DECUS
PROGRAM LIBRARY

DECUS NO.	FOCAL8-72
TITLE	GENERAL LEAST SQUARES FIT
AUTHOR	Harold Metcalf
COMPANY	SUNY Stony Brook Stony Brook, New York
DATE	Submitted: December 11, 1969
SOURCE LANGUAGE	FOCAL

GENERAL LEAST SQUARES FIT

DECUS Program Library Write-up DECUS No. FOCAL8-72

We have written a non-linear least squares fit that could be used to fit any curve that we specify to a set of data. We can fit a Lorentzian, a polynomial, an exponential, or any other curve to our data.

Suppose there is an experimental signal which depends on one variable, for example, light intensity as a function of wavelength and we want to fit a theoretical function characterized by (NV) parameters to it. Suppose there are (ND) data points, and we want to fit to every $(IN)^{th}$ point. (Usually IN=1 and we fit to every point, but sometimes for the sake of speed we fit to every second or third point and IN=2 or IN=3.) Label the NV parameters by V(I) where I=1,NV, and denote the theoretical function by f(x,V(I)). For example, if we wanted to fit a Lorentzian to a set of data, the V(I) would be the center, width, amplitude, and baseline. (This is the case for the sample program shown below.) The function would be f = V(3)/(1.0+(V(2)*(V(1)-x))^2) + V(4) where V(1) = center, V(2) = width, V(3) = amplitude, and V(4) = baseline. One could add a fifth parameter for some type of asymmetry, and a sixth parameter for some baseline slope, etc. The object of the least squares fit is to find values of the parameters so that the sum of the squares of all ND values,

$$\sum_{i=1}^{ND} (f_{fitted} - f_{experimental})^2 = F,$$

is a minumum. A useful way of doing this is to make a Taylor series expansion of f about each value of x for which there is a data point. Then the partial derivatives of F with respect to each of the V(I) evaluated at the x coordinate of each data point must be calculated. The partial derivatives are then multiplied in pairs, summed, and arranged to form a matrix. The inverse of this matrix is used to calculate the V(I). The routine is programmed almost directly from W. E. Wentworth's review in J. Chem. Ed., vol 42, p. 96 (1965).

In order to do the Taylor expansion we need a trial set of parameters which must be a reasonably close approximation. For an experimental signal with signal to noise ratio larger than 3, these trial parameters can often be guessed by just looking at the signal. (Statements 1.12 through 1.16 of the sample proved sufficiently accurate estimates for our particular data. The width is usually 0.1 to within a factor of 3, and this is good enough.) Statements 1.18 through 1.28 type out the initial values the program will start with, as well as the sum of the squares of the residuals. It is this sum which must be minimized, and the sample outputs which follow the program indeed show this sum

converging to a minimum. Statement 1.30 is the main control statement: QQ is indexed for six iterations; group 2 evaluates the partial derivatives, finds the normal matrix, inverts it (via the D 9 of 2.18), and calculates the new values of the V(I). The rest of group 1 evaluates the uncertainties of the parameters after the last iteration.

The Lorentzian line shape is evaluated in group 3. It should be apparent that any line shape could be used, and it could be described by any (reasonable) number of parameters. A spectrum with multiple peaks can be used just as easily; it is only necessary to specify the centers at the start to an accuracy of about a line width. The G(J) and SI which group 3 calculates are different because one of the V(I) has been varied by DV(I). The DV(I)'s are set in 2.02. The difference of G(J) and SI divided by DV(I) is the partial derivative of F with respect to V(I) (statement 2.6). These partial derivatives are multiplied and then summed to form the matrix B = A. Matrix B is saved but could be eliminated if more space were needed.

Group 3 could be as long and complicated as necessary. For example, if the data are obtained using modulation of the x variable and a phase sensitive detector; group 3 could evaluate the function f at several different points along the modulation wave form, rectify the ac signal thus obtained, integrate it, amplify it, and use the result to compute the partial derivatives. If the modulation amplitude is small compared to the structure of the signal, then the modulated signal is simply the derivative but usually this is not the case. The lockin amplifier simulation described here will easily separate the natural width of the line from modulation broadening even for large modulation amplitudes.

The program has been written for equally spaced values of the x variable, and for equal weight on each data point. Neither of these restrictions is necessary. In order to use arbitrary x values for a Lorentzian line, simply replace the I in 3.01 by X(I) and alter 1.05 to get the X's into the machine, otherwise, alter group 3 as necessary. There is room altogether for about 120 numbers. If both X(I) and Y(I) are measured, the limit is about 55 to 60 data points. I have recently received an overlay to FOCAL 69 called MODV written by Arnold Fish of DEC. It enables FOCAL to store variables in upper core as well as text and thereby lifts this restriction. It allows about 500 points or 250 pairs, etc. However, for any reasonable theoretical curve, this is usually more than an adequate number of points (this is a PDP8/L, not a 360!!) The data can

be weighted by multiplying the G(J) and SI in group 3 by the respective weights (see Wentworth's paper), but remember if the weights are also being stored it cuts down further on the number of data points that can be fitted.

The program is shown below with a few examples of its output. The last few characters on line 1.21 are SUM",! but they have been overtyped. FOCAL is perfectly happy to read such a line as you can see from the titles on the output. The program, needless to say, cannot be used with 4K FOCAL.

```
C-8K  MODV 11-219

Ø1.Ø5  *;A J;*;F I=1,99;*;A Y(I);*
Ø1.1Ø  S  ND=99;S NV=5;S SU=Ø;S LA=1;S N=NV; S IN=1
Ø1.12  F  I=1,IN,ND; S SU=SU+Y(I); IF (Y(I)-Y(LA)) 1.14,1.14; S LA=I
Ø1.14  C  FOR INVERSE LINE (DECREASE AT CENTER), 'IF (Y(LA)-Y(I))' IN 1.12
Ø1.16  S  V(1)=LA; S V(2)=.1; S V(4)=SU*IN/ND; S V(3)=Y(LA)-V(4); S V(5)=Ø
Ø1.18  S  SU=Ø; S Q=1; F I=1,IN,ND; D 3;S SU=SU+(SI-Y(I))↑2
Ø1.21  T  !!!!, "    CENTER    WIDTH    AMPLI    BASE    DISPER    SUM
Ø1.26  F  I=1,NV;T %7.Ø4,V(I)
Ø1.28  T  %12.Ø4,SU,!
Ø1.3Ø  F  QQ=1,6;D 2;D 1.26;D 1.28
Ø1.32  F  J=1,NV;S DV(J)=FSQT(A((NV+1)*(J-2)+(NV+2))*SU/(ND-NV))
Ø1.34  T  !!, "UNCERTAINTIES IN LAST ITERATION",!;F I=1,NV;T %7.Ø4,DV(I)
Ø1.35  S  D=V(3)/FSQT(SU/(ND/IN))
Ø1.38  T  !!,"SIGNAL TO NOISE RATIO",D,!!!!!!!!!!
Ø1.4Ø  GO TO 1.Ø5

Ø2.Ø2  F  J=1,NV; S C(J)=Ø.Ø; S DV(J)=.Ø1*V(J); S DV(5)=.Ø1
Ø2.Ø3  F  J=1,NV; F K=1,NV; S B(NV*(J-1)+K)=Ø.Ø
Ø2.Ø4  F  I=1,IN,ND; S Q=1; D 3; D 2.Ø8; F J=1,NV; F K=1,NV;D 2.14
Ø2.Ø5  G  2.18
Ø2.Ø8  F  J=1,NV; S V(J)=V(J)+DV(J);S Q=2; D 3;D 2.12
Ø2.12  S  V(J)=V(J)-DV(J); S C(J)=C(J)+( (G(J)-SI)/DV(J))*(SI-Y(I))
Ø2.14  S  H=NV*(J-1)+K; D 2.16
Ø2.16  S  B(H)=B(H)+((G(J)-SI)/DV(J))*((G(K)-SI)/DV(K)); S A(H)=B(H)
Ø2.18  D  9
Ø2.22  F  J=1,NV; F K=1,NV; S V(J)=V(J)-A(NV*(J-1)+K)*C(K)
Ø2.24  S  SU=Ø.Ø;S Q=1; F I=1,IN,ND;D 3;S SU=SU+(SI-Y(I))↑2
Ø2.32  S  N=N

Ø3.Ø1  S  Z=(I-V(1))*V(2)
Ø3.5Ø  S  D=V(3)*(1+V(5)*Z)/(1+Z*Z)+V(4)
Ø3.7Ø  IF (Q-1.5) 3.72;S  G(J)=D;G 3.9
Ø3.72  S  SI=D
Ø3.9Ø  S  N=N
```

THIS IS TO BE FOLLOWED BY MATRIX INVERSION IN GROUPS 9 AND 10.

CENTER	WIDTH	AMPLI	BASE	DISPER	SUM
= 5Ø.ØØØØ=	Ø.Ø7ØØ=	1ØØØØ.ØØ=	5ØØØØØ.Ø=	Ø.ØØØØ=	865476ØØØØ.ØØ
= 47.5734=	Ø.Ø645=-4617.8ØØ=		496931.Ø=	Ø.15Ø4=	8355920Ø.ØØØØ
= 55.7263=	Ø.Ø767=-4781.84Ø=		497Ø24.Ø=-	Ø.34Ø7=	5127250Ø.ØØØØ
= 55.Ø333=	Ø.Ø822=-4428.9ØØ=		496ØØ9.Ø=-	Ø.3426=	4782ØØØ.ØØØØ
= 54.9728=	Ø.Ø825=-4453.64Ø=		496924.Ø=-	Ø.3412=	477934Ø0.ØØØØ
= 54.9666=	Ø.Ø826=-4454.72Ø=		496924.Ø=-	Ø.34Ø8=	47793ØØØ.ØØØØ
= 54.9718=	Ø.Ø826=-4453.44Ø=		496924.Ø=-	Ø.3414=	477934Ø0.ØØØØ

UNCERTAINTIES IN LAST ITERATION

= 1.Ø721=	Ø.ØØ92=	264.751Ø=	167.3Ø1Ø=	Ø.Ø848	

SIGNAL TO NOISE RATIO=- 6.4Ø96

THIS IS OD #21

CENTER	WIDTH	AMPLI	BASE	DISPER	SUM
= 55.ØØØØ=	Ø.1ØØØ=	28.799Ø=	17.3Ø1Ø=	Ø.ØØØØ=	8149.41ØØ
= 51.3412=	Ø.1687=	28.9271=	14.18Ø6=	Ø.4364=	1Ø91.86ØØ
= 52.5412=	Ø.1922=	3Ø.7475=	12.6695=	Ø.4546=	5Ø2.852Ø
= 52.3598=	Ø.1997=	31.3524=	12.7Ø94=	Ø.4656=	495.826Ø
= 52.3858=	Ø.1995=	31.4417=	12.6915=	Ø.4623=	495.795Ø
= 52.3862=	Ø.1995=	31.4371=	12.6922=	Ø.4623=	495.796Ø
= 52.3862=	Ø.1995=	31.4369=	12.6922=	Ø.4623=	495.796Ø

UNCERTAINTIES IN LAST ITERATION

= Ø.2631=	Ø.Ø111=	1.2754=	Ø.32Ø1=	Ø.Ø492

SIGNAL TO NOISE RATIO= 14.Ø477

We have found this program to be extremely flexible. To begin, it is easy to alter the statements in group 1 which select the initial parameters from which the program starts. Also, a single, short, alteration is all that is necessary to switch from a Lorentzian to an inverted Lorentzian (see 1.14).

Sometimes it is obvious that the program is not converging and that it should be started with different values of the trial parameters. This can be done as follows: give the direct command F I=1,NV;A V(I). CR. Then FOCAL will ask for the V(I)'s, and after the last one has been typed in, it will type an *. The give it GOTO 1.18.

Frequently our data shows a slight asymmetry and we cannot tell whether our

Lorentzian should include an asymmetric component (as shown here) or whether we have a sloping baseline arising from some instrumental effect. It is very easy to check this. Simply alter group 3 so that the lineshape has a sloping baseline instead of an asymmetric component, and, after convergence, examine the sum of the squares of the residuals and compare it with the sum from the asymmetric case. If they are within a few percent of one another, then there is no significant distinction between either of the explanations for the asymmetry; however, a sizeable difference between the sums implies that one of the asymmetries is a considerably more accurate description of the data than the other one.

It is amusing to produce a set of data which is not a Lorentzian and ask the program to fit it anyway. For example, one could use a Gaussian, a Lorentzian convolved with a Gaussian, a sum of two Lorentzians with different widths and strengths, etc. If an experimenter suspects some sort of adulteration of his signal, it is a simple matter to adulterate a pure signal in the computer and feed it back to the fitting program to determine the effect of the adulteration. This simulation of experimental artifacts is very useful with complicated lineshapes, especially when the S/N is not extremely high.