UNIVERSITY OF ILLINOIS
DIGITAL COMPUTER

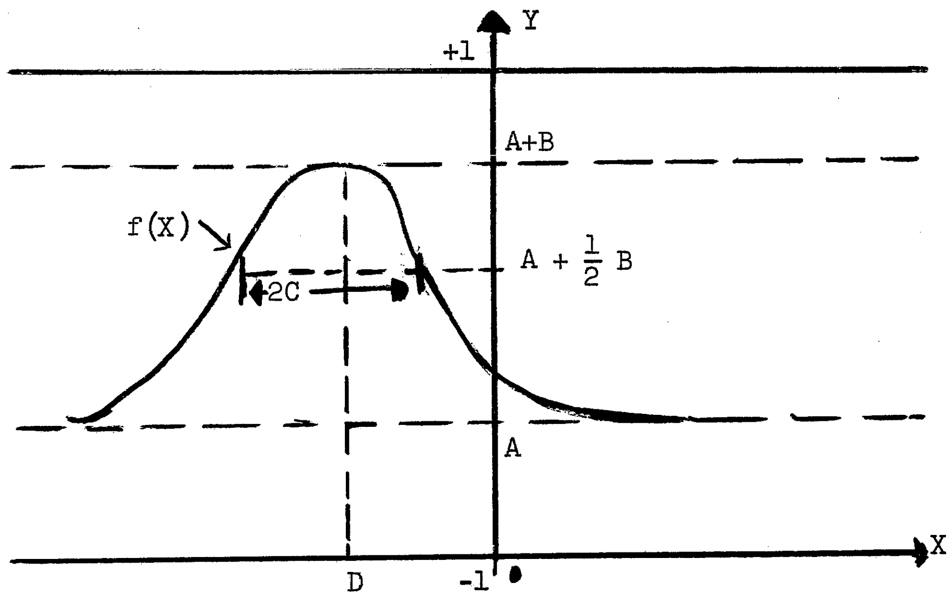
AUXILIARY
LIBRARY ROUTINE K18-330

TITLE: Least Squares Fit of a Lorentz Function to a set of Experimental Points (SADOI Only)

TYPE: Complete program

DESCRIPTION: This routine fits a set of experimental points with the function

$$f(X) = A + \frac{B}{1 + \left(\frac{X-D}{C}\right)^2} = A + \frac{BC^2}{C^2 + (X-D)^2} .$$



A is the level of the background, B is the height of the peak above background, C is half the full width at half the height of the peak, and D is that value of X for which f attains its peak value.

Optimum values of the parameters A, B, C, and D are found by minimizing the sum of the residuals R:

$$R = \sum_{i=1}^{N} [Y_i - f(X_i)]^2 W_i ,$$

where

N is the number of experimental points, $4 \le N \le 205$:,

X_i is the X-coordinate of the ith experimental point,

Y_i is the Y-coordinate of the ith experimental point,

W_i is the weight associated with the ith point.

The quantities W_i may be:

1) $W_i = (1/N)$ for all i,
2) $W_i = (1/N) Z_i$
3) $W_i = (1/N)\sqrt{Z_i}$

where the Z_i are specified by the user. The third alternative provides for the weighting of an experimental point by the square root of the number of observations or counts at that X_i.

At the start, the user provides the routine with the values X_i and Y_i (and Z_i if desired), initial **estimates** of A, B, C, and D, and two parameters, δ_1 and ϵ. To find the minimum of R, the routine proceeds as follows:

1) In order to achieve greater accuracy, the value of R used by the minimization routine is scaled up by some power P of two. Before minimization begins, the routine computes R using the given values of A_0, B_0, C_0, and D_0, and tests to see if $R_0 < 2^{-5}$. If $R_0 \ge 2^{-5}$, the computer stops on an FF002 error stop, which usually indicates either that the experimental points cannot be fit by the function f, or that the scatter of the experimental points is so great that the routine may have difficulty in determining optimum values of A, B, C, and D. If $R_0 < 2^{-5}$ P is determined so that $1/16 \le 2^P R_0 < 1/8$, and the quantity of $2^P R = R'$ is minimized.
2) R' is computed at the points $A \pm \delta_1$, $B \pm \delta_1$, $C \pm \delta_1$, $D \pm \delta_1$. From these values, the approximate direction of maximum decrease of R' is determined. The routine then changes A, B, C, and D until a point is reached where the values of R' at the points $A_1 \pm \delta_1$, $B_1 \pm \delta_1$, $C_1 \pm \delta_1$, and $D_1 \pm \delta_1$ are all greater than the values R', at (A_1, B_1, C_1, D_1).

3) The numbers R', A_1, B_1, C_1, and D_1 are then printed out by the routine. A new value of δ is computed, $\delta_2 = 1/10\ \delta_1$, and if $\delta_2 \geq \epsilon$, return to step (2) using the new δ.

4) If $\delta_{k+1} < \epsilon$, the final tolerance, the optimum values of R', A, B, C, and D are printed, followed by -P as an integer. The user can then have the routine compute and print f(X) at any desired X, using these optimum values of A, B, C, and D. Thus, the final δ_k used will be $\epsilon \leq \delta_k < 10\epsilon$. Note that $\epsilon \geq 10^{-12}$.

Because the same δ is used for each variable, this routine may be deceived into thinking it has found a minimum in all four variables. The routine will work best when the variation in R' is nearly the same for a change δ in each variable; this may be achieved by scaling the data so that $A_0 \simeq B_0 \simeq 2C_0$. Also, it is required that $|A_0 \pm \delta_1| < 1$, $|B_0 \pm \delta_1| < 1$, $|C_0 \pm \delta_1| < 1$, and $|D_0 \pm \delta_1| < 1$.

The quantity δ_1 must be chosen with some care. If δ_1 is too large, the non-linearity of the functions f and R may cause the routine to hunt in the wrong direction, in which case it will eventually hangup; if δ_1 is too small, the routine will spend too much time searching for any minimum at all. δ_1 should be large enough so that the true minimum lies within $A_0 \pm \delta_1$, $B_0 \pm \delta_1$, $C_0 \pm \delta_1$, $D_0 \pm \delta_1$ with a fairly high certainty. A good rule of thumb is to take δ to be the largest of the uncertainties in A_0, B_0, C_0, and D_0, or, 1 to 10 percent of the value of the largest of A, B, C, or D.

DIRECTIONS FOR USE

All data is read by routine N12. All numbers are fractions < 1, signed, with up to 12 decimal digits. The decimal point is assumed to lie immediately after the sign.

The data tape is prepared as follows:

1) A <u>single</u> one-hole delay
2) A title, consisting of any characters except a one-hole delay.
3) Any number of one-hole delays to terminate the title.

The routine will copy all characters between the one-hole delays onto the output tape. If no title is desired, punch <u>two</u> one-hole delays.

4) The list of values X_i, in order, punched for N 12.

5) A termination symbol, any one of N, J, F, L:

6) The list of values Y_i, in exactly the same order as the X_i. There must be one and only one Y_i for each X_i.

7) A termination symbol, one of N, J, F, L:

   a) If N, $W_i = (1/N)$ for all i. The routine then skips to step 10.

   b) If J, F, or L, a list of Z_i is to be read, at step 8.

8) The list of Z_i, in the same number and order as the X_i.

9) A termination symbol, one of N, J, F, L:

   a) If N, $W_i = (1/N) Z_i$. The routine then proceeds to step 10.

   b) If J, F, or L, $W_i = (1/N)\sqrt{Z_i}$.

10) The initial values A_0, B_0, C_0, D_0 of the variables A, B, C, D.

11) A termination symbol, one of N, J, F, L:

   a) If N, optimum values will be found for A, B, C, and D.

   b) If J, F, or L, D_0 is set to zero, and it is assumed that $D = 0$ in all further calculations.

12) The values of δ_1, the initial mesh size, and ϵ, the final tolerance.

13) A termination symbol, one of N, J, F, or L:

   a) If N, then after the final values of A, B, C, and D have been found, the routine will print back X_i, and $f(X_i)$ using the final values of A, B, C, and D. Then the routine will proceed to step 14.

   b) If J, the routine will proceed to step 14.

   c) If F, the routine will do as in (a) above, then proceed to step 16.

   d) If L, the routine skips to step 16.

14) A list of points $\overline{X}_j$. The computer will print $\overline{X}_j$ and $f(\overline{X}_j)$ as in 13a. The number of points $\overline{X}_j$ may be different from N, the number of points X_i. There may be up to 617 $\overline{X}_j$'s.

15) A termination symbol, any one of N, J, F, or L.

16) STOP on 24090. Raising the black switch returns to step 1, to read in a new data tape.

STOPS

24090 raise black switch to read data tape.

FF000 Sum check fails on readin

FF001 wrong number of variables in this list } Skip start

FF002 Initial sum of residuals exceeds 2^{-5} } goes to 24090.

NOTE

In 11b it is specified that D should be zero throughout the computation. The final results may have $D \neq 0$; this is an indication that rounding errors are at least as large as the final value of D.

TIME REQUIRED

The running time of this routine is fairly sensitive to the initial choices of A, B, C, D, and δ. A rough estimate for N points is

$$\text{Time} \doteq N[1 + \log_{10} \frac{\delta_1}{\epsilon}] \text{ seconds.}$$

To this should be added 1 second for each pair of X, f(X) values printed out.

DATE	September 22, 1961
PROGRAMMED BY	John Ehrman
APPROVED BY	J N Snyder

nj

LOCATION			ORDER	NOTES PAGE 1 K 18
Abs.	Rel.	Sym.	003K	
3			00F 0010F	S-Parameters for H4
4			00F 0015F	
5			00F 0020F	
6			00F 00 1000 0000 0000 J	
7			00F 00100F	
8			00F 00(2)	intermediate print routine
9			00F 00(1)	Function auxiliary
			0025K	
25		(N)	00F 00F	N
26			00F 00F	1/N
27			00F 00S6	1/10
28		(0)	00F 00F	Zero
29			274(12) 00F	
30			L14S5 101F	
31		(L)	00F 00(X)	Location of X_i
32			00F 00F	Y_i
33			00F 00F	W_i
34			00F 00S3	
35		(F)	00F 00F	A_0, B_0, C_0, D_0 termination
36			00F 00F	δ_1 ϵ termination
37			00F 00F	P
38		(θ1)	80F 00F	
39		(N12)	00K	Input Routine
78		(P16)	00K	Print Routine
134		(R1)	00K	Square Root Routine
			00K	
143		(100)	93191F 241S0	Main Routine
144		(99)	914F 40F	
145			L7F 36(99)	Search for 1-hole delay
146			92135F 50S0	2 CRLF
147			26(C) 92707F	Jump to tape copy
148			L52(0) 402(12)	Set up to vary D
149		(103)	50(X) 50S0	
150			26(N12) 41(N)	Input X_i

LOCATION			ORDER	NOTES PAGE 2 K 18
Abs.	Rel.	Sym.		
151			L521(N12) 46(101)	Set Y_i addresses
152			1020F 421(L)	
153			L521(N12) L0(103)	
154			1020F 42(N)	Store N
155			1938F 66(N)	
156			S5F 401(N)	Store 1/N
157		(101)	50[ ]F 50S0	Input Y_i
158			26(N12) 403F	Save termination at 3
159			41F L521(N12)	
160			L0(101) 1020F	
161			42F L5(N)	Compute number of Y_i's,
162			L0F 401F	test whether = N
163			L31F 36(102)	
164		(105)	FF1F 26(100)	Error stop, wrong number of elements
165		(102)	L521(N12) 46(104)	Set Z_i addresses
166			1020F 422(L)	
167			L33F 36(106)	
168		(104)	50[ ]F 50S0	If termination = J, F, or L
169			26(N12) 404F	input Z_i, save termination at 4
170			L521(N12) L0(104)	
171			1020F 423F	compute number of Z_i's
172			L5(N) L03F	
173			40F L3F	test
174			361S0 26(105)	hang up if unequal
175		(106)	L52(L) 42(107)	
176			42(108) L1(N)	Set Z_i addresses
177		(107)	408F L5[ ]F	Set count. Get Z_i
178			402F L33F	Check whether W_i = 1/N
179			36(109) L34F	jump if yes. Test $W_i = 1/N \sqrt{Z_i}$?
180			322S0 L52F	
181			50F 50S0	yes
182			26(R1) 502F	put $\sqrt{Z_i}$ or Z_i in Q, x 1/N
183		(108)	7J1F 40[ ]F	Store W_i
184			F5(108) 42(107)	
185			42(108) F58F	
186			36(110) 26(107)	

LOCATION			ORDER	NOTES PAGE 3 K18
Abs.	Rel.	Sym.		
187		(109)	L51(N) 22(108)	get $W_i = 1/N$
188		(110)	501S3 50S0	Input A_0, B_0, C_0, D_0
189			26(N12) 40(F)	save termination
190			L3(F) 36(111)	
191			L51(0) 402(12)	if J, F, or L, set to keep D=0
192			414S3 50F	and set $D_0 = 0$
193		(111)	503F 50S0	read δ_1 and ϵ
194			26(N12) 401F	
195			L53F 40S4	store δ_1
196			L54F 4073(H4)	store ϵ
197			F5(0) 42(15)	set one shift in auxiliary
198			L51S3 401S5	Set A_0 in place for auxiliary
199			L52S3 402S5	B_0
200			L53S3 403S5	C_0
201			L54S3 404S5	D_0
202			50F 50S0	
203			26(1) 40F	compute R_0
204		(98)	411F LJF	clear 1 for count
205			361S0 26(97)	
206			L5F 001F	shift up once
207			40F F51F	and count
208			401F 22(98)	
209		(97)	L51F L06(C)	
210			L06(C) 362S0	see if $R_0 > 2^{-3}$; jump if not
211			FF2F 26(100)	hang up. Skip start to begin again
212			L46(C) 42(15)	get to scale up
213			402(F) 50S0	
214			26(H4) L53(L)	begin minimization
215			42(112) 92139F	Set address of final values
216			92515F 194F	
217		(112)	403F L5[ ]F	
218			5012F 50S0	print final values
219			26(P16) 92131F	
220			92515F F5(112)	
221			42(112) L53F	
222			001F 36(112)	count

LOCATION Abs.	Rel.	Sym.	ORDER	NOTES PAGE 4 K18
223			L12(F) 50F	
224			521F 50S0	print P
225			26(P16) 92131F	
226			92143F L51(F)	
227			0039F 36(113)	Test for X_i, $f(X_i)$ print
228		(115)	L51(F) 0038F	
229			36(117) 26(100)	Test for $\overline{X}_j$, $f(\overline{X}_j)$ print
230		(113)	L1(N) 408F	print X_i, $f(X_i)$
231			L5(L) 42(114)	
232			50F 92131F	
233		(114)	92515F L5[ ]F	get X_i
234			407F 221S0	
235			508F 50S0	print X_i to 8 places
236			26(P16) 92967F	2 spaces
237			L57F 50S0	
238			26(1.) 406F	compute $f(X_i)$
239			5010F 50S0	print $f(X_i)$ to 10 places
240			26(P16) F58F	
241			408F 36(115)	count
242			F5(114) 221(113)	
243		(117)	50(X) 50S0	read $\overline{X}_j$
244			26(N12) L521(N12)	
245			1020F 42(118)	set end constant
246			L5(L) 42(119)	
247			92131F 92131F	
248		(119)	92515F L5[ ]F	
249			407F 221S0	
250			508F 50S0	print $\overline{X}_j$
251			26(P16) 92967F	
252			L57F 50S0	
253			26(1.) 221S0	
254			5010F 50S0	
255			26(P16) 92131F	print $f(\overline{X}_j)$
256			F5(119) 42(119)	
257			L0(118) 36(119)	
258			92191F 26(100)	

LOCATION Abs.	Rel.	Sym.	ORDER	NOTES PAGE 5 K18
259		(118)	12515F L5[ ]F	end constant
			00K	
260	0	(1.)	40F K5F	Subroutine to compute f(X)
261	1		42(1.0) 502S3	with final values of A, B, C, D
262	2		753S3 401F	
263	3		7J3S3 501F	
264	4		743S3 102F	
265	5		402F S5F	form $\frac{1}{4}$ BC^2 double precision
266	6		403F 503S3	
267	7		753S3 101F	
268	8		F4(0) 101F	form $\frac{1}{4}$ C^2, rounded
269	9		404F L5F	
270	10		101F 40F	
271	11		L14S3 101F	
272	12		L4F 40F	
273	13		50F 7JF	$\frac{1}{4}(X-D)^2 + \frac{1}{4}C^2$
274	14		L44F 40F	
275	15		L52F 503F	
276	16		66F S5F	
277	17	(1.0)	L41S3 22[ ]F	
			00K	
278	0	(2)	L53(L) 422L	Intermediate printout
279	1		194F 403F	
280	2		92515F L5[ ]F	
281	3		5012F 50S0	print R_k', A_k, B_k, C_k, D_k
282	4		26(P16) 92131F	
283	5		F52L 422L	
284	6		L53F L43F	
285	7		321L 92131F	
286	8		2669(H4) 00F	
			00K	Auxiliary subroutine to
287	0	(1)	K5F 42(11)	evaluate R' (A, B, C, D)
288	1		502S5 753S5	
289	2		40F 7J3S5	
290	3		50F 7431S5	$\frac{1}{4}$ BC^2, double precision

LOCATION			ORDER	NOTES PAGE 6 K18
Abs.	Rel.	Sym.		
291	4		102F 402F	
292	5		S5F 403F	
293	6		503S5 753S5	
294	7		101F F4(0)	
295	8		101F 404F	$\frac{1}{4} C^2$ at 4
296	9		L5(L) 42(12)	initialize list addresses
297	10		L51(L) 42(13)	
298	11		L52(L) 42(14)	
299	12		416F 497F	clear sum boxes
300	13		L1(N) 408F	set up count
301	14	(12)	50(0) L5[ ]F	get X_i
302	15		101F 40F	
303	16		[L14S5 101F]	treat D appropriately
304	17		L4F 40F	
305	18		50F 7JF	
306	19		L44F 401F	$\frac{1}{4} (X-D)^2$ at 1
207	20		L52F 503F	
308	21	(13)	661F L1[ ]F	$-Y_i$
309	22		S4F L41S5	$+ \frac{BC^2}{C^2 + (X-D)^2} + A$
310	23	(14)	405F 50[ ]F	
311	24		755F 40F	W_i
312	25		7J5F 50F	$[Y_i - f(X_i)]^2 W_i$
313	26	(15)	745F 00[ ]F	shift left P places
314	27		L46F 406F	accumulate sum
315	28		S5F L47F	
316	29		407F 363S0	
317	30		L4(-1) 407F	correct overflow
318	31		F56F 406F	
319	32		F5(12) 42(12)	advance addresses
320	33		F5(13) 42(13)	
321	34		F5(14) 42(14)	
322	35		F58F 408F	count
323	36		361S0 22(12)	
324	37	(11)	L56F 22[ ]F	exit

LOCATION Abs.	Rel.	Sym.	ORDER	NOTES PAGE 7 K18
			00K	Tape copy routine
325	0	(C)	K5F 425L	
326	1		501L 914F	
327	2		363L 503L	
328	3		026F 424L	
329	4		L06L 92[ ]F	
330	5		361L 22[ ]F	
331	6		00F 002F	
			00K	
332		(H4)		Minimization subroutine
407			00K(x) 26999N	Start of X_i list
			00996K	
996			L3F 36(100)	Test sum check
997			FFF 26(100)	Hang up; skip start to go
998			L94090F NF3564F	Sum check
			26996N	