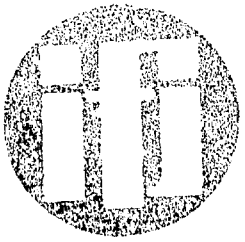
INSTYTUT FIZYKI JĄDROWEJ
INSTITUTE OF NUCLEAR PHYSICS
ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ



KRAKÓW

RAPORT No 865/PL

HAUSER-FESHBACH CROSS-SECTION CALCULATIONS FOR ELASTIC AND INELASTIC SCATTERING OF ALPHA PARTICLES PROGRAM CORA

A. HARTMAN, M. SIEMASZKO, W. ZIPPER

1975

HAUSER-FESHBACH CROSS-SECTION CALCULATIONS FOR ELASTIC AND INELASTIC SCATTERING OF ALPHA PARTICLES - PROGRAM CORA

OBLICZENIA PRZEKROJU CZYNNEGO NA ELASTYCZNE I NIEELASTYCZNE ROZPRASZANIE CZĄSTEK ALFA NA GRUNCIE FORMALIZMU HAUSERA-FESHBACHA, PROGRAM CORA

ВЫЧИСЛЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ СЕЧЕНИЙ ДЛЯ УПРУГОГО И НЕУПРУГОГО РАССЕЯНИЯ АЛЬФА ЧАСТИЦ НА БАЗЕ ФОРМАЛИЗМА ХАУЗЕРА-ФЕШБАХА ПРОГРАММА CORA

A. Hartman M. Siemaszko W. Zipper
Institute of Physics, Silesian University, Katowice
Poland
and
Institute of Nuclear Physics, Cracow, Poland

C R A C O W
February 1974

The program CORA was prepared on the basis of Hauser and Feshbach compound reaction formalism. It allows the differential cross-section distributions for the elastic and inelastic scattering of alpha particles / via compound nucleus state/ to be calculated. The transmission coefficients are calculated on the basis of a four parameter optical model. The search procedure / for ρ and σ parameters / is also included.

Program CORA został napisany w oparciu o formalizm Hausera i Feshbacha dla rozpraszania przez jądro złożone. Umożliwia on obliczanie różniczkowego przekroju czynnego na elastyczne i nieelastyczne rozpraszanie cząstek alfa przez jądro złożone. Współczynniki transmisji liczone są w oparciu o czteroparametrowy model optyczny. Program posiada możliwość dopasowywania obliczonych rozkładów kątowych do danych eksperymentalnych przez automatyczną zmianę parametrów ρ i σ .

Представлено программу построену на базе формализма Хаузера-Фешбаха для реакции идущих через составное ядро. Программа CORA вычисляет дифференциальные сечения упруго и неупруго рассеянных альфа частиц, испускаемых составным ядром. Коэффициенты прохождения вычисляется при помощи четырё-параметрной оптической модели. Через автоматическое изменение параметров ρ и σ возможен подбор вычисленных сечений к экспериментальным.

1. Introduction.

The program CORA is written in ALGOL 1204 for the ODRA 1204 computer.

Section 2 gives the basic formalism of compound reaction cross-section calculations.

In section 3 the simplified block scheme of the program is presented and the data input is shown. The optical model procedure comes from reference [6].

Section 4 gives the data description.

Section 5 and 6 contain the results of the test run and listing of the program, respectively.

2. Formalism.

The energy averaged differential cross-section for compound reactions can be calculated following Hauser and Feschbach [1], by the use of optical model transmission coefficients.

Let an initial state be specified by $c = (\alpha, I, i, s, l, J_c)$. Here α represents the pair of interacting particles and their excitation states; I and i are the internal

spins of the nucleus and incoming particle, respectively;

s is the channel spin, which is given by

$$\vec{s} = \vec{I} + \vec{i}, \qquad /1/$$

l is the orbital angular momentum of the incoming particle; π_c is the channel parity, defined by

$$\pi_c = (-1)^l \pi_\alpha . \qquad /2/$$

The compound nucleus stete has the spin J , according to the relation

$$\vec{J} = \vec{l} + \vec{s} = \vec{l} + \vec{I} + \vec{i} \qquad /3/$$

The final state is denoted by $c' = (\alpha', I', i', s', l', \pi_c')$. Therefore the energy averaged differential cross-section is given by

$$\left\langle \frac{d\sigma_{\alpha\alpha'}}{d\Omega}(\theta) \right\rangle = \frac{1}{4k_\alpha^2 (2I+1)(2i+1)} \sum_{\substack{s l s' l' \\ J \pi}} W_{cc'} \frac{T_l(\alpha)\, T_{l'}(\alpha')}{\sum_{s'' l'' \alpha''} T_{l''}(\alpha'')} \qquad /4/$$

$$\times Z(lJlJ;sL)\, Z(l'Jl'J;s'L)\, (-1)^{s-s'} P_L(\cos\theta),$$

where k_α is the wave number in the entrance channel; T_l and $T_{l'}$ are the transmission coefficients, which are connected with phase shifts by the relation

$$T_l(\alpha) = 1 - |e^{2i\delta_{\alpha l}}|^2 . \qquad /5/$$

$W_{cc'}$ is the so-called "width correlation factor" [2]; Z is the coefficient defined by Blatt and Biedenharn [3];

$P_L(\cos\theta)$ is the Legendre polynomial of order L. The Z coefficients are defined as follows

$$Z(l_1 J_1 l_2 J_2; sL) = i^{L-l_1+l_2} [(2l_1+1)(2l_2+1)(2J_1+1)(2J_2+1)]^{1/2} \quad /6/$$
$$\times W(l_1 J_1 l_2 J_2 sL)(l_1 l_2 00|L0),$$

where W is the Racah coefficient. The Z coefficients are equal zero, when the sum l_1+l_2+L is odd, because of vanishing of the Clebsh-Gordan coefficient $(l_1 l_2 00|L0)$.

The main difficulty connected with the application of the formula /4/ is evaluation of the term

$$\sum_{S'' l'' \alpha''} T_{l''}(\alpha''). \quad /7/$$

This sum involves all open exit channels into which the compound nucleus can decay. Eberhard et al. [5] proposed the calculation of this sum on the basis of the statistical model [4]. This procedure gives the following result

$$\sum_{S'' l'' \alpha''} T_{l''}(\alpha'') = 2\pi \frac{\Gamma_0}{D_0} (2J+1)\, e^{-J(J+1)/(2\sigma_{res}^2(1+\omega_{res}))}. \quad /8/$$

Here Γ_0 and D_0 are the mean level width and the mean level spacing of compound nucleus states with spin $J=0$, respectively. The spin cutoff parameter σ_{res} represents an averaged value for the various residual nuclei, which are formed by the decay of the compound nucleus; ω_{res}

is a correction factor [5], approximated by $\omega_{res} \approx \frac{5}{3}\frac{A_\nu}{A_{res}}$. Here A_ν is the mass of the emitted particle and A_{res} is the mass of the corresponding residual nucleus.

Substituting /8/ into /4/ we obtain

$$\left\langle \frac{d\sigma_{\alpha\alpha'}}{d\Omega}(\theta) \right\rangle = \frac{1}{4\rho k_\alpha^2 (2I+1)(2i+1)} \sum_{\substack{Sls'l' \\ LJ\pi}} W_{cc'} \frac{T_l(\alpha) T_{l'}(\alpha')}{(2J+1)\exp[-J(J+1)/2\sigma^2]} \quad /9/$$

$$\times Z(lJlJ;sL)\, Z(l'Jl'J;s'L)(-1)^{s-s'} P_L(\cos(\theta)),$$

where $\sigma^2 = \sigma_{res}^2 (1+\omega_{res})$; $\rho = 2\pi \frac{\Gamma_e}{D_o}$.

The authors are deeply indebted to Professor A. Strzałkowski and Dr. L. Jarczyk for very helpful discussions.

References:

1. W.Hauser and H.Feshbach, Phys.Rev. 87 /1952/ 366,
H.Feshbach, in Nuclear Spectroscopy, Part B
/Academic Press, New York, 1960/ p. 625.

2. P.A.Moldauer, Phys.Rev. 123 /1961/ 968,
Phys.Letters 8 /1964/ 249,
Phys.Letters 19 /1967/ 1047,
Phys.Rev. 171 /1968/ 1164,
Phys.Rev. 177 /1969/ 1841.

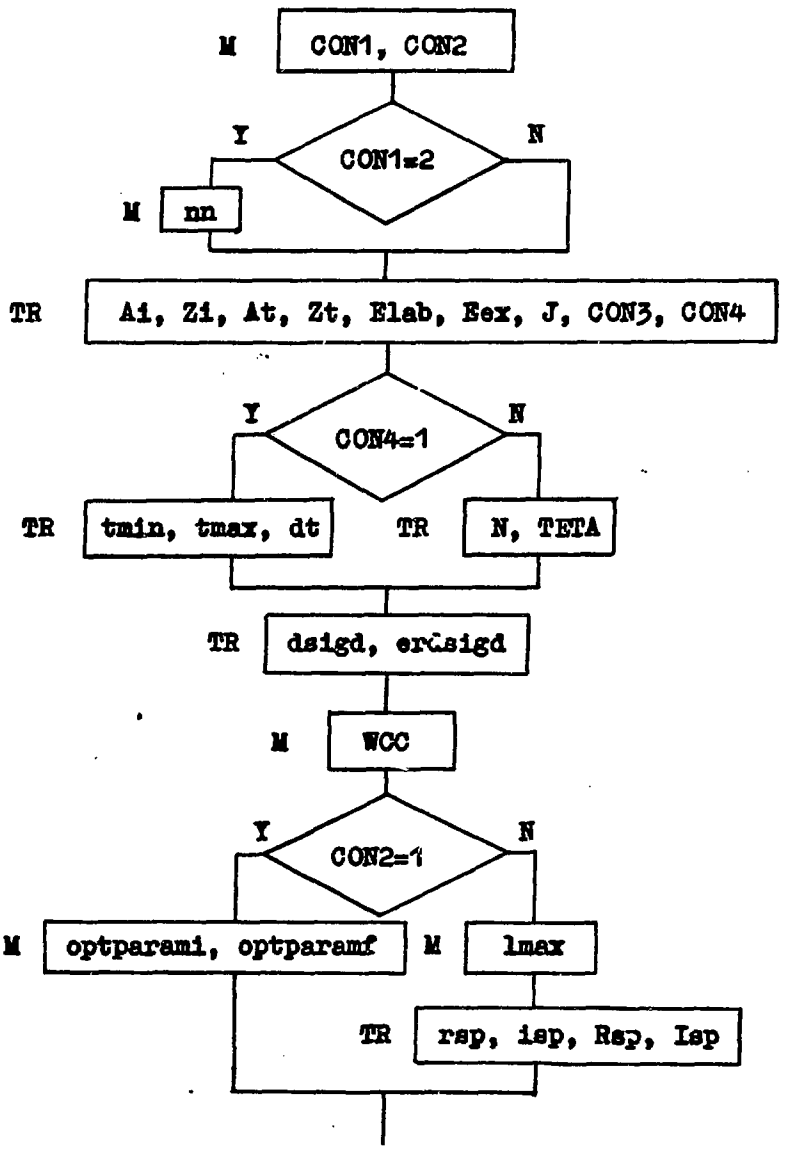
3. J.M.Blatt and L.C.Biedenharn, Rev.Mod.Phys. 24 /1952/
258.

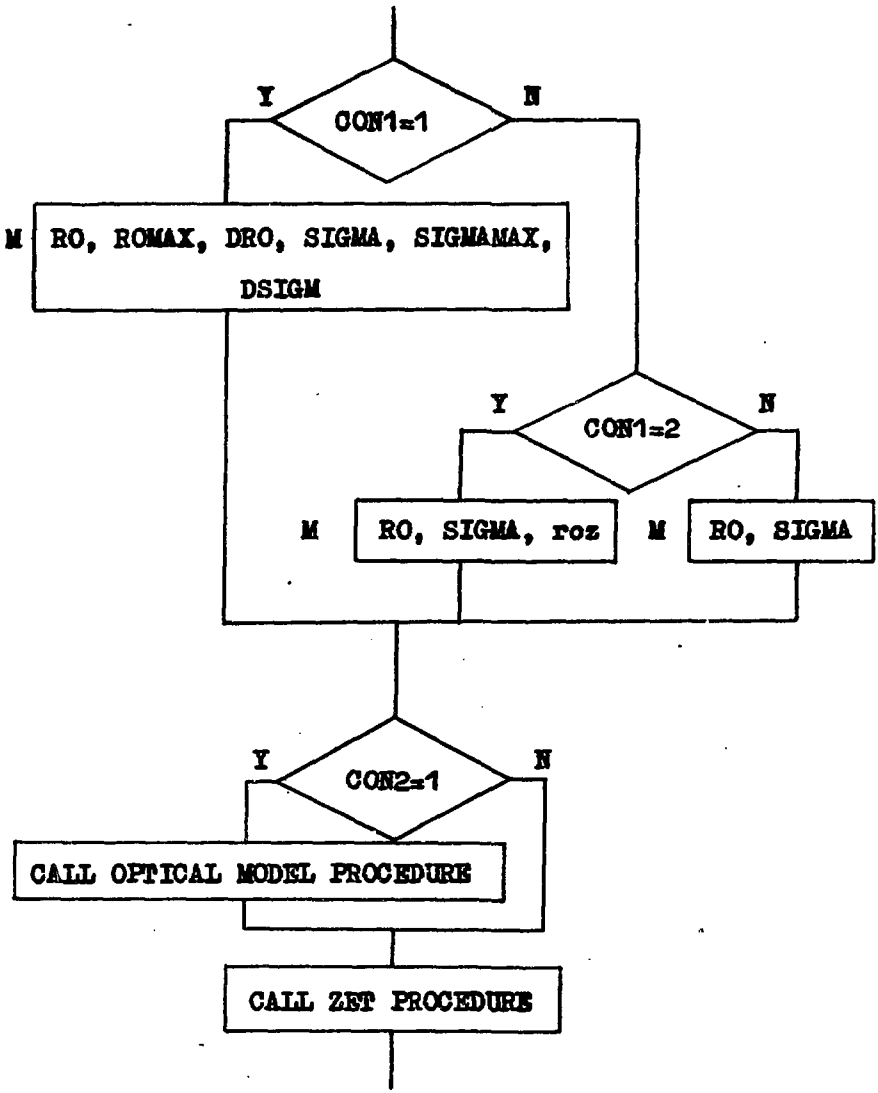
4. W. von Witsch, P. von Brentano, T.Mayer-Kuckuk and
A.Richter, Nucl.Phys. 80 /1966/ 394.

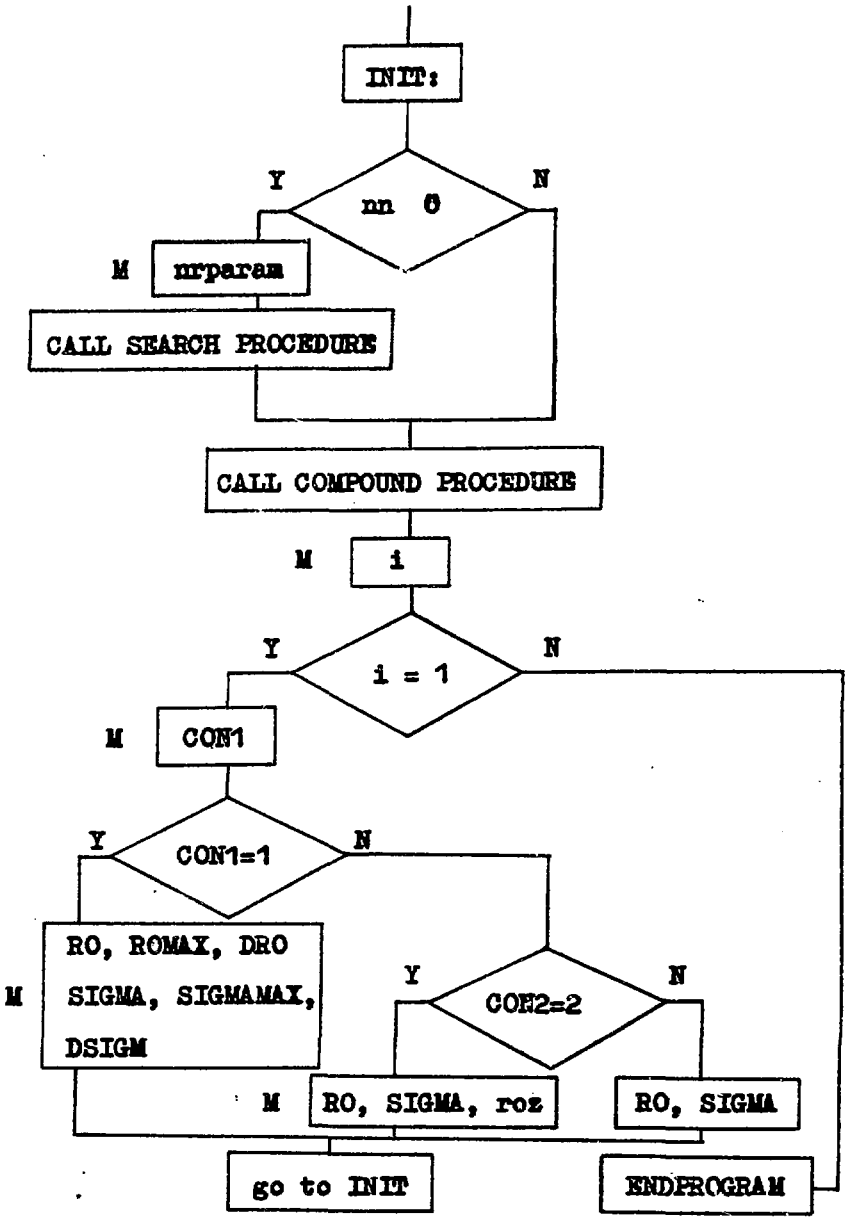
5. K.A.Eberhard, P. von Brentano, M.Böhring and
R.O.Stephen, Nucl.Phys. A125 /1969/ 673.

6. M.Ziemkiewicz, Report No. 741/PL, Cracow 1971.

3. Data Input and Program Scheme.




Y
CON1=1
N
M
RO, ROMAX, DRO, SIGMA, SIGMAMAX, DSIGM
Y
CON1=2
N
M
RO, SIGMA, roz
M
RO, SIGMA
Y
CON2=1
N
CALL OPTICAL MODEL PROCEDURE
CALL ZET PROCEDURE



M - monitor TR - tape reader

4. Input Data Description.

CON1
- 1 grid procedure is used
- 2 search procedure is used
- 3 one run for ρ and σ

CON2
- 1 transmission coefficients are calculated
- 2 transmission coefficients are read from tape

CON3
- 1 final state parity is odd
- 2 final state parity is even

CON4
- 0 regular step of the angles
- 1 irregular step of the angles

nn - number of parameters used in the search procedure / nn= 1 or 2 /

nrparam- the parameters, which are searched / if nrparam=1, then σ , if nrparam=2, then ρ /

Ai - mass
Zi - charge
— of the incident particle

At - mass
Zt - charge
— of the target nucleus

Elab - energy of the incoming particle / in laboratory system / [MeV]

Eex - excitation energy of the final state / in CM system / [MeV]

J - spin of the final state

tmin - minimal, tmax - maximal } value of the angle for which the angular distributions are calculated

dt - step of the angle

N - number of angles for which the experimental cross-section was measured

TETA - table of angles / TETA=TETA[i], where i=1,2, ... ,N

dsigd - table of experimental differential cross-section data / in CM system /

erdsigd - table of errors of the experimental data

optparami - optical model parameters: U_i, W_i, r_i, a_i for the entrance channel

optparamf - optical model parameters: U_f, W_f, r_f, a_f for the exit channel

lmax - maximum l-value of the calculated η_l coefficients

rsp - table of real parts of η_l coefficients for the entrance channel

isp - table of imaginary parts of η_l coefficients for the entrance channel

Rsp - table of real parts of η_l coefficients for the exit channel

Isp - table of imaginary parts of η_l coefficients for the exit channel

WCC - "width correlation factor"

RO - initial value of the ρ parameter

ROMAX - final value of the ρ parameter

DRO - step of ρ in the grid procedure

SIGMA - initial value of the σ parameter

SIGMAMAX - final value of the σ parameter

DSIGMA - step of σ in the grid procedure

roz - if the difference between two successive values of χ^2 is less then roz, search procedure is stopped

Description of the keys.

	key down	key up
KL 2	must be used during the search	search interrupts
KL 10	graph will be printed	without graph
KL 11	experimental and calculated cross-sections will be printed	without printing
KL 12	reflection and transmission coefficients will be printed	without printing

5. Test Run.

The following input data were used in the test run of the CORA program:

```
3,1,
'28Si  alpha, alpha+  28Si  BINGHAM EXPERIMENTAL DATA '
4,2,28,14,24,6.27,3,2,0,
25,157.5,2.5,
0.4675,0.4675,0.4,0.475,0.4,0.537,0.485,
0.375,0.375,0.415,0.34,0.355,0.242,0.3,
0.325,0.361,0.43,0.408,0.355,0.229,0.203,
0.26,0.271,0.286,0.287,0.265,0.265,
0.275,0.26,0.235,0.2,0.186,0.223,0.296,
0.341,0.333,0.341,0.35,0.282,0.235,
0.32,0.312,0.322,0.415,0.339,0.432,0.52,
0.595,0.575,0.582,0.89,0.91,0.93,1.01,

0.148,0.095,0.115,0.125,0.11,0.147,0.2,
0.165,0.094,0.085,0.06,0.08,0.076,0.098,
0.08,0.092,0.165,0.153,0.125,0.09,0.7,
0.04,0.038,0.061,0.08,0.047,0.135,
```

0.095,0.09,0.048,0.03,0.032,0.039,0.041,

0.046,0.044,0.045,0.085,0.095,0.1,

0.1,0.08,0.085,0.085,0.072,0.08,0.07,

0.08,0.115,0.122,0.13,0.11,0.11,0.101,

1,

50.875,10.501,1.699,0.505,

50.875,10.501,1.699,0.505,

1750,3.1,

The program requires one unit of magnetic drum storage. With the above input data the test run takes about 25 minutes of ODRA 1204 computer time. It depends strongly on maximum l - value used and the spin of the final state.

CORA-73 PROGRAM

INELASTIC SCATTERING OF ALPHA PARTICLES WITH HAUSER-FESHBACH FORMALISM

REACTION: 28Si (alpha, alpha+) 28Si BINGHAM EXPERIMENTAL DATA

INCIDENT PARTICLE ENERGY E= 24.000 MEV (LAB)

FINAL STATE ENERGY Eex= 6.270 MEV (CM)

SPIN AND PARITY J= 3(+)

OPTICAL MODEL PARAMETERS

ENTRANCE CHANNEL	U= 50.875	W=10.501	R=1.699	A= .505
EXIT CHANNEL	U= 50.875	W=10.501	R=1.699	A= .505

COMPOUND NUCLEUS PARAMETERS

RO=1.750$_{*}$03 SIGMA= 3.100

THETA	DCS THEOR	DCS EXP	ERR DCS EXP
29.13	4.63316$_{*}$-01	4.67500$_{*}$-01	1.47500$_{*}$-01
32.02	4.11138$_{*}$-01	4.67500$_{*}$-01	9.50000$_{*}$-02
34.89	3.72797$_{*}$-01	4.00000$_{*}$-01	1.15000$_{*}$-01
37.76	3.42573$_{*}$-01	4.75000$_{*}$-01	1.25000$_{*}$-01
40.61	3.18750$_{*}$-01	4.00000$_{*}$-01	1.10000$_{*}$-01
43.46	3.00584$_{*}$-01	5.37000$_{*}$-01	1.47000$_{*}$-01
46.29	2.86552$_{*}$-01	4.85000$_{*}$-01	2.00000$_{*}$-01
49.12	2.74595$_{*}$-01	3.75000$_{*}$-01	1.65000$_{*}$-01
51.93	2.63192$_{*}$-01	3.57000$_{*}$-01	9.40000$_{*}$-02
54.72	2.52021$_{*}$-01	4.15000$_{*}$-01	8.50000$_{*}$-02

57.51	2.41754$_{w}$-01	3.40000$_{w}$-01	6.00000$_{w}$-02
60.28	2.33287$_{w}$-01	3.55000$_{w}$-01	8.00000$_{w}$-02
63.03	2.26987$_{w}$-01	2.42000$_{w}$-01	7.60000$_{w}$-02
65.77	2.22432$_{w}$-01	3.00000$_{w}$-01	9.75000$_{w}$-02
68.49	2.18731$_{w}$-01	3.25000$_{w}$-01	8.00000$_{w}$-02
71.20	2.15113$_{w}$-01	3.61000$_{w}$-01	9.20000$_{w}$-02
73.89	2.11363$_{w}$-01	4.30000$_{w}$-01	1.65000$_{w}$-01
76.57	2.07829$_{w}$-01	4.08000$_{w}$-01	1.53000$_{w}$-01
79.22	2.05063$_{w}$-01	3.55000$_{w}$-01	1.25000$_{w}$-01
81.86	2.03397$_{w}$-01	2.29000$_{w}$-01	9.00000$_{w}$-02
84.48	2.02739$_{w}$-01	2.03000$_{w}$-01	7.00000$_{w}$-01
87.09	2.02670$_{w}$-01	2.60000$_{w}$-01	4.00000$_{w}$-02
89.67	2.02727$_{w}$-01	2.71000$_{w}$-01	3.80000$_{w}$-02
92.24	2.02688$_{w}$-01	2.86000$_{w}$-01	6.10000$_{w}$-02
94.78	2.02683$_{w}$-01	2.87000$_{w}$-01	8.00000$_{w}$-02
97.31	2.03097$_{w}$-01	2.65000$_{w}$-01	4.70000$_{w}$-02
99.82	2.04329$_{w}$-01	2.65000$_{w}$-01	1.35000$_{w}$-01

102.31	2.06542$_{10}$-01	2.75000$_{10}$-01	9.50000$_{10}$-02
104.78	2.09555$_{10}$-01	2.60000$_{10}$-01	9.00000$_{10}$-02
107.24	2.12943$_{10}$-01	2.35000$_{10}$-01	4.80000$_{10}$-02
109.67	2.16299$_{10}$-01	2.00000$_{10}$-01	3.00000$_{10}$-02
112.09	2.19496$_{10}$-01	1.86000$_{10}$-01	3.20000$_{10}$-02
114.48	2.22806$_{10}$-01	2.23000$_{10}$-01	3.90000$_{10}$-02
116.86	2.26784$_{10}$-01	2.96000$_{10}$-01	4.10000$_{10}$-02
119.22	2.31990$_{10}$-01	3.41000$_{10}$-01	4.60000$_{10}$-02
121.57	2.38697$_{10}$-01	3.33000$_{10}$-01	4.40000$_{10}$-02
123.89	2.46755$_{10}$-01	3.41000$_{10}$-01	4.50000$_{10}$-02
126.20	2.55675$_{10}$-01	3.15000$_{10}$-01	8.50000$_{10}$-02
128.49	2.64901$_{10}$-01	2.82000$_{10}$-01	9.50000$_{10}$-02
130.77	2.74140$_{10}$-01	2.35000$_{10}$-01	1.00000$_{10}$-01
133.03	2.83589$_{10}$-01	3.20000$_{10}$-01	1.00000$_{10}$-01
135.28	2.93953$_{10}$-01	3.12000$_{10}$-01	8.00000$_{10}$-02
137.51	3.06212$_{10}$-01	3.22000$_{10}$-01	8.50000$_{10}$-02
139.72	3.21269$_{10}$-01	4.15000$_{10}$-01	8.50000$_{10}$-02

141.93	$3.39657_{10}-01$	$3.92000_{10}-01$	$7.20000_{10}-02$
144.12	$3.61589_{10}-01$	$4.32000_{10}-01$	$8.00000_{10}-02$
146.29	$3.87452_{10}-01$	$5.20000_{10}-01$	$7.00000_{10}-02$
148.46	$4.18600_{10}-01$	$5.95000_{10}-01$	$8.00000_{10}-02$
150.61	$4.57976_{10}-01$	$5.75000_{10}-01$	$1.15000_{10}-01$
152.76	$5.09883_{10}-01$	$5.82000_{10}-01$	$1.22200_{10}-01$
154.89	$5.78420_{10}-01$	$8.90000_{10}-01$	$1.30000_{10}-01$
157.02	$6.64633_{10}-01$	$9.10000_{10}-01$	$1.10000_{10}-01$
159.13	$7.63281_{10}-01$	$9.30000_{10}-01$	$1.10000_{10}-01$
161.24	$8.60740_{10}-01$	$1.01000_{10}00$	$1.01000_{10}-01$

$CHISQ=8.414295_{10}01$

```
begin
 integer Zi,Zt,CON1,CON2,CON3,CON4,J,lmax,l,nn,N,nn1;
 real Ai,At,Elab,tmin,tmax,dt,Eex,RO,ROMAX,DRO,SIGMA,
 SIGMAMAX,DSIGM,WCC,roz;
 array param[1:2],optparami,optparamf[1:4],isp,rsp,
Isp,Rsp[0:30],TETA[1:100];
 integer array text[1:50];
 boolean el,druk;
 setinput(0);
 read(CON1,CON2);
 if CON1=2
  then read(nn)
  else
  begin
   nn=0;
   nn1=nn+1;
  end;
 setinput(1);
 instring(text[1]);
 read(Ai,Zi,At,Zt,Elab,Eex,J,CON3,CON4);
 if CON4=0
  then
  begin
   read(tmin,tmax,dt);
   N=entier((tmax-tmin)/dt+1.1);
  end
```

```
 else
 begin
  read(N);
  read(TETA);
 end;
begin
 array dsigd,erdsigd[1:N],EXA[1:N,-1:nn],RA[0:30,0:30];
 integer m,i,L,l1,zapis;
 real rspl,Rspl,Ispl,ispl,ra,rb,ib,chisq;
 real procedure ZET(a,b,c,d,e,f);
  integer a,b,c,d,e,f;
  drum;
 procedure COMPOUND(chisq);
  real chisq;
  begin
   integer pp1,pp2;
   real k,q,f,pl,plp,plm,dce,ia,a,a2,b,cx,Wns,pomo,
        gamma;
   array B[0:30];
   procedure dlaF;
    if b≥100
     then pp1=100
     else
     begin
      pp1=0;
      for m=50 step -25 until 0 do
```

```
        if b≥1
         then
         begin
          pp1=entier(ln(b)/2.3026×25)+m;
          go to cd
         end
         else b=b×10;
cd:   end;
    procedure dlaW;
     print('
w-2                          w-1                          w0
w1                           w2?');
    procedure dlaV;
     print('
+-------------------------+-------------------------+
-------------------------+-------------------------+?');
    b=At/(Ai+At);
    gamma=sqrt(Ai×Ai×Elab/(At×(At×Elab-(At+Ai)×Eex)));
    k=sqrt(Ai×Elab)×b×.2187;
    if drukAkey(11)
     then
     begin
      print('??
---THETA------DCS-THEOR----DCS-EXP-------ERR-DCS-EXP');
      line(2);
     end;
```

```
q=2.5/(k×k×param[2])×WCC;
pomo=.5/(param[1]×param[1]);
for L=0 step 2 until 2×lmax do
 begin
  dce=0;
  for l=0 step 1 until lmax do
   begin
    dce=dce+RA[l,L+2]×exp(l×(l+1)×pomo)/(2×l+1);
   end l;
  B[L+2]=dce×(-1)↑J;
 end L;
chisq=0;
for i=1 step 1 until N do
 begin
  if CON4=1
   then a=tmin+(i-1)×dt
   else a=TETA[i];
  a=tmin+(i-1)×dt;
  a=.01745329×a;
  a2=sin(a);
  a=arctan(gamma×a2/sqrt(1-gamma×gamma×a2×a2))+a;
  a2=sin(a);
  if drukAkey(11)
   then
   begin
    format('␣␣111.11');
```

```
        print('?',a/.01745329);
       end;
      cx=cos(a);
      ia=0;
      pl=0;
      plm=0;
      for L=0 atan 2 until 2*lmax do
       begin
        if L=0
         then
         begin
          plp=1;
          go to C;
         end;
        a=1/L;
        b=1/(L+1);
        plm=pl;
        pl=plp;
        plp=(2-a)*cx*pl+(a-1)*plm;
C;      ia=ia+B[L/2]*plp;
        plm=pl;
        pl=plp;
        plp=(2-b)*cx*pl+(b-1)*plm;
        if L=0
         then
         begin
```

```
        pl=1;
        plp=ox;
       end;
     end L;
    EXA[i,zapis]=Wns=ix*q;
    chisq=(abs(Wns-dsigd[i])/erdsigd[i])↑2+chisq;
    if druk^key(11)
     then
     begin
      format('␣␣␣1.12345␣12');
      print(Wns,dsigd[i],erdsigd[i]);
     end;
  end i;
 if druk^key(11)
  then
  begin
   line(1);
   format('?␣␣␣␣␣␣␣␣␣␣␣␣CHISQ=1.123456␣12');
   print(chisq);
   print('?␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣');
   line(2);
  end;
 if druk^key(10)
  then
  begin
   print('?␣␣␣␣␣␣␣␣␣␣␣␣␣EXPERIMENT␣␣␣␣␣␣␣␣THEORY');
```

```
line(1);
dlaW;
dlaV;
for i=1 step 1 until N do
 begin
  b=dsigd[i];
  dlaF;
  pp2=pp1;
  b=EXA[i,zapis];
  dlaF;
  line(1);
  print('.');
  if pp1≤pp2
   then
   begin
    space(pp1);
    print('x');
    pp2=pp2-pp1-1;
    if pp2≥0
     then
     begin
      space(pp2);
      print('o');
     end
   end
   else
```

```
      begin
       space(pp2);
       print('o');
       space(pp1-pp2-1);
       print('x');
      end
    end i;
   dlaV;
   dlaW;
   line(1);
   format('?    CHISQ=1.123456_12');
   print(chisq);
   line(2);
  end wykres;
 if druk^key(12)
  then
  begin
   print('?                        REFLECTION AND TR
         ANSMISSION COEFFICENTS');
   line(2);
   print('?          ENTRANCE CHANNEL                 
         .....EXIT CHANNEL');
   print('??  I    RE ETA I      IM ETA I      TL     
         .....RE ETA I      IM ETA I      TL');
  line(1);
   format('?11  1.12345_12  1.12345_12  1.12345_12
```

```
␣␣␣-1.12345₁₀12␣␣-1.12345₁₀12␣␣1.12345₁₀12');
      for l:=0 step 1 until lmax do
       begin
        rspl:=rsp[l];
        ispl:=isp[l];
        Rspl:=Rsp[l];
        Ispl:=Isp[l];
        print(l,rspl,ispl,(1-rspl×rspl-ispl×ispl),Rspl,
              Ispl,(1-Rspl×Rspl-Ispl×Ispl));
       end l;
    end ETA L;
  end COMPOUND;
 procedure OPTICALMODEL(lmax,Zi,Zt,Ai,At,Elab,Q,rsp,isp,
                        param,el);
  integer lmax,Zi,Zt;
  real Q,Elab,Ai,At;
  array rsp,isp,param;
  Boolean el;
  drum;
 read(dsigd,erdsigd);
 setinput(0);
 read(WCC);
 if CON2=1
  then
  begin
   read(optparami);
```

```
    read(optparamf);
  and
  else
  begin
   read(lmax);
   setinput(1);
   for l:=0 step 1 until lmax do
    read(rsp[l],isp[l]);
   for l:=0 step 1 until lmax+J do
    read(Rsp[l],Isp[l]);
  end;
if CON1=1
 then
 begin
  setinput(0);
  read(RO,ROMAX,DRO);
  read(SIGMA,SIGMAMAX,DSIGM);
 end;
setinput(0);
if CON1=2
 then read(RO,SIGMA,ros);
print('?----------------------------------------CORA-
      73 PROGRAM');
if CON1=3
 then read(RO,SIGMA);
```

```
param[1]=SIGMA;
param[2]=RO;
print('??␣␣␣␣␣␣␣␣␣␣␣␣INELASTIC␣␣SCATTERING␣␣OF␣␣ALPHA
      ␣␣PARTICLES␣␣WITH␣␣HAUSER-FESHBACH␣FORMALISM');
line(2);
print('?␣␣␣␣␣␣␣␣␣␣␣␣REACTION:');
outstring(text[1]);
format('111.111');
print('?␣␣␣␣␣␣␣␣␣␣␣␣INCIDENT␣PARTICLE␣ENERGY␣E=',
      Elab,'␣␣MEV␣(LAB)');
print('?␣␣␣␣␣␣␣␣␣␣␣␣FINAL␣STATE␣ENERGY␣␣␣␣␣␣␣Eex=',
      Eex,'␣MEV␣(CM)');
format('11');
print('?␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣SPIN␣AND␣PARITY␣J=',J);
if CON3=2
 then print('(+)')
 else print('(-)');
line(2);
print('?␣␣␣␣␣␣␣␣␣␣␣␣OPTICAL␣MODEL␣PARAMETERS');
line(1);
format('?␣␣␣␣␣␣␣␣␣␣␣␣ENTRANCE␣CHANNEL␣␣U=111.111␣␣␣
       W=11.111␣␣␣R=1.111␣␣␣A=1.111');
print(optparami[1],optparami[2],optparami[3],
      optparami[4]);
format('?␣␣␣␣␣␣␣␣␣␣␣␣EXIT␣CHANNEL␣␣␣␣␣␣U=111.111␣␣␣
       W=11.111␣␣␣R=1.111␣␣␣A=1.111');
```

```
print(optparamf[1],optparamf[2],optparamf[3],
      optparamf[4]);
line(2);
if CON1=2 V CON1=3
 then
 begin
  print('?~~~~~~~~~~~~COMPOUND~NUCLEUS~PARAMETERS');
  line(1);
  format('?~~~~~~~~~~~~RO=1.111~11~~~~SIGMA=11.111');
  print(RO,SIGMA);
 end;
if CON2=1
 then
 begin
  el=true;
  OPTICALMODEL(lmax,Zi,Zt,Ai,At,Elab,.0,rsp,isp,
               optparami,el);
  el=false;
  OPTICALMODEL(lmax+J,Zi,Zt,Ai,At,Elab,-Eex,Rsp,Isp,
               optparamf,el);
 end CON2;
Elab=Elab+Eex*(Ai+At)/At;
for L=0 step 2 until 2*lmax do
 for l=0 step 1 until lmax do
  begin
   ra=0;
```

```
      rspl=rsp[l];
      ispl=isp[l];
      if L≤2×l
       then rb=ZET(l,l,l,l,C,L);
      for l1=abs(l-J) step 1 until l+J do
       begin
        Rspl=Rsp[l1];
        Ispl=Isp[l1];
        if l<l1∧L>2×l∨l≥l1∧L>2×l1
         then go to D;
        if ((l+l1)/2=(l+l1)÷2∧CON3=1)∨((l+l1)/2≠(l+l1)÷2
            ∧ CON3=2)
         then go to D;
        ib=ZET(l1,l,l1,l,J,L);
        ra=ra+rb×ib×(1-Rspl×Rspl-Ispl×Ispl);
D:     end L1;
      RA[l,L+2]=ra×(1-rspl×rspl-ispl×ispl);
     end l;
INIT:
  if CON1=1
   then
   begin
    druk:=false;
    zapis:=-1;
    line(5);
    print('?␣␣.....␣GRID␣PROCEDURE␣CORA-73␣.....');
```

```
   line(2);
   print('?    CHISQ       RO       SIGMA');
   line(2);
   for param[1]:=SIGMA step DSIGM until SIGMAMAX do
    begin
     format('                        11.111');
     print('?',param[1]);
     for param[2]:=RO step DRO until ROMAX do
      begin
       COMPOUND(chisq);
       format('?   1.11  11   1.111  11');
       print(chisq,param[2]);
      end 2;
     line(4);
    end 1;
   line(2);
   print('      ..........  END PROCEDURE   ...   CORA ');
   go to RETURN;
  end;
 if CON1=3
  then
  begin
   druk:=true;
   zapis:=1;
   COMPOUND(chisq);
   go to RETURN;
```

```
    and;
   if nn>0
    then
    begin
     integer ll,it,j;
     real F1,F2,F3,aa,z,z2,zm,ZL,FL,a,b;
     array paramzm,popr[1:nn];
     integer array nrparam[1:nn];
     for i=1 step 1 until nn do
      begin
       read(nrparam[i]);
      paramzm[i]=param[nrparam[i]];
      and;
     print('?nr-param?',nrparam);
     line(1);
     druk=false;
   it=zapis=0;
    COMPOUND(F1);
POCZ:
    for zapis=1 step 1 until nn do
     begin
      aa=paramzm[zapis];
      if aa#0
       then
       begin
        ll=nrparam[zapis];
```

```
      param[11]=aa×1.001;
      COMPOUND(chisq);
      param[11]=aa;
      aa=.001×aa;
      for j=1 step 1 until N do
       EXA[j,zapis]=(EXA[j,zapis]-EXA[j,0])/aa;
     end
     else
     for j=1 step 1 until N do
      EXA[j,zapis]=0
  end;
zapis=-1;
begin
 array MAC[1:nn,1:nn1];
 for i=1 step 1 until nn do
  begin
   for j=i step 1 until nn do
    begin
     aa=0;
     for l=1 step 1 until N do
      aa=EXA[l,i]×EXA[l,j]/erdsigd[l]↑2+aa;
     MAC[j,i]=MAC[i,j]=aa
    end;
   aa=0;
   for l=1 step 1 until N do
    aa=(dsigd[l]-EXA[l,0])×EXA[l,i]/erdsigd[l]↑2+aa;
```

```
    MAC[i,nn1]:=aa
  end;
begin
  integer array p[1:nn1];
  integer ph,pj;
  for i:=1 step 1 until nn1 do
   p[i]:=i;
  for l:=1 step 1 until nn do
   begin
    ll:=l+1;
    a:=0;
    for i:=l step 1 until nn do
     begin
      b:=abs(MAC[l,p[i]]);
      if b>a
       then
       begin
        a:=b;
        j:=i
       end
     end;
    if a>=-5
     then
     begin
      ph:=p[j];
      a:=MAC[l,ph];
```

```
      p[j]=p[l];
      p[l]=ph;
      for j=ll step 1 until nn1 do
       begin
        pj=p[j];
        b=MAC[l,pj]=MAC[l,pj]/a;
        for i=ll step 1 until nn do
         MAC[i,pj]=MAC[i,pj]-MAC[i,ph]×b
       end
     end
     else
     for j=ll step 1 until nn1 do
      MAC[l,p[j]]=0;
   end l;
  for l=nn step -1 until 1 do
   begin
    a=MAC[l,nn1];
    for j=l+1 step 1 until nn do
     begin
      pj=p[j];
      a=a-MAC[l,pj]×popr[pj]
     end;
    popr[p[l]]=a
   end
 end
end MAC;
```

```
 a=0;
for i=1 step 1 until nn do
 begin
  b=paramzm[i];
  if b≠0
   then
   begin
    b=abs(popr[i]/b);
    if b>a
     then a=b
   end
 end;
z=if a<.1 then 1 else if a<.3 then .5 else if a<.5
   then .2 else if a<1 then .1 else .05/a;
for i=1 step 1 until nn do
 param[nrparam[i]]=paramzm[i]+popr[i]×z;
COMPOUND(F2);
z2=if F1>F2 then z+z else .5×z;
for i=1 step 1 until nn do
 param[nrparam[i]]=paramzm[i]+popr[i]×z2;
COMPOUND(F3);
if F1>F2
 then
 begin
  Z1=z;
  F1=F2;
```

```
      for i:=1 step 1 until M do
       EXA[i,0]:=EXA[i,-1];
      if F3≥F2
       then go to PARABOLA
       else zw:=z+z2
     end
     else
     begin
      ZL:=0;
      FL:=F1;
      if F3≥F1+(F2-F1)×.5
       then
       begin
        if F1=F3
         then zw:=z2
         else zw:=-z2
       end
       else go to PARABOLA
     end;
    go to PARABOLI;
PARABOLA:
    a:=(F3-F2)×z×z-(F2-F1)×(z2×z2-z×z);
    b:=(F2-F1)×(z-z2)+(F3-F2)×z;
    if b≠0
     then zw:=.5×a/b
     else
```

```
      begin
       print('parab.niemoz');
       zm=z2
      end;
     if zm<-z
      then zm=-z;
PARABOLI:
     for i=1 step 1 until nn do
      param[nrparam[i]]=paramzm[i]+popr[i]*zm;
     if F3<FL
      then
      begin
       FL=F3;
       ZL=z2;
       for i=1 step 1 until N do
        EXA[i,0]=EXA[i,-1];
      end;
     COMPOUND(chisq);
     it=it+1;
     if chisq<FL
      then
      begin
       FL=chisq;
       ZL=zm;
       for i=1 step 1 until N do
        EXA[i,0]=EXA[i,-1]
```

```
    end;
   for i:=1 step 1 until nn do
    paramzm[i]:=param[nrparam[i]]:=paramzm[i]+popr[i]*ZL;
   format('?it=11    chisq=1.12345,12');
   print(it,chisq);
   format('?  search-param-=11111.11');
   print(paramzm);
   line(1);
   if abs(F1-FL)>rez^key(2)
    then
    begin
     F1:=FL;
     go to POCZ
    end
  end nn;
 druk:=true;
 zapis:=-1;
 COMPOUND(chisq);
RETURN:
 wait('?    GO-ON---1    END---0');
 setinput(0);
if ininteger=0
  then go to ENDPROG;
 read(CON1);
 if CON1=1
  then
```

```
    begin
     read(RO,ROMAX,DRO);
     read(SIGMA,SIGMAMAX,DSIGM);
    end;
   if CON1=2
    then read(RO,SIGMA,roz);
   if CON1=3
    then
    begin
     read(RO,SIGMA);
    end;
   if CON1=2 V CON1=3
    then
    begin
     print('?____________COMPOUND_NUCLEUS_PARAMETERS');
     line(1);
     format('?____________RO=1.111,11____SIGMA=11.111');
     print(RO,SIGMA);
    end;
   param[1]=SIGMA;
   param[2]=RO;
   go to INIT;
ENDPROG:
 end
end
```

```
procedure OPTICALMODEL(lmax,Zi,Zt,Ai,At,Elab,Q,rsp,isp,
          param,el);
 integer lmax,Zi,Zt;
 real Q,Elab,Ai,At;
 array rsp,isp,param;
 boolean el;
 begin
  integer l,m,i,uu,lmax1,p1,p2,pp1,pp2,lmax3;
  real k,gamma,chisq,rhomax,EE,Un,Wn,r1,a1,k2,a,b,c,d,e,
       f,pom,ro,ri,pomo,rc,rck,gamma2,r3,r4,Wns;
  a=At↑.333333333;
  b=At/(Ai+At);
  EE=Elab×b+Q;
  Elab=EE/b;
  k=sqrt(Ai×Elab)×b×.2187;
  gamma=sqrt(Ai/Elab)×Zi×Zt×.15745;
  gamma2=gamma+gamma;
  Un=param[1]/EE;
  r1=param[3]×a;
  a1=param[4];
  Wn=param[2]/EE;
  ro=1.34;
  ri=.2;
  rc=ro×a;
  rck=rc×k;
  pomo=gamma/(rck×rck×rck);
```

```
pom=1-3×gamma/rck;
a=Un;
d=Wn;
e=r1;
f=a1;
rhomax=if (k×(e+f×ln(a)))≥(k×(e+f×ln(d)))
       then k×(e+f×ln(a))+8.059047825
       else k×(e+f×ln(d))+8.059047825;
if el
 then lmax=entier(rhomax)+2;
lmax1=lmax+1;
lmax3=lmax1;
pp1=lmax3+lmax3;
pp2=lmax3+1;
begin
 array RS,IS[1:pp1];
 procedure CDIV(a,b,c,d,e,f);
  value c,d,e,f;
  real a,b,c,d,e,f;
  begin
   b=e×e+f×f;
   if b=0
    then wait('CDIV imposs');
   a=(c×e+d×f)/b;
   b=(d×e-c×f)/b;
  end;
```

```
procedure CMUL(a,b,c,d,e,f);
 value c,d,e,f;
 real a,b,c,d,e,f;
 begin
  a=c×e-d×f;
  b=d×e+c×f
 end;
begin
 real RVC,IVC,e1,e2,ee1,ee2,eh1,eh2,r,phA,pIA,hr,hc,
      hc1,ri2,rik,ri12,aa,bb,s,chisq2,cc,grck;
 array pRS,pIS[1:pp1];
 procedure FUN;
  begin
   e1=e1×eh1;
   hc=hc+hc1;
   chisq=chisq+rik;
   chisq2=chisq×chisq;
   RVC=Un/(1+e1);
   IVC=Wn/(1+e1);
   s=RVC+1-(if r≥rc then gamma2/chisq else
     (3-hc×hc)×grck)
  end FUN;
 hc=chisq=0;
 hc1=ri/rc;
 rik=ri×k;
 ri2=rik×rik;
```

```
ri12=ri2/12;
grck=gamma/rok;
eh1=exp(ri/a1);
e1=ee1=exp(-r1/a1);
begin
 real Ra2,Ia2;
 m=1;
 for r=ri,ri+ri do
  begin
   FUN;
   b=pom+RVC;
   a=2;
   c=6;
   d=12;
   e=20;
   f=chisq;
   for l=0 step 1 until lmax do
    begin
     cc=a-(l+1)×l/chisq2;
     Ra2=-b/c;
     Ia2=-IVC/c;
     aa=(((((IVC×Ia2-b×Ra2-pomo)/e×chisq)×chisq+
        Ra2)×chisq)×chisq+1)×f;
     bb=(((((-IVC×Ra2-b×Ia2)/e×chisq)×chisq+Ia2)×
        chisq)×chisq)×f;
     RS[m]=pRA=cc;
```

```
      IS[m]:=pIA:=IWC;
      CMUL(pRS[m],pIS[m],1+pRA×ri12,pIA×ri12,ea,bb);
      m:=m+1;
      a:=a+2;
      c:=c+4;
      d:=d+6;
      e:=e+8;
      f:=f×chisq;
     end 1
   end r
end;
begin
 integer n,n3,n6,n1,n2,n4,n5;
 n3:=entier(rhomax/rik);
 n:=n3+3;
 n6:=n-6;
 n5:=n-5;
 n4:=n-4;
 n2:=n-2;
 n1:=n-1;
 rhomax:=n3×rik;
 begin
  array y,iy[pp2:pp1,n6:n];
  real procedure dlaS(y);
   array y;
   dlaS:=((y[m,n]-y[m,n6])/60+(y[m,n5]-y[m,n1])×.15+
```

```
        (y[m,n2]-y[m,n4])×.75)/r1k;
for i=3 step 1 until n do
 begin
  r=r+r1;
  FUN;
  p1=1;
  p2=pp2;
  for l=0 step 1 until lmax do
   begin
    cc=s-(l+1)×l/chisq2;
    e=pRS[p2];
    f=pIS[p2];
    CDIV(a,b,RS[p2]×r12,IS[p2]×r12,RS[p2]×r112+1,
         IS[p2]×r112);
    RS[p2]=pRA=cc;
    IS[p2]=pIA=lVC;
    CMUL(c,d,2-a,-b,e,f);
    pRS[p2]=c-pRS[p1];
    pIS[p2]=d-pIS[p1];
    pRS[p1]=e;
    pIS[p1]=f;
    if i≥n6
     then
     begin
      CDIV(y[p2,i],iy[p2,i],pRS[p2],pIS[p2],1+
           pRA×r112,pIA×r112);
```

```
            if i=n3
            then
            begin
             RS[p1]=y[p2,1];
             IS[p1]=iy[p2,i]
            end
          end;
         p1=p1+1;
         p2=p2+1;
        end 1
      end 1;
     for m=pp2 step 1 until pp1 do
      begin
       RS[m]=dlaS(y);
       IS[m]=dlaS(iy);
      end
    end
   end
end RVC;
begin
 real sigma0,sig2,alpha,FF,Fd,GG,Gd,RA,IA,RB,IB,RC,IC,
 RD,ID,RE,IE;
 array t[0:lmax+10];
 procedure SLJ(RA,IA);
  real RA,IA;
  begin
```

```
        p1=p1+1;
        p2=p2+1;
        CDIV(a,b,RS[p2],IS[p2],RS[p1],IS[p1]);
        CMUL(c,d,a,b,FF,GG);
        CMUL(e,f,a,b,FF,-GG);
        CDIV(RA,IA,Fd-c,Gd-d,e-Fd,f+Gd);
      end:
    begin
      real rhom,sig,rho,eta,sto;
      rho=rhomax;
      eta=gamma;
      sto=16+eta×eta;
      sig=-eta+eta/2×ln(sto)+3.5×arctan(eta/4)-(arctan(eta)
          +arctan(eta/2)+arctan(eta/3))-eta/(12×sto)×
          (1+1/30×(eta↑2-48)/(sto×sto)+1/105×(eta↑4-
          160×eta×eta+1280)/(sto×sto×sto×sto));
      if (rho≤175/12∧eta≤4)∨rho≥(eta×eta+4×eta+3)×5/12
      then
       begin
        real sx,ls,tx,lt,ss,sl,tt,tl,ns,nt,sn,tn,an,bn,
             theta,cth,sth;
        if rho≥(eta×eta+4×eta+3)×5/12
         then rhom=rho
         else rhom=(eta×eta+4×eta+3)×5/12;
MM:     sx=ns=1;
        tx=nt=0;
```

```
ss=sn=0;
tt=tn=1-eta/rhom;
for i=0 step 1 until 10,11,i+1
while (abs(ns)>=7×abs(sx)∨abs(nt)>=7×abs(tx)∨
       abs(sn)>=7×abs(ss)∨abs(tn)>=7×abs(tt))∧
       (abs(ns)<abs(ls)∧abs(nt)<abs(lt)∧
 abs(sn)<abs(sl)∧abs(tn)<abs(tl)) do
 begin
  an=(2×i+1)×eta/(2×(i+1)×rhom);
  bn=(eta×eta-i×(i+1))/(2×(i+1)×rhom);
  ls=ns;
  lt=nt;
  sl=sn;
  tl=tn;
  ns=an×ls-bn×lt;
  nt=an×lt+bn×ls;
  sn=an×sl-bn×tl-ns/rhom;
  tn=an×tl+bn×sl-nt/rhom;
  sx=sx+ns;
  tx=tx+nt;
  ss=ss+sn;
  tt=tt+tn;
 end;
if abs(sx×tt-ss×tx-1)>=5
 then
 begin
```

```
      rhom=rhom+2.5;
      go to MM;
     end;
    theta=-eta*ln(2*rhom)+rhom+sig;
    cth=cos(theta);
    sth=sin(theta);
    GG=sx*cth-tx*sth;
    Gd=ss*cth-tt*sth
  end
  else
  begin
   GG=1.22340416*etaf(1/6)*(1+0.0495957017/etaf(4/3)-
      0.0088888889/(eta*eta)+0.00245519918/etaf(10/3)-
      0.00091089580 6/(eta*eta*eta*eta)+
      0.0002534684/etaf(16/3));
   Gd=-0.707881773*etaf(-1/6)*(1-0.172826037/etaf(2/3)
      +0.000317460317/(eta*eta)-0.00358121485/
      etaf(8/3)+0.000311782468/(eta*eta*eta*eta)-
      0.000907396643/etaf(14/3));
   rhom=2*eta
  end;
 if rhom<rho
  then
  begin
   integer nh;
   real x,y,yd,h,hh,d,k0,k1,k2,k3;
```

```
    nh=entier(abs(rhom-rho)×10+1);
    h=(rho-rhom)/nh;
    hh=h/2;
    x=rhom;
    y=GG;
    yd=Gd;
    d=hh×(2×eta/x-1);
    for i=1 step 1 until nh do
     begin
      k1=d×y;
      k0=hh×(yd+k1/2);
      x=x+hh;
      d=hh×(2×eta/x-1);
      k2=d×(y+k0);
      k0=h×(yd+k2);
      x=x+hh;
      d=hh×(2×eta/x-1);
      k3=d×(y+k0);
      y=y+h×(yd+(k1+2×k2)/3);
      yd=yd+(k1+4×k2+k3)/3
     end;
    GG=y;
    Gd=yd
  end;
m=lmax+10;
begin
```

```
   real fd0;
   integer w;
   t[m]=0;
   t[m-1]=1;
   for w=m-1 step -1 until 1 do
    t[w-1]=w/sqrt(eta×eta+w×w)×(((2×w+1)×eta/(w×(w+1))
           +(2×w+1)/rho)×t[w]-sqrt(eta×eta+(w+1)×(w+1))
           /(w+1)×t[w+1]);
   fd0=(eta+1/rho)×t[0]-sqrt(eta×eta+1)×t[1];
   a=(-Gd+(1/rho+eta)×GG)/sqrt(1+eta×eta);
   alpha=1/(sqrt(1+eta×eta)×(t[0]×a-t[1]×GG));
   FF=alpha×t[0];
   Fd=alpha×fd0;
  end;
  sigg22=sigma0=sigg+sigg;
end Coulomb;
p1=0;
p2=lmax3;
SLJ(RA,IA);
rsp[0]=RA;
isp[0]=IA;
k2=gamma×gamma;
for l=1 step 1 until lmax do
 begin
  c=FF;
  d=GG;
```

```
      FF=alpha*t[1];
      a=sqrt(k2+l*l)/l;
      b=gamma/l+1/rhomax;
      Fd=a*c-b*FF;
      GG=(b*GG-Gd)/a;
      Gd=a*d-b*GG;
      SLJ(RA,IA);
      rsp[1]=RA;
      isp[1]=IA;
    end l;
  end
 end
end OPTICALMODEL;
```

```
real procedure ZET(a,b,c,d,e,f);
 integer a,b,c,d,e,f;
 begin
  integer k,g,gl;
  real x,ISJS;
  real procedure SJS(J1,J2,J3,L1,L2,L3);
   value J1,J2,J3,L1,L2,L3;
   integer J1,J2,J3,L1,L2,L3;
   begin
    real KAPPA,PH,SJ;
    integer IWMIN,IWMAX,IW,IW1,IW2,IW3,IW4,IW5,IW6,IW7,IW8;
    boolean cond;
    cond:=if J1+J2<J3Vabs(J1-J2)>J3VJ1+L2<L3Vabs(J1-L2)>L3
            VL1+J2<L3Vabs(L1-J2)>L3VL1+L2<J3Vabs(L1-L2)>J3
         then true else false;
    SJ:=.0;
    if cond
     then
     begin
      setoutput(3);
      print('cond=true');
      go to END;
     end;
    KAPPA:=.0;
    IWMIN:=J1+J2+J3;
```

```
if IWMIN<J1+L2+L3
 then IWMIN=J1+L2+L3
 else
  if IWMIN<L1+J2+L3
   then IWMIN=L1+J2+L3
   else
    if IWMIN<L2+J3+L1
     then IWMIN=L1+L2+J3;
IWMAX=J1+J2+L1+L2;
if IWMAX>J2+J3+L2+L3
 then IWMAX=J2+J3+L2+L3
 else
  if IWMAX>J1+J3+L1+L3
   then IWMAX=J1+J3+L1+L3
   else
    if IWMIN>IWMAX
     then go to END;
for IW=IWMIN step 2 until IWMAX do
 begin
  IW1=IW/2+2;
  IW2=(IW-J1-J2-J3)/2+1;
  IW3=(IW-J1-L2-L3)/2+1;
  IW4=(IW-L1-J2-L3)/2+1;
  IW5=(IW-L1-L2-J3)/2+1;
  IW6=(J1+J2+L1+L2-IW)/2+1;
  IW7=(J1+J3+L1+L3-IW)/2+1;
```

```
        IW8=(J2+J3+L2+L3-IW)/2+1;
        if (IW/4-entier(IW/4))≠0
         then PH:=-1
         else PH:=1;
        KAPPA:=KAPPA+PH×SN(IW1)/SN(IW2)/SN(IW3)/SN(IW4)/
              SN(IW5)/SN(IW6)/SN(IW7)/SN(IW8);
      end IW;
     SJ:=KAPPA×DELTA(J1,J2,J3)×DELTA(J1,L2,L3)×
        DELTA(L1,J2,L3)×DELTA(L1,L2,J3);
END:SJS:=SJ×10;
   end SJS;
  real procedure DELTA(J1,J2,J3);
   value J1,J2,J3;
   integer J1,J2,J3;
   begin
    integer IW1,IW2,IW3,IW4;
    real G;
    IW1=(J1+J2-J3)/2+1;
    IW2=(J1-J2+J3)/2+1;
    IW3=(-J1+J2+J3)/2+1;
    IW4=(J1+J2+J3+2)/2+1;
    G:=sqrt(SN(IW1)×SN(IW2)×SN(IW3)/SN(IW4));
    DELTA:=G/3.16227765;
   end DELTA;
  real procedure SN(N);
   integer N;
```

```
    begin
     real SL;
     switch GN=G0,G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,
              G11,G12,G13,G14,G15,G16,G17,G18,G19,G20,
              G21,G22,G23,G24,G25,G26,G27,G28,G29,G30,
              G31,G32,G33,G34,G35,G36,G37,G38,G39,G40,
              G41,G42,G43,G44,G45,G46,G47,G48,G49,G50,
              G51,G52,G53,G54,G55,G56,G57,G58,G59,G60,
              G61,G62,G63,G64,G65,G66,G67,G68,G69,G70,
              G71,G72,G73,G74,G75,G76,G77,G78,G79,G80,
              G81,G82,G83,G84,G85,G86,G87,G88,G89,G90,
              G91,G92,G93,G94,G95,G96,G97;
     go to GN[N];
G0:  SL=1.000000000,000;
     go to FC;
G1:  SL=1.000000000,-001;
     go to FC;
G2:  SL=2.000000000,-002;
     go to FC;
G3:  SL=5.999999999,-003;
     go to FC;
G4:  SL=2.400000000,-003;
     go to FC;
G5:  SL=1.200000000,-003;
     go to FC;
G6:  SL=7.199999998,-004;
```

```
    go to PC;
G7: SL=5.040000000₁₀-004;
    go to PC;
G8: SL=4.031999999₁₀-004;
    go to PC;
G9: SL=3.628799999₁₀-004;
    go to PC;
G10:SL=3.628799999₁₀-004;
    go to PC;
G11:SL=3.991679998₁₀-004;
    go to PC;
G12:SL=4.790015997₁₀-004;
    go to PC;
G13:SL=6.227020800₁₀-004;
    go to PC;
G14:SL=8.717829119₁₀-004;
    go to PC;
G15:SL=1.307674368₁₀-003;
    go to PC;
G16:SL=2.092278988₁₀-003;
    go to PC;
G17:SL=3.556874279₁₀-003;
    go to PC;
G18:SL=6.402373701₁₀-003;
    go to PC;
G19:SL=1.216451003₁₀-002;
```

```
    go to FC;
G20:SL=2.432902007₁₀-002;
    go to FC;
G21:SL=5.109094213₁₀-002;
    go to FC;
G22:SL=1.124000727₁₀-001;
    go to FC;
G23:SL=2.585201671₁₀-001;
    go to FC;
G24:SL=6.204484010₁₀-001;
    go to FC;
G25:SL=1.551121002₁₀000;
    go to FC;
G26:SL=4.032914611₁₀000;
    go to FC;
G27:SL=1.088886945₁₀001;
    go to FC;
G28:SL=3.048883445₁₀001;
    go to FC;
G29:SL=8.841761989₁₀001;
    go to FC;
G30:SL=2.652528596₁₀002;
    go to FC;
G31:SL=8.222838648₁₀002;
    go to FC;
G32:SL=2.631308367₁₀003;
```

```
    go to FC;
G33:SL=8.683317610₁₀003;
    go to FC;
G34:SL=2.952327987₁₀004;
    go to FC;
G35:SL=1.033314795₁₀005;
    go to FC;
G36:SL=3.719933263₁₀005;
    go to FC;
G37:SL=1.376375307₁₀006;
    go to FC;
G38:SL=5.230226166₁₀006;
    go to FC;
G39:SL=2.039788206₁₀007;
    go to FC;
G40:SL=8.159152821₁₀007;
    go to FC;
G41:SL=3.345252657₁₀008;
    go to FC;
G42:SL=1.405006116₁₀009;
    go to FC;
G43:SL=6.041526296₁₀009;
    go to FC;
G44:SL=2.658271570₁₀010;
    go to FC;
G45:SL=1.196222206₁₀011;
    go to FC;
```

```
G46:SL=5.502622148w011;
    go to FC;
G47:SL=2.586232409w012;
    go to FC;
G48:SL=1.241391556w013;
    go to FC;
G49:SL=6.082818625w013;
    go to FC;
G50:SL=3.041409313w014;
    go to FC;
G51:SL=1.55118749w015;
    go to FC;
G52:SL=8.065817516w015;
    go to FC;
G53:SL=4.274883283w016;
    go to FC;
G54:SL=2.308436973w017;
    go to FC;
G55:SL=1.269640335w018;
    go to FC;
G56:SL=7.109985874w018;
    go to FC;
G57:SL=4.052691948w019;
    go to FC;
G58:SL=2.350561330w020;
    go to FC;
```

```
G59:SL=1.386831184₁₀021;
    go to FC;
G60:SL=8.320987105₁₀021;
    go to FC;
G61:SL=5.075802133₁₀022;
    go to FC;
G62:SL=3.146997322₁₀023;
    go to FC;
G63:SL=1.982608313₁₀024;
    go to FC;
G64:SL=1.268869320₁₀025;
    go to FC;
G65:SL=8.247650579₁₀025;
    go to FC;
G66:SL=5.443449381₁₀026;
    go to FC;
G67:SL=3.647111085₁₀027;
    go to FC;
G68:SL=2.480035538₁₀028;
    go to FC;
G69:SL=1.711224521₁₀029;
    go to FC;
G70:SL=1.197857164₁₀030;
    go to FC;
G71:SL=8.504785866₁₀030;
    go to FC;
```

```
G72:SL=6.123445823₁₀031;
    go to PC;
G73:SL=4.470115450₁₀032;
    go to PC;
G74:SL=3.307885433₁₀033;
    go to PC;
G75:SL=2.480914074₁₀034;
    go to PC;
G76:SL=1.885494696₁₀035;
    go to PC;
G77:SL=1.451830920₁₀036;
    go to PC;
G78:SL=1.132428116₁₀037;
    go to PC;
G79:SL=8.946182137₁₀037;
    go to PC;
G80:SL=7.156945690₁₀038;
    go to PC;
G81:SL=5.797126023₁₀039;
    go to PC;
G82:SL=4.753643326₁₀040;
    go to PC;
G83:SL=3.945523970₁₀041;
    go to PC;
G84:SL=3.314240126₁₀042;
    go to PC;
```

```
G85:SL=2.817104114,043;
    go to PC;
G86:SL=2.422709532,044;
    go to PC;
G87:SL=2.107757298,045;
    go to PC;
G88:SL=1.854826417,046;
    go to PC;
G89:SL=1.650795515,047;
    go to PC;
G90:SL=1.485715960,048;
    go to PC;
G91:SL=1.352001527,049;
    go to PC;
G92:SL=1.243841401,050;
    go to PC;
G93:SL=1.156772506,051;
    go to PC;
G94:SL=1.087366153,052;
    go to PC;
G95:SL=1.032997848,053;
    go to PC;
G96:SL=9.916779310,053;
    go to PC;
G97:SL=9.619275954,054;
    go to PC;
```

```
FC; SN=SL;
   end PROCEDURE SLN;
  k=f-a+c;
  if (k/2-entier(k/2))≠0
   then
   begin
    setoutput(0);
    print('?COEFFICENT␣␣IS␣␣COMPLEX');
    stop;
   end
   else g=(a+c+f)/2;
  if k-4×(k+4)≠0
   then k=1
   else k=0;
  gl=g+f-a-b-d-c+k;
  x=2×a+1;
  x=x×(2×b+1);
  x=x×(2×c+1);
  x=x×(2×d+1);
  x=x×(2×f+1);
  if a<b
   then ISJS=SJS(2×d,2×c,2×e,2×a,2×b,2×f)
   else ISJS=SJS(2×a,2×b,2×e,2×d,2×c,2×f);
  ZET=(-1)↑gl×sqrt(x)×SN(g)/SN(g-a)/SN(g-c)/SN(g-f)×
      DELTA(2×a,2×c,2×f)×ISJS;
end ZET;
```

Z dostarczonego maszynopisu druk i oprawę wykonano w Zakładzie Graficznym Politechniki Krakowskiej w Krakowie. Nakład 120 egz.

Zam. 567/74