16) | Error in $\delta y_i$ from (16) |
|---|---|---|---|---|
| $hf_i$ | 1 | $a+b+c+d$ | 1 | 0 |
| $h^2 f_j f_i^j$ | $\frac{1}{2}$ | $bm+cn+dp$ | $\frac{1}{2}$ | 0 |
| $h^3 f_j f_k f_i^{jk}$ | $\frac{1}{6}$ | $\frac{1}{2}(bm^2+cn^2+dp^2)$ | $\frac{1}{6}$ | 0 |
| $h^3 f_j f_k^j f_i^k$ | $\frac{1}{6}$ | $crn+d(sm+tn)$ | $\frac{1}{6}$ | 0 |
| $h^4 f_j f_k f_l f_i^{jkl}$ | $\frac{1}{24}$ | $\frac{1}{6}(bm^3+cn^3+dp^3)$ | $\frac{1}{24}$ | 0 |
| $h^4 f_j f_k f_l^j f_i^{kl}$ | $\frac{1}{8}$ | $crmn+d(sm+tn)p$ | $\frac{1}{8}$ | 0 |
| $h^4 f_j f_k f_l^{jk} f_i^l$ | $\frac{1}{24}$ | $\frac{1}{2}\{crm^2+d(sm^2+tn^2)\}$ | $\frac{1}{24}$ | 0 |
| $h^4 f_j f_k^j f_l^k f_i^l$ | $\frac{1}{24}$ | $dtrm$ | $\frac{1}{24}$ | 0 |
| $h^5 f_j f_k f_l f_m f_i^{klm}$ | $\frac{1}{120}$ | $\frac{1}{24}(bm^4+cn^4+dp^4)$ | $\frac{5}{378}$ | $+\frac{1}{2860}$ |
| $h^5 f_j f_k f_l f_m^l f_i^{km}$ | $\frac{1}{60}$ | $\frac{1}{2}\{crmn^2+d(sm+tn)p^2\}$ | $\frac{5}{360}$ | $+\frac{1}{486}$ |
| $h^5 f_j f_k f_l^{kl} f_m^m f_i^m$ | $\frac{1}{30}$ | $\frac{1}{2}\{crm^2n+d(sm^2+tn^2)p\}$ | $\frac{1}{252}$ | $-\frac{1}{486}$ |
| $h^5 f_j f_k f_l^k f_m^l f_i^m$ | $\frac{1}{30}$ | $dtrmp$ | $\frac{1}{24}$ | $+\frac{1}{120}$ |
| $h^5 f_j f_k f_l f_m^{kl} f_i^m$ | $\frac{1}{120}$ | $\frac{1}{6}\{crm^3+d(sm^3+tn^3)\}$ | $\frac{1}{144}$ | $-\frac{1}{720}$ |
| $h^5 f_j f_k f_l^l f_m^k f_i^m$ | $\frac{1}{40}$ | $dtrmn$ | $\frac{1}{48}$ | $-\frac{1}{240}$ |
| $h^5 f_j f_k f_l^m f_m^{kl} f_i^l$ | $\frac{1}{120}$ | $\frac{1}{2}dtrm^2$ | $\frac{1}{96}$ | $+\frac{1}{480}$ |
| $h^5 f_j f_k^l f_l^k f_m^l f_i^m$ | $\frac{1}{120}$ | 0 | 0 | $-\frac{1}{120}$ |
| $h^5 f_j f_k f_l^k f_m^l f_i^m$ | $\frac{1}{40}$ | $\frac{1}{2}(cr^2m^2+d(sm+tn)^2)$ | $\frac{1}{24}\mp\frac{1}{48}\sqrt{\frac{1}{3}}$ | $\frac{1}{40}\mp\frac{1}{48}\sqrt{\frac{1}{3}}$ |

$$\left.\begin{aligned} a+b+c+d &= 1,\\ bm+cn+dp &= \tfrac{1}{2},\\ bm^2+cn^2+dp^2 &= \tfrac{1}{3},\\ cmr+d(nt+ms) &= \tfrac{1}{6},\\ bm^3+cn^3+dp^3 &= \tfrac{1}{4},\\ cmnr+d(nt+ms)p &= \tfrac{1}{8},\\ cm^2r+d(n^2t+m^2s) &= \tfrac{1}{12},\\ dmrt &= \tfrac{1}{24}, \end{aligned}\right\} \qquad (13)$$

which imply

$$p = 1. \qquad (14)$$

We have here two degrees of freedom. It is possible to express each quantity rationally in terms of $m$ and $n$, but the expressions are clumsy and difficult to use in any further analysis. However, Kutta suggested five special cases in which, whilst one degree of freedom is retained, the quantities can be expressed in particularly simple forms. These are listed in Table 2. The Runge-Kutta rule, the simplest particular solution, may be obtained by putting $m = \frac{1}{2}$ in case (i), or $t = 1$ in case (v).

The main disadvantage of this type of process is that four substitutions have to be made into the given equations (10) for each step of the integration, and in this respect it compares unfavourably with other types of similar accuracy. This is part of the price that has to be paid for abandoning preceding function values, and if the equations (10) are complicated this may be a serious drawback. However, systems consisting of many equations are, in practice, usually simple in form, or may be made so by the introduction of additional variables, using techniques similar to those employed on the Differential Analyser.

Table 2

| Quantity | Case (i) | Case (ii) | Case (iii) | Case (iv) | Case (v) |
|---|---|---|---|---|---|
| $m$ | $m$ | $m$ | $\frac{1}{2}$ | 1 | $\frac{1}{2}$ |
| $n$ | $1-m$ | $\frac{1}{2}$ | 0 | $\frac{1}{2}$ | $\frac{1}{2}$ |
| $p$ | 1 | 1 | 1 | 1 | 1 |
| $r$ | $\dfrac{1-m}{2m}$ | $\dfrac{1}{8m}$ | $r$ | $\frac{1}{8}$ | $\dfrac{1}{2t}$ |
| $s$ | $\dfrac{(1-m)(2m-1)}{2m[6m(1-m)-1]}$ | $-\dfrac{1}{2m}$ | $\frac{3}{2}$ | $-\frac{t}{4}$ | $1-t$ |
| $t$ | $\dfrac{m}{6m(1-m)-1}$ | 2 | $\dfrac{1}{2r}$ | $t$ | $t$ |
| $a$ | $\dfrac{6m(1-m)-1}{12m(1-m)}$ | $\frac{1}{6}$ | $\frac{1}{6}-\dfrac{1}{12r}$ | $\frac{1}{6}$ | $\frac{1}{6}$ |
| $b$ | $\dfrac{1}{12m(1-m)}$ | 0 | $\frac{2}{3}$ | $\frac{1}{6}-\dfrac{1}{3t}$ | $\dfrac{2-t}{3}$ |
| $c$ | $\dfrac{1}{12m(1-m)}$ | $\frac{2}{3}$ | $\dfrac{1}{12r}$ | $\frac{2}{3}$ | $\frac{t}{3}$ |
| $d$ | $\dfrac{6m(1-m)-1}{12m(1-m)}$ | $\frac{1}{6}$ | $\frac{1}{6}$ | $\dfrac{1}{3t}$ | $\frac{1}{6}$ |

3. *Choice of process.* We have now to consider the number of storage registers per equation required by the process as given by (12). At the beginning of the step we need $n+1$ registers to hold the quantities $Y_0, \ldots, Y_n$, i.e. one register per equation. For the first stage we need one more register per equation, to receive the quantities $k_{i0}$. For the second stage we need $Y_i + mk_{i0}$, and a register to receive $k_{i1}$. We must also store in some form $Y_i + [n-r]k_{i0}$ which will be needed at the next stage, $Y_i + [p-s-t]k_{i0}$ which will be needed at the last stage, and $Y_i + ak_{i0}$ which will be needed at the end of the step. However, these five quantities are linearly dependent and can certainly be represented by three, and so three registers will suffice for the second stage.

It is at the third stage that the maximum number of registers is required. We have to store in some form

$$Y_i + [n-r]k_{i0} + rk_{i1}, \quad Y_i + [p-s-t]k_{i0} + sk_{i1}, \quad Y_i + ak_{i0} + bk_{i1} \quad \text{and} \quad k_{i2},$$

and in general four registers are necessary to each equation. At the last stage three are certainly sufficient, because we do not have to consider forming a value of $y$ for a following stage. Thus if we can reduce the number of registers required in the third stage to three per equation, we need never exceed this number in the whole process.

Clearly, three registers will suffice for the third stage if the quantities to be stored are linearly dependent, i.e. if

$$\begin{vmatrix} 1 & n-r & r \\ 1 & p-s-t & s \\ 1 & a & b \end{vmatrix} = 0. \tag{15}$$

This condition is not incompatible with (13), and the combined equations have a single infinity of solutions. To find the simplest particular solutions we can apply (15) to each of the cases listed in Table 2. The results are as follows.

*Case* (i). $3m^3 - 9m^2 + 6m - 1 = 0$, whence

$$m_N = 1 + \sqrt{\tfrac{4}{3}} \cos(120N + 10)^\circ = 0{\cdot}258,\ 0{\cdot}605 \text{ or } 2{\cdot}137.$$

None of these cases is very attractive as the basis of a practical process.

*Case* (ii). There is no solution.

*Case* (iii). $4r^2 + 4r + 5 = 0$. There is no real solution.

*Case* (iv). $2t^2 - 6t + 5 = 0$. There is no real solution.

*Case* (v). $2t^2 - 4t + 1 = 0$, whence $t = 1 \pm \sqrt{\tfrac{1}{2}}$.

The corresponding values for the other constants are

$$\left.\begin{aligned} m &= \tfrac{1}{4}, \quad r = 1 \mp \sqrt{\tfrac{1}{2}}, \quad a = \tfrac{1}{4}, \\ n &= \tfrac{1}{4}, \quad s = \mp\sqrt{\tfrac{1}{2}}, \quad b = \tfrac{1}{4}(1 \mp \sqrt{\tfrac{1}{2}}), \\ p &= 1, \quad t = 1 \pm \sqrt{\tfrac{1}{2}}, \quad c = \tfrac{1}{6}(1 \pm \sqrt{\tfrac{1}{2}}), \\ & \qquad\qquad d = \tfrac{1}{6}. \end{aligned}\right\} \tag{16}$$

These two solutions are probably the simplest that exist. They possess, in common with the Runge-Kutta case, the advantage that the independent variable assumes only values corresponding to the beginning and end of each step, and the midpoint of each

step. This is valuable in cases where the equations involve a function of $x$ which is defined by a table at equal intervals.

To decide which form of (16) to use, we appeal to the fifth-order terms in the expansion of $y$. The last two columns of Table 1 show that only the last of those terms is affected by our choice, and clearly indicate the adoption of the upper sign for the square root throughout (16). The rest of the paper will deal solely with this case.

Substituting the values given by (16) into the equations (12), and introducing new symbols for the intermediate values of $y$, we have

$$\left.\begin{aligned} k_{i0} &= hf_i(y_{00}, y_{10}, y_{20}, \ldots),\\ k_{i1} &= hf_i(y_{01}, y_{11}, y_{21}, \ldots),\\ k_{i2} &= hf_i(y_{02}, y_{12}, y_{22}, \ldots),\\ k_{i3} &= hf_i(y_{03}, y_{13}, y_{23}, \ldots), \end{aligned}\right\} \tag{17}$$

where

$$\left.\begin{aligned} y_{i0} &= Y_i,\\ y_{i1} &= Y_i + \tfrac{1}{2}k_{i0},\\ y_{i2} &= Y_i + [-\tfrac{1}{2} + \sqrt{\tfrac{1}{2}}]\,k_{i0} + [1 - \sqrt{\tfrac{1}{2}}]\,k_{i1},\\ y_{i3} &= Y_i + [-\sqrt{\tfrac{1}{2}}]\,k_{i1} + [1 + \sqrt{\tfrac{1}{2}}]\,k_{i2}, \end{aligned}\right\} \tag{18}$$

and the adopted value of $y_i$ at the end of the step is

$$y_{i4} = Y_i + \tfrac{1}{6}k_{i0} + \tfrac{1}{3}[1 - \sqrt{\tfrac{1}{2}}]\,k_{i1} + \tfrac{1}{3}[1 + \sqrt{\tfrac{1}{2}}]\,k_{i2} + \tfrac{1}{6}k_{i3}. \tag{18a}$$

In calculating $k_{ip}$ $(i = 0, \ldots, n)$ for a particular $p$, we require a register to hold each $y_{ip}$, and a register to receive each $k_{ip}$. For the quantities stored in the third set of registers, any linear combination of $Y_i$ and the preceding values of $k_i$, such as will enable us to derive the quantities required in succeeding stages, will meet the case. There is, however, as will appear later, some advantage in choosing a quantity which is purely a combination of the preceding values of $k_i$ and does not contain $Y_i$; such a quantity will therefore be of the order of $h$. This determines it apart from an arbitrary factor, and we shall adopt the following:

$$\left.\begin{aligned} q_{i1} &= k_{i0},\\ q_{i2} &= [-2 + 3\sqrt{\tfrac{1}{2}}]\,k_{i0} + [2 - \sqrt{2}]\,k_{i1},\\ q_{i3} &= [-\tfrac{1}{2}]\,k_{i0} + [-1 - \sqrt{2}]\,k_{i1} + [2 + \sqrt{2}]\,k_{i2}. \end{aligned}\right\} \tag{19}$$

Introducing the quantities defined by (18) and (19) into the process as given by (17), we obtain

$$\left.\begin{aligned} k_{i0} &= hf_i(y_{00}, y_{10}, \ldots),\\ y_{i1} = y_{i0} + \tfrac{1}{2}k_{i0}, \qquad & q_{i1} = k_{i0},\\ k_{i1} &= hf_i(y_{01}, y_{11}, \ldots),\\ y_{i2} = y_{i1} + [1 - \sqrt{\tfrac{1}{2}}](k_{i1} - q_{i1}), \qquad & q_{i2} = [2 - \sqrt{2}]\,k_{i1} + [-2 + 3\sqrt{\tfrac{1}{2}}]\,q_{i1},\\ k_{i2} &= hf_i(y_{02}, y_{12}, \ldots),\\ y_{i3} = y_{i2} + [1 + \sqrt{\tfrac{1}{2}}](k_{i2} - q_{i2}), \qquad & q_{i2} = [2 + \sqrt{2}]\,k_{i2} + [-2 - 3\sqrt{\tfrac{1}{2}}]\,q_{i2},\\ k_{i3} &= hf_i(y_{03}, y_{13}, \ldots),\\ y_{i4} = y_{i3} + \tfrac{1}{6}k_{i3} - \tfrac{1}{3}q_{i2}. \end{aligned}\right\} \tag{20}$$

The sequence of events during one step is as follows:

(i) Calculate $k_{i0}$ $(i = 0, \ldots, n)$.
(ii) Calculate $y_{i1}$ and $q_{i1}$ $(i = 0, \ldots, n)$. The most convenient order in which t do this is $y_{01}, q_{01}; y_{11}, q_{11}; y_{21}, q_{21}; \cdots$
(iii) Calculate $k_{i1}$ $(i = 0, \ldots, n)$.
(iv) Calculate $y_{i2}$ and $q_{i2}$ $(i = 0, \ldots, n)$.

.....

(viii) Calculate $y_{i4}$ $(i = 0, \ldots, n)$.

As each quantity is calculated it is stored in the register formerly holding the corr sponding quantity of the previous stage, which is no longer required.

It would be quite possible to use the process in this form, but one or two refinemen can be made to give greater accuracy whilst using the same storage space, with litt increase in the complexity of the process. The rest of the paper is devoted to t development and study of the more accurate form.

4. *Obtaining maximum accuracy.* The process as it stands suffers from the disa vantage that many rounding-off errors are accumulated during each step. If the err committed in the determination of each of the eleven quantities in (20) is represent by the operator $e$, then the total error in $y_t$ accumulated during one step is

$$\begin{aligned} e(\delta y_i) = {} & \tfrac{1}{6}e(k_{i0}) + \tfrac{1}{3}[1 - \sqrt{\tfrac{1}{2}}]\,e(k_{i1}) + \tfrac{1}{3}[1 + \sqrt{\tfrac{1}{2}}]\,e(k_{i2}) + \tfrac{1}{6}e(k_{i3}) \\ & + e(y_{i1}) + e(y_{i2}) + e(y_{i3}) + e(y_{i4}) \\ & - \tfrac{1}{2}e(q_{i1}) - \tfrac{1}{2}e(q_{i2}) - \tfrac{1}{2}e(q_{i3}). \end{aligned} \tag{2}$$

Now the quantities $k$ are small, being of the order of $h$. It is possible therefore store them to a higher degree of absolute accuracy than can be achieved in $y$, b introducing a scale factor $g$, i.e. by storing $g^{-1}k$ instead of $k$. If we assume that the sar number of digits is available for $k$ as for $y$, then $g$ can be made comparable with without causing $g^{-1}k$ to exceed the capacity of the register. Moreover, the extra digi thus stored are significant provided that the $f_t^i$ are of the order of unity or less, for this condition holds, then $f_t$ can be obtained to the same order of accufacy as $y$, and will have an error of the order of $h$ times that of $y$. The same factor may be appli to $q$, because

$$q_{i1} = k_{i0}, \quad q_{i2} \sim (\sqrt{\tfrac{1}{2}})\,k, \quad q_{i3} \sim \tfrac{1}{2}k, \tag{2}$$

and hence if $g^{-1}k$ is within capacity, so is $g^{-1}q$. The developments now to be describ depend on the fact that $k$ and $q$ are stored in this way.

The rounding-off errors of $k$ and $q$ may thus be made negligible compared with tho of $y$, which are then the only large terms remaining on the right-hand side of (2) However, we are still at liberty to adjust $q$ if we please, and the possibility now aris of introducing 'deliberate errors' in the estimations of $q$, to compensate almost exact for the rounding-off errors which we are forced to make in $y$. This is possible becau we can choose $q$ from an array of values which has a much closer spacing than t available values for representing $y$. In fact, if we make the errors in $q$ satisfy as near as possible the following:

$$e(q_{i1}) = 3e(y_{i1}), \quad e(q_{i2}) = 3e(y_{i2}), \quad e(q_{i3}) = 3e(y_{i3}), \tag{2}$$

then the resultant $e(\delta y_i)$ as given by (21) contains only quantities of the same order as the rounding-off errors of $k$ and $q$, with the exception of the single term $e(y_{i4})$. Before considering how further reductions of error may be made, let us see how the process appears with these modifications.

We first define the auxiliary quantities

$$\left.\begin{aligned} r_{i1} &= y_{i1} - y_{i0} = \tfrac{1}{2}k_{i0} + e(r_{i1}), \\ r_{i2} &= y_{i2} - y_{i1} = [1 - \sqrt{\tfrac{1}{2}}]\,(k_{i1} - q_{i1}) + e(r_{i2}), \\ r_{i3} &= y_{i3} - y_{i2} = [1 + \sqrt{\tfrac{1}{2}}]\,(k_{i2} - q_{i2}) + e(r_{i3}). \end{aligned}\right\} \tag{24}$$

The last term in each case represents an error which is deliberately introduced in order to round-off $r$ to the same digital accuracy as $y$. The new value of $y$ is now found by adding $r$ to the previous value, no further rounding-off being necessary; in fact, $e(r_{ip}) = e(y_{ip})$. If now we arrange that the corresponding $q$ contains a term $3r$, we shall have achieved our object. The values of $q$ are now defined thus:

$$\left.\begin{aligned} q_{i1} &= 3r_{i1} - \tfrac{1}{2}k_{i0}, \\ q_{i2} &= q_{i1} + 3r_{i2} - [1 - \sqrt{\tfrac{1}{2}}]\,k_{i1}, \\ q_{i3} &= q_{i2} + 3r_{i3} - [1 + \sqrt{\tfrac{1}{2}}]\,k_{i2}. \end{aligned}\right\} \tag{25}$$

Note that the quantities $r$ are only used temporarily during the formation of each $y$ and $q$; we do not need an extra register for each variable.

We now return to consider the remaining large term in $e(\delta y_i)$, namely, $e(y_{i4})$. With our present scheme, $e(y_{i4})$ can be neither eliminated nor compensated for because we have no quantity $q_{i4}$ which can be adjusted accordingly. We have, in fact, achieved a high degree of accuracy within each step, but we are throwing away some of this accuracy at the end of the step by rounding-off $y_{i4}$ without keeping any record of the rounding-off error. There is, however, no objection to introducing a quantity $q_{i4}$, solely for the purpose of retaining the high accuracy of the calculation from one step to the next. This 'bridging' value of $q$ will be much smaller than the values encountered within the step, and in fact it will be equal, as nearly as possible, to three times the rounding-off error of $y_{i4}$. It becomes $q_{i0}$ for the following step.

We now have:

$$\left.\begin{aligned} k_{i0} &= hf_i(y_{00}, y_{10}, \ldots), & r_{i1} &= \tfrac{1}{2}k_{i0} - wq_{i0} + e(r_{i1}), \\ y_{i1} &= y_{i0} + r_{i1}, & q_{i1} &= q_{i0} + 3r_{i1} - \tfrac{1}{2}k_{i0}, \\ k_{i1} &= hf_i(y_{01}, y_{11}, \ldots), & r_{i2} &= [1 - \sqrt{\tfrac{1}{2}}]\,(k_{i1} - q_{i1}) + e(r_{i2}), \\ y_{i2} &= y_{i1} + r_{i2}, & q_{i2} &= q_{i1} + 3r_{i2} - [1 - \sqrt{\tfrac{1}{2}}]\,k_{i1}, \\ k_{i2} &= hf_i(y_{02}, y_{12}, \ldots), & r_{i3} &= [1 + \sqrt{\tfrac{1}{2}}]\,(k_{i2} - q_{i2}) + e(r_{i3}), \\ y_{i3} &= y_{i2} + r_{i3}, & q_{i3} &= q_{i2} + 3r_{i3} - [1 + \sqrt{\tfrac{1}{2}}]\,k_{i2}, \\ k_{i3} &= hf_i(y_{03}, y_{13}, \ldots), & r_{i4} &= \tfrac{1}{6}(k_{i3} - 2q_{i3}) + e(r_{i4}), \\ y_{i4} &= y_{i3} + r_{i4}, & q_{i4} &= q_{i3} + 3r_{i4} - \tfrac{1}{2}k_{i3}. \end{aligned}\right\} \tag{26}$$

The coefficient $w$ in the expression for $r_{i1}$ is not very critical. It appears at first sight that we should take $w = \frac{1}{2}$, for this would give us the best possible estimate for $y_{i1}$, but it will be shown later that in fact the overall optimum value is $w = 1$. However,

since $q_{i0}$ is of the order of the last digit of $y$, and errors in $r$ are compensated for over t step, we introduce no appreciable error by changing $w$, if this is convenient. The er thus introduced is considered in the next section.

This modification has not destroyed the facility of automatic starting possessed the process. The bridging $q$ serves merely to ensure that rounding-off errors in $y$ do n build up appreciably over a large range of integration. At the beginning of a range can with little loss of accuracy set $q_{i0} = 0$; in fact, in many cases, the initial value of will be given exactly with no rounding-off error. The facility of being able to chan $h$ between any two steps is still available, provided we do not wish to change $g$. $g$ is changed we must either change every $g^{-1}q_{i0}$ correspondingly, or suffer a sma persistent error of the order of the rounding-off error in $y$ times the relative change in

The quantities $q$ now play a double role; they serve to give the process the analytic form of (20), and they also effectively carry a few extra digits of $y$. The process is he in its final form. Apart from the economy of registers, two features make the proce particularly suitable for use in an automatic machine: first, since each value of $k$, and $q$ may be 'written over' its predecessor, wasteful shifting operations are avoide and secondly, the four stages composing one step are very similar in form and can b carried out by substantially the same sequence of instructions.

5. *Analysis of errors.* We now have to make a more thorough study of the erro occurring in the final form of the process. The quantity with which we are primaril concerned now is the value of $y_{i4} - \frac{1}{3}q_{i4}$ at the end of each step, for this represents th nearest estimate we possess to the analytical solution. The quantity we denote b $y_i$ will have rounding-off errors, but these will not accumulate; they will nearly canc out over a short range, and the cumulative error over a large range will be the sum o the errors in $y_{i4} - \frac{1}{3}q_{i4}$.

These errors are of two kinds: those due to rounding-off, and 'truncation' erro caused by the fact that the process yields only fourth-order accuracy. The latter hav been listed in Table 1, and little more need be said about them here. It must be remem bered that each of the fifth-degree terms in that table is the sum of $(n+1)^4$ produc of the functions $f$ and their derivatives. In practice, however, where $n$ is large, such a in the solution of a partial differential equation as a set of ordinary equations, th majority of these products will vanish. In most cases the last term but one will probably be dominant, since it contains only first-order derivatives.

Coming now to rounding-off errors, let us consider first the errors introduced b using rounded-off forms of the coefficients $\sqrt{\frac{1}{2}}$ and $\frac{1}{6}$. If $\sqrt{\frac{1}{2}}$ is replaced by $\sqrt{\frac{1}{2}} + e_1$, in vestigation shows that the resultant error in $y_{i4} - \frac{1}{3}q_{i4}$ is $\frac{1}{3}(k_{i2} - k_{i1})e_1$. Expressed a a power series in $h$, the largest term is

$$\tfrac{1}{6}[1 - \sqrt{\tfrac{1}{2}}]h^2 f_i f_i' f_i^2 e_1.$$

Thus if $\sqrt{\frac{1}{2}}$ is stored to the same accuracy as $y$, the resultant error per step is of the orde of $h^2u$, where $u$ is the value of one unit in the last digit of $y$; over a finite range the erro is $O(h^2u)$ and is quite negligible.

If $\frac{1}{6} + e_2$ is used in place of $\frac{1}{6}$ in forming $r_{i4}$, no error is produced in $y_{i4} - \frac{1}{3}q_{i4}$, i.e. n cumulative error is caused directly. However, indirect errors arise from using in

correct values of $y_{i4}$ in the calculation of $k_{j4}$. The effect of the errors in $y_i$ on the values of $k_j$ will be considered shortly, but for the present we need only note that the effect of $e_2$ will be entirely negligible. Owing to rounding-off it is meaningless to attempt to keep the error in $y_{i4}$ much below about $\frac{1}{2}u$. The error in $y_{i4}$ due to $e_2$ is actually

$$(k_{i0}+[2+2\sqrt{2}]k_{i1}+[-4-2\sqrt{2}]k_{i2}+k_{i3})e_2,$$

which is of the order of $h^3e_2$.

We now proceed to estimate the effect on $y_{i4}-\frac{1}{3}q_{i4}$ of rounding-off errors in the other quantities. The accumulated error is due entirely to the errors in $k_i$ and $q_i$, because the terms in $e(r_i)$ cancel out. However, errors in $k_i$ are caused partly by using erroneous values of $y_j$, and so it is convenient to begin by studying the behaviour of $y$ itself. If we assume that $y_{i0}-\frac{1}{3}q_{i0}$ represents the correct value of the function at the beginning of the step, the differences between the stored values $y_i$ and the correct solution are due almost entirely to the terms $e(r_i)$, which are larger than the errors in $k_i$ and $q_i$ by a factor of the order of $h^{-1}$. Now we have seen that each $e(r_i)$ has no direct effect on quantities in following steps. It can and does, however, affect following values of $r_i$, $q_i$ and $y_i$ up to the end of the present step. Thus $e(r_{i1})$, for example, affects not only $y_{i1}$ but $y_{i2}$ and $y_{i3}$ as well. The errors in $y$ within one step are therefore not independent, and in fact we shall find that they tend to cancel one another out. Table 3 shows the effect on each quantity within one step of the errors $e(r_{i0})$, $e(r_{i1})$, $e(r_{i2})$ and $e(r_{i3})$, where $r_{i0}$ is the quantity denoted by $r_{i4}$ in the preceding step.

We represent total errors by $E$, the error in any $k_i$ due to errors in $y_j$ is

$$h\sum_j f'_i E(y_j).$$

The error in $y_{i4}-\frac{1}{3}q_{i4}$ due to errors in $k_i$ is, by (18$a$),

$$\tfrac{1}{6}E(k_{i0})+\tfrac{1}{3}[1-\sqrt{\tfrac{1}{2}}]E(k_{i1})+\tfrac{1}{3}[1+\sqrt{\tfrac{1}{2}}]E(k_{i2})+\tfrac{1}{6}E(k_{i3}), \qquad (27)$$

and the contribution to this due to errors in $y_j$ is therefore

$$h\sum_j f'_i\{\tfrac{1}{6}E(y_{j0})+\tfrac{1}{3}[1-\sqrt{\tfrac{1}{2}}]E(y_{j1})+\tfrac{1}{3}[1+\sqrt{\tfrac{1}{2}}]E(y_{j2})+\tfrac{1}{6}E(y_{j3})\},$$

Table 3

| Quantity | Coefficient of errors | | | |
|---|---|---|---|---|
| | $e(r_{i0})$ | $e(r_{i1})$ | $e(r_{i2})$ | $e(r_{i3})$ |
| $r_{i0}$ | 1 | 0 | 0 | 0 |
| $q_{i0}$ | 3 | 0 | 0 | 0 |
| $y_{i0}$ | 1 | 0 | 0 | 0 |
| $r_{i1}$ | $-3w$ | 1 | 0 | 0 |
| $q_{i1}$ | $3(1-3w)$ | 3 | 0 | 0 |
| $y_{i1}$ | $1-3w$ | 1 | 0 | 0 |
| $r_{i2}$ | $-3(1-\sqrt{\frac{1}{2}})(1-3w)$ | $-3(1-\sqrt{\frac{1}{2}})$ | 1 | 0 |
| $q_{i2}$ | $3(-2+3\sqrt{\frac{1}{2}})(1-3w)$ | $3(-2+3\sqrt{\frac{1}{2}})$ | 3 | 0 |
| $y_{i2}$ | $(-2+3\sqrt{\frac{1}{2}})(1-3w)$ | $(-2+3\sqrt{\frac{1}{2}})$ | 1 | 0 |
| $r_{i3}$ | $3(\frac{1}{2}-\sqrt{\frac{1}{2}})(1-3w)$ | $3(\frac{1}{2}-\sqrt{\frac{1}{2}})$ | $-3(1+\sqrt{\frac{1}{2}})$ | 1 |
| $q_{i3}$ | $-\frac{3}{2}(1-3w)$ | $-\frac{3}{2}$ | $-3(2+3\sqrt{\frac{1}{2}})$ | 3 |
| $y_{i3}$ | $-\frac{1}{2}(1-3w)$ | $-\frac{1}{2}$ | $-(2+3\sqrt{\frac{1}{2}})$ | 1 |
| $r_{i4}$ | $\frac{1}{2}(1-3w)$ | $\frac{1}{2}$ | $(2+3\sqrt{\frac{1}{2}})$ | $-1$ |
| $q_{i4}$ | 0 | 0 | 0 | 0 |
| $y_{i4}$ | 0 | 0 | 0 | 0 |

neglecting changes in $f_i^j$ during the step. Substituting the expressions for $E(y_j)$ from Table 3 we obtain

$$h\sum_j f_i^j\{\tfrac{1}{6}(1-w)\,e(r_{j0})+\tfrac{1}{12}e(r_{j1})-\tfrac{1}{6}\sqrt{\tfrac{1}{2}}\,e(r_{j2})+\tfrac{1}{6}e(r_{j3})\}, \tag{28}$$

which is somewhat smaller than would be obtained if the errors in $y_j$ were independent. Assuming that the values of $e(r_j)$ are independent and randomly distributed between $-\frac{1}{2}u$ and $+\frac{1}{2}u$, we find that the standard deviation in $y_{i4}-\frac{1}{3}q_{i4}$ from this source is equal to

$$\tfrac{1}{12}[(3[1-w]^2+\tfrac{7}{3})\sum_j (f_i^j)^2]^{\frac{1}{2}}\,hu. \tag{29}$$

This expression leads to the optimum value of $w = 1$ quoted in the last section.

It remains to estimate the errors due to rounding-off $k_i$ and $q_i$. The behaviour of $k_i$ may be abnormal in some cases, depending on the form of the given functions $f$, but we shall assume that the rounding-off errors in $k_i$ are randomly distributed between $-\frac{1}{2}gu$ and $+\frac{1}{2}gu$. By (27) they cause a standard deviation in $y_{i4}-\frac{1}{3}q_{i4}$ equal to $\frac{1}{6}\sqrt{\frac{7}{6}}\,gu$.

The rounding-off errors in $q_i$ are simply additive over the step. Referring to (26) we see that they are due entirely to the last term in each expression for $q_i$, since, assuming that we have chosen $g^{-1}$ to be an integer, the first two terms are both exact to the number of digits to which $g^{-1}q_i$ is stored. In $q_{i2}$ and $q_{i3}$ the coefficient of $k_i$ in the last term is not a simple ratio, and therefore the rounding-off error will be random between $-\frac{1}{2}gu$ and $+\frac{1}{2}gu$. In $q_{i1}$ and $q_{i4}$, however, the coefficient is in each case $-\frac{1}{4}$, and we cannot round off without introducing a bias of $+\frac{1}{4}gu$ or $-\frac{1}{4}gu$. It is important that the total bias should be removed by rounding-off $q_{i1}$ and $q_{i4}$ in opposite directions, one up and one down. The standard deviation in each case is $\frac{1}{4}gu$. The resultant standard deviation in $y_{i4}-\frac{1}{3}q_{i4}$ from the rounding-off of $q_i$ is thus $\frac{1}{6}\sqrt{\frac{7}{6}}gu$.

The combined standard deviation in $y_i-\frac{1}{3}q_i$ over one step from all rounding-off errors is

$$\tfrac{1}{6}[\tfrac{7}{12}g^2+\tfrac{1}{12}(3[1-w]^2+\tfrac{7}{3})\,h^2\sum_j (f_i^j)^2]^{\frac{1}{2}}\,u. \tag{30}$$

In most cases $\Sigma(f_i^j)^2$ will be sufficiently small to make the second term in the root unimportant, and the value of $w$ will not appreciably affect the accuracy of the process. If $\Sigma(f_i^j)^2$ is exceptionally large the effect of taking, say, $w = 0$ instead of $w = 1$ may just be noticeable.

If $w = 1$, (30) becomes

$$\tfrac{1}{6}[\tfrac{7}{12}\{g^2+\tfrac{1}{12}h^2\sum_j (f_i^j)^2\}]^{\frac{1}{2}}\,u, \tag{31}$$

which is the same standard deviation as would be caused by a single rounding-off to the nearest multiple of

$$v = \tfrac{1}{6}[7\{g^2+\tfrac{1}{12}h^2\sum_j (f_i^j)^2\}]^{\frac{1}{2}}\,u \simeq gu. \tag{32}$$

The overall accuracy of the process, if truncation errors are negligible, is therefore approximately the same as would be obtained if $y$ were rounded-off at each step to the nearest multiple of $gu$, so that we are effectively keeping a number of extra digits of $y$ corresponding to the value of $g$. It is important to note that $q_{i1}$ and $q_{i4}$ must be oppositely rounded-off to eliminate bias.

6. *Example.* As a simple example, the single equation $y' = y$ will be integrated from $x = 0$ to $x = 1$ in steps of $h = 0{\cdot}1$. We shall suppose that we are using a decimal machine

with registers of six-figure capacity, and that the decimal point is at the left-hand end of the register. The starting value will be $y(0) = 0{\cdot}1$. We can take $g = h = 0{\cdot}1$, for this will make $g^{-1}k = g^{-1}hy' = y$, so that $g^{-1}k$ will be within capacity. We shall take the value 0·7071 for $\sqrt{\frac{1}{2}}$; the error is about $7 \times 10^{-6}$, which is just small enough to have negligible effect. For $\frac{1}{6}$, the value 0·1667 is sufficiently accurate. We shall use the case $w = 1$.

Table 4

| $x$ | Stage | $r$ | $y = g^{-1}k$ | $g^{-1}q$ |
|---|---|---|---|---|
| 0·0 | 0 | | 100 000 | 0 |
| | 1 | 5 000 | 105 000 | 100 000 |
| | 2 | 146 | 105 146 | 73 626 |
| | 3 | 5 381 | 110 527 | 55 561 |
| 0·1 | 4 | − 10 | 110 517 | − 3 |
| | 1 | 5 526 | 116 043 | 110 519 |
| | 2 | 162 | 116 205 | 81 390 |
| | 3 | 5 943 | 122 148 | 61 306 |
| 0·2 | 4 | − 8 | 122 140 | − 8 |
| | 1 | 6 108 | 128 248 | 122 162 |
| | 2 | 178 | 128 426 | 89 938 |
| | 3 | 6 570 | 134 996 | 67 802 |
| 0·3 | 4 | − 10 | 134 986 | + 4 |
| | 1 | 6 749 | 141 735 | 134 981 |
| | 2 | 198 | 141 933 | 99 407 |
| | 3 | 7 260 | 149 193 | 74 913 |
| 0·4 | 4 | − 11 | 149 182 | − 14 |
| | 1 | 7 461 | 156 643 | 149 225 |
| | 2 | 217 | 156 860 | 109 854 |
| | 3 | 8 024 | 164 884 | 82 798 |
| 0·5 | 4 | − 12 | 164 872 | − 4 |
| | 1 | 8 244 | 173 116 | 164 880 |
| | 2 | 241 | 173 357 | 121 404 |
| | 3 | 8 869 | 182 226 | 91 536 |
| 0·6 | 4 | − 14 | 182 212 | + 3 |
| | 1 | 9 110 | 191 322 | 182 197 |
| | 2 | 267 | 191 589 | 134 169 |
| | 3 | 9 802 | 201 391 | 101 167 |
| 0·7 | 4 | − 16 | 201 375 | − 9 |
| | 1 | 10 070 | 211 445 | 201 404 |
| | 2 | 294 | 211 739 | 148 292 |
| | 3 | 10 831 | 222 570 | 111 762 |
| 0·8 | 4 | − 16 | 222 554 | − 3 |
| | 1 | 11 128 | 233 682 | 222 560 |
| | 2 | 326 | 234 008 | 163 895 |
| | 3 | 11 969 | 245 977 | 123 490 |
| 0·9 | 4 | − 17 | 245 960 | − 9 |
| | 1 | 12 299 | 258 259 | 245 981 |
| | 2 | 360 | 258 619 | 181 137 |
| | 3 | 13 227 | 271 846 | 136 459 |
| 1·0 | 4 | − 18 | 271 828 | − 4 |

The truncation error will consist entirely of the last term but one in Table 1, w gives

$$-\tfrac{1}{120}h^5y = -8{\cdot}3\times10^{-8}y \quad \text{per step,}$$

i.e.

$$-8{\cdot}3\times10^{-8}\Sigma y = -14\times10^{-8} \quad \text{over the range.}$$

In this particular example there is no rounding-off error introduced by $k$. Under this condition, the theory of the last section predicts a standard deviation of

$$\tfrac{1}{24}\sqrt{(21)}gu = 2\times10^{-8} \quad \text{over one step,}$$

i.e.

$$\sqrt{(10)}\times2\times10^{-8} = 6\times10^{-8} \quad \text{over the range.}$$

To remove bias, $q_1$ will be rounded up and $q_4$ down. Since $g^{-1}k = y$, the common value need only be recorded once.

The results of the calculation are shown in Table 4. It will be seen that the value of $y - \frac{1}{3}q$ at the end of the range is 0·27182813, which has an error of $-5\times10^{-8}$.

7. *Application.* A routine has been prepared by the author for applying the process on the EDSAC (Electronic Delay Storage Automatic Calculator) at the Mathematical Laboratory. The routine handles any number of equations within the storage capacity of the machine; it consists of sixty-seven instructions (single address), and has to be supplemented by a routine which evaluates every $k$ as required by the given system of equations. The time required to perform each step is 0·21 sec. per variable, plus four times the time required to evaluate the $k$'s.

*Acknowledgements.* The author wishes to thank Mr D. J. Wheeler for many helpful suggestions, Dr M. V. Wilkes for advice in preparing the manuscript, and the Department of Scientific and Industrial Research for a Maintenance Allowance.

REFERENCE

(1) KUTTA, W. *Z. Math. Phys.* 46 (1901), 435.

UNIVERSITY MATHEMATICAL LABORATORY
CAMBRIDGE