PRACE NAUKOWE Wyższej Szkoły Pedagogicznej w Częstochowie
Seria: FIZYKA. CHEMIA 1987 z. II

DETLEF GAUDYN

GAUSS: PROGRAM DO ROZKŁADU KRZYWEJ DOŚWIADCZALNEJ NA FUNKCJE GAUSSA

Nazwa programu: GAUSS
Komputer: SM-4, Instalacja: WSP Częstochowa
Język programowania: FORTRAN
Pamięć operacyjna: 56 kB
Problem fizyczny:
Rozkład krzywej doświadczalnej na funkcje Gaussa.
Metoda rozwiązania:
Gaussiany dopasowywane są metodą najmniejszych kwadratów typu Marquardt'a.
Ogólne ograniczenia programu:
Możliwość wpisania 7 gaussianów o nieznanych parametrach i 5 gaussianów o stałych parametrach w widmo składające się maksimum z 200 punktów pomiarowych.

W wielu dziedzinach fizyki doświadczalnej otrzymujemy krzywe pomiarowe, które są sumą różnych funkcji składowych. Najczęściej interesuje nas problem rozłożenia takiej krzywej na funkcje Gaussa.

Program GAUSS rozwiązuje to zagadnienie przez zastosowanie metody najmniejszych kwadratów z wykorzystaniem techniki iteracyjnej Marquardt'a [1]. Program napisany jest w języku FORTRAN i przetestowany na minikomputerze SM-4.

1. Metoda matematyczna

1.1. Model

W przyjętym modelu zakłada się, że krzywa pomiarowa jest sumą:

(a) N_g gaussianów o nieznanych parametrach

(b) N_{gfix} gaussianów o znanych parametrach

(c) tła opisanego funkcją liniową $y_t = a_0 + b_0$

oraz, że

(d) wyniki pomiarów y_i podlegają rozkładowi normalnemu $N(y_i, S_i)$.

Teoretyczna wartość F_i dla zmiennej niezależnej x_i jest równa

$$(1) \qquad F_i = \sum_{j=1}^{N_g} A_j \cdot \exp\left\{ -\frac{1}{2} \left[\frac{(x_i - x_j{}^0)}{\delta_j} \right]^2 \right\}$$

$$+ \sum_{j=1}^{N_{gfix}} A_j{}^F \cdot \exp\left\{ -\frac{1}{2} \left[\frac{(x_i - x_j{}^F)}{\delta_j{}^F} \right]^2 \right\} + a_0 \cdot x_i + b_0$$

gdzie:

N_g — ilość gaussianów o nieznanych parametrach

N_{gfix} — ilość gaussianów o znanych parametrach

A_j, $A_j{}^F$ — maksima funkcji Gaussa odpowiednio o nieznanych i stałych parametrach

δ_j, $\delta_j{}^F$ — odchylenia standardowe gaussianów nieznanych i stałych

$x_j{}^0$, $x_j{}^F$ — współrzędne położeń maksimów nieznanych i stałych gaussianów.

a_0, b_0 —— współczynniki równania tła.

Ponieważ estymować będziemy tylko parametry nieznanych wielkości, więc interesująca nas funkcja będzie opisana równaniem

$$(2) \qquad f_i = F_i - \sum_{j=1}^{N_{gfix}} A_j{}^F \cdot \exp\left\{ -\frac{1}{2} \left[\frac{(x_i - x_j{}^F)}{\delta_j{}^F} \right]^2 \right\}$$

Aby otrzymać ostateczną postać funkcji f_i (2) należy odjąć od niej, jeżeli znane są wartości parametrów a_0 lub b_0, odpowiednio człon a_0x_i lub b_0.

Dysponując teoretyczną wartością funkcji f_i oraz wartościami zmierzonymi y_i możemy zastosować metodę najmniejszych kwadratów w celu wyznaczenia estymatorów nieznanych parametrów funkcji (2).

1.2. Półliniowa metoda najmniejszych kwadratów

Niech n będzie ilością punktów pomiarowych y_i, których położenie jest określone zmienną niezależną x_i. Stosując ważoną metodę najmniejszych kwadratów, żądamy aby dla wektora nieznanych parametrów

$$(3) \qquad b = (A_1, A_2, \ldots, A_{N_g}, a_0, b_0, x_1{}^0, x_2{}^0, \ldots, x^0_{N_g}, \delta_1, \delta_2, \ldots, \delta_{N_g})^T$$

(T — transpozycja)

funkcja

$$(4) \qquad \Phi = \sum_{n}^{i=1} w_i \cdot [y_i - f_i(b)]^2 = \min$$

osiągnęła wartość minimalną.
$f_i(b)$ jest teoretyczną wartością zmiennej zależnej y_i w punkcie x_i dla ustalonych parametrów wektora b, a w_i są wagami statystycznymi punktów pomiarowych.

Przedstawiony model (2) jest liniowy w parametrach określających maksima gaussianów i w parametrach tła, a nieliniowy w położeniach maksimów gaussianów oraz ich odchyleniach standardowych.

W programie GAUSS do rozwiązania powyższego problemu (4) wykorzystana jest technika Marquardt'a [1]. Zastosowanie tej techniki pozwala stosować model tylko częściowo nieliniowy.
Rozkładając wektor b na liniowe i nieliniowe składniki otrzymujemy

$$(5) \qquad b = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

gdzie:

$$(6) \qquad \alpha = (A_1, A_2, ..., A_{N_g}, a_0, b_0)^T$$

i

$$(7) \qquad \beta = (x_1^0, x_2^0, ..., x_{N_g}^0, \delta_1, \delta_2, ..., \delta_{N_g})^T$$

Z (6) i (7) wynika, że prezentowany model jest liniowy w α i nieliniowy w β parametrach. Wektor liniowy parametrów ma wymiar $k_\alpha = N_g + 2 - IAFIX - IBFIX$, gdzie $IAFIX = 1$, gdy wartość a_0 jest stała i $IBFIX = 1$, gdy wartość b_0 jest stała. Wymiar wektora β jest równy $k_\beta = 2N_g$.

Jeżeli zdefiniujemy

$$(8) \qquad u_{ij} = \exp -\left(\frac{(x_i - x_j^0)^2}{2\delta_j^2} \right) \quad \left(j \leqslant N_g \right)$$

$$(9) \qquad u_{ij} = x_i \ , \quad \alpha_{N_g+1} = a_0 \qquad (j = N_g+1 \text{ i } IAFIX = 0)$$

$$(10) \qquad u_{ij} = 1 \ , \quad \alpha_{k_\alpha} = b_0 \quad (j = k_\alpha \text{ i } IBFIX = 0)$$

wówczas funkcję (2) możemy zapisać w postaci

$$f_i = \sum_{j=1}^{k_\alpha} \alpha_j \cdot u_{ij} \tag{11}$$

Postać (11) funkcji f_i wykorzystuje się do estymacji parametrów wektora b.

Zastosowanie metody najmniejszych kwadratów w celu wyznaczenia liniowych parametrów prowadzi do rozwiązania liniowego układu równań [2]

$$\underset{k_\alpha}{\overset{k_\alpha}{}} [C]\,[\alpha] = [\gamma] \tag{12}$$

Macierz C ma elementy równe $c_{jj_1} = \sum w_i u_{ij} u_{ij_1}$, a wektor składowe równe

$$\gamma_j = \sum_i w_i y_i u_{ij}$$

Do obliczenia estymowanych, nieliniowych parametrów będą potrzebne pochodne $\partial f_i / \partial \beta_{j'}$

$$\frac{\partial f_i}{\partial \beta_{j'}} = \sum_{j_1=1}^{k_\alpha} \left[\frac{\partial \alpha_{j_1}}{\partial \beta_{j'}} u_{ij_1} + \alpha_{j_1} \frac{\partial u_{ij_1}}{\partial \beta_{j'}} \right] \qquad (j' = 1, 2, ..., k_\beta) \tag{13}$$

Ponieważ w naszym przypadku $\partial \alpha_{j_1} / \partial \beta_{j'} = 0$ więc

$$p_{ij'} = \frac{\partial f_i}{\partial \beta_{j'}} = \sum_{j_1=1}^{k_\alpha} \alpha_{j_1} \cdot \frac{\partial u_{ij_1}}{\partial \beta_{j'}} = \tag{14}$$

$$= \sum_{j_1=1}^{N_g} A_{j_1} \frac{\partial}{\partial \beta_{j'}} \exp\left(- \frac{(x_i - x_{j_1}^0)^2}{2\delta_{j_1}^2} \right)$$

$$p_{ij'} = \alpha_j \frac{(x_i - x_j^0)}{\delta_j^2} \cdot u_{ij} \qquad (j \leqslant N_g, \quad j' \leqslant N_g) \tag{15}$$

$$(16) \qquad p_{ij'} = \alpha_j \frac{(x_i - x_j^0)^2}{\delta_j^3} \cdot u_{ij} \qquad j \leqslant N_g\,, \quad N_g < j' \leqslant k_\beta$$

Stosując technikę Marquardt'a [1, 2] dla wektora

$$\beta = \left(x_1^0, x_2^0, \ldots, x_{N_g}, \delta_1, \delta_2, \ldots, \delta_{N_g}\right)^T$$

minimalizujemy funkcję

$$\Phi = \sum_{i=1}^{n} w_i [y_i - f(x_i, \alpha, \beta)]^2$$

Metoda ta wywodzi się z metod iteracyjnych Gaussa-Newtona, które sprowadzają się do rozwiązania układu równań

$$(17) \qquad A \cdot d = g$$

Macierz A jest symetryczna i dodatnio określona i równa się iloczynowi P^TWP. P, W, g mają elementy odpowiednio równo $p_{ij} = \partial f_i / \partial \beta_j$,

$$w_{ii'} = \delta_{ii'} \cdot w_i$$

$$g_j = \sum_{i=1}^{n} w_i (y_i - f_i) \frac{\partial f_i}{\partial \beta_j} \qquad (j \leqslant k_\beta\ )$$

($\delta_{ii'}$ jest deltą Kroneckera). Wektor d jest wektorem poprawek do wektora β. Modyfikacja Marquardt'a układu (17) prowadzi do nowego układu równań

$$(18) \qquad (A^* + kI) d^* = g^*$$

I jest macierzą jednostkową, macierz A^* otrzymuje się przez skalowanie macierzy A

$$(19) \qquad A^* = (a^*_{jj'}) = \left(\frac{a_{jj'}}{\sqrt{a_{j'j'}} \sqrt{a_{jj}}} \right)$$

a wektor g^* przez skalowanie wektora g

$$(20) \qquad g^* = (g_j^*) = \left(\frac{g_j}{\sqrt{a_{jj}}} \right)$$

Wektor poprawek d wyraża się wzorem

$$(21) \qquad d_j = d_j^* / \sqrt{a_{jj}}$$

Parametr k jest parametrem interpolacyjnym pomiędzy metodą Gaussa-Newtona ($k=0$) a metodą gradientu ($k=\infty$). Pierwsza z metod jest szybka, ale może być rozbieżna, druga jest bezpieczniejsza ale dużo wolniejsza.

Marquardt [1] poleca następującą strategię doboru parametru k. Niech υ będzie liczbą >1 (tutaj $\upsilon=10$). Niech $k^{(r)}$ będzie k-tą wartością r-tej iteracji (tutaj $k^{(0)}=0,01$). $k^{(r+1)}$ będzie determinowane wartością sumy najmniejszych kwadratów Φ, przed i po iteracji. Jeżeli $\Phi(k^{(r)}/\upsilon) \leqslant \Phi^{(r)}$, to $k^{(r+1)}=k^{(r)}$ (iteracja zwana iteracją „typu A"). Przy niespełnieniu tego warunku za $k^{(r+1)}$ podstawiamy $k^{(r)}\upsilon^m$ ($m=0, 1, 2, \ldots$), która dla najmniejszych wartości m spełnia nierówność $\Phi(k^{(r)}\upsilon^m) \leqslant \Phi^{(r)}$ (iteracja zwana iteracją „typu B"). Iteracje są prowadzone do momentu, gdy $\Phi^{(r)}$ i $\beta^{(r)}$ są stałe przy wzroście r.

1.3. Analiza statystyczna

Analiza statystyczna obejmuje:

(i) model idealny

(ii) małe fluktuacje danych pomiarowych wokół ich wartości średnich

(iii) „wagi statystyczne".

Z uwzględnieniem (i) — (iii) możliwe jest obliczenie macierzy kowariancji Q estymowanych parametrów wektora b (3),

$$Q=(P^TWP)^{-1}, \quad p_{ij}=\partial f_i/\partial b_j \tag{22}$$

oraz założenie, że Φmin (minimalna wartość Φ) podlega rozkładowi χ^2 o $q=n-k_{free}$ stopniach swobody. k_{free} jest ilością estymowanych parametrów wektora b,

$$k_{free}=k_\alpha + k_\beta \tag{23}$$

Wielkość

$$S^2=\Phi\text{min}/q \tag{24}$$

jako wartość χ^2 na stopień swobody, jest miarą dopasowania. Wartość S^2 znacznie większa od 1 wskazuje, że model teoretyczny nie jest dobrą reprezentacją zmierzonych wielkości.

1.4. Rozwiązanie równań liniowych

Podstawową częścią przedstawionej metody najmniejszych kwadratów jest rozwiązanie symetrycznego układu równań liniowych (12, 18)

$$A\cdot x=b \tag{25}$$

Układ ten ze względu na symetryczność macierzy A daje się w prosty sposób rozwiązać przez zastosowanie rozkładu Cholesky'ego [5, 6]. Jeżeli symetryczna macierz A jest dodatnio określona stosuje się rozkład Cholesky'ego w postaci

(26) $$A=G\cdot G^T$$

G jest macierzą dolną trójkątną. W przypadkach, gdy macierz A nie jest dodatnio określona, stosuje się rozkład Cholesky'ego w postaci

(27) $$A=G\cdot J\cdot G^T$$

gdzie G jest identyczna jak w (26), a J jest macierzą diagonalną z elementami ± 1.

W tej pracy zastosowano rozkład w postaci (27).

2. Charakterystyka programu

2.1. Struktura

Program GAUSS składa się: z segmentu głównego oraz podprogramów: MARQAR, STATIS, COLDEC, COLSOL, DRUK. Segment główny zawiera instrukcje czytania danych oraz wywołuje i realizuje poszczególne podprogramy.

Podprogram MARQAR wykonuje estymację nieznanych parametrów metodą najmniejszych kwadratów.

STATIS przeprowadza analizę statystyczną.

COLDEC i COLSOL służą do rozwiązywania symetrycznego układu równań liniowych. COLDEC dokonuje rozkładu Cholesky'ego, a COLSOL na bazie tego rozkładu rozwiązuje dany układ równań. Te dwa podprogramy są wykorzystane w podprogramie analizy statystycznej STATIS i w podprogramie MARQAR.

Podprogram DRUK wyprowadza wyniki na drukarkę.

2.2. Wejście

Dane wejściowe podzielone zostały na cztery bloki. W przedstawionej wersji czytane są one ze zbioru o nazwie INGAUS, umieszczonego na dysku.

BLOK 1 jest pojedyńczym rekordem, format 6I3, zawierającym następujące dane: N, NG, NGFIX, IAFIX, IBFIX, IPRINT. N — jest ilością punktów pomiarowych; NG — ilością gaussianów o estymowanych parametrach; NGFIX — ilością gaussianów stałych; IAFIX i IBFIX są parametrami sterującymi tła, a IPRINT jest parametrem sterującym wyprowadzaniem iteracji Marquardt'a. Jeżeli IPRINT=0 iteracje nie są drukowane, natomiast IPRINT=1 wyprowadza iteracje.

BLOK 2 zawiera kolejno rekordy z danymi:

a) (B(J),J=1,KBET), format 6E12.5; B(J) jest wektorem przybliżonych wartości nieliniowych parametrów zgodnie z (7), z tym że, odchylenia standardowe zastąpione zostają odpowiadającymi im szerokościami połówkowymi FWHM. Pomiędzy szerokościami połówkowymi a odchyleniami standardowymi istnieje zależność $FWHM=2\cdot(2\cdot \ln 2)^{1/2}\cdot\delta$

b) (X(I),I=1,N), format 6E12.5; X(I) jest wektorem wartości zmiennych niezależnych
c) (Y(I),I=1,N), format 6E12.5; Y(I) jest wektorem wartości punktów pomiarowych
d) (W(I),I=1,N), format 6E12.5; W(I) jest wektorem wag statystycznych punktów pomiarowych.

BLOK 3 jest blokiem niepustym, jeżeli NGFIX≠0. Zawiera on parametry gaussianów stałych (AF(J),XF(J),FFWHM(J),J=1,NGFIX), format 6E12.5; AF(J) — są wartościami maksimów, XF(J) — wartościami ich położeń, a FFWHM(J) — szerokościami połówkowymi.

Istnienie BLOKU 4 jest uzależnione od wartości parametrów sterujących IAFIX i IBFIX. Jeżeli IAFIX=1, to pierwszy rekord BLOKU 4, format E12.5, zawiera wartość a_0 współczynnika równania tła (1) natomiast, gdy IAFIX=0 obliczany będzie jako estymator. Analogicznie drugi rekord, format E12.5, zawiera wartość współczynnika b_0, jeżeli IBFIX=1 lub, gdy IBFIX=0 współczynnik będzie estymowany.

3. Test

Program został przetestowany na danych generowanych zgodnie z (1) oraz na danych pochodzących z eksperymentu. W każdym z przypadków uzyskano zgodność modelu rzeczywistego z teoretycznym.

Dwa przedstawione testy dotyczą tej samej krzywej i obrazują działanie programu. Pierwszy przebieg pozwala ustalić jako stałe parametry tła i wybrać „lepsze" wartości początkowe parametrów nieliniowych dla testu 2.

3.1. Wyjście

Program wyprowadza wczytane (przybliżone) wartości położeń maksimów oraz wartości szerokości połówkowych FWHM i odpowiadające im δ. Jeżeli parametr sterujący IPRINT=1 zostają wyprowadzone wartości charakterystyczne iteracji: współczynniki nieliniowe (x_1^0, x_2^0, ..., x^0_{Ng}, δ_1, δ_2, ..., δ_{Ng}), współczynniki liniowe (A_1, A_2, ..., A_{Ng}, a_0 — jeżeli IAFIX=0, b_0 — jeżeli IBFIX=0), wartość sumy najmniejszych kwadratów Φ, k — Marquardt'a oraz typ iteracji (A lub B).

Po informacji, po ilu iteracjach uzyskano dopasowanie, wyprowadzane są estymatory wszystkich parametrów i ich odchylenia standardowe. Podawany jest również procentowy udział każdego gaussianu w analizowanej krzywej oraz wartość funkcji χ^2 na stopień swobody. Jeżeli NGFIX≠0 zastają wyprowadzone wartości parametrów gaussianów stałych. Ostatnią wyprowadzaną informacją jest pole pod krzywą teoretyczną oraz pole pod krzywą doświadczalną.

```
50  2  1  0  0  0
0.11000E 02 0.31000E 02 0.60000E 01 0.80000E 01
0.00000E 00 0.10000E 01 0.20000E 01 0.30000E 01 0.40000E 01 0.50000E 01
0.60000E 01 0.70000E 01 0.80000E 01 0.90000E 01 0.10000E 02 0.11000E 02
0.12000E 02 0.13000E 02 0.14000E 02 0.15000E 02 0.16000E 02 0.17000E 02
0.18000E 02 0.19000E 02 0.20000E 02 0.21000E 02 0.22000E 02 0.23000E 02
0.24000E 02 0.25000E 02 0.26000E 02 0.27000E 02 0.28000E 02 0.29000E 02
0.30000E 02 0.31000E 02 0.32000E 02 0.33000E 02 0.34000E 02 0.35000E 02
0.36000E 02 0.37000E 02 0.38000E 02 0.39000E 02 0.40000E 02 0.41000E 02
0.42000E 02 0.43000E 02 0.44000E 02 0.45000E 02 0.46000E 02 0.47000E 02
0.48000E 02 0.49000E 02
0.46366E 02 0.69776E 02 0.10554E 03 0.15795E 03 0.23390E 03 0.34864E 03
0.53125E 03 0.81156E 03 0.11720E 04 0.15020E 04 0.16400E 04 0.15060E 04
0.11800E 04 0.82356E 03 0.54725E 03 0.36864E 03 0.25792E 03 0.18601E 03
0.13780E 03 0.10674E 03 0.89459E 02 0.84564E 02 0.93518E 02 0.12163E 03
0.17758E 03 0.27001E 03 0.40109E 03 0.55930E 03 0.71661E 03 0.83478E 03
0.88000E 03 0.83877E 03 0.72459E 03 0.57122E 03 0.41689E 03 0.28948E 03
0.20027E 03 0.14658E 03 0.11885E 03 0.10689E 03 0.10309E 03 0.10296E 03
0.10427E 03 0.10607E 03 0.10801E 03 0.11000E 03 0.11200E 03 0.11400E 03
0.11600E 03 0.11800E 03
0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01
0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01
0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01
0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01
0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01
0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01
0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01
0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01 0.10000E 01
0.10000E 01 0.10000E 01
0.60000E 03 0.10000E 02 0.94193E 01
WPROWADZONE PARAMETRY
        POLOZ. MAXIM=     11.000    31.000
        FWHM =             6.000     8.000
        SIGMA=             2.548     3.397
*****************************************************************************

DOPASOWANIE PO 25 ITERACJACH

        ROWNANIE TLA Y= 0.200E 01 * X +  0.200E 02
        ODCH. STAND.    0.117E-01        0.382E 00

        PIK=             999.972   800.001
        ODCH. STAND.       0.682     0.551

        POLOZ. MAXIM=     10.000    30.000
        ODCH. STAND.       0.002     0.002

        SIGMA=             2.000     3.000
        ODCH. STAND.       0.002     0.003

        FWHM =             4.710     7.064
        ODCH. STAND.       0.004     0.006
```

```
WZGLEDNE NATEZENIA W %      29.411    35.294
         ODCH. STAND.        0.032     0.039

CHI=  0.00005  NA STOPIEN SWOBODY
*****************************************************************************

PARAMETRY GAUSSIANOW STALYCH

         PIK=             600.000

         POLOZ. MAXIM=     10.000

         SIGMA=             4.000

         FWHM =             9.419

WZGLEDNE NATEZENIA W %     35.294

         POLE POD KRZYWA - WG MODELU      =     0.2046939E 05

         POLE POD KRZYWA - RZECZYWISTE =     0.2046938E 05
*****************************************************************************
WPROWADZONE PARAMETRY
         POLOZ. MAXIM=     10.000    30.000
         FWHM =             4.710     7.064
         SIGMA=             2.000     3.000

ITERACJA NO  1
WSPOLCZYNNIKI NIELINIOWE:
    0.1000000E 02    0.3000000E 02      0.2000153E 01      0.2999805E 01
WSPOLCZYNNIKI LINIOWE:
    0.9999338E 03    0.8000256E 03
PHI =    0.19122621E-01
KAPPA =     0.2000E-01
A

ITERACJA NO  2
WSPOLCZYNNIKI NIELINIOWE:
    0.1000000E 02    0.3000000E 02      0.2000050E 01      0.2999929E 01
WSPOLCZYNNIKI LINIOWE:
    0.9999595E 03    0.8000091E 03
PHI =    0.38428947E-02
KAPPA =     0.2000E-02
A

ITERACJA NO  3
WSPOLCZYNNIKI NIELINIOWE:
    0.1000000E 02    0.3000000E 02      0.2000015E 01      0.2999971E 01
WSPOLCZYNNIKI LINIOWE:
    0.9999682E 03    0.8000038E 03
PHI =    0.21573356E-02
KAPPA =     0.2000E-03
A
```

```
ITERACJA NO  4
WSPOLCZYNNIKI NIELINIOWE:
    0.1000000E 02    0.3000000E 02    0.2000004E 01    0.2999934E 01
WSPOLCZYNNIKI LINIOWE:
    0.9999710E 03    0.8000018E 03
PHI =    0.19709293E-02
KAPPA =    0.2000E-04
A

ITERACJA NO  5
WSPOLCZYNNIKI NIELINIOWE:
    0.1000000E 02    0.3000000E 02    0.2000000E 01    0.2999939E 01
WSPOLCZYNNIKI LINIOWE:
    0.9999719E 03    0.8000012E 03
PHI =    0.19525553E-02
KAPPA =    0.2000E-05
A

ITERACJA NO  6
WSPOLCZYNNIKI NIELINIOWE:
    0.1000000E 02    0.3000000E 02    0.1999999E 01    0.2999991E 01
WSPOLCZYNNIKI LINIOWE:
    0.9999722E 03    0.8000009E 03
PHI =    0.19399404E-02
KAPPA =    0.2000E-06
A

ITERACJA NO  7
WSPOLCZYNNIKI NIELINIOWE:
    0.1000000E 02    0.3000000E 02    0.1999999E 01    0.2999991E 01
WSPOLCZYNNIKI LINIOWE:
    0.9999724E 03    0.8000007E 03
PHI =    0.19387770E-02
KAPPA =    0.2000E-07
B
B
B
B
B
B
B
A

ITERACJA NO  8
WSPOLCZYNNIKI NIELINIOWE:
    0.1000000E 02    0.3000000E 02    0.1999999E 01    0.2999991E 01
WSPOLCZYNNIKI LINIOWE:
    0.9999724E 03    0.8000007E 03
PHI =    0.19387770E-02
KAPPA =    0.2000E 02
B
A
```

```
ITERACJA NO  9
WSPOLCZYNNIKI NIELINIOWE:
   0.1000000E 02    0.3000000E 02    0.1999999E 01    0.2999991E 01
WSPOLCZYNNIKI LINIOWE:
   0.9999724E 03    0.8000007E 03
PHI =    0.19387770E-02
KAPPA =    0.2000E 02
B
A
************************************************************************

DOPASOWANIE PO  9 ITERACJACH

        ROWNANIE TLA Y= 0.200E 01 * X +  0.200E 02
        ODCH. STAND.         FIX                  FIX

        PIK=                  999.972   800.001
        ODCH. STAND.            0.650     0.531

        POLOZ. MAXIM=          10.000    30.000
        ODCH. STAND.            0.002     0.002

        SIGMA=                  2.000     3.000
        ODCH. STAND.            0.002     0.002

        FWHM =                  4.710     7.064
        ODCH. STAND.            0.004     0.005

 UZGLEDNE NATEZENIA W %        29.411    35.294
        ODCH. STAND.            0.029     0.036

 CHI=   0.00005  NA STOPIEN SWOBODY

************************************************************************

 PARAMETRY GAUSSIANOW STALYCH

        PIK=              600.000

        POLOZ. MAXIM=      10.000

        SIGMA=              4.000

        FWHM =              9.419

 UZGLEDNE NATEZENIA W %    35.294

        POLE POD KRZYWA - WG MODELU    =    0.2046939E 05

        POLE POD KRZYWA - RZECZYWISTE =     0.2046938E 05
```

```
FORTRAN V004A                     00:00:00    06-MAY-85   PAGE    1

      C     ************************** - PROGRAM GAUSS - *****************
            COMMON/MMSD/A(9),B(14),PHIR(25)
            COMMON/MMS/X(200),W(200),U(200,9),P(200,14),AA(23,23),G(23,23),
            1SGN(23),RHS(23),DEL(23),AA1(23,23)
            COMMON/MMD/B1(14),SCALE(14),GS(14),F(200),Y(200)
            COMMON/DRU/NGFIX,X2,A0,B0,AREAT,AREAF,FPOLE(5),AF(5),FFWHM(5),
            1SIF(5),XF(5)
            COMMON/CMMSD/N,NG,NGP1,KALP,KBET,K,IAFIX,IBFIX,PHI,IC
            BYTE UICK(2)
            INTEGER UIC
            EQUIVALENCE (UICK(1),UIC)
            DATA UICK/"27,"27/
            CALL SETFIL(1,'INGAUS',IDI,'DK',1,UIC)
      C     BLOK 1 ***********************************************************
            READ(1,300)N,NG,NGFIX,IAFIX,IBFIX,IPRINT
            KBET=2*NG
            NGP1=NG+1
            X2=2.*SQRT(2.*ALOG(2.))
      C     BLOK 2 ***********************************************************
            READ(1,305)(B(J),J=1,KBET)
            READ(1,305)(X(I),I=1,N)
            READ(1,305)(Y(I),I=1,N)
            READ(1,305)(W(I),I=1,N)
            WRITE(5,400)
            WRITE(5,405)
            WRITE(5,410)(B(J),J=1,NG)
            WRITE(5,420)(B(J),J=NGP1,KBET)
            DO 5 J=NGP1,KBET
      5     B(J)=B(J)/X2
            WRITE(5,415)(B(J),J=NGP1,KBET)
            AREAT=0.
            AREAF=0.
            DO 10 I=1,N
      10    AREAT=AREAT+Y(I)
      C     BLOK 3 ***********************************************************
            IF(NGFIX.EQ.0)GO TO 30
            READ(1,305)(AF(J),XF(J),FFWHM(J),J=1,NGFIX)
            DO 15 J=1,NGFIX
      15    SIF(J)=FFWHM(J)/X2
            DO 25 J=1,NGFIX
            DO 25 I=1,N
            X3=-((X(I)-XF(J))**2/(2.*SIF(J)**2))
            IF(X3.GE.-88.6)GO TO 20
            X3=-91
      20    X1=AF(J)*EXP(X3)
            AREAF=AREAF+X1
      25    Y(I)=Y(I)-X1
      C     BLOK 4 ***********************************************************
      30    IF(IAFIX.EQ.0)GO TO 35
            READ(1,305)A0
            IF(A0.EQ.0.)GO TO 35
            DO 40 I=1,N
            X1=A0*X(I)
            AREAF=AREAF+X1
      40    Y(I)=Y(I)-X1
```

```
FORTRAN V004A                              00:00:00   06-MAY-85   PAGE   2

      35   IF(IBFIX.EQ.0)GO TO 45
           READ(1,305)B0
           IF(B0.EQ.0.)GO TO 45
           DO 50 I=1,N
           AREAF=AREAF+B0
      50   Y(I)=Y(I)-B0
      45   KALP=NG+2-IAFIX-IBFIX
           CALL MARQAR(ISING,IPRINT)
           IF(ISING.EQ.1)STOP
           CALL STATIS(ISING)
           IF(ISING.EQ.1)STOP
           CALL DRUK
      C    ************************************************************
      300  FORMAT(6I3)
      305  FORMAT(6E12.5)
      C    ************************************************************
      400  FORMAT(1X,130(1H*)/)
      405  FORMAT(1X,'WPROWADZONE PARAMETRY')
      410  FORMAT(10X,'POLOZ. MAXIM=',8F10.3)
      415  FORMAT(10X,'SIGMA=',7X,8F10.3)
      420  FORMAT(10X,'FWHM =',7X,8F10.3)
           END
FORTRAN V004A                              00:00:00   04-MAY-84   PAGE   1

      C    ************************************************************
           SUBROUTINE MARQAR(ISING,IPRINT)
           COMMON/MMSD/A(9),B(14),PHIR(25)
           COMMON/MMD/B1(14),SCALE(14),GS(14),F(200),Y(200)
           COMMON/MMS/X(200),W(200),U(200,9),P(200,14),AA(23,23),G(23,23),
           1SGN(23),RHS(23),DEL(23),AA1(23,23)
           COMMON/CMMSD/N,NG,NGP1,KALP,KBET,K,IAFIX,IBFIX,PHI,IC
      C    ************************************************************
           DATA EPS,RKAPO,RNY/1.E-6,2.E-2,1.E1/
           LIPHIF=0
           IRMAX=25
           OEPS=1.+EPS
           IF(IAFIX.EQ.1)GO TO 5
           DO 1 I=1,N
      1    U(I,NGP1)=X(I)
      5    IF(IBFIX.EQ.1)GO TO 10
           DO 6 I=1,N
      6    U(I,KALP)=1.
      10   DO 15 I=1,N
           DO 15 J=1,NG
           X1=(X(I)-B(J))**2/(2.*B(J+NG)**2)
           X1=-X1
           IF(X1.GE.-88.6)GO TO 15
           X1=-91
      15   U(I,J)=EXP(X1)
           DO 30 J=1,KALP
           DO 20 J1=J,KALP
           H1=0.
           DO 25 I=1,N
      25   H1=H1+W(I)*U(I,J)*U(I,J1)
      20   AA1(J1,J)=H1
           H1=0.
           DO 35 I=1,N
```

```
35      H1=H1+W(I)*U(I,J)*Y(I)
30      RHS(J)=H1
        CALL COLDEC(KALP,AA1,G,SGN,ISING)
        IF(ISING.EQ.1)RETURN
        CALL COLSOL(KALP,G,RHS,A,SGN)
        PHI=0.
        DO 45 I=1,N
        H1=0.
        DO 40 J=1,KALP
40      H1=H1+A(J)*U(I,J)
        F(I)=H1
45      PHI=PHI+W(I)*((Y(I)-H1)**2)
        IF(LIPHIF)50,50,95
50      LIPHIF=1
        RKAP=RKAP0
        IR=0
        IF(KBET.GT.0)GOTO55
        IC=0
        RETURN
55      IR=IR+1
        PHIR(IR)=PHI
        IF(IPRINT.EQ.0)GO TO 60
```

FORTRAN V004A 00:00:00 04-MAY-84 PAGE 2

```
        WRITE(5,300)IR
        WRITE(5,330)(B(J),J=1,KBET)
        WRITE(5,305)(A(J),J=1,KALP)
        WRITE(5,310)PHIR(IR)
        WRITE(5,315)RKAP
60      DO 65 I=1,N
        DO 70 J=1,NG
        X1=X(I)-B(J)
70      P(I,J)=A(J)*X1/B(J+NG)**2*U(I,J)
        DO 75 J=NGP1,KBET
        X1=X(I)-B(J-NG)
75      P(I,J)=A(J-NG)*X1*X1/B(J)**3*U(I,J-NG)
65      CONTINUE
        DO 80 J=1,KBET
        B1(J)=B(J)
        DO 80 J1=J,KBET
        H1=0.
        DO 85 I=1,N
85      H1=H1+W(I)*P(I,J)*P(I,J1)
80      AA(J1,J)=H1
        DO 90 J=1,KBET
90      SCALE(J)=SQRT(AA(J,J))
        DO 100 J=1,KBET
        H1=0.
        DO 105 I=1,N
105     H1=H1+(Y(I)-F(I))*W(I)*P(I,J)
100     G8(J)=H1/SCALE(J)
        DO 110 J=1,KBET
```

```
            DO 110 J1=J,KBET
      110   AA(J1,J)=AA(J1,J)/SCALE(J)/SCALE(J1)
            RKAP=RKAP/RNY
      120   DO 125 J=1,KBET
      125   AA(J,J)=1.+RKAP
            CALL COLDEC(KBET,AA,G,SGN,ISING)
            IF(ISING.EQ.1)RETURN
            CALL COLSOL(KBET,G,GS,RHS,SGN)
            DO 130 J=1,KBET
            DEL(J)=RHS(J)/SCALE(J)
            B(J)=B1(J)+DEL(J)
            IF(B(J).GT.0.)GOTO 130
            WRITE(5,335)
            B(J)=B1(J)/2.
      130   CONTINUE
            GOTO 10
      95    IF(PHI.LE.PHIR(IR)*OEPS)GOTO 135
            RKAP=RKAP*RNY
            IF(IPRINT.EQ.1)WRITE(5,325)
            GOTO 120
      135   IF(IPRINT.EQ.1)WRITE(5,320)
            IF(IR.LT.3)GOTO 55
            ZARGU1=PHIR(IR)-PHIR(IR-1)
            ZARGU2=PHIR(IR-1)-PHIR(IR-2)
            DELTA=ABS(ZARGU1)+ABS(ZARGU2)
            DELTA=DELTA/PHIR(IR)/8.
            IF(DELTA.GT.EPS)GOTO 140
            IC=IR

FORTRAN V004A                  00:00:00    04-MAY-84    PAGE    3

            RETURN
      140   IF(IR.LT.IRMAX)GOTO 55
            IC=-1
            RETURN
      300   FORMAT(1HO,'ITERACJA NO',I3)
      305   FORMAT(1X,'WSPOLCZYNNIKI LINIOWE: '/(5E18.7))
      310   FORMAT(6H PHI =,E18.8)
      315   FORMAT(8H KAPPA =,E14.4)
      320   FORMAT(2H A)
      325   FORMAT(2H B)
      330   FORMAT(1X,'WSPOLCZYNNIKI NIELINIOWE: '/(5E18.7))
      335   FORMAT(15H SIGN EXCURSION)
            END
```

```
FORTRAN V004A                     00:00:00    04-MAY-84   PAGE    1

            SUBROUTINE STATIS(ISING)
            COMMON/MMSD/A(9),B(14),VAR(25)
            COMMON/MMS/X(200),W(200),U(200,9),P(200,14),AA(23,23),G(23,23),
            1SGN(23),RHS(23),DEL(23),Q(23,23)
            COMMON/CMMSD/N,NG,NGP1,KALP,KBET,K,IAFIX,IBFIX,PHI,IC
            NK=N-3-3*NG+IAFIX+IBFIX
            SK=NK
            PHI=PHI/SK
            K=KALP+KBET
            DO 10 I=1,N
            DO 5 J=1,NG
            X1=X(I)-B(J)
      5     P(I,J)=A(J)*X1/B(J+NG)**2*U(I,J)
            DO 15 J=NGP1,KBET
            X1=X(I)-B(J-NG)
      15    P(I,J)=A(J-NG)*X1*X1/B(J)**3*U(I,J-NG)
      10    CONTINUE
            DO 20 J=1,K
            DO 20 J1=J,K
            H1=0.
            DO 25 I=1,N
            IF(J.LE.KALP)U1=U(I,J)
            IF(J.GT.KALP)U1=P(I,J-KALP)
            IF(J1.LE.KALP)U2=U(I,J1)
            IF(J1.GT.KALP)U2=P(I,J1-KALP)
      25    H1=H1+U1*U2*W(I)
      20    AA(J1,J)=H1
            CALL COLDEC(K,AA,G,SGN,ISING)
            IF(ISING.EQ.1)RETURN
            DO 30 J=1,K
            DO 35 J1=1,K
      35    RHS(J1)=0.
            RHS(J)=1.
            CALL COLSOL(K,G,RHS,DEL,SGN)
            DO 40 J1=1,K
      40    Q(J1,J)=DEL(J1)
      30    CONTINUE
            DO 45 J=1,K
            H1=Q(J,J)
      45    VAR(J)=SQRT(ABS(H1))
            RETURN
            END
FORTRAN V004A                     00:00:00    03-MAY-85   PAGE    1

            SUBROUTINE DRUK
            BYTE ABY1(10),ABY2(10)
            COMMON/MMSD/A(9),B(14),VAR(25)
            COMMON/MMD/FWHM(14),POLE(14),SFWHM(14),SPOLE(200),Y(200)
            COMMON/DRU/NGFIX,X2,A0,B0,AREAT,AREAF,FPOLE(5),AF(5),FFWHM(5),
            1SIF(5),XF(5)
            COMMON/CMMSD/N,NG,NGP1,KALP,KBET,K,IAFIX,IBFIX,PHI,IC
            KAPNG=KALP+NG
            KALPP1=KALP+1
            NGP2=NG+2
            KANGP1=KALP+NG+1
```

```
      DO 4 J=1,10
      ABY1(J)=' '
    4 ABY2(J)=' '
      DO 5 I=1,N
    5 AREAF=AREAF+SPOLE(I)
      WRITE(5,420)
      IF(IC.GT.0)GO TO 10
      WRITE(5,470)
      GO TO 15
   10 WRITE(5,455)IC
   15 IF(IAFIX.EQ.0)GO TO 20
      A(NGP1)=A0
      ABY1(4)='F'
      ABY1(5)='I'
      ABY1(6)='X'
      GO TO 25
   20 ENCODE(10,425,ABY1)VAR(NGP1)
   25 IF(IBFIX.EQ.0)GO TO 30
      A(NGP2)=B0
      ABY2(4)='F'
      ABY2(5)='I'
      ABY2(6)='X'
      GO TO 35
   30 ENCODE(10,425,ABY2)VAR(KALP)
   35 WRITE(5,430)A(NGP1),A(NGP2)
      WRITE(5,435)ABY1,ABY2
      WRITE(5,405)(A(J),J=1,NG)
      WRITE(5,440)(VAR(J),J=1,NG)
      WRITE(5,410)(B(J),J=1,NG)
      WRITE(5,440)(VAR(J),J=KALPP1,KAPNG)
      WRITE(5,415)(B(J),J=NGP1,KBET)
      WRITE(5,440)(VAR(J),J=KANGP1,K)
      H1=0.
      DO 40 J=1,NG
      FWHM(J)=X2*B(J+NG)
      SFWHM(J)=X2*VAR(J+KAPNG)
      POLE(J)=2.5066283*B(J+NG)*A(J)
      H1=H1+POLE(J)
   40 SPOLE(J)=2.5066283*SQRT(A(J)**2*VAR(J+KAPNG)**2+B(J+NG)**2*
     1VAR(J)**2)
      IF(NGFIX.EQ.0)GO TO 55
      DO 45 J=1,NGFIX
      FPOLE(J)=2.5066283*AF(J)*SIF(J)
   45 H1=H1+FPOLE(J)
```

```
FORTRAN V004A                          00:00:00     08-MAY-85    PAGE    2

                DO 50 J=1,NGFIX
         50     FPOLE(J)=FPOLE(J)/H1*100.
         55     DO 60 J=1,NG
                POLE(J)=POLE(J)/H1*100.
         60     SPOLE(J)=SPOLE(J)/H1*100.
                WRITE(5,460)(FWHM(J),J=1,NG)
                WRITE(5,440)(SFWHM(J),J=1,NG)
                WRITE(5,465)(POLE(J),J=1,NG)
                WRITE(5,440)(SPOLE(J),J=1,NG)
                WRITE(5,445)PHI
                IF(NGFIX.EQ.0)GO TO 65
                WRITE(5,420)
                WRITE(5,450)
                WRITE(5,405)(AF(J),J=1,NGFIX)
                WRITE(5,410)(XF(J),J=1,NGFIX)
                WRITE(5,415)(SIF(J),J=1,NGFIX)
                WRITE(5,460)(FFWHM(J),J=1,NGFIX)
                WRITE(5,465)(FPOLE(J),J=1,NGFIX)
         65     WRITE(5,475)AREAF
                WRITE(5,480)AREAT
                WRITE(5,420)
         C      **********************************************************
         405    FORMAT(1H0,9X,'PIK=',9X,8F10.3)
         410    FORMAT(1H0,9X,'POLOZ. MAXIM=',8F10.3)
         415    FORMAT(1H0,9X,'SIGMA=',7X,8F10.3)
         420    FORMAT(1X,130(1H*)/)
         425    FORMAT(E10.3)
         430    FORMAT(10X,'ROWNANIE TLA Y=',E10.3,' * X + ',E10.3)
         435    FORMAT(10X,'ODCH. STAND.',3X,10A1,7X,10A1)
         440    FORMAT(10X,'ODCH. STAND. ',8F10.3)
         445    FORMAT(1H0,4HCHI=,F10.5,'  NA STOPIEN SWOBODY'/)
         450    FORMAT(1X,'PARAMETRY GAUSSIANOW STALYCH'/)
         455    FORMAT(1X,'DOPASOWANIE PO ',I2,' ITERACJACH'/)
         460    FORMAT(1H0,9X,'FWHM =',7X,8F10.3)
         465    FORMAT(1H0,'WZGLEDNE NATEZENIA W %',8F10.3)
         470    FORMAT(1X,'BRAK DOPASOWANIA PO 25 ITERACJACH'/)
         475    FORMAT(1H0,9X,'POLE POD KRZYWA - WG MODELU   =',E18.7)
         480    FORMAT(1H0,9X,'POLE POD KRZYWA - RZECZYWISTE =',E18.7)
                RETURN
                END
FORTRAN V004A                          00:00:00     04-MAY-84    PAGE    1

                SUBROUTINE COLDEC(N,A,G,SGN,ISING)
                DIMENSION A(23,23),G(23,23),SGN(23)
                ISING=0
                DO 5J=1,N
         5      SGN(J)=1.
                H1=A(1,1)
                IF(H1)10,55,15
         10     SGN(1)=-1.
         15     G(1,1)=SQRT(ABS(H1))
                IF(N.EQ.1)RETURN
```

```
      DO 20I=2,N
   20 G(I,1)=SGN(1)*A(I,1)/G(1,1)
      DO 50J=2,N
      SUM=0.
      JM1=J-1
      DO 25K=1,JM1
   25 SUM=SUM+SGN(K)*(G(J,K)**2)
      H1=A(J,J)-SUM
      IF(H1)30,55,35
   30 SGN(J)=-1.
   35 G(J,J)=SQRT(ABS(H1))
      IF(J.EQ.N)RETURN
      JP1=J+1
      DO 45I=JP1,N
      SUM=0.
      DO 40K=1,JM1
   40 SUM=SUM+SGN(K)*G(I,K)*G(J,K)
   45 G(I,J)=SGN(J)*(A(I,J)-SUM)/G(J,J)
   50 CONTINUE
   55 WRITE(6,60)
   60 FORMAT(37H0MATRIX SINGULAR. THIS JOB IS SKIPPED)
      ISING=1
      RETURN
      END
```

```
FORTRAN V004A                    00:00:00    04-MAY-84    PAGE 1

      SUBROUTINE COLSOL(N,G,B,X,SGN)
      DIMENSION G(23,23),B(23),X(23),SGN(23)
      IF(N.EQ.1)GOTO30
      X(1)=B(1)/G(1,1)
      DO 10I=2,N
      IM1=I-1
      SUM=0.
      DO 5 J=1,IM1
    5 SUM=SUM+G(I,J)*X(J)
   10 X(I)=(B(I)-SUM)/G(I,I)
      DO 15I=1,N
   15 X(I)=SGN(I)*X(I)
      X(N)=X(N)/G(N,N)
      NP1=N+1
      DO 25IBACK=2,N
      I=NP1-IBACK
      IP1=I+1
      SUM=0.
      DO 20J=IP1,N
   20 SUM=SUM+G(J,I)*X(J)
   25 X(I)=(X(I)-SUM)/G(I,I)
      RETURN
   30 X(1)=SGN(1)*B(1)/((G(1,1))**2)
      RETURN
      END
```

BIBLIOGRAFIA

[1]. D.W. Marquardt, J. Soc. Indust. Appl. Math. 11, 431 (1963).
[2] P. Kirkegaard, Some Aspects of the General Least — Squares Problem for Data Fitting, Riso-M-1399, p. 16 (1971).
[3]. P. Kirkegaard and M. Eldrup, Comput. Phys. Commun. 3, 240 (1972).
[4] H. Szydłowski (red.), Teoria pomiarów, PWN 1981.
[5]. J. Legras, Praktyczne metody analizy numerycznej, WNT 1974.
[6] Z. Fortuna, Metody numeryczne, WNT 1982.

DETLEF GAUDYN

GAUSS: PROGRAM FOR THE EXPERIMENTAL CURVE DISTRIBUTION ON GAUSS FUNCTIONS

SUMMARY

In the article, the introduces a mathematic method of the experimental curve resolution into Gauss functions (which can be solved by FORTRAN lanquage describing a computer programme) and a computer programme in FORTRAN lanquage solving the question by means of a computer.

The static analysis as well as the estimators of unknown parameters are counted on the basis of theoretical half-linear method of the smallest squares of Marquardt's type.