NBS TECHNICAL NOTE

410

Improved Fortran Program for Single Particle Energy Levels and Wave Functions in Nuclear Structure Calculations

RANDALL S. CASWELL

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards

UNITED STATES DEPARTMENT OF COMMERCE • John T. Connor, Secretary
NATIONAL BUREAU OF STANDARDS • A. V. Astin, Director

TECHNICAL NOTE 410

ISSUED SEPTEMBER 30, 1966

Improved Fortran Program for Single Particle Energy Levels and Wave Functions in Nuclear Structure Calculations

Randall S. Caswell

Institute for Basic Standards
National Bureau of Standards
Washington, D.C., 20234

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C., 20402
Price 25 cents.

Contents

Improved Fortran Program for Single Particle Energy Levels and Wave Functions in Nuclear Structure Calculations

Randall S. Caswell

An improved program has been developed for numerical calculations of single particle energy levels and wave functions for Woods-Saxon or other real potential wells. The program is designed to be used as a subroutine of a larger program for nuclear structure calculation but may be used separately if desired. Improvements over the previous program include: (1) for bound states the wave function is calculated by integrating outward from zero radius and inward from a maximum radius beyond the nuclear potential well with matching at an intermediate radius; (2) a first order correction has been made to the starting conditions at small radius for the integration of the wave function; (3) a large step size is used until the calculation has nearly converged on the eigenvalue; then a small step size is used to provide maximum accuracy; (4) the correction of a small error in the Kutta-Runge integration procedure has been made. As a result of the changes the program is approximately five times faster than the previous version. Checks with harmonic oscillator potentials give an accuracy of .01% in the energy level values. Listings of Fortran II and Fortran IV versions are given.

Key Words: Eigenvalues, eigenfunctions, energy levels, wave functions, single particle shell model.

1. Introduction

Nuclear shell model calculations usually have employed harmonic oscillator wave functions for the evaluation of radial integrals for both bound and continuum states. Deeply bound states can be well represented by harmonic oscillator wave functions. For continuum wave functions it appears desirable to use more realistic wave functions, for example, those of a real Woods-Saxon well. A previous program (Caswell, 1962) was developed to provide these eigenvalues and wave functions. The older program, however, used a method of matching inside and outside wave functions at successively larger radii which is inherently slow. In some large nuclear structure programs

the calculation of the single particle wave functions occupies an undesirably large fraction of the computer time. The program described here speeds up the calculation by a factor of roughly 5, provides increased accuracy, and gives normalized wave functions at up to 201 radial points.

2. Description of the Calculation

The chief calculational steps are: (1) calculation of the Woods-Saxon real potential well including a spin-orbit term; (2) numerical integration of the wave function, $u(r) = r\psi(r)$, from zero to a matching radius, RMATCH and, for bound states, from a maximum radius beyond the nuclear potential, RMAX, backwards in radius to the same matching radius -- for scattering states the maximum radius is used as the matching radius; and (3) calculation of the logarithmic derivatives, f_ℓ(inside) and f_ℓ(outside). For scattering states f_ℓ(outside) is determined from the criterion of a 90° phase shift. It is also possible to read in from the input data values for f_ℓ(outside) which are taken as the value at the maximum radius, RMAX. The approximate location of the eigenvalues may be determined by a survey versus energy of f_ℓ(inside), f_ℓ(outside), and the difference between them, called "signature of the well". An automatic search procedure finds the energy for which f_ℓ(inside) = f_ℓ(outside), and the wave function vs. radius is calculated and stored in memory.

The potential is calculated by subroutine VR3 as a function of radius at up to 399 points, which is the number required for the numerical integration routine for up to 201 radial points for the wave functions. Spacing between points is RMAX/NINTVL where RMAX is the maximum radius and NINTVL is the number of intervals (one less than the number of points). The potential used is:

$$V(r) = V_c\rho(r) - \alpha V_c \left(\frac{\hbar}{2Mc}\right)^2 (\vec{\ell}\cdot\vec{\sigma})\frac{1}{r}\frac{d\rho}{dr} \qquad (1)$$

where $\rho(r) = \dfrac{1}{1 + \exp\left(\dfrac{r-R_o}{a}\right)}$, the "Saxon" potential; α is the strength

of the spin-orbit interaction compared to the "Thomas" interaction for a nucleon; M is the nucleon mass; V_c is the central real potential; $\hbar/Mc$ is the Compton wave length for a nucleon; $\vec{\ell}$ is the orbital angular momentum of the neutron; $\vec{\sigma}$ is the Pauli spin operator of the incident neutron; R_o is the nuclear radius; and a is the "diffuseness" parameter of the potential. If a negative value is given for α, it is taken as V_{so}, the depth of the spin-orbit well in MeV where $\alpha V_c\left(\frac{\hbar}{2Mc}\right)^2$ is replaced by $V_{so}\left(\frac{\hbar}{m_\pi c}\right)^2$, m_π being the mass of the π-meson. If $j = \ell + \frac{1}{2}$, then $\vec{\sigma}\cdot\vec{\ell} = \ell$, and the values of the variable VPLUS (nuclear potential) for different radii are calculated. If $j = \ell - \frac{1}{2}$, then $\vec{\sigma}\cdot\vec{\ell} = -(\ell+1)$, and potential is calculated for the spin-orbit anti-parallel case. Other potential shapes can be used by changes in this subroutine. For example, the program has been tested with a harmonic oscillator potential.

The calculation of f_ℓ(inside) and f_ℓ(outside), and the search procedure for matching the f_ℓ values is controlled by EIGENV, the main subroutine of the program. First, the radial part of the non-relativistic Schrödinger wave equation is integrated out in radius for the appropriate ℓ:

$$\frac{d^2u_\ell(r)}{dr^2} + \frac{2m}{\hbar^2}\left[E - \frac{\ell(\ell+1)\hbar^2}{2mr^2} - V(r)\right]u_\ell(r) = 0. \qquad (2)$$

The values of the function and the first derivative at the first radial point (radius = H) are determined from the relation $u_\ell(r) = C_\ell r^{\ell+1}$ as $r \to 0$, where C_ℓ is a constant. We make a first order correction to the starting value of the wave function, u, and the derivative $\frac{du}{dr}$ at the first radial point as follows: Equation (2) may be rewritten (omitting subscript ℓ's):

$$\frac{d^2u}{dr^2} = \frac{\ell(\ell+1)u}{r^2} - \frac{2m}{\hbar^2}(E-V)u. \qquad (3)$$

As $r \to 0$, the last term on the right is negligible compared to the first and our solution is just $u = r^{\ell+1}$ as indicated above, taking C_ℓ equal to one. Taking into account the second term and using our approximate solution for u, we may find the derivative at the first point r (= H) by integration of (3).

$$\frac{du}{dr} = (\ell+1)r^{\ell} - \frac{2m}{\hbar^2}\frac{(E-V)r^{\ell+2}}{(\ell+2)} \quad . \tag{4}$$

Integrating again,

$$u = r^{\ell+1} - \frac{2m}{\hbar^2}\frac{(E-V)}{(\ell+2)(\ell+3)}r^{\ell+3} \quad . \tag{5}$$

Equations (4) and (5) are used as the initial values of wave function and derivative to be used by the wave function integration subroutine. V is taken as the value of the potential at the point r = H .

A similar problem arises in the backward integration from a radius RMAX for bound states. At this distance the centrifugal potential term is small and the differential equation may be written:

$$\frac{d^2u}{dr^2} = \frac{2m}{\hbar^2}[V(r)-E]u = \gamma^2 u \tag{6}$$

where the expression in the square bracket is positive for a bound state. If we consider V(r) as constant, then the descending exponential solution to this equation is $u = \exp(-\gamma r)$, $du/dr = -\gamma \exp(-\gamma r)$, and $(du/dr)/r = -\gamma$. Since normalization is initially arbitrary, we set u = 1 and $(du/dr) = -\gamma$ at RMAX for starting the backward integration.

The second order Runge-Kutta method is used for the numerical integration (see Zurmühl, 1961). An error in the previous code (Caswell, 1962) which led to errors of about 1 percent in the eigenvalues and eigenfunctions by causing a slightly wrong potential to be

used at certain steps has been corrected. The integration is controlled by subroutine INTEG3 and carried out by subroutines KRR1 and FR.

Modes of the calculation. The calculation may be run in three alternative modes. When MODE = 1, if the energy is negative (bound state), the wave function is integrated outward from zero to RMATCH and inward from RMAX to RMATCH, using the starting conditions given above. For convergence on an eigenvalue it is required that the "inside" and "outside" f_ℓ's match at RMATCH. If the energy is positive (continuum state), the criterion of a 90° phase shift is applied to the wave function at the radius RMAX where the inside and outside wave functions must join smoothly (as indicated by having the same value of f_ℓ). For the 90° phase shift continuum case, the "outside" or asymptotic solutions are spherical Neumann functions, $G_\ell = -xn_\ell(x)$, since Coulomb effects are not considered. The argument $x = kr$. The evaluation of the logarithmic derivatives, f_ℓ, is given in Appendix 1.

When MODE = 2, values of f_ℓ(outside) at a radius beyond the nuclear potential radius, RMAX, may be given the EIGENV program by the calling program. The variable FLOUT is used to read in the value of f_ℓ(outside). In the case of a bound state, after integration inward from RMAX starting at the value FLOUT, the variable FLOUT is actually matched to the f_ℓ(inside).

When MODE = 3, the program just calculates and normalizes the wave function at the given energy EN. No search is carried out. EN must have been determined on a previous run. This mode is provided to minimize running time by avoiding unnecessary searches.

3. Search Procedure

To begin the search procedure, one needs a trial energy. In principle this may be selected by guess. The guess will usually converge to some eigenvalue if enough search steps are allowed. A more systematic way to find approximate energies for starting the automatic search procedure is to use the method called the "signature of the well." This is carried out by setting the variable NOMAX = 1, and making a run with a large number of trial energies, for example, at 1 MeV steps

throughout the energy range of interest. When NOMAX = 1, the radius of the well at the energy, EN, is used as the matching radius for bound states, and RMAX is used for scattering states. Values are calculated for f_ℓ(inside), f_ℓ(outside), and for the difference between the two (see, for example, Figure 1). The integration in radius is carried out only once for each value of EN. The energy values at which the difference, DIF, equals zero are close to the energy of the final converged eigenvalue which is to be found, and they serve as suitable starting values for the automatic search procedure.

Automatic search procedure. First an integration of the wave function is carried out at the trial eigenvalue energy (the value of EN when EIGENV is called). A second integration is carried out at a nearby energy (one percent difference in energy). Using the differences (between the inside and outside f_ℓ's found at the matching radius) found in these two trial calculations, a linear interpolation or extrapolation is made to an energy for which the difference is predicted to be zero (corresponding to smooth joining of the inside and outside wave functions). Integration in radius is carried out at this predicted eigenvalue energy, and the difference is again found. We now know three energies and the three corresponding differences. Using this information, the program predicts by quadratic interpolation a fourth energy at which the difference should be closer to zero. The calculation continues, energy prediction being made by quadratic interpolation on the last three values, until either the difference between f_ℓ's is closer to zero than the difference specified by the value of EPS given to the subroutine, or until the maximum number of tries, NOMAX, has been used up. The wave functions during the convergence procedure are shown in Figure 2 for a typical case. If the maximum number of tries has been used up without converging on an eigenvalue, the value of EN and values of the wave function at radial points, WAVEFN(I), are set equal to zero to insure that the calling program does not use incorrect information. When the converged eigenvalue has been found, the part of the wave function between RMATCH and RMAX is renormalized so that it has the proper absolute magnitude to join smoothly to the inner wave

function at RMATCH. A radial integration of the $\int u^2 dr$ is then carried out by subroutine SIMPSN, and the complete wave function is normalized so that $\int_0^{RMAX} u^2 dr = 1$.

To save computer time an integration step size five times the final step size is used for the initial steps of the search (when DIF>0.1). Integration step size is controlled by the variable KSTEP which is set equal to 5 in the early, rough part of the search and set equal to 1 for the final determination of the eigenvalue and wave function.

4. Sample Results

Sample results for eigenvalues and wave functions are shown in Figure 3. Eigenvalues are shown for the $1s_{\frac{1}{2}}$, $2s_{\frac{1}{2}}$, $3s_{\frac{1}{2}}$, $1p_{\frac{3}{2}}$, $1p_{\frac{1}{2}}$, $1d_{\frac{5}{2}}$ state. Wave functions are shown for the $15_{\frac{1}{2}}$, $25_{\frac{1}{2}}$, and $1d_{\frac{5}{2}}$ states.

5. Accuracy and Checks of the Code

To test the accuracy of the program, subroutine VR3 was rewritten to use an harmonic oscillator potential:

$$V(r) = -100 \left[1 - (r/5)^2 \right] \text{ MeV}$$

where r is in fermis. For this particular potential $\hbar\omega$ = 18.810 MeV for AMASS = 15.0. Based on this value we may compare eigenvalues given by the code for various ℓ values with the usual relation $E = (n+\frac{1}{2})\hbar\omega$, all energies being in MeV.

Harmonic Oscillator Levels	s states	p states	d states
-15.355		-15.352(2p)	
-34.165	-34.162(2s)		-34.163(1d)
-52.975		-52.974(1p)	
-71.785	-71.784(1s)		

It can be seen that the code predicts the level separations to an accuracy of about 0.01% which is about the accuracy of various constants in the code.

6. References

Abramowitz, M. and Stegun, I.A., (1964). Handbook of Mathematical Functions, National Bureau of Standards Applied Mathematics Series No. 55.

Blatt, J. M. and Weisskopf, V. F., (1952). Theoretical Nuclear Physics, 334, Wiley (New York).

Caswell, R. S. (1962). A Fortran code for calculation of eigenvalues and eigenfunctions in real potential wells, NBS Technical Note 159. The present report replaces this reference, which should no longer be used.

Schiff, L. I. (1955). Quantum Mechanics, 77, McGraw-Hill (New York).

Zurmühl, R. (1961). Praktische Mathematic für Ingenieure and Physiker, 412, Springer-Verlag (Berlin, Gottingen, Heidelberg).

The author would like to thank Michael Danos for suggestions and his continued interest in this problem.

Appendix 1. Evaluation of f_ℓ(outside) for Scattering States

For continuum states (E>0) we may write for the logarithmic derivative:

$$f_\ell(\text{outside}) = R\,\frac{(du_\ell/dr)_R}{(u_\ell)_R} = R\,\frac{\left[(d/dr)(rn_\ell)\right]_R}{[rn_\ell]_R} \quad , \qquad (1)$$

where n_ℓ is the spherical Neumann function, which corresponds to a 90^o phase shift, as defined in Schiff (1955).

But

$$\frac{d}{dx}(xn_\ell) = \frac{d}{dx}\left(x^{\ell+1}x^{-\ell}n_\ell(x)\right) = (\ell+1)\left[n_\ell(x)\right] - xn_{\ell+1}(x). \qquad (2)$$

If $X = kR = \left(\frac{2m}{\hbar^2}E\right)^{\frac{1}{2}}R$, we have for f_ℓ(outside):

$$f_\ell = \ell+1 - \frac{Xn_{\ell+1}(X)}{n_\ell(X)}\left(= X\,\frac{G_\ell'}{G_\ell}\right), \qquad (3)$$

where the prime refers to differentiation with respect to X. The quantities F_ℓ and G_ℓ are the "regular" and "irregular" wave functions which for neutrons (no Coulomb interaction) are $F_\ell = X\,j_\ell(X)$ and $G_\ell = -X\,n_\ell(X)$.

For values of X>1, "forward" recursion formulae are used (see, for example, Abramowitz and Stegun, 1964):

$$n_o(X) = -\frac{\cos X}{X} \quad ,$$

$$n_1(X) = -\frac{\cos X}{X^2} - \frac{\sin X}{X} \quad , \text{ and}$$

$$n_{\ell+1}(X) = (2\ell+1)\,\frac{n_\ell(X)}{X} - n_{\ell-1}(X).$$

For $\ell \leq 9$ the values accurately reproduce tabular values from the same reference, there appearing to be no need for use of "backward" recursion formulas (from large to small ℓ).

For $X < 1$ a series expansion is used (see Abramowitz and Stegun, 1964):

$$n_\ell(X) = -\frac{1\cdot 3\cdot 5\cdot \ldots\ (2\ell-1)}{X^{\ell+1}} \left\{1-\frac{\frac{1}{2}X^2}{1!(1-2\ell)} + \frac{(\frac{1}{2}X^2)^2}{2!(1-2\ell)(3-2\ell)} - \cdots\right\}.$$

Appendix 2. Instructions for use of EIGENV subroutine and meaning of input variables.

To call the EIGENV subroutine values must be assigned to the following list of variables (with meanings given) and then the statement CALL EIGENV calls the subroutine. The subroutine returns as the answer a value of EN which is the energy eigenvalue, and of a normalized WAVEFN(I) at up to 201 equally spaced points. The wave function is $r\psi$.

VARIABLE	TYPICAL VALUE	MEANING
RADIUS	3.15	radius of the Saxon potential in fermis, typically 1.25 $A^{1/3}$.
A	.65	diffuseness parameter of the Saxon potential in fermis.
VC	-50.0	central real potential in MeV (is negative for attractive potentials).
ALPHA	20.0	magnitude of the spin-orbit potential expressed in number of times larger than for the Thomas term of a nucleon. If a negative number is given, it will be taken as V_{so} the depth of the spin-orbit term in MeV.
AMASS	15.0	atomic mass in a.m.u. of the nucleus excluding the single neutron whose states are being considered.
RMAX	8.0	maximum radius to which wave function will be calculated in fermis (typically about 5 fm greater than the average radius of the well).
EPS	.0001	the difference between the value of f_ℓ (inside) and f_ℓ(outside) must be less than EPS for the eigenvalue to be considered determined.
MODE	1	normal search for bound or scattering states. If a bound state, f_ℓ(outside) is determined by integration inward from RMAX to the matching radius RMATCH (which is automatically determined). For a scattering state the matching is at RMAX and f_ℓ(outside) is determined from the criterion for a 90° phase shift.
MODE	2	f_ℓ(outside) is read in from the input data. It is considered as the value of f_ℓ(outside) at radius RMAX.

VARIABLE	TYPICAL VALUE	MEANING
MODE	3	just calculate the wave function for the given energy, EN. Do not search for a better eigenvalue.
NOMAX	20	maximum number of automatic search steps seeking the eigenvalue. If NOMAX=1, one point is determined for the "signature of the well."
NINTVL	100	number of intervals in radius over which the wave function is to be determined. In the example 101 radial points will be calculated, equally spaced between the first point at R=0 and the last point at R=RMAX. NINTVL must be a multiple of 10.
L	1	orbital angular momentum of the single particle state sought.
JDBLD	3	twice the j-value of the state sought. In the example $j=\frac{3}{2}$ so "j-doubled" is 3.
JPRINT	1	no printing of intermediate results.
	2	print intermediate results for code checking.
EN	-11.0	value given to subroutine is the initial energy. Value returned by subroutine is the eigenvalue (if successful). Value is in MeV.
FLOUT	4.	read-in value of f_ℓ(outside) for the radius RMAX.

Appendix 3. Fortran II listing.

The subroutine and their functions are:

MAINEV	A main program for testing the EIGENV program and a sample for how to call the program.
EIGENV	Controlling subroutine which performs matching and eigenvalue prediction.
VR3	Calculates V(r).
INTEG3	Controls integration in radius.
KRR1	Kutta-Runge integration subroutine.
FR	Calculates f(r) for differential equation $d^2u/dr^2 = f(r)$.
SIMPSON	Simpson rule integration subroutine for normalization of the wave function.

The DIMENSION and COMMON statements appearing in the subroutines must also appear in the calling program.

```
*     LABEL
CMAINEV
*     LIST
C     MAIN PROGRAM TO TEST EIGENV SINGLE PARTICLE EIGENVALUE AND
C     WAVE FUNCTION PROGRAM
      DIMENSION RPRINT(5),WAVE(5),ENRGY(100),FL(100)
      DIMENSION WAVEFN(201),VPLUS(399),XINTGD(201)
      COMMON WAVEFN,RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,NOMAX,NINTVL,
     1L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR,VPLUS,V,NMAX,
     2F,RMATCH,I,XPLUS,XPLUSP,FRR,INOUT,KSTEP,REFL,XINTGD,YINTG,XINT1,H,
     3XINT2,X
    5 WRITE OUTPUT TAPE 6,7
    7 FORMAT(1H1)
      READ INPUT TAPE 5,10
   10 FORMAT(72H
     1                                )
      WRITE OUTPUT TAPE 6,10
      WRITE OUTPUT TAPE 6,4501
 4501 FORMAT(70H    RADIUS          A         VC  ALPHA(VSO)    AMASS
     1RMAX        EPS )
      READ INPUT TAPE 5,1,RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS
      WRITE OUTPUT TAPE 6,1,RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS
    1 FORMAT(7F10.5)
      WRITE OUTPUT TAPE 6,4503
 4503 FORMAT(70H  ENERGIES       MODE      NOMAX     NINTVL          L      J
     1DBLD     JPRINT )
      READ INPUT TAPE 5,20,KZ,MODE,NOMAX,NINTVL,L,JDBLD,JPRINT
      WRITEOUTPUTTAPE 6,20,KZ,MODE,NOMAX,NINTVL,L,JDBLD,JPRINT
   20 FORMAT(7I10)
      NPOINT=NINTVL+1
      GO TO (4081,4082,4081),MODE
 4081 WRITE OUTPUT TAPE 6,4504
 4504 FORMAT(15H TRIAL ENERGIES )
      READ INPUT TAPE 5,1,(ENRGY(KY),KY=1,KZ)
      WRITEOUTPUT TAPE6,1,(ENRGY(KY),KY=1,KZ)
      GO TO 4085
 4082 WRITE OUTPUT TAPE 6,4083
 4083 FORMAT(25H  ENERGY        FLOUT,RMAX )
      READ INPUT TAPE 5,    1,(ENRGY(KY),FL(KY),KY=1,KZ)
      WRITE OUTPUT TAPE 6,    1,(ENRGY(KY),FL(KY),KY=1,KZ)
 4085 DO 100 KY=1,KZ
      EN=ENRGY(KY)
      FLOUT=FL(KY)
      WRITE OUTPUT TAPE 6,6180
 6180 FORMAT( 68H          ENERGY     DIFFERENCE      FL INSIDE       FL OUT
     1SIDE    RMATCH )
      CALL EIGENV
      IF(ABSF(EN)-1.E-10) 100,4088,4088
 4088 WRITE OUTPUTTAPE 6,4089,EN
 4089 FORMAT (34H  WAVE FUNCTION VS RADIUS, ENERGY=,F10.5,4H MEV)
      R=-HI
```

```
      INMAX=5
      DO 200 N=1,NPOINT,5
      DO 150 IN=1,INMAX
      NN=N+IN-1
      WAVE(IN)=WAVEFN(NN)
      R=R+HI
      RPRINT(IN)=R
      IF(NN-NPOINT)150,150,155
  150 CONTINUE
      GO TO 159
  155 INMAX=IN-1
  159 WRITE OUTPUT TAPE 6,160,(RPRINT(IN),WAVE(IN),IN=1,INMAX)
  160 FORMAT(5(F9.3,E15.6))
  200 CONTINUE
      GO TO (100,170),JPRINT
  170 WRITE OUTPUT TAPE 6,180,YINTG
  180 FORMAT(7H YINTG=,F10.5)
  100 CONTINUE
      READ INPUT TAPE 5,20,NDUMMY
      IF (NDUMMY)1000,1000,5
 1000 CALL EXIT
      END
*     LABEL
CEIGENV
*     LIST
      SUBROUTINE EIGENV
      DIMENSION Y(11)
      DIMENSION WAVEFN(201),VPLUS(399),XINTGD(201)
      COMMON WAVEFN,RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,NOMAX,NINTVL,
     1L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR,VPLUS,V,NMAX,
     2F,RMATCH,I,XPLUS,XPLUSP,FRR,INOUT,KSTEP,REFL,XINTGD,YINTG,XINT1,H,
     3XINT2,X
      K=L+1
      HI=RMAX/FLOATF(NINTVL)
      NPOINT=NINTVL+1
      IF(ALPHA) 1,2,2
    1 ALPHA=ALPHA*181.1/VC
    2 CALL VR3
      KSTEP=1
      H=HI
      GO TO (15,15,6130),MODE
   15 IF(EN) 180,180,170
  170 RMATCH=RMAX
      GO TO 190
  180 RMATCH=(5.*FLOATF(NINTVL/15+1))*HI
  190 MPRINT=2
      IF(NOMAX-1) 25,30,25
   25 KSTEP=5
      H=5.*HI
   30 CALL INTEG3
      NO=1
  203 ENERGY=EN
      IF(EN) 213,213,220
  213 FLOMT=R*XPLUSP/XPLUS
      DIF=REFL-FLOMT
      GO TO 6001
  220 PMOM=SQRTF(.04826*F*EN)
      DIF=REFL-FLOUT
      FLOMT=FLOUT
      GO TO (2201,6001,6130),MODE
```

```
2201 XX=R*PMOM
     KPLUS1=K+1
     IF(XX-1.) 300,300,400
 300 NPASS=1
 301 SUM=1.
     TERM=1.
     DO 310 N=1,20
     TERM=-TERM*0.5*XX**2/(FLOATF(N)*FLOATF(2*N-1-2*L))
     SUM=SUM+TERM
     IF(ABSF(TERM)-1.E-8)320,310,310
 310 CONTINUE
 320 IF(K-1) 325,335,325
 325 DO 330 IK=2,K
     AK=IK
 330 SUM=SUM*(2.*AK-3.)
 335 GO TO (340,350),NPASS
 340 YL=-SUM/(XX**K)
     K=K+1
     L=L+1
     NPASS=2
     GO TO 301
 350 YLPLUS=-SUM/(XX**K)
     K=K-1
     L=L-1
     GO TO 500
 400 Y(1)=-COSF(XX)/XX
     Y(2)=-COSF(XX)/XX**2-SINF(XX)/XX
     DO 410 IK=3,KPLUS1
     AK=IK
 410 Y(IK)=(2.*AK-3.)*Y(IK-1)/XX-Y(IK-2)
     YL=Y(K)
     YLPLUS=Y(KPLUS1)
 500 FLOMT=FLOATF(L)+1.-XX*YLPLUS/YL
     DIF=REFL-FLOMT
6001 IF(NO-NOMAX) 6004,6100,6100
6004 IF(NO-2) 6005,6080,6090
6005 EN=0.99*EN
     GO TO 6100
6080 EN    =(ENERG1*DIF-ENERGY*DIFM1)/(DIF-DIFM1)
     GO TO 6100
6090 EN    =(((ENERG2*DIFM1-ENERG1*DIFM2)*DIF/(DIFM1-DIFM2))-((ENERG1
    1*DIF-ENERGY*DIFM1)*DIFM2/(DIF-DIFM1)))/(DIF-DIFM2)
6100 IF(ABSF((EN    /ENERGY)-1.)-.2) 6110,6110,6101
6101 EN=ENERGY+ABSF(ENERGY)*SIGNF(0.1,(EN-ENERGY))
6110 WRITE OUTPUT TAPE 6,6160,ENERGY,DIF,REFL,FLOMT,R
6160 FORMAT(F15.5,3E15.7,F8.3)
     IF (NO-NOMAX) 6114,6140,6140
6114 IF(ABSF(DIF    )-EPS) 6115,6115,6120
6115 IF(KSTEP-1) 105,6116,105
6116 EN=ENERGY
     GO TO 6130
6120 IF(KSTEP-1) 100,110,100
 100 IF(ABSF(DIF    )-.1) 105,110,110
 105 KSTEP=1
     H=HI
 110 CALL INTEG3
     DIFM2=DIFM1
     DIFM1=DIF
     ENERG2=ENERG1
     ENERG1=ENERGY
```

```
      NO=NO+1
      GO TO 203
 6130 MPRINT=1
      CALL INTEG3
      GO TO 6150
 6140 EN=0.
      DO 6145 N=1,NPOINT
 6145 WAVEFN(N)=0.
 6150 CONTINUE
      RETURN
       END
*     LABEL
CVR
*     LIST
      SUBROUTINE VR3
      DIMENSION WAVEFN(201),VPLUS(399),XINTGD(201)
      COMMON WAVEFN,RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,NOMAX,NINTVL,
     1L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR,VPLUS,V,NMAX,
     2F,RMATCH,I,XPLUS,XPLUSP,FRR,INOUT,KSTEP,REFL,XINTGD,YINTG,XINT1,H,
     3XINT2,X
      R=HI
      H=HI/2.
      GO TO (2090,2000),JPRINT
 2000 WRITE OUTPUT TAPE 6,2091
 2091 FORMAT(25H                V          R )
 2090 C= .0110270/A
      NMAX=2*NINTVL-1
      DO 2210 I=1,NMAX
      REXP=EXPF((R-RADIUS)/A)
      VCC=VC/(1.0+REXP)
      VS=ALPHA*VC*C*REXP/(((1.0+REXP)**2)*R)
      AK=K
      IF(JDBLD-2*L) 22,22,21
   21 VPLUS(I)=VCC+VS*(AK-1.0)
      GO TO 25
   22 VPLUS (I)=VCC-VS*AK
   25 GO TO (40,28),JPRINT
   28 WRITE OUTPUT TAPE 6,30,VPLUS(I),R
   30 FORMAT( E15.7,F7.3)
   40 R=R+H
 2210 CONTINUE
      RETURN
       END
*     LABEL
CINTEG
*     LIST
      SUBROUTINE INTEG3
      DIMENSION WAVEFN(201),VPLUS(399),XINTGD(201)
      COMMON WAVEFN,RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,NOMAX,NINTVL,
     1L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR,VPLUS,V,NMAX,
     2F,RMATCH,I,XPLUS,XPLUSP,FRR,INOUT,KSTEP,REFL,XINTGD,YINTG,XINT1,H,
     3XINT2,X
      GO TO (1,15),MPRINT
    1 GO TO (5,5,15),MODE
    5 IF(EN) 5215,4221,4221
   15 WAVEFN(1)=0.
      AK=K
      F=AMASS/(AMASS+1.00898)
      R=H
      RR=R
```

```
      V=VPLUS(2*KSTEP-1)
      XPLUS=R**K
      X=XPLUS
      ELL=0.
      CALL FR
      ELL=K-1
      XPLUS=XPLUS+FRR*R**2/((ELL+2.)*(ELL+3.))
      WAVEFN(2)=XPLUS
      XPLUSP   =AK*(R **(K-1)) +FRR*R/(ELL+2.)
      GO TO (4090,4090,4080),MODE
4080  NOMAX=2
4090  INOUT=1
      DO 4210 I=KSTEP,1000,KSTEP
      I=I
      CALL KRR1
      R=R+H
4127  IF(NOMAX-1) 4130,4128,4130
4128  RMATCH=RMAX
      IF (R-0.2*RADIUS) 4130,4129,4129
4129  IF(EN-V)     10,10,4132
  10  RMATCH=R
      GO TO 4220
4130  WAVEFN(I+2)=XPLUS
4132  IF(R-RMATCH+.001) 4210,4220,4220
4210  CONTINUE
4220  REFL=R*XPLUSP/XPLUS
      WMATCH=XPLUS
      IF(EN) 4211,4216,4216
4211  XPLUS=1.
      WAVEFN(NPOINT)=1.
      GO TO (4212,4213,4213),MODE
4212  XPLUSP=-SQRTF(VPLUS(NMAX)-EN)
      GO TO 4214
4213  XPLUSP=FLOUT/RMAX
4214  H=-H
      R=RMAX
      INOUT=-1
      DO 5210 J=1,1000,KSTEP
      I=NINTVL-J
      CALL KRR1
      R=R+H
      WAVEFN(I+1)=XPLUS
      IF(R-RMATCH-.001) 4215,4215,5210
5210  CONTINUE
4215  H=-H
      GO TO (5215,4230),MPRINT
5215  I=I+1
      DO 5216 J=I,NPOINT
5216  WAVEFN(J)=WAVEFN(J)*WMATCH/XPLUS
4216  GO TO (4221,4230),MPRINT
4221  YINTG=0.
      XINT1=0.
      XINT2=RMAX
      DO 4225 N=1,NPOINT
4225  XINTGD(N)=WAVEFN(N)**2
      CALL SIMPSN
      DO 4228  N=1,NPOINT
4228  WAVEFN(N)=WAVEFN(N)/SQRTF(YINTG)
4230  RETURN
      END
```

```
*      LABEL
CKRR
*      LIST
       SUBROUTINE KRR1
       DIMENSION WAVEFN(201),VPLUS(399),XINTGD(201)
       COMMON WAVEFN,RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,NOMAX,NINTVL,
      1L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR,VPLUS,V,NMAX,
      2F,RMATCH,I,XPLUS,XPLUSP,FRR,INOUT,KSTEP,REFL,XINTGD,YINTG,XINT1,H,
      3XINT2,X
       GO TO (56,40),JPRINT
   40  WRITE OUTPUT TAPE 6,45,H,R,XPLUS,XPLUSP,AI,EN,V,RMATCH
   45  FORMAT(3H H=,F8.5,3H R=,F8.5,7H XPLUS=,E12.5,8H XPLUSP=,E12.5,4H A
      1I=,F8.5,4H EN=,F10.4,3H V=,F9.4,8H RMATCH=,F6.3)
   56  RR=R
       INDEX=2*I-INOUT
       V=VPLUS(INDEX)
       X=XPLUS
       CALL FR
       AI=FRR*H**2/2.0
       RR=R+H/2.0
       X=XPLUS+ XPLUSP*H/2.0+AI/4.0
        INDEX=2*I+INOUT*(KSTEP-1)
       V=VPLUS(INDEX)
       CALL FR
       AII =FRR*(H**2)/2.0
       RR=R+H
       X=XPLUS+XPLUSP*H+AII
       INDEX=2*I+INOUT*(2*KSTEP-1)
       V=VPLUS(INDEX)
       CALL FR
       AIII=FRR*H**2/2.0
       XPLUS=XPLUS+H*XPLUSP+(AI+2.0*AII)/3.0
       XPLUSP =XPLUSP +(AI +4.0*AII+AIII)/(3.0*H)
       GO TO (4125,190),JPRINT
  190  IF(ABSF(RR-RMATCH)-.001) 195,4125,4125
  195  WRITE OUTPUT TAPE 6,45,H,RR,XPLUS,XPLUSP,AI,EN,V,RMATCH
 4125  CONTINUE
       RETURN
       END
*      LABEL
CFR
*      LIST
       SUBROUTINE FR
       DIMENSION WAVEFN(201),VPLUS(399),XINTGD(201)
       COMMON WAVEFN,RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,NOMAX,NINTVL,
      1L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR,VPLUS,V,NMAX,
      2F,RMATCH,I,XPLUS,XPLUSP,FRR,INOUT,KSTEP,REFL,XINTGD,YINTG,XINT1,H,
      3XINT2,X
       FRR=-.04826*F *(EN     -20.721*ELL*(ELL+1.0)/(F *RR**2)-V)*X
       RETURN
       END
*      LABEL
CSIMPSN
*      LIST
       SUBROUTINE SIMPSN
       DIMENSION WAVEFN(201),VPLUS(399),XINTGD(201)
       COMMON WAVEFN,RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,NOMAX,NINTVL,
      1L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR,VPLUS,V,NMAX,
      2F,RMATCH,I,XPLUS,XPLUSP,FRR,INOUT,KSTEP,REFL,XINTGD,YINTG,XINT1,H,
      3XINT2,X
```

```
      DELINT=(XINT2-XINT1)/FLOATF(NINTVL)
      NPOINT=NINTVL+1
      YINTG3=XINTGD(1)
      NPT=1
      DO 20 NPT=2,NINTVL,2
      YINTG3=YINTG3+XINTGD(NPT)*4.0
   20 CONTINUE
      DO 30 NPT=3,NINTVL,2
      YINTG3=YINTG3+XINTGD(NPT)*2.0
   30 CONTINUE
      YINTG3=YINTG3+XINTGD(NPOINT)
      YINTG=YINTG+YINTG3*DELINT/3.0
      RETURN
      END
```

Appendix 4. Fortran IV listing.

A block of COMMON called MEMEIG is common to MAINEV, EIGENV and to the subroutines called by EIGENV. It must appear in the calling program.

```
$IBFTC MAINEV  LIST,M94,XR7
C      MAIN PROGRAM TO TEST EIGENV SINGLE PARTICLE EIGENVALUE AND WAVE
C      FUNCTION PROGRAM
       DIMENSION RPRINT(5),WAVE(5),ENRGY(100),FL(100)
       COMMON/MEMEIG/WAVEFN(201),RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,
      1NOMAX,NINTVL,L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR,
      2H,VPLUS(399),V,NMAX,F,RMATCH,I,     XPLUS,XPLUSP,FRR,INOUT,KSTEP,
      3REFL,XINTGD(201),YINTG,XINT1,XINT2,X
    5 WRITE(6,7)
    7 FORMAT(1H1)
      READ(5,10)
   10 FORMAT(72H
     1                                )
      WRITE(6,10)
      WRITE(6,4501)
 4501 FORMAT(70H    RADIUS          A          VC  ALPHA(VSO)    AMASS
     1RMAX        EPS )
      READ(5,1) RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS
      WRITE(6,1)RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS
    1 FORMAT(7F10.5)
      WRITE(6,4503)
 4503 FORMAT(70H  ENERGIES       MODE      NOMAX     NINTVL          L      J
     1DBLD     JPRINT )
      READ (5,20) KZ,MODE,NOMAX,NINTVL,L,JDBLD,JPRINT
      WRITE(6,20) KZ,MODE,NOMAX,NINTVL,L,JDBLD,JPRINT
   20 FORMAT(7I10)
      NPOINT=NINTVL+1
      GO TO (4081,4082,4081),MODE
 4081 WRITE(6,4504)
 4504 FORMAT(15H TRIAL ENERGIES )
      READ(5,1)    (ENRGY(KY),KY=1,KZ)
      WRITE(6,1)    (ENRGY(KY),KY=1,KZ)
      GO TO 4085
 4082 WRITE(6,4083)
 4083 FORMAT(25H  ENERGY        FLOUT,RMAX )
      READ(5,1)    (ENRGY(KY),FL(KY),KY=1,KZ)
      WRITE(6,1)     (ENRGY(KY),FL(KY),KY=1,KZ)
 4085 DO 100 KY=1,KZ
      EN=ENRGY(KY)
      FLOUT=FL(KY)
      WRITE(6,6180)
 6180 FORMAT( 68H          ENERGY    DIFFERENCE     FL INSIDE      FL OUT
     1SIDE   RMATCH )
      CALL EIGENV
      IF(ABS (EN)-1.E-10) 100,4088,4088
 4088 WRITE(6,4089) EN
 4089 FORMAT (34H  WAVE FUNCTION VS RADIUS, ENERGY=,F10.5,4H MEV)
      R=-HI
      INMAX=5
      DO 200 N=1,NPOINT,5
      DO 150 IN=1,INMAX
      NN=N+IN-1
```

```
      WAVE(IN)=WAVEFN(NN)
      R=R+HI
      RPRINT(IN)=R
      IF(NN-NPOINT)150,150,155
  150 CONTINUE
      GO TO 159
  155 INMAX=IN-1
  159 WRITE(6,160) (RPRINT(IN),WAVE(IN),IN=1,INMAX)
  160 FORMAT(5(F9.3,E15.6))
  200 CONTINUE
      GO TO (100,170),JPRINT
  170 WRITE(6,180) YINTG
  180 FORMAT(7H YINTG=,F12.5)
  100 CONTINUE
      GO TO 5
      END
$IBFTC EV      LIST,M94,XR7
      SUBROUTINE EIGENV
      DIMENSION Y(11)
      COMMON/MEMEIG/WAVEFN(201),RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,
     1NOMAX,NINTVL,L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR,
     2H,VPLUS(399),V,NMAX,F,RMATCH,I,     XPLUS,XPLUSP,FRR,INOUT,KSTEP,
     3REFL,XINTGD(201),YINTG,XINT1,XINT2,X
      K=L+1
      HI=RMAX/FLOAT(NINTVL)
      NPOINT=NINTVL+1
      IF(ALPHA) 1,2,2
    1 ALPHA=ALPHA*181.1/VC
    2 CALL VR3
      KSTEP=1
      H=HI
      GO TO (15, 15,6130),MODE
   15 IF(EN) 180,180,170
  170 RMATCH=RMAX
      GO TO 190
  180 RMATCH=(5.*FLOAT(NINTVL/15+1))*HI
  190 MPRINT=2
      IF(NOMAX-1) 25,30,25
   25 KSTEP=5
      H=5.*HI
   30 CALL INTEG3
      NO=1
  203 ENERGY=EN
      IF(EN) 213,213,220
  213 FLOMT=R*XPLUSP/XPLUS
      DIF=REFL-FLOMT
      GO TO 6001
  220 PMOM=SQRT(.04826*F*EN)
      DIF=REFL-FLOUT
      FLOMT=FLOUT
      GO TO (2201,6001,6130),MODE
 2201 XX=R*PMOM
      KPLUS1=K+1
      IF(XX-1.) 300,300,400
  300 NPASS=1
  301 SUM=1.
      TERM=1.
      DO 310 N=1,20
      TERM=-TERM*0.5*XX**2/(FLOAT(N)*FLOAT(2*N-1-2*L))
      SUM=SUM+TERM
```

```
      IF(ABS(TERM)-1.E-8) 320,310,310
  310 CONTINUE
  320 IF(K-1) 325,335,325
  325 DO 330 IK=2,K
      AK=IK
  330 SUM=SUM*(2.*AK-3.)
  335 GO TO (340,350),NPASS
  340 YL=-SUM/(XX**K)
      K=K+1
      L=L+1
      NPASS=2
      GO TO 301
  350 YLPLUS=-SUM/(XX**K)
      K=K-1
      L=L-1
      GO TO 500
  400 Y(1)=-COS(XX)/XX
      Y(2)=-COS(XX)/XX**2-SIN(XX)/XX
      DO 410 IK=3,KPLUS1
      AK=IK
  410 Y(IK)=(2.*AK-3.)*Y(IK-1)/XX-Y(IK-2)
      YL=Y(K)
      YLPLUS=Y(KPLUS1)
  500 FLOMT=FLOAT(L)+1.-XX*YLPLUS/YL
      DIF=REFL-FLOMT
 6001 IF(NO-NOMAX) 6004,6100,6100
 6004 IF(NO-2) 6005,6080,6090
 6005 EN=0.99*EN
      GO TO 6100
 6080 EN    =(ENERG1*DIF-ENERGY*DIFM1)/(DIF-DIFM1)
      GO TO 6100
 6090 EN    =(((ENERG2*DIFM1-ENERG1*DIFM2)*DIF/(DIFM1-DIFM2))-((ENERG1
     1*DIF-ENERGY*DIFM1)*DIFM2/(DIF-DIFM1)))/(DIF-DIFM2)
 6100 IF(ABS((EN    /ENERGY)-1.)-.2) 6110,6110,6101
 6101 EN=ENERGY+ABS(ENERGY)*SIGN(0.1,(EN-ENERGY))
 6110 WRITE (6,6160) ENERGY,DIF,REFL,FLOMT,R
 6160 FORMAT(F15.5,3E15.7,F8.3)
      IF (NO-NOMAX) 6114,6140,6140
 6114 IF(ABS(DIF    )-EPS) 6115,6115,6120
 6115 IF(KSTEP-1) 105,6116,105
 6116 EN=ENERGY
      GO TO 6130
 6120 IF(KSTEP-1) 100,110,100
  100 IF(ABS(DIF    )-.1) 105,110,110
  105 KSTEP=1
      H=HI
  110 CALL INTEG3
      DIFM2=DIFM1
      DIFM1=DIF
      ENERG2=ENERG1
      ENERG1=ENERGY
      NO=NO+1
      GO TO 203
 6130 MPRINT=1
      CALL INTEG3
      GO TO 6150
 6140 EN=0.
      DO 6145 N=1,NPOINT
 6145 WAVEFN(N)=0.
 6150 CONTINUE
```

```
      RETURN
       END
$IBFTC VR      LIST,M94,XR7
      SUBROUTINE VR3
      COMMON/MEMEIG/WAVEFN(201),RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,
     1NOMAX,NINTVL,L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR,
     2H,VPLUS(399),V,NMAX,F,RMATCH,I,     XPLUS,XPLUSP,FRR,INOUT,KSTEP,
     3REFL,XINTGD(201),YINTG,XINT1,XINT2,X
      R=HI
      H=HI/2.
      GO TO (2090,2000),JPRINT
 2000 WRITE(6,2091)
 2091 FORMAT(25H                V          R )
 2090 C= .0110270/A
      NMAX=2*NINTVL-1
      DO 2210 I=1,NMAX
      REXP=EXP ((R-RADIUS)/A)
      VCC=VC/(1.0+REXP)
      VS=ALPHA*VC*C*REXP/(((1.0+REXP)**2)*R)
      AK=K
      IF(JDBLD-2*L) 22,22,21
   21 VPLUS(I)=VCC+VS*(AK-1.0)
      GO TO 25
   22 VPLUS (I)=VCC-VS*AK
   25 GO TO (40,28),JPRINT
   28 WRITE(6,30)  VPLUS(I),R
   30 FORMAT( E15.7,F7.3)
   40 R=R+H
 2210 CONTINUE
      RETURN
       END
$IBFTC INTEG   LIST,M94,XR7
      SUBROUTINE INTEG3
      COMMON/MEMEIG/WAVEFN(201),RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,
     1NOMAX,NINTVL,L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR,
     2H,VPLUS(399),V,NMAX,F,RMATCH,I,     XPLUS,XPLUSP,FRR,INOUT,KSTEP,
     3REFL,XINTGD(201),YINTG,XINT1,XINT2,X
      GO TO (1,15),MPRINT
    1 GO TO (5,5,15),MODE
    5 IF(EN) 5215,4221,4221
   15 WAVEFN(1)=0.
      AK=K
      F=AMASS/(AMASS+1.00898)
      R=H
      RR=R
      V=VPLUS(2*KSTEP-1)
      XPLUS=R**K
      X=XPLUS
      ELL=0.
      CALL FR
      ELL=K-1
      XPLUS=XPLUS+FRR*R**2/((ELL+2.)*(ELL+3.))
      WAVEFN(2)=XPLUS
      XPLUSP   =AK*(R **(K-1)) +FRR*R/(ELL+2.)
      GO TO (4090,4090,4080),MODE
 4080 NOMAX=2
 4090 INOUT=1
      DO 4210 I=KSTEP,1000,KSTEP
      I=I
      CALL KRR1
```

```
      R=R+H
 4127 IF(NOMAX-1) 4130,4128,4130
 4128 RMATCH=RMAX
      IF (R-0.2*RADIUS) 4130,4129,4129
 4129 IF(EN-V)      10,10,4132
   10 RMATCH=R
      GO TO 4220
 4130 WAVEFN(I+2)=XPLUS
 4132 IF(R-RMATCH+.001) 4210,4220,4220
 4210 CONTINUE
 4220 REFL=R*XPLUSP/XPLUS
      WMATCH=XPLUS
      IF(EN) 4211,4216,4216
 4211 XPLUS=1.
      WAVEFN(NPOINT)=1.
      GO TO (4212,4213,4213),MODE
 4212 XPLUSP =-SQRT(VPLUS(NMAX)-EN)
      GO TO 4214
 4213 XPLUSP=FLOUT/RMAX
 4214 H=-H
      R=RMAX
      INOUT=-1
      DO 5210 J=1,1000,KSTEP
      I=NINTVL-J
      CALL KRR1
      R=R+H
      WAVEFN(I+1)=XPLUS
      IF(R-RMATCH-.001) 4215,4215,5210
 5210 CONTINUE
 4215 H=-H
      GO TO (5215,4230),MPRINT
 5215 I=I+1
      DO 5216 J=I,NPOINT
 5216 WAVEFN(J)=WAVEFN(J)*WMATCH/XPLUS
 4216 GO TO (4221,4230),MPRINT
 4221 YINTG=0.
      XINT1=0.
      XINT2=RMAX
      DO 4225 N=1,NPOINT
 4225 XINTGD(N)=WAVEFN(N)**2
      CALL SIMPSN
      DO 4228  N=1,NPOINT
 4228 WAVEFN(N)=WAVEFN(N)/SQRT(YINTG)
 4230 RETURN
      END
$IBFTC KRR     LIST
      SUBROUTINE KRR1
      COMMON/MEMEIG/WAVEFN(201),RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,
     1NOMAX,NINTVL,L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR,
     2H,VPLUS(399),V,NMAX,F,RMATCH,I,     XPLUS,XPLUSP,FRR,INOUT,KSTEP,
     3REFL,XINTGD(201),YINTG,XINT1,XINT2,X
      GO TO (56,40),JPRINT
   40 WRITE(6,45)   H,R,XPLUS,XPLUSP,AI,EN,V,RMATCH
   45 FORMAT(3H H=,F8.5,3H R=,F9.5,7H XPLUS=,E12.5,8H XPLUSP=,E12.5,4H A
     1I=,F9.5,4H EN=,F10.4,3H V=,F9.4,8H RMATCH=,F6.3)
   56 RR=R
      INDEX=2*I-INOUT
      V=VPLUS(INDEX)
      X=XPLUS
      CALL FR
```

```
      AI=FRR*H**2/2.0
      RR=R+H/2.0
      X=XPLUS+ XPLUSP*H/2.0+AI/4.0
      INDEX=2*I+INOUT*(KSTEP-1)
      V=VPLUS(INDEX)
      CALL FR
      AII =FRR*(H**2)/2.0
      RR=R+H
      X=XPLUS+XPLUSP*H+AII
      INDEX=2*I+INOUT*(2*KSTEP-1)
      V=VPLUS(INDEX)
      CALL FR
      AIII=FRR*H**2/2.0
      XPLUS=XPLUS+H*XPLUSP+(AI+2.0*AII)/3.0
      XPLUSP =XPLUSP +(AI +4.0*AII+AIII)/(3.0*H)
      GO TO (4125,190),JPRINT
  190 IF(ABS(RR-RMATCH)-.001) 195,4125,4125
  195 WRITE(6,45)  H,RR,XPLUS,XPLUSP,AI,EN,V,RMATCH
 4125 CONTINUE
      RETURN
      END
$IBFTC FR      LIST,M94,XR7
      SUBROUTINE FR
      COMMON/MEMEIG/WAVEFN(201),RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,
     1NOMAX,NINTVL,L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR
     2H,VPLUS(399),V,NMAX,F,RMATCH,I,     XPLUS,XPLUSP,FRR,INOUT,KSTEP,
     3REFL,XINTGD(201),YINTG,XINT1,XINT2,X
      FRR=-.04826*F *(EN    -20.721*ELL*(ELL+1.0)/(F *RR**2)-V)*X
      RETURN
      END
$IBFTC SIMPSN  LIST,M94,XR7
      SUBROUTINE SIMPSN
      COMMON/MEMEIG/WAVEFN(201),RADIUS,A,VC,ALPHA,AMASS,RMAX,EPS,MODE,
     1NOMAX,NINTVL,L,ELL,K,JDBLD,JPRINT,MPRINT,NPOINT,EN,FLOUT,HI,R,RR,
     2H,VPLUS(399),V,NMAX,F,RMATCH,I,     XPLUS,XPLUSP,FRR,INOUT,KSTEP,
     3REFL,XINTGD(201),YINTG,XINT1,XINT2,X
      DELINT=(XINT2-XINT1)/FLOAT (NINTVL)
      YINTG3=XINTGD(1)
      NPT=1
      DO 20 NPT=2,NINTVL,2
      YINTG3=YINTG3+XINTGD(NPT)*4.0
   20 CONTINUE
      DO 30 NPT=3,NINTVL,2
      YINTG3=YINTG3+XINTGD(NPT)*2.0
   30 CONTINUE
      YINTG3=YINTG3+XINTGD(NPOINT)
      YINTG=YINTG+YINTG3*DELINT/3.0
      RETURN
      END
```

Figure Captions

Fig. 1. Results of the "signature of the well" run for the ℓ=0 well shown in Figure 3. Results are given for f_ℓ(inside), f_ℓ(outside), and for the difference (DIF) between them. Note that the energy at which the difference is zero is close to the energy of the converged eigenvalue, indicated by the short arrow.

Fig. 2. Improvement of the wave function match during the automatic search procedure. Note that for proper normalization of the bound state wave function that RMAX must be sufficiently large that there is negligible contribution to $u^2 dr$ beyond RMAX.

Fig. 3. Single particle states and the $1s_{\frac{1}{2}}$, $2s_{\frac{1}{2}}$, and $1d_{\frac{5}{2}}$ wave functions in a Woods-Saxon potential well. The parameters of the well are: V_c = -50 MeV, RADIUS = 3.15 fermis, a = 0.650 fermis, and the spin-orbit term is 20 times the Thomas term for a nucleon (ALPHA = 20). The shape of the well shown is for ℓ=0. During the "signature of the well" preliminary run, the matching radius is given by the heavy black line - the edge of the well for bound states, and the maximum radius (8 fermis in this example) for continuum states.

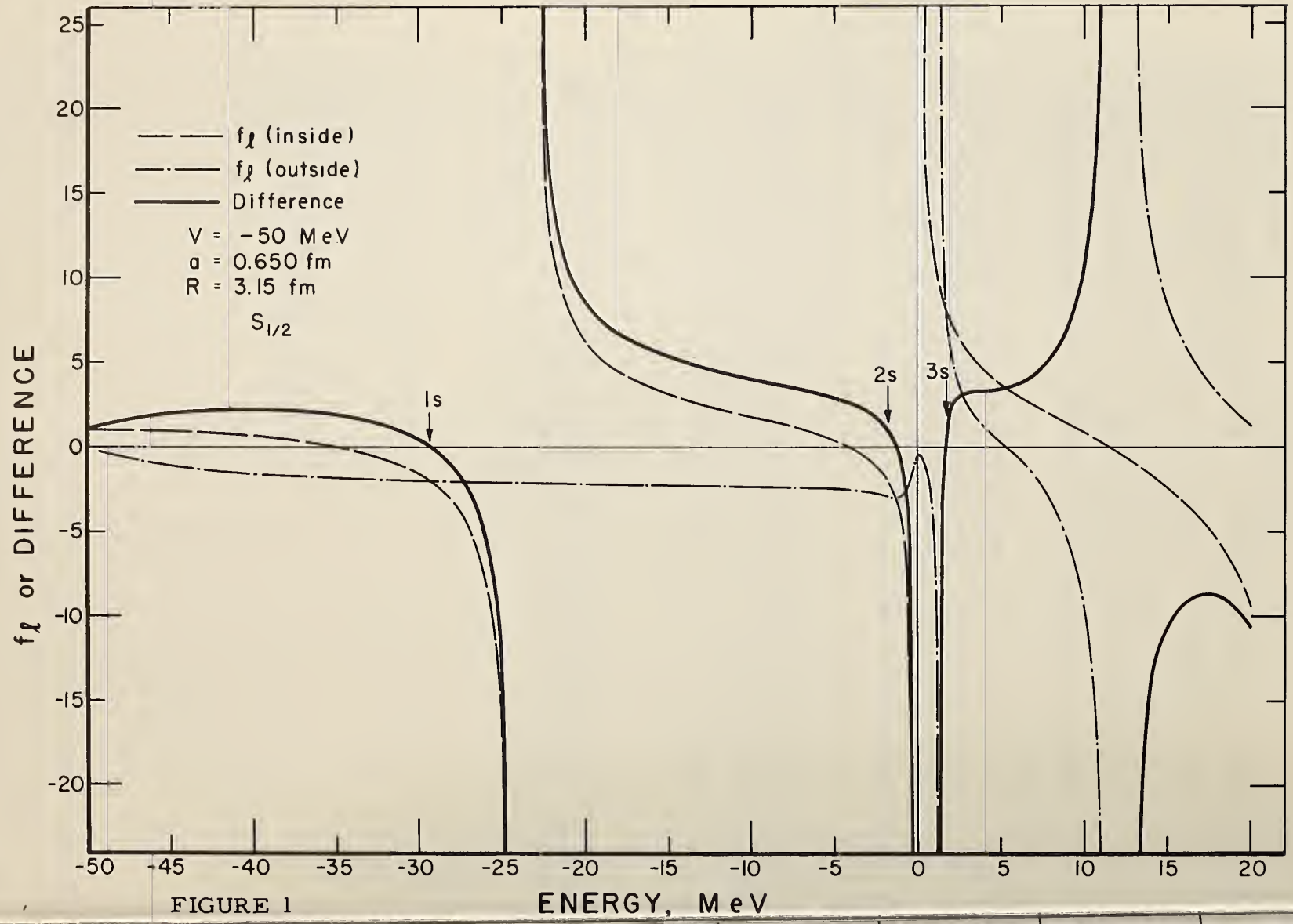
f_ℓ (inside)
f_ℓ (outside)
Difference
V = −50 MeV
a = 0.650 fm
R = 3.15 fm
$S_{1/2}$
1s
2s
3s
f_ℓ or DIFFERENCE
ENERGY, MeV
25
20
15
10
5
0
-5
-10
-15
-20
-50
-45
-40
-35
-30
-25
-20
-15
-10
-5
0
5
10
15
20

FIGURE 1

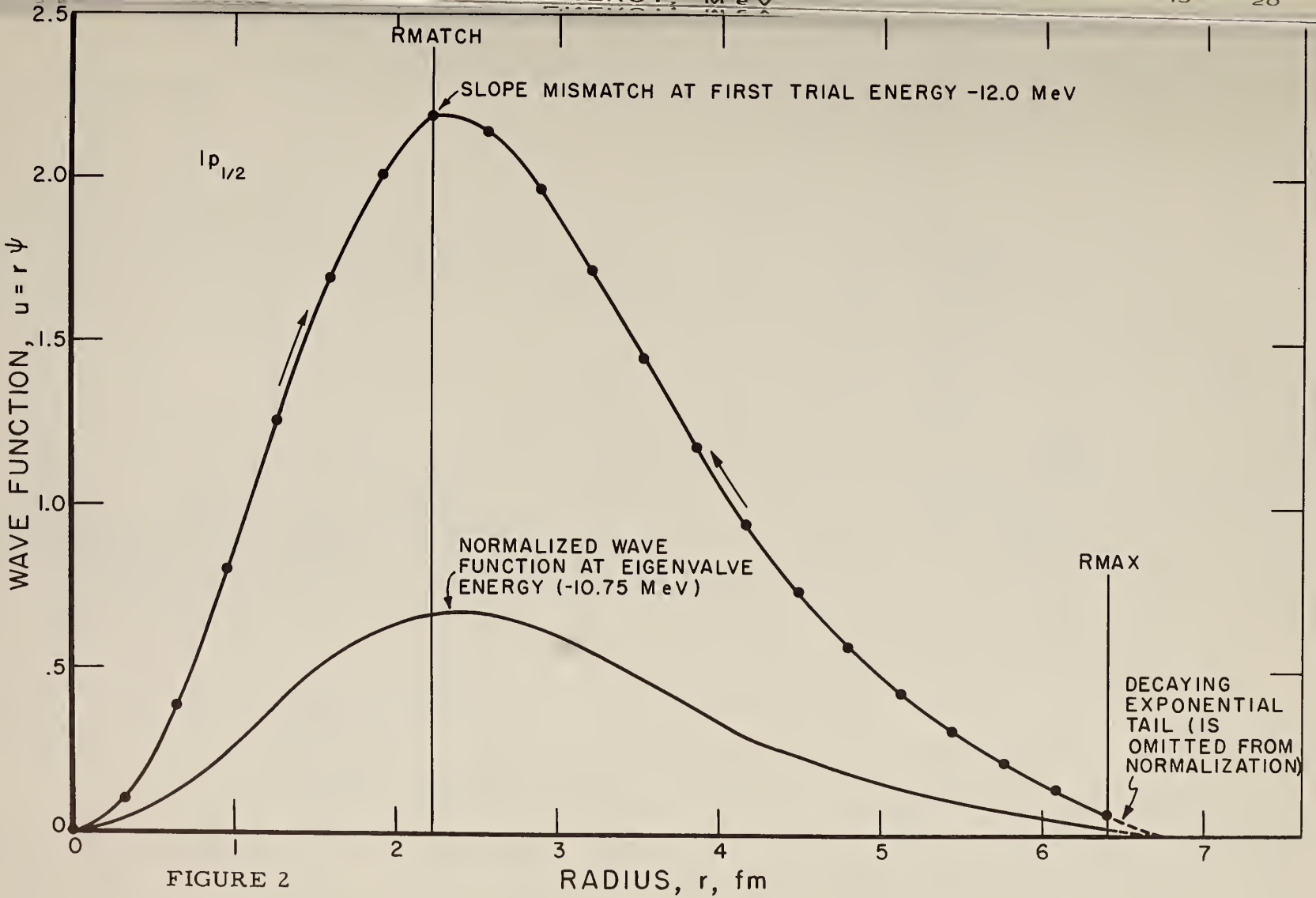
RMATCH
SLOPE MISMATCH AT FIRST TRIAL ENERGY -12.0 MeV
$1p_{1/2}$
NORMALIZED WAVE FUNCTION AT EIGENVALVE ENERGY (-10.75 MeV)
RMAX
DECAYING EXPONENTIAL TAIL (IS OMITTED FROM NORMALIZATION)
WAVE FUNCTION, u = r ψ
RADIUS, r, fm
2.5
2.0
1.5
1.0
.5
0
0
1
2
3
4
5
6
7

FIGURE 2

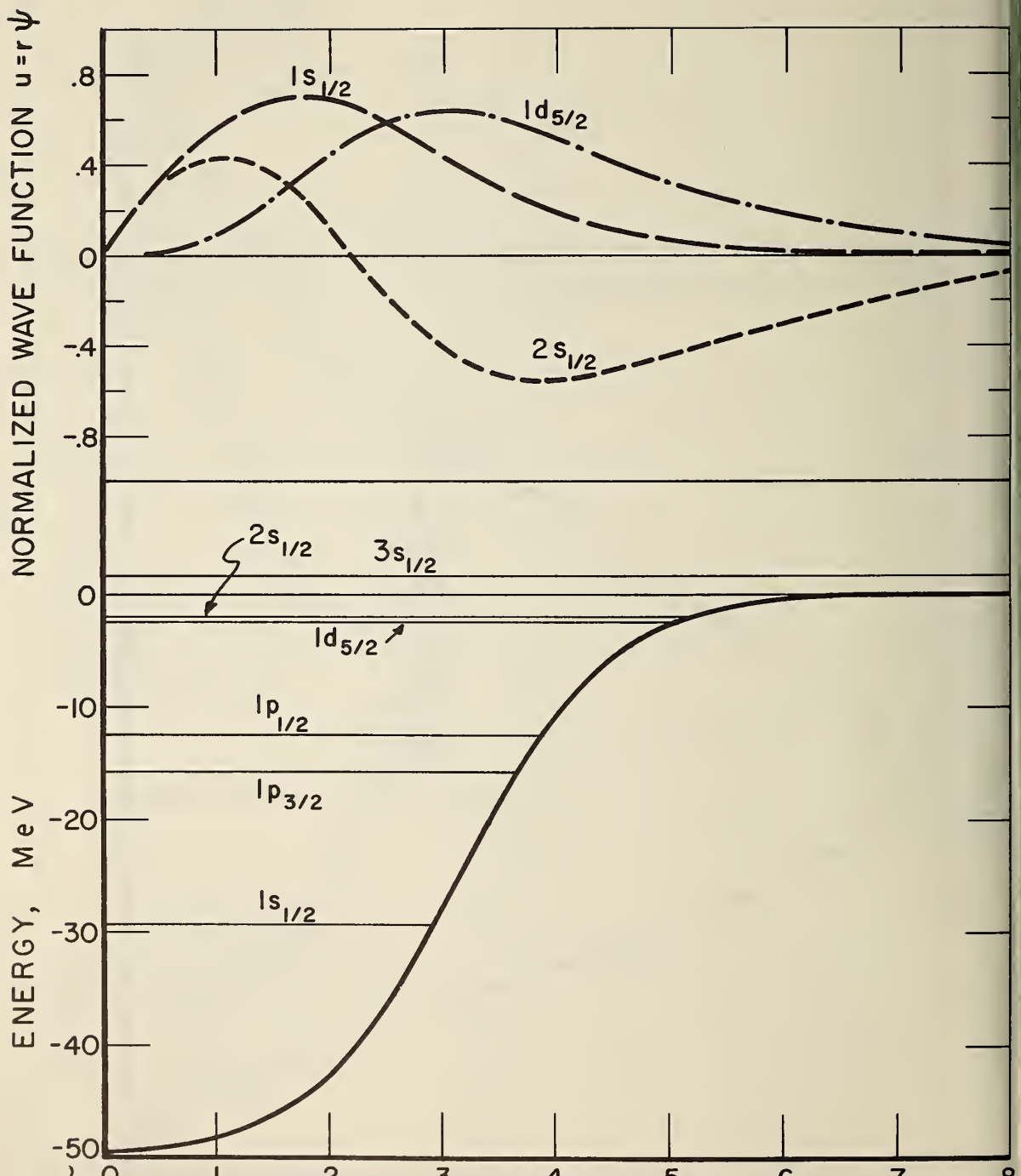

FIGURE 3

* U.S. GOVERNMENT PRINTING OFFICE: 1966—251-019/27