2797/2-79

сообщения
объединенного
института
ядерных
исследований
дубна

Ц840а
К-144

Р11 - 12360

Г.С.Казача

ПРОГРАММА ВЫЧИСЛЕНИЯ РЕЗОНАНСНЫХ СОСТОЯНИЙ ДЛЯ ПОТЕНЦИАЛА КОНЕЧНОГО РАДИУСА ДЕЙСТВИЯ

1979

Р11 - 12360

Г.С.Казача

ПРОГРАММА ВЫЧИСЛЕНИЯ РЕЗОНАНСНЫХ СОСТОЯНИЙ ДЛЯ ПОТЕНЦИАЛА КОНЕЧНОГО РАДИУСА ДЕЙСТВИЯ

Объединенный институт
ядерных исследований
БИБЛИОТЕКА

Казача Г.С. P11 - 12360

Программа вычисления резонансных состояний для потенциала конечного радиуса действия

В работе рассматривается метод вычисления собственных значений волновых функций уравнения Шредингера с комплексной энергией и с обрезанным потенциалом Саксона-Вудса. Задача решается конечно-разностным методом, уравнение и граничные условия аппроксимируются разностными уравнениями четвертого порядка аппроксимации. Описывается алгоритм поиска корней на комплексной плоскости, позволяющий найти заданное число собственных значений. Даются описание и текст программы, написанной на ФОРТРАНе.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1979

Kazacha G.S. P11 - 12360

Program for Calculating Resonance States in the Case of a Cutoff Potential

A method for calculating eigenvalues and wave functions of Schroedinger equation with a complex energy and the cutoff Wood-Saxon potential is considered. The problem is solved by the finite-difference method. The equation and boundary conditions are approximated by difference equations with the fourth order accuracy. The procedure for determining roots on a complex plane, that allows one to calculate a given number of eigenvalues is written. The description and text of the corresponding FORTRAN program are given.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1979

© 1979 Объединенный институт ядерных исследований Дубна

I. В настоящей работе рассматривается задача вычисления комплексных собственных значений (резонансных состояний) и соответствующих им волновых функций радиального уравнения Шредингера с потенциалом конечного радиуса действия R.

$$\frac{d^2U}{dr^2} + \frac{2m}{\hbar^2}(E-V(r))U + \frac{\ell(\ell+1)}{r^2}U = 0, \quad r\in[0,R], \tag{I}$$

$$U(0) = 0, \tag{2}$$

$$\frac{U'(R)}{U(R)} = \left.\frac{\frac{d}{dr}h_\ell^+(kr)}{h_\ell^+(kr)}\right|_{r=R} = d_\ell(k), \tag{3}$$

где E – комплексная энергия, $E = E_0 - i\frac{\Gamma}{2}$,

$k = \sqrt{\frac{2m}{\hbar^2}E} = k_0 + ix$,

$h_\ell^+(z)$ – функция Риккати-Ганкеля /I/, имеющая при $z \to \infty$ следующую асимптотику:

$$h_\ell^+(z) \sim e^{i(z-\ell\pi/2)}.$$

Волновые функции U_n удовлетворяют условию нормировки

$$\int_0^R U_n^2(r)dr + \left.\frac{d}{dK}d_\ell(K)\right|_{K=K_n}\frac{U_n^2(R)}{2K_n} = 0. \tag{4}$$

В качестве потенциала был выбран "обрезанный" потенциал Саксона-Вудса, который хорошо описывает среднее поле сферических ядер.

$$V(r) = \begin{cases} -\dfrac{V_o}{1 + \exp[(r-R_o)/\alpha]} + V_{\ell S}, & r \le R, \\ 0, & r > R, \end{cases} \qquad (5)$$

где $V_{\ell S}$ – спин-орбитальный член

$$V_{\ell S} = - \frac{\eta(\bar{\ell}\cdot\bar{S})}{\alpha\cdot r} \frac{V_o \exp[(r-R_o)/\alpha]}{\{1 + \exp[(r-R_o)/\alpha]\}^2} .$$

Следует отметить, что собственные значения k_n существенным образом зависят от радиуса действия потенциала R. Лишь в двух случаях

1) $k_n = ix, \quad x > 0,$

2) $k_n = k_o + ix, \quad x < 0, \quad 2x \ll \frac{1}{\alpha},$

существует сходимость собственных значений при увеличении R. Первый случай соответствует связанным состояниям с отрицательной энергией, второй – подбарьерным состояниям с малой шириной резонанса.

Для решения задачи (I)-(3) одновременно с двух концов решаются две задачи Коши (при заданном k). Из условия равенства логарифмических производных в некоторой внутренней точке x_o получается уравнение для определения собственных значений

$$Q(k) = \frac{U'}{U}\bigg|_{x_o-0} - \frac{U'}{U}\bigg|_{x_o+0} = 0. \qquad (6)$$

Для решения уравнения (6) используется алгоритм, в основе которого лежит метод Ньютона /2/,/3/, позволяющий отыскать заданное число наименьших по модулю собственных значений.

II. Метод решения

Рассмотрим разностную аппроксимацию задачи (I)-(3). Разностные уравнения

$$d_{n-1}U_{n-1} - C_nU_n + d_{n+1}U_{n+1} = 0, \quad n=1,\dots N-1, \qquad (7)$$

где

$$U_n = U(h\cdot n), \quad r_n = h\cdot n,$$

$$d_n = 1.- \frac{(V_э(r_n)-k^2)h^2}{12},$$

$$C_n = 2. + \frac{10(V_э(r_n) - k^2)h^2}{12},$$

$$V_э(r) = \frac{2m}{h^2} V(r) + \frac{\ell(\ell+1)}{r^2},$$

аппроксимируют уравнение (I) на равномерной сетке с шагом h с точностью порядка h^4 /4/.

Производные в (6) также заменяются разностными формулами четвертого порядка аппроксимации.

$$Q(k) = (S_{M+1}U^{\text{л}}_{M+1} - S_{M-1}U^{\text{л}}_{M-1})\, U^{\text{пр}}_M - (S_{M+1}U^{\text{пр}}_{M+1} - S_{M-1}U^{\text{пр}}_{M-1})\, U^{\text{л}}_M = 0, \qquad (8)$$

где $U^{\text{л}}$ и $U^{\text{пр}}$ – значения функции U, полученные при решении задачи Коши слева и справа,

$$M = \frac{X_o}{h},$$

$$S_n = 1 - \frac{(V_э(r_n) - k^2)h^2}{6}.$$

Зададим k_o, $U_o = 0$, $U_1 = (\frac{\pi\alpha h}{2})^{\frac{1}{2}} J_{\ell+\frac{1}{2}}(\alpha h)$,

$$U_N = h^+_\ell(kR), \quad U_{N-1} = h^+_\ell(kr_{N-1}),$$

где

$$\alpha = \sqrt{-\frac{2m}{h^2} V(0) + k^2},$$

$J_\lambda(z)$ – функция Бесселя.

Тогда из (7) можно последовательно определить все $U^{\text{л}}_n$ и $U^{\text{пр}}_n$ и вычислить $Q(k_o)$.

Для процесса Ньютона

$$Z_{K+1} = Z_K - \frac{Q(Z_K)}{Q'(Z_K)} \qquad (9)$$

необходимо знать производную $Q'(k)$. Продифференцируем (7) и (8) по k:

$$d_{n-1}U'_{n-1} - C_nU'_n + d_{n+1}U'_{n+1} + \frac{kh^2}{6}(U_{n-1} + 10U_n + U_{n+1}) = 0, \qquad (10)$$

$$U'_o = 0, \quad U'_1 = \frac{d}{dk}\left\{ (\frac{\pi\alpha h}{2})^{\frac{1}{2}} J_{\ell+\frac{1}{2}}(\alpha h)\right\}, \qquad (11)$$

$$U'_N = \frac{d}{dk} h_\ell^+ (kR), \quad U'_{N-1} = \frac{d}{dk} h_\ell^+ (kr_{N-1}), \tag{12}$$

$$Q'(k) = (S_{M+1}U^{\wedge}_{M+1} - S_{M-1} U^{\wedge}_{M-1})U'^{np}_M - (S_{M+1}U^{np}_{M+1} - S_{M-1}U^{np}_{M-1})U'^{\wedge}_M +$$
$$+ U^{np}_M \left\{ S_{M+1}U'^{\wedge}_{M+1} - S_{M-1}U'^{\wedge}_{M-1} + \frac{kh^2}{3}(U^{\wedge}_{M+1} - U^{\wedge}_{M-1}) \right\} - \tag{13}$$
$$- U^{\wedge}_M \left\{ S_{M+1}U'^{np}_{M+1} - S_{M-1}U'^{np}_{M-1} + \frac{kh^2}{3}(U^{np}_{M+1} - U^{np}_{M-1}) \right\} .$$

Зная U_n и решая задачу Коши справа и слева, из (I0)-(I3) можно найти все $U_n'^{\wedge}$ и $U_n'^{np}$ и вычислить $Q'(k)$.

Считая $Q(k)$ полиномом по k, можно утверждать /2/, что если начальное приближение z_0 удовлетворяет условию $|z_0| \le \min_n |k_n|$, то процесс Ньютона (9) сходится к минимальному по модулю корню k_1. Так как на поиск связанных состояний нередко требуется очень много итераций, а, следовательно, и машинного времени, программа ограничивается поиском корней в нижней полуплоскости. Известно /5/, что при больших n

$$k_n \approx \frac{1}{R} \left\{ S_n - \frac{1}{2S_n} \ln\left[(2S_n)^2/A^2 \right] - \frac{i}{2} \ln\left[(2S_n)^2/A^2 \right] \right\} ,$$

где

$$S_n = n\pi - \frac{\pi}{2}\xi \quad , \qquad \xi = \begin{cases} 0, & (-1)^\ell \, V(R) > 0, \\ 1, & (-1)^\ell \, V(R) < 0, \end{cases}$$

$$A^2 = R^2 |V(R)| .$$

Тогда, если найдены первые n корней $k_1, k_2, \ldots k_n$, в качестве начального приближения для следующего корня можно взять

$$z_0 = k_n + \frac{\pi}{R} + \frac{i}{R} \ln \frac{n}{n+1}$$

и вместо $Q(k)$ рассматривать функцию

$$Q_1 = \frac{Q(k)}{\prod_{m=1}^{n} (k-k_m)} .$$

Процесс Ньютона в этом случае будет иметь вид

$$z_{K+1} = z_K - \frac{Q(z_K)}{Q'(z_K) - Q(z_K)\cdot P} , \tag{I4}$$

где $$P = \sum_{m=1}^{n} \frac{1}{z_K - k_m} .$$

После каждой итерации по формулам (9), (I4) проверяется условие

$$|Q(z_{K+1})| < |Q(z_K)| . \tag{I5}$$

Если это условие не выполнено, далее процесс продолжается по формуле

$$z_{K+1}^{(i)} = z_K - t_i \frac{Q(z_K)}{Q'(z_K) - Q(z_K)\cdot P} , \quad t_i = \frac{t_{i-1}}{2} , \quad t_0 = 1,$$

пока не будет выполнено условие (I5).

Если величина $$\frac{Q(z_K)}{Q'(z_K) - Q(z_K)\cdot P} > \text{Const},$$

то в качестве следующего приближения берется точка на окружности $|z - z_K| = \text{Const}$, где выполнено условие (I5).

Описанный выше алгоритм был использован для нахождения резонансных состояний сферических ядер. Параметры потенциала задавались по формулам

$$V_0 = (V_1 \cdot N + V_2 Z)/A \ , \quad V_1 = 19.7, \quad V_2 = 87,$$

$$\alpha = 0.63, \quad \eta = 0{,}263\left(1 + \frac{2(TN-Z)}{A} \right),$$

$$R_0 = 1{,}24 \cdot A^{1/3} ,$$

где A – номер ядра, Z – заряд ядра, N – число нейтронов.

Ш. Описание программы

Программа вычисления резонансных состояний сферических ядер написана на языке ФОРТРАН для ЭВМ CDC-6500. При числе узлов разностной сетки $N \le 1001$ она занимает 105000В слов центральной памяти. Программа состоит из главной программы RQUASI и следующих подпрограмм: QFUN, Q, HL, BJ, EIGFUN, SPLIN3, DHL.

Д а н н ы е

Для работы программы необходимо ввести следующие величины:

A – атомный вес,
Z – заряд ядра,
TN – число нейтронов,
R – радиус действия потенциала,
H – шаг разностной сетки,
L – орбитальный момент,
PJ – полный момент,
NPR,NPR1 – константы управления печатью.

На первой карте пробиваются значения A,Z,TN,R,H по формату (3F5.0, F6.0, F7.4), на второй – L,PJ,NP,NPR,NPR1 по формату (I3, F5.1,I3, 2I2).

Печать результатов

В программе печатаются собственные значения k_n и соответствующие им значения энергии. Если $NPR1 \neq 0$, то после каждой итерации печатаются значения Z_K, $Q(Z_K)$ и $Q'(Z_K)$. Если $NPR \neq 0$, то печатаются собственные функции, удовлетворяющие условию нормировки (4).

Описание программ

PROGRAM RQUASI(INPUT, OUTPUT) – главная программа. Вычисляет заданное число NP наименьших собственных значений k_n, $n=1,2,\dots,NP$ и, если необходимо, соответствующие им собственные функции.

В тексте программы приняты следующие обозначения:

A – атомный вес,
$AK = \frac{2m}{\hbar^2}$,
E – энергия,
H – шаг разностной сетки,
KN – массив для собственных значений,
L – орбитальный момент,
NP – число состояний, которое необходимо вычислить,
PJ – полный момент,
R – радиус действия потенциала,
TK – значение k в процессе итераций,
TKN – начальное значение для процесса Ньютона,
TN – число нейтронов,
V – массив для значений $V_э(r)$ в узлах разностной сетки,

VO,RO,DZ=α, ZT=η – параметры потенциала, WQ=$Q(Z_K)$, DQ=$Q'(Z_K)$
Z – заряд ядра.
COMPLEX FUNCTION QFUN(W)-вычисляет значения $Q(k_0)$ и $Q'(k_0)$. W = k_0, QFUN = $Q(k_0)$, DERIV = $Q'(k_0)$.

U, U1 – массивы для значений функций $U^\wedge$ и U^{np} в узлах разностной сетки.

UD, UD1 – массивы для значения производной по k от функций $U^\wedge$ и U^{np}.

M = XO/H.

FUNCTION Q(x) – вычисляет $V_э = \frac{2m}{\hbar^2} V(x) + \frac{\ell(\ell+1)}{x^2}$.
SUBROUTINE HL(S,DL,Z)– вычисляет $h_\ell^+(Z)$ и $\frac{d}{dz} h_\ell^+(z)$.
Z – аргумент, $S = h_\ell^+(z)$, $DL = \frac{d}{dz}(z)$.
SUBROUTINE BJ(S,DL,Z)– вычисляет функцию $j_\ell(z) = (\frac{\pi z}{2})^{\frac{1}{2}} J_{\ell+\frac{1}{2}}(z)$ и $\frac{d}{dz} j_\ell(z)$.

Z – аргумент, $S = j_\ell(z)$, $DL = \frac{d}{dz} j_\ell(z)$.

SUBROUTINE EIGFUN(W) – нормирует и печатает собственную функцию для заданного $W=k_n$.
W – собственное значение k.
U – массив для значений собственной функции.

SUBROUTINE SPLIN3(X,Y,DERIV,N,NC,VOFINT) – упрощенный вариант стандартной программы с тем же именем. Вычисляет интеграл от таблично заданной функции, используя сплайн-аппроксимацию третьего порядка.

X – массив значений переменной интегрирования.
Y – массив значений подынтегральной функции.
N – число точек.
NC = N + 1
DERIV – рабочий массив размерности (NC, 2).
VOFINT – значение интеграла.
COMPLEX FUNCTION DHL(Z) – вычисляет $\frac{d}{dk} \frac{k h_\ell^+(z)}{h_\ell^+(z)}$.

Текст программы приводится в приложении.

Автор выражает благодарность А.А.Корнейчуку, С.И.Сердюковой и Ф.А.Гарееву за постоянное внимание к работе и полезные обсуждения.

Приложение

```
      PROGRAM PROS(INPUT,OUTPUT)
 THIS PROGRAM SOLVE COMPLEX EIGEN VALUE PROBLEM FOR
 SHRODINGER EQUATION
   D2U/DX2+K**2*U-V(X)U=0,U(0)=0,((DU/DX)/U)X=R=D(K)
 A-ATOMIC WEIGHT, Z-NUCLEUS CHARGE, TN-NUMBER OF NEUTRONS
 R-RADIUS OF INTERACTION, H-STEP ALONG X
 N-NUMBER OF POINTS(N.LT.1001)
 L-ORBITAL MOMENT, PJ-WHOLE MOMENT
 NP-NUMBER OF EIGEN VALUES SHOULD BE CALCULATED
 NPR,NPP1-CONTROL PRINT CONSTANT
 KN-ARRAY FOR EIGEN VALUES, E-EIGEN ENERGY
 V-ARRAY FOR POTENTIAL
      COMPLEX TK,TK1,TR,WO,DO,OFUN,DERIV,KN,HK,WO1,AI,TKN,DTK,E
      DIMENSION KN(100)
      COMMON/BL1/R0,R,DZ,ZT,AK,V0,L,SL
      COMMON/BL2/H,N,V(1001),DERIV
      READ 10,A,Z,TN,R,H
      READ 11,L,PJ,NP,NPP,NPR1
10    FORMAT(3F5.0,F6.2,F7.4)
11    FORMAT(I3,F5.1,I3,2I3)
      AI=(0.,1.)
      V1=19.7    $     V2=37.
      V0=(V1*TN+V2*Z)/A
      DZ=0.63    $     ZT=0.263*(1.+2.*(TN-Z)/A)
      R0=1.24
      R0=R0*A**0.3333333333
      AK=0.0482
       N=R/H+1   $   R=H*(N-1)
      PRINT 12,A,Z,V0,R0,DZ,ZT,R,H
      PRINT 13,L,PJ
12    FORMAT(20X,2HA=,F4.0,2X,2HZ=,F4.0,2X,3HV0=,F5.2,2X,3HR0=,
     1  F5.2,2X,3HDZ=,F4.2,2X,3HZT=,F4.2,/,40X,2HR=,F5.2,2X,
     2  2HH=,F6.4)
13    FORMAT(40X,2HL=,I2,2X,2HJ=,F4.1)
      N=N+1
      SL=PJ*(PJ+1)-L*(L+1)-0.75
      DO 20 I=2,N
      X=H*(I-1)
20      V(I)=O(X)
 CALCULATIONS OF THE ROOTS OF EQUATION WO(K)=0
      I=1
55    TKN=(0.01,0.)
      IF(I.EQ.1)GO TO 56
      TKN=KN(I-1)+CMPLX(3./R,ALOG(I/(I+1.))/R)
56    TK=TKN
      DO 50 J=1,100
      WO=OFUN(TK)   $   DO=DERIV   $  TR=0.
      IF(NPR1.NE.0) PRINT 110,TK,WO,DO
      IF(I.EQ.1)GO TO 45
      DO 40 II=2,I
40    TR=TR+1./(TK-KN(II-1))
45    HK=WO/(DO-WO*TR)
      IF(CABS(HK).LT.0.1)GO TO 51
      NK=10   $   HK=0.1
48    DO 52 II=2,NK
      HK=HK*CEXP(2.*AI*3.14*(II-1)/NK)
      TK1=TK-HK   $   WO1=OFUN(TK1)
```



```
      IF(AIMAG(TK1).GT.0.)GO TO 33
      IF(CABS(WO1).LT.CABS(WO)) GO TO 46
52    CONTINUE
      NK=2*NK  $  GO TO 48
51    TK1=TK-HK  $  WO1=OFUN(TK1)
      IF(AIMAG(TK1).GT.0.) GO TO 30
      IF(CABS(WO1).LT.CABS(WO)) GO TO 46
      HK=HK/2.
      IF(CABS(HK).GT.0.001)GO TO 51
      HK=0.001  $  NK=10  $  GO TO 48
46    IF(CABS(TK-TK1).LT.0.0001)GO TO 60
      TK=TK1
110   FORMAT(2X,3HTK=,2E12.4,2X,2HO=,2E12.4,2X,3HDO=,2E12.4)
50    CONTINUE
80    DTK=(0.,-0.1)
      IF(AIMAG(TKN).LT.-1.5)DTK=(0.1,0.)
      TKN=TKN+DTK  $  GO TO 56
60    KN(I)=TK1
      IF(REAL(KN(I)).LT.0.)KN(I)=-CONJG(KN(I))
      E=KN(I)*KN(I)/AK
      PRINT 100,KN(I),E
      IF(NPR.NE.0) CALL EIGFUN(KN(I))
100   FORMAT(10X,3HKN=,2E12.4,2X,2HE=,2E12.4)
      I=I+1
      IF(I.GT.NP) GO TO 30
      GO TO 55
30     CONTINUE
      END

      COMPLEX FUNCTION OFUN(W)
 THIS FUNCTION CALCULATES VALUE
      WO(K)=(DUL(X0)/DX)*UR(X0)-(DUR(X0)/DX)*UL(X0),
 WHERE UL-SOLUTION OF INITIAL PROBLEM
 D2UL/DX2+K**2*U-V(X)UL=0, U(0)=0
 UR-SOLUTION OF INITIAL VALUE PROBLEM
 D2 UR/DX2+K**2*UR-V(X)UR=0, (D UR(R)/DX)/UR(R)=D(K)
 U,U1-ARRAYS FOR UL AND UR RESPECTIVLY
 UD,UD1-ARRAYS FOR D UL/DK AND DUR/DK
 OFUN=WO(W),   DERIV=DWO(W)/DK
      COMPLEX U,U1,W,FUN,UD,UD1,X,D,D1,C,S,S1,KAPPA1,DERIV,SL1,SL2
      DIMENSION UD(1001),UD1(1001),FUN(1001)
      COMMON/BL1/R0,R,DZ,ZT,AK,V0,L,SL
      COMMON/BL2/H,N,V(1001),DERIV
      COMMON/BL3/U(1001),U1(1001),M,M1
      DO 10 I=2,N
10    FUN(I)=V(I)-W*W
      U(1)=0. $ KAPPA1=CSQRT(-FUN(3)+L*(L+1)/4./H/H)
      X=KAPPA1*H $ CALLBJ(U(2),UD(2),X) $ UD(2)=UD(2)*H
      X=KAPPA1*2.*H $ CALL BJ(U(3),UD(3),X) $ UD(3)=UD(3)*2.*H
      UD(2)=UD(2)*W/KAPPA1 $ UD(3)=UD(3)*W/KAPPA1
      X=W*R $ CALL HL(U1(1),UD1(1),X)  $  UD1(1)=UD1(1)*R
      X=W*(R-H) $ CALL HL(U1(2),UD1(2),X) $ UD1(2)=UD1(2)*(R-H)
      M=1.3*R0/H   $    M2=M+1
      M1=N+1-M     $    M3=M1+1
      DO 14 NN=4,M2
      C=1.-FUN(NN)*H*H/12.   $  D1=1.-FUN(NN-2)*H*H/12.
      C=2.+10.*FUN(NN-1)*H*H/12.
```

```
      U(NN)=(C*U(NN-1)-D1*U(NN-2))/D
14    UD(NN)=(C*UD(NN-1)-D1*UD(NN-2))/D-
     1   W*H*H*(U(NN-2)+10.*U(NN-1)+U(NN))/6./D
      CO 13 MM=3,M3
      NN=N-MM+2
      C=1.-FUN(NN+1)*H*H/12.  $  D1=1.-FUN(NN-1)*H*H/12.
      C=2.+10.*FUN(NN)*H*H/12.
      U1(MM)=(C*U1(MM-1)-D*U1(MM-2))/D1
13    UD1(MM)=(C*UD1(MM-1)-D*UD1(MM-2))/D1-
     1  W*H*H*(U1(MM-2)+10.*U1(MM-1)+U1(MM))/6./D1
      S=1.-FUN(M+1)*H*H/6.  $ S1=1.-FUN(M-1)*H*H/6.
      SL1=S*U(M+1)-S1*U(M-1)  $ SL2=S*U1(M1-1)-S1*U1(M1+1)
      CERIV=(S*UD(M+1)-S1*UD(M-1)+W*H*H*(U(M+1)-U(M-1))/3.)*U1(M1)
     1 +SL1*UD1(M1)-(S*UD1(M1-1)-S1*UD1(M1+1)+W*H*H*(U1(M1-1)-U1(M1
     2 +1))/3.)*U(M)-SL2*UD(M)
      CFUN=SL1*U1(M1)-SL2*U(M)
      RETURN
      END

      FUNCTION Q(X)
C THIS FUNCTION CALCULATES THE POTENTIAL
      COMMON/BL1/RØ,R,DZ,ZT,AK,VØ,L,SL
      Y=X
      EX=EXP((Y-RØ)/DZ)
      EX1=1.+EX
      U=AK*VØ
      C=-U/EX1
      C=Q+Q*ZT*SL*EX/(EX1*DZ*Y)
      C=Q+L*(L+1)/(Y*Y)
      RETURN
      END

      SUBROUTINE HL(S,DL,Z)
C CALCULATES HANKEL FUNCTION HL(Z)
C Z-ARGUMENT, S-VALUE OF HANKEL FUNCTION, DL=DHL/DZ
      COMPLEX S,DL,SUM,SUM1,CK,CK1,AI,CE,Z
      COMMON/BL1/RØ,R,DZ,ZT,AK,VØ,L,SL
      PI=3.14 $SUM=1.$SUM1=Ø. $CK=1. $ AI=(Ø.,1.)
      CO 1 I=1,L
      CK=CK*(I*(I-1)-L*(L+1))/2./AI/Z/I
      CK1=CK*I/Z
      SUM=SUM+CK
1     SUM1=SUM1+CK1
      CE=CEXP(AI*(Z-PI*L/2.)) $ DL=CE*(AI*SUM-SUM1)
      S=SUM*CE
      RETURN $ END

      SUBROUTINE BJ(S,DL,Z)
C CALCULATES FUNCTION J=(PI*Z/2)**(1/2)*JL(Z)
C JL(Z)-BESSEL FUNCTION OF L+1/2 ORDER
C Z-ARGUMENT, S=J(Z), DL=DJ(Z)/DZ
      COMPLEX S,DL,Z,SUM,SUM1,CK,CK1,ZL
      COMMON/BL1/RØ,R,DZ,ZT,AK,VØ,L,SL
      SUM=1.  $ SUM1=Ø.  $ CK=1.
      DO 1 I=2,6,2
      CK=-CK*Z**2/I/(I+2.*L+1.) $ CK1=CK*I $ SUM=SUM+CK
1     SUM1=SUM1+CK1
```



```
      ZL=Z**L $ S=ZL*Z*SUM $ DL=ZL*((L+1.)*SUM+SUM1)
      RETURN  $  END

      SUBROUTINE EIGFUN(W)
      COMPLEX U,U1,W,CONST
      CIMENSION RFI(1001),AFI(1001),X(1001),DRV(1001,2)
      COMMON/BL1/RØ,R,DZ,ZT,AK,VØ,L,SL
      COMMON/BL2/H,N,V(1001),DERIV
      COMMON/BL3/U(1001),U1(1001),M,M1
      CONST=U(M)/U1(M1)  $ N1=N-1
      CO 2 I=1,N
2     X(I)=H*(I-1)
      CO 1 I=M,N
      J=N-I+1
1     U(I)=CONST*U1(J)
      CO 7 I=1,N
      RFI(I)=REAL(U(I)*U(I))
7     AFI(I)=AIMAG(U(I)*U(I))
      CALL SPLIN3(X,RFI,DRV,N1,N,RF)
      CALL SPLIN3(X,AFI,DRV,N1,N,AF)
      CONST=CMPLX(RF,AF)+DHL(W*R)*U(N1)*U(N1)/2./W
      CONST=CSQRT(1./CONST)
      DO 3 I=1,N
      U(I)=U(I)*CONST $ RFI(I)=REAL(U(I))
3     AFI(I)=AIMAG(U(I))
      PRINT 4
      PRINT 5,(RFI(I),I=1,N)
      PRINT 6
      PRINT 5,(AFI(I),I=1,N)
4     FORMAT(/,10X,27HREAL PART OF EIGEN FUNCTION)
6     FORMAT(/,10X,30HIMAGARY PART OF EIGEN FUNCTION)
5     FORMAT(10E12.4)
      RETURN $ END

      SUBROUTINE SPLIN3(X,Y,DERIV,N,NC,VOFINT)
      CIMENSION X(NC) , Y(NC) , DERIV(NC,2)
      DATA ZERO,HALF,ONE,THREE/Ø.,.5,1.,3./
      DATA THIRD , SIXTH / .333333333333333 , .166666666666667 /
      NXY=N
      SECD1=ZERO
      SECDN = ZERO
      BET1=ONE/(ONE+HALF*(X(2)-X(1))/(X(3)-X(2)))
      ALF1=BET1*(ONE- ((X(2)-X(1))/(X(3)-X(2)))**2)
      BETN=ONE/(ONE+HALF*(X(N)-X(N-1))/(X(N-1)-X(N-2)))

      ALFN=BETN*(ONE- ((X(N)-X(N-1))/(X(N-1)-X(N-2)))**2)
1015  DERIV(1,2)=SECD1
      DERIV(N,2)=SECDN
      DERIV(1,1)=ZERO
      DXPLUS=X(2)-X(1)
      IF ( DXPLUS.GT.ZERO) GO TO 1020
      IN=1
1020  DYPLUS=(Y(2)-Y(1))/DXPLUS
      IU=N-1
      DO 1040 I=2,IU
      DXMIN =DXPLUS
      DYMIN =DYPLUS
```

```
      DXPLUS=X(I+1)-X(I)
      IF (DXPLUS.GT.ZERO) GO TO 1030
      IN=I
 1030 DXINV =ONE/(DXPLUS+DXMIN)
      DYPLUS=(Y(I+1)-Y(I))/DXPLUS
      DIVDIF=DXINV*(DYPLUS-DYMIN)
      ALF    =HALF*DXINV*DXMIN
      BET    =HALF-ALF
      IF (I.EQ.2)  DIVDIF=DIVDIF-THIRD*ALF*DERIV(1,2)
      IF (I.EQ.IU) DIVDIF=DIVDIF-THIRD*BET*DERIV(N,2)
      IF (I.EQ.2) ALF=ZERO
      IF (I.NE.2) GO TO 1032
      BET=BET*ALF1
      DIVDIF=DIVDIF*BET1
      GO TO 1035
 1032 IF (I.NE.IU) GO TO 1035
      ALF=ALF*ALFN
      DIVDIF=DIVDIF*BETN
 1035 DXINV =ONE/(ONE+ALF*DERIV(I-1,1))
      DERIV(I,1)=-DXINV*BET
      DERIV(I,2)= DXINV*(THREE*DIVDIF-ALF*DERIV(I-1,2))
 1040 CONTINUE
 1050 DO 1060 I=2,IU
      J=N-I
      DERIV(J,2)=DERIV(J,1)*DERIV(J+1,2)+DERIV(J,2)
 1060 CONTINUE
      DERIV(1,2)=((X(3)-X(1))/(X(3)-X(2)))*DERIV(2,2)-((X(2)-X(1))/(X(3)
     1-X(2)))*DERIV(3,2)
      DERIV(N,2)=-((X(N)-X(N-1))/(X(N-1)-X(N-2)))*DERIV(N-2,2)+((X(N)-X
     1(N-2))/(X(N-1)-X(N-2)))*DERIV(N-1,2)
 1070 VOFINT=ZERO
      DO 1080 I=1,IU
      DXPLUS=X(I+1)-X(I)
      DYPLUS=Y(I+1)-Y(I)
      DIVDIF=DYPLUS/DXPLUS
      DERIV(I,1)=DIVDIF-DXPLUS*(THIRD*DERIV(I,2)+SIXTH*DERIV(I+1,2))
      DXPLUS=HALF*DXPLUS
      VOFINT=VOFINT+DXPLUS*(Y(I+1)+Y(I)-THIRD*(DERIV(I+1,2)+DERIV(I,2))
     1DXPLUS**2)
 1080 CONTINUE
 2000 CONTINUE
      RETURN
      END

      COMPLEX FUNCTION DHL(Z)
      COMPLEX Z,SUM,SUM1,SUM2,CK,CK1,CK2,AI,DL,DL2
      COMMON/BL1/P0,P,DZ,ZT,AK,V0,L,SL
      SUM=1. $ SUM1=0. $ SUM2=0. $CK=1. $AI=(0.,1.)
      DO 1 I=1,L
      CK=CK*(I*(I-1)-L*(L+1))/2./AI/I/Z $ CK1=CK*I/Z $ CK2=CK1*(I+1)/Z
      SUM=SUM+CK $ SUM1=SUM1+CK1
    1 SUM2=SUM2+CK2
      DL=AI-SUM1/SUM
      DL2=-1.-2.*AI*SUM1/SUM+SUM2/SUM
      DHL=DL+Z*(DL2-DL*DL)
      RETURN $ END
```


Литература

I. Дж.Тейлор. Теория рассеяния. М., "Мир", 1975, с.216-219.

2. Акишин П.Г. и Пузынин И.В. ОИЯИ, 5-10992, Дубна, 1977.

3. Воеводин В.В. ЖВМФ, т.I, № 2, 196I.

4. Самарский А.А. Теория разностных схем. М.,"Наука", 1977, с. 91-93.

5. H.M.Nussenzveig. Causality and Dispersion Relations, Academic Press, New York, 1972.

Рукопись поступила в издательский отдел
2 апреля 1979 года.