ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

Р5 - 3712

Н.Ю.Ширикова

# ВЫЧИСЛЕНИЕ УРОВНЕЙ ЭНЕРГИЙ И ВОЛНОВЫХ ФУНКЦИЙ СФЕРИЧЕСКИХ ЯДЕР

ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И АВТОМАТИЗАЦИИ

1968

Объединенный институт
ядерных исследований
ЛВТА

Р5 - 3712

Н.Ю.Ширикова

# ВЫЧИСЛЕНИЕ УРОВНЕЙ ЭНЕРГИЙ И ВОЛНОВЫХ ФУНКЦИЙ СФЕРИЧЕСКИХ ЯДЕР

Уровни энергии $E \equiv E_{n\ell j}$ являются решением уравнения

$$\int_{r_1(E)}^{r_2(E)} \sqrt{\frac{2m}{h^2}(E - V(r))}\, dr = (n + \frac{1}{2})\pi , \qquad (1)$$

где $r_1(E) < r_2(E)$ – решения уравнения

$$E - V(r) = 0 . \qquad (2)$$

В качестве среднего поля, определяющего одночастичные уровни и волновые функции, берем потенциал Саксона-Вудса. Тогда эффективный потенциал взаимодействия для нейтронов имеет вид .

$$V(r) = -\frac{V_0}{1 + \exp(\alpha(r - R_0))}\left(1 + \frac{\kappa\alpha \exp(\alpha(r - R_0))(\vec{\sigma}\vec{\ell})}{r(1 + \exp(\alpha(r - R_0)))}\right) + \frac{h^2(\ell + ½)^2}{2mr^2} . \qquad (3)$$

Для протонов в эффективный потенциал (3) необходимо добавить член, описывающий кулоновское взаимодействие

$$V(r) = (3) + k\frac{z-1}{r} f(r), \qquad f(r) = \begin{cases} \frac{3}{2}\frac{r}{R_0} - \frac{1}{2}\left(\frac{r}{R_0}\right)^3, & r < R_0 \\ 1, & r \geq R_0 \end{cases} \qquad (4)$$

Радиус ядра определяется по формуле

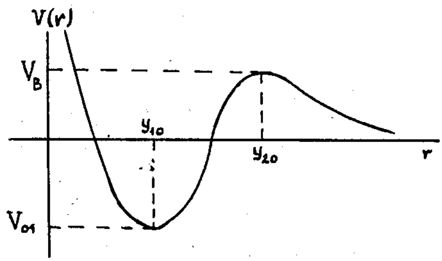
$$R_0 = r_0 A^{1/3}, \tag{5}$$

$$(\vec{\sigma}\vec{\ell}) = \begin{cases} \ell, & j = \ell + \frac{1}{2}, \\ -(\ell+1), & j = \ell - \frac{1}{2}. \end{cases} \tag{6}$$

Для заданных значений $\ell, j$ сначала исследуются свойства $V(r)$ на отрезке $[\frac{R_0}{40}, 2R_0]$.

Отыскиваются корни $V'(r)$ на этом отрезке. Этих корней два. Обозначим их через $y_{10}$ и $y_{20}$. Первый корень находится на отрезке $[\frac{R_0}{40}, R_0]$, а второй - на $[R_0, 2R_0]$.

Обозначим $V_{01} = V(y_{10})$, $V_B = V(y_{20})$.




Если $V_B < 1$, то полагаем $V_B = 1$. Задаем $E = V_B - 1$. Отыскиваем корни уравнения (2). Обозначим их через $z_{10}$, $z_{20}$. Вычисляем

$$L_2 = \int_{z_{10}}^{z_{20}} \sqrt{\frac{2m}{h^2}(V_B - 1 - V(r))}\, dr . \tag{7}$$

Для заданных значений $\ell, j, n$ отыскание $E_{n\ell j}$ производится следующим образом. Проверяется выполнение условия

$$L_2 - (n + \frac{1}{2})\pi > 0 . \tag{8}$$

Если (8) не выполняется, то состояние с заданными $\ell, j, n$ отсутствует. В случае выполнения условия (8) $E_{n\ell j}$ отыскивается на отрезке $[V_{01}-\epsilon_2, V_B-1]$, если разыскивается первый уровень. Если же будут разыскиваться уровни с другим значением $n$ ( $\ell$ и $j$ не меняются), то $E_{(n+1)\ell j}$ разыскивается на отрезке

$$[E_{n\ell j}, V_B - 1].$$

Для выбранного значения $E$ корни уравнения (2) $r_1 \equiv r_1(E), r_2 \equiv r_2(E)$ отыскиваются на отрезках

$$\left[\frac{R_0}{40}, y_{10}\right], \quad [y_{10}, y_{20}].$$

Найдя уровень $E \equiv E_{n\ell j}$, удовлетворяющий (1), определяем величины, характеризующие радиальную часть волновой функции. Запишем радиальную часть в виде:

$$R_{n\ell j}(r) = \frac{N_{n\ell j}}{r} \left(\frac{a}{c}\right)^{1/2} H_n(S) \exp\left(-\frac{S^2}{2}\right). \tag{9}$$

Параметр $a$ определяется как корень уравнения

$$\int_{r_1}^{a} \sqrt{\frac{2m}{h^2}(E_{n\ell j} - V(r))}\, dr = \frac{\pi}{2}\left(n + \frac{1}{2}\right). \tag{10}$$

Этот корень разыскивается на отрезке $[r_1, r_2]$. Функция $S(r)$ является решением уравнения

$$\int_{-\sqrt{2n+1}}^{S} (2n+1-\sigma^2)^{1/2}\, d\sigma = \int_{r_1}^{r} \sqrt{\frac{2m}{h^2}(E_{n\ell j} - V(r))}\, dr. \tag{11}$$

Попытаемся аппроксимировать $S(r)$ в виде

$$S(r) = \begin{cases} b_1 \ln(r/a), & r \le a \\ b \ln(r/a), & r \ge a. \end{cases} \tag{12}$$

Для нахождения параметра $b_1$ ищем $S(r_1 + 0{,}7)$. Это значение $S_1 = S(r_1 + 0{,}7)$ ищем на отрезке $[-\sqrt{2n+1}, 0]$. Тогда уравнение (11) можно переписать в виде:

$$\frac{1}{2} S(2n+1-S^2)^{1/2} + \left(n+\frac{1}{2}\right)\left(\arcsin\frac{S}{\sqrt{2n+1}} + \frac{\pi}{2}\right) = \int_{r_1}^{r_1+0{,}7} \sqrt{\frac{2m}{h^2}(E-V(r))}\,dr . \qquad (13)$$

Решив это уравнение, находим

$$b_1 = \frac{S(r_1 + 0{,}7)}{\ln\left(\frac{r_1 + 0{,}7}{a}\right)} .$$

Параметр $c$ определяется следующей формулой:

$$c = \frac{1}{2}\left(a\,S(a+1) + \left|\frac{a\,S(r_1 + 0{,}7)}{a - r_1 - 0{,}7}\right|\right). \qquad (14)$$

Значение $S_2 = S(a+1)$ находим на отрезке $[0, \sqrt{2n+1}]$, решая уравнение, аналогичное уравнению (13) с той только разницей, что верхний предел интегрирования равен $a+1$.

Для нахождения параметра $b$ ищем $S_3 = S(r_2 + 3{,}3)$ на отрезке $[\sqrt{2n+1}, 10]$. В этом случае уравнение (11) можно записать в виде:

$$\frac{\sqrt{2n+1}}{2} S\left(\frac{S^2}{2n+1} - 1\right)^{1/2} - \left(n+\frac{1}{2}\right)\ln\left(\frac{S}{\sqrt{2n+1}} + \sqrt{\frac{S^2}{2n+1} - 1}\right) = \int_{r_2}^{r_2+3{,}3} \sqrt{\frac{2m}{h^2}(E-V(r))}\,dr . \qquad (15)$$

Тогда

$$b = \frac{S(r_2 + 3{,}3)}{\ln\left(\frac{r_2 + 3{,}3}{a}\right)} . \qquad (16)$$

Параметр $N_{n\ell j}$ определяется по формуле

$$N_{n\ell j} = \sqrt{\frac{c}{a\int_{\epsilon_6}^{2R_q} H_n^2(S)\exp(-S^2)\,dr}} , \qquad (17)$$

где

$$H_0 = 1,$$

$$H_1(S) = 2S,$$

$$H_2(S) = 4S^2 - 2,$$

$$H_3(S) = 8S^3 - 12S,$$

$$\ldots\ldots\ldots\ldots\ldots$$

$$H_{n+1}(S) = 2SH_n(S) - 2nH_{n-1}(S),$$

$\epsilon_6$ – величина порядка $10^{-6}$.

Таким образом, найденный уровень будем охарактеризовывать следующими параметрами

$$-E,\ n+1,\ \ell,\ j,\ a,\ b_1,\ c,\ b,\ N_{n\ell j}\ .$$

Отметим, что нахождение корня уравнения типа (1,2,10,13,15) производится методом деления отрезка пополам.

Программа, написанная на языке Алгол для транслятора ТА-1М, реализует метод вычисления уровней энергий и волновых функций сферических ядер, изложенный в работе /1/.

В программе использованы следующие процедуры ввода, вывода – р 0477, р 0042, р 1041, р 1050, р 0040. Процедура bisec для нахождения корня функции на отрезке взята из /2/, стр 12. Для вычисления интегралов использована процедура simps, составленная И.Н.Силиным.

Для программы задаются следующие величины:

$\lambda = \frac{2m}{h2}$ ,

$\epsilon_1$ – абсолютная точность (примерно $10^{-3}$), с которой отыскиваются корни $r_1(E)$, $r_2(E)$ ,

$\epsilon_2$ – абсолютная точность для отыскания $E_{n\ell j}$,

$\epsilon_3$ – относительная точность вычисления интегралов,

$\epsilon_4$ – абсолютная точность для определения $a$, $S_1$, $S_2$, $S_3$ ,

$\epsilon_6$ – нижний предел (примерно $10^{-6}$) в формуле (17),

$\epsilon_7$ – маленькое число порядка $10^{-7}$,

$k_1$ – узлы для определения S

$k_2$ (заданы $k_1 = 0{,}7$. $k_2 = 3{,}3$),

$k$ – константа из формулы (4).

Для счёта задаются следующие величины

$$A, z, r_0, V_0, \kappa, \alpha$$

и варианты счёта, содержащие

$$\ell, j, n1, n2.$$

Последним числом задается $\ell$, равное 20. Для заданного ядра будут отыскиваться состояния с заданными $\ell, j$ и $n$, принимающим значения от $n1$ до $n2$.

На печать выдается шапка ядра:

$$A, z, R_0, V_0, \kappa, \alpha.$$

Если программа не находит уровень, то печатается

$$\ell, j, n, n2, L_2 - (n + \frac{1}{2})\pi.$$

Последняя величина всегда в этом случае отрицательна. Эта печать указывает, что уровней с $\ell, j, n$ до $\ell, j, n2$ не существует.

После просчёта всех вариантов для заданного ядра происходит упорядочение найденных уровней по величинам $E$ и печатаются параметры уровней. Для каждого уровня печатается

$$-E, n+1, \ell, j, a, b_1, c, b, N.$$

Эти величины выдаются в десятичном виде на перфокарте.

Примечание: В программе предусмотрена аварийная выдача для случая, когда не находится какой-либо из корней

$$E, r_1, r_2, a, S_1, S_2, S_3.$$

В этом случае печатается

$$\ell, j, n, n2, 0.$$

Для ядра

$$A = 237, z = 93, r_0 = 1.24, V_0 = 60.5; \kappa = 0.381; \alpha = 1.44$$

с заданными значениями величин

$$\lambda = \frac{2m}{h^2} = 0.04824 \; ; \quad \epsilon_1 = 10^{-3} \; ; \; \epsilon_2 = 10^{-2} \; ; \; \epsilon_3 = 10^{-3} \; ;$$

$$\epsilon_4 = 10^{-3} \; ; \; \epsilon_6 = 10^{-3} \; ; \quad \epsilon_7 = 10^{-8} \; ; \; k_1 = 0.7 \; ; \quad k_2 = 3.3 \; ;$$

$$k = 1.44 \, ,$$

для уровня $\ell = 9 , \quad j = \frac{19}{2} \quad , \; n = 0$

были получены следующие результаты:

$$- E = - 18.0200512$$
$$n + 1 = 1$$
$$\ell = 9$$
$$j = 9.5$$
$$a = 6.99623364$$
$$b_1 = 5.40383050$$
$$c = 5.57629496$$
$$b = 4.95629827$$
$$N = 0.57102703$$

*Литература*


1. *Б.Н.Калинкин, Я.Грабовский, Ф.А. Гареев, Об уровнях среднего поля ядер, Препринт ОИЯИ Р-2682, Дубна 1966.*
2. *М.И.Агеев, В.П.Алик, Р.М.Галис, Алгоритмы (1-50) ВЦ АН СССР, Москва, 1966.*



Рукопись поступила в издательский отдел

15 февраля 1968 года.

```
begin  array re(1:9),re (1:400);
real h,y10,y20,r 0,r 02,npi,npi1,X1,X2,
v 01,v b ,l 2,l,j,n,r,
n1,n2,pi,he1,he2,he3,he4,e ,hl,v 0,
hk,ha,a ,k,z,kz,k1,k2,
hkhas,hks,a,b1,c,b,n ,cl2,
      cn,cn1,pi2,cn2,hmr,mine ,s ,y,s 1,yr,
he6,ca,h n,h 0,h 1
,he7
,kr1,hmr1,hmr2,z10,z20,s 2,s 3
   ;
integer hn,k ,i,hm,minhm,hr
   ;
real  procedure v (r);  value r;  real r;
    begin  real t,r4,r5,r6;
    t:=eXp(hax(r-r 0)); r4:=t+1;
    r5:=-v 0x(1+hkhasxt/(rxr4))/r4+
      cl2/(rxr);
    if z=0 then  go to m;
    r6:=r/r 0; r5:=r5+kzx
    ( if r<r 0 then (1.5xr6-.5xr6xr6xr6)
    else 1)/r;
m: v :=r5
    end  procedure v ;
real  procedure dv (r);  value r;  real r;
    begin  real t,r4,r5,r6,r2,r7,r3;
    t:=eXp(hax(r-r 0)); r4:=t+1; r2:=rxr;
    r5:=v 0xhaxtx(1+hks/r2-hkhasx(1-t)/
    (rxr4))/(r4xr4)-2xcl2/(r2xr);
    if z=0 then  go to mm;
    r6:=r/r 0;
    if r<r 0 then  begin r3:=1.5xr6-
    .5xr6xr6xr6; r7:=1.5x(1-r6xr6)/r 0
    end  else  begin r3:=1; r7:=0 end ;
    r5:=r5-kzx(r3/r2-r7/r);
mm: dv :=r5
    end  procedure dv ;
```



```
procedure bisece(a,b,eps,eps1,f,X);
value a,b;  real a,b,eps,eps1,X;
real  procedure f;
begin  real y,z;
       procedure fun(y);
       real y;
       begin y:=f(X);
       if abs(y)<eps then  go to fin
       end ;
X:=a; fun(y);
X:=b; fun(z);
if sign(y)=sign(z) then  begin
  p1041(l,j,n,n2,0);  go to pm3 end ;
iter: X:=(a+b)/2; fun(y);
if sign(y)=sign(z) then b:=X else a:=X;
if abs(a-b)>eps1 then  go to iter;
fin: end bisece;
procedure bisec(a,b,eps,eps1,f,X);
value a,b;  real a,b,eps,eps1,X;
real  procedure f;
begin  real y,z;
       procedure fun(y);
       real y;
       begin y:=f(X);
       if abs(y)<eps then  go to fin
       end ;
X:=a; fun(y);
X:=b; fun(z);
if sign(y)=sign(z) then  begin
  p1041(l,j,n,n2,0);  go to pm3 end ;
iter: X:=(a+b)/2; fun(y);
if sign(y)=sign(z) then b:=X else a:=X;
if abs(a-b)>eps1 then  go to iter;
fin: end bisec;
```

```
real procedure p(r); value r; real r;
begin p:=e-v(r) end ;
real procedure int(e ); value e ; real e ;
begin
if e =v b -1 and kr1=/=0 then begin hmr:=l 2;
x1:=z10; x2:=z20; go to m end ;
bisec(h,y10,he7,he1,p,x1);
bisec(y10,y20,he7,he1,p,x2);
simps(x1,x2,h,he3,he7,funct,hmr,hmr1,hmr2);
m:int:=hmr-npi end procedure int;
real procedure finda(a); value a; real a;
begin if a=x1 then begin hmr:=0; go to met
end else simps(x1,a,h,he3,he7,funct,hmr,
hmr1,hmr2);
met:finda:=hmr-npi1 end finda;
real procedure negs(s ); value s ; real s ;
begin real y; y:=s /cn2; negs:=.5xcn2xs x
  sqrt(1-yxy)+cn1x(arcsin(y)+pi2)-hmr end ;
real procedure poss(s ); value s ; real s ;
begin real y,yr; y:=s /cn2;
yr:=sqrt(yxy-1); poss:=.5xs xcn2xyr-cn1x
  ln(y+yr)-hmr end poss;
procedure funct(r,y); value r; real r,y;
y:=sqrt(hlxabs(e -v (r)));
procedure norm(r,y); value r; real r,y;
begin
switch p :=mn0,mn1,mn2,mn3;
s :=(ln(r)-ca)x
    ( if r<a then b1 else b);
if n<4 then go to p (n+1);
    h 0:=1; h 1:=y:=2xs ;
    for i:=1 step 1 until n-1 do begin
      h n:=yxh 1-2xnxh 0; h 0:=h 1; h 1:=h n end ;
    go to pm10;
mn0: h n:=1; go to pm10;
mn1: h n:=2xs ; go to pm10;
mn2: h n:=4xs xs -2; go to pm10;
mn3: h n:=s x(8xs xs -12);
pm10:y:=exp(-s xs )xh nxh n end ;
```



```
procedure simps(a,b,h,reps,aeps,funct,
i,ih,iabs);
value a,b,reps,aeps,h;
real a,b,reps,aeps,h,i,ih,iabs;
procedure funct;
begin integer k; array f(1:7),p(2:5);
real x,x0,di1,di2,di3,c,y,eps,delta;
i:=ih:=iabs:=0; p(2):=p(4):=4; p(3):=2;
p(5):=1; if b-a=0 then go to exit;
reps:=abs(reps); aeps:=abs(aeps);
  for k:=1 step 1 until 7 do f(k):=1018;
  x:=a; c:=0; funct(x,y); f(1):=y/3;
forward: x0:=x; if (x0+4xh-b)xsign(h)>0
then begin h:=(b-x0)/4; if h=0 then
go to exit; for k:=2 step 1 until 7 do
  f(k):=1018; c:=1 end ;
back: di2:=f(1); di3:=abs(f(1));
for k:=2 step 1 until 5 do
    begin x:=x+h; if f(k)=1018 then
begin funct(x,y); f(k):=y/3 end ;
di2:=di2+p(k)xf(k);
di3:=di3+p(k)xabs(f(k)) end k;
di1:=(f(1)+4xf(3)+f(5))x2xh;
di2:=di2xh; di3:=di3xh;
if reps=0 and aeps=0 then go to fix;
eps:=(iabs+di3)xreps;
if eps<aeps then eps:=aeps;
delta:=abs(di2-di1);
if delta<eps then
begin if delta<eps/8 then
begin h:=2xh; f(1):=f(5); f(2):=f(6);
f(3):=f(7); f(4):=f(5):=f(6):=f(7):=1018
end else
fix: begin f(1):=f(5); f(3):=f(6); f(5):=
  f(7); f(2):=f(4):=f(6):=f(7):=1018
end h;
di1:=di2+(di2-di1)/15; i:=i+di1;
ih:=ih+di2; iabs:=iabs+di3
end else
begin h:=h/2; f(7):=f(5); f(6):=f(4);
```

```
f(5):=f(3); f(3):=f(2);
   f(2):=f(4):=1018; X:=X0; c:=0;
go to back end half h;
if c=0 then go to forward;
exit: end simps;
   P0477(t);
P0042(hl,he1,he2,he3,he4,he6,he7,k1,k2,k);
   pi:=3.14159265; pi2:=pi/2;
pm6:P0042(a,z,r 0,v 0,hk,ha);
kz:=kx(z-1);
   r 0:=r 0xa↑(1/3); r 02:=r 0+r 0;
k:=0; hn:=1; P1041(a,z,r 0,v 0,hk,ha);
   h:=r 0/40;
pm3: P0042(l); if l=20 then go to pm5;
P0042(j,n1,n2); cl2:=l+.5; if j=cl2 then
   begin hkhas:=hkxhaxl; hks:=hkxl end else
   begin hkhas:=-hkxhax(l+1); hks:=-hkx(l+1)
   end; cl2:=(cl2xcl2)/hl;
npi:=kr1:=0;
bisec(h,r 0,he7,he1,dv,y10);
bisec(r 0,r 02,he7,he1,dv,y20);
v 01:=v (y10); v b:=v (y20);
if v 01>0 and v b<1 then begin P1041(l,j,v 01,
v b); go to pm3 end;
if v b<1 then v b:=1; e:=v b-1; l 2:=int(e);
z10:=X1; z20:=X2;
e:=v 01+he2; n:=n1;
kr1:=1;
pm4: cn:=n+n+1; cn2:=sqrt(cn);
   cn1:=n+.5; npi:=cn1xpi; npi1:=npi/2;
   if l 2-npi<0 then begin
   P1041(l,j,n,n2,l 2-npi);
   go to pm3 end;
bisece(e,v b-1,he7,he2,int,e);
   comment find enlj;
bisec(X1,X2,he7,he4,finda,a);
simps(X1,X1+k1,h,he3,he7,funct,hmr,hmr1,hmr2)
;
```



```
bisec(-cn2+he4,0,he7,he4,negs,s 1);
   b1:=s 1/ln((X1+k1)/a);
simps(X1,a+1,h,he3,he7,funct,hmr,hmr1,hmr2);
bisec(0,cn2-he4,he7,he4,negs,s 2);
c:=.5x(axs 2+abs(axs 1/(a-X1-k1)));
simps(X2,X2+k2,h,he3,he7,funct,hmr,hmr1,
hmr2);
bisec(cn2,10,he7,he4,poss,s 3);
b:=s 3/ln((X2+k2)/a);
ca:=ln(a);
simps(he6,r 02,h,he3,he7,norm,hmr,hmr1,hmr2);
n:=sqrt(c/(axhmr));
   re(1):=-e; re(2):=n+1; re(3):=l;
   re(4):=j; re(5):=a; re(6):=b1;
re(7):=c; re(8):=b; re(9):=n;
P1050(111,k,re); re(hn):=-e; hn:=hn+1;
   k:=k+10; if hn>400 then go to pm5;
pm7: n:=n+1; if n<n2 then go to pm4
   else pm3;
pm5: for hm:=1 step 1 until hn-1 do begin
mine:=400; for hr:=1 step 1 until hn-1 do
if re(hr)<mine then
begin mine:=re(hr); minhm:=hr end;
   k:=(minhm-1)x10; P1050(101,k,re);
    P1041(re); P0040(re); re(minhm):=400;
   end hm;
   if l=20 then go to pm6;
   k:=0; hn:=1; go to pm7;
t:P1041(l,j,n,y10,y20,z10,z20,
r 0,npi,npi1,e,X1,X2,v 01,v b,l 2,kz,a,b1,
c,b,n,hmr,hmr1,hmr2,s,s 1,s 2,s 3,mine,hn,k,h,
i,hm,minhm,hr,re); stop
   end
```