

Internal Report No. 5

Identification: LSQFT/General Least-Squares Fitting Subroutine

Revised From a Tested Program of Unknown Origin

M. Radomski

August 1968

Purpose

This program least-square fits a functional form given by the user in a subroutine to data consisting of up to 300 X and Y values (with associated statistical error bars on the Y's). Parameters need not enter linearly into the functional form (e.g. $y = x^{p_1} + p_2$); however, if they do, the iterative numerical method used will converge to an analytically correct answer in one iteration. Printout at each iteration includes: current values of parameters, most recent trial increment in parameters, current value of chi-squared for the fit. Final printout includes: least square values of parameters, statistical error bars on those values, and chi-squared for the least square fit. These final values and the error matrix of the parameters are available to the calling program through labeled common.

Method

The user supplies, by means of a subroutine, a functional form for theoretical values of the dependent variable y depending on the parameters p_1, p_2, \dots, p_n and on x :

$$y = F(p_1, \dots, p_n, x).$$

The program is to find the values of p_1, \dots, p_n which minimize chi-squared

for the given set of x_i, y_i, σ_i for $i = 1, 2, \dots, \text{NPT}$.

$$\chi^2(\bar{p}) = \sum_{i=1}^{\text{NPT}} (y_i - F(\bar{p}, x_i))^2 / \sigma_i^2$$

This is done by numerically solving the equations

$$\frac{\partial \chi^2}{\partial p_m}(\bar{p}) = 0; m = 1, 2, \dots, N$$

i.e.

$$\sum_{i=1}^{\text{NPT}} \left(\frac{y_i - F(\bar{p}, x_i)}{\sigma_i} \right)^2 D_{im}(\bar{p}) = 0; m = 1, 2, \dots, N \quad (1)$$

where D_{im} are defined as

$$D_{im}(\bar{p}) = \frac{\partial F}{\partial p_m}(\bar{p}, x_i); m = 1, \dots, N; i = 1, \dots, \text{NPT}$$

In (1) now only p_1, p_2, \dots, p_N are variables.

Numerical solution is carried out by a multivariable analog of Newton's method. F and D are approximated to first order in the neighborhood of initial guesses p_1^0, \dots, p_n^0 supplied by the calling program, i.e. where we denote $F(\bar{p}^0, x_i)$ by F_i^0 and $D_{im}(\bar{p}^0)$ by D_{im}^0 :

$$F(\bar{p}, x_i) = F_i^0 + \sum_{m=1}^n D_{im}^0(\delta p_m) + O(\delta p^2); i = 1, \dots, \text{NPT} \quad (2)$$

$$D_{im}(\bar{p}) = D_{im}^o + \sum_{j=1}^N \frac{\partial D_{im}}{\partial p_j} (\bar{p}^o) (\delta p_j) + o(\delta p^2); \quad i = 1, \dots, \text{NPT}; \quad (3)$$

$$m = 1, \dots, N.$$

Here we have used

$$\delta \bar{p} = \bar{p} - \bar{p}_o.$$

Substitution of (2) into (1) yields equation (4) below; of both (2) and (3) into (1) yields equation (5) correct to first order in $\delta \bar{p}$:

$$\sum_{i=1}^{\text{NPT}} \frac{y_i - F_i^o}{\sigma_i^2} D_{im}(\bar{p}) = \sum_{i=1}^{\text{NPT}} \sum_{\ell=1}^N \frac{D_{im}^o D_{i\ell}^o}{\sigma_i^2} (\delta p_\ell); \quad m = 1, \dots, N \quad (4)$$

$$\sum_{i=1}^{\text{NPT}} \frac{y_i - F_i^o}{\sigma_i^2} D_{im}^o = \sum_{i,\ell} \left[\frac{D_{im}^o D_{i\ell}^o}{\sigma_i^2} + \frac{y_i - F_i^o}{\sigma_i^2} \frac{\partial D_{im}}{\partial p_\ell} (\bar{p}^o) \right] (\delta p_\ell); \quad m = 1, \dots, N. \quad (5)$$

Defining the NPT dimensional vector \bar{X} and the N by N dimensional matrix \bar{M} by

$$X_i = (y_i - F_i^o) / \sigma_i^2; \quad i = 1, \dots, \text{NPT}$$

$$M_{m\ell} = \sum_{i=1}^{\text{NPT}} D_{im}^o D_{i\ell}^o / \sigma_i^2; \quad m, \ell = 1, \dots, N$$

(4) becomes

$$\bar{X} \bar{D} (\bar{p}) = \bar{M} (\delta \bar{p}). \quad (4')$$

But if the terms in (5) in $\partial D_{im} / \partial p$ are dropped (5) becomes

$$\bar{X} \bar{D}^o = \bar{M} (\delta \bar{p}). \quad (5')$$

The dropping of these terms is seen to be equivalent to replacing $\bar{D} (\bar{p})$ in (4)' by $\bar{D} (\bar{p}^o)$.

The program, since it cannot know the values of \bar{D} at the unknown true least square fit values of \bar{p} , computes $\delta \bar{p}$ using (5)'. Dropping the terms in $\partial D_{im} / \partial p$ seems justifiable if $D_{im} (\bar{p})$ varies slowly compared to $F (\bar{p}; x_i)$. Although this omission may affect the rate of convergence of the iterative method, it should not affect the values of p_1, \dots, p_n converged to, since at $\bar{p}^o = \bar{p}$, $\bar{D} (\bar{p}^o) = D (\bar{p})$.

The values of $\delta \bar{p}$ which, according to (5)', produce a $\bar{p} = \bar{p}^o + \delta \bar{p}$ nearly satisfying equation (1) are given by

$$\delta \bar{p} = \bar{M}^{-1} \bar{X} \bar{D}^o.$$

A new set of trial values p_1', \dots, p_n' is computed according to the following scheme: for $j = 1, \dots, N$

$$p_j^1 = p_j^o + \delta p_j \times \text{DAMP} (j)$$

unless

$$|\delta p_j' - \delta p_j| \geq 2 |\delta p_j'|$$

in which case

$$p_j^1 = p_j^0 + \delta p_j \times \text{DAMP}(j)/2$$

where DAMP (j) are damping factors given by the calling program and δp_j^1 is the delta of p_j from the previous iteration. The algorithm is repeated from the beginning using p_j^1 in place of p_j^0 until convergence criteria are met or until MAXCYC iterations have been executed.

Convergence criteria depend on the error matrix. When \bar{p}^0 is at least square value of \bar{p} , then \bar{M}^{-1} is the error matrix for p_1, \dots, p_n . In particular the error bar on p_j for $j = 1, \dots, N$ is $\sqrt{\bar{M}_{jj}^{-1}}$. Convergence is judged to have occurred when

$$\delta p_j \leq \text{TEST} \times \sqrt{\bar{M}_{jj}^{-1}}; j = 1, \dots, N.$$

where TEST is a number (presumably less than 1) predetermined by the calling program.

If a singular matrix \bar{M} is encountered, an error message is printed and control returns to the calling program. FLAG (see below) is set to -1.0.

Usage

CALL LSQFT (FUNCT, FLAG)

FLAG is (before calling): zero if all DAMP (J) are desired to be 1 (normal usage), in which case they need not be set to 1 by the calling program.

: non-zero if the calling program has supplied N values of DAMP (J) and any of them is desired to be not 1.

(after calling): unchanged if no singular matrix \bar{M} is encountered on this call.

: -1.0 if a singular \bar{M} was encountered.

FUNCT is a subroutine whose name is mentioned in an EXTERNAL statement in the calling program. FUNCT must compute values of $F(I)$, $I = 1, \dots, NPT$; $D(I, M)$, $I = 1, \dots, NPT$, $M = 1, \dots, N$; PARTIT (J), $J = 1, \dots, 2N$ (see below). FUNCT must contain the common block FFUN described below, identical to that required in the calling program.

The calling program must contain the following labeled common blocks NFUN, FFUN and OUT (OUT block is optional, may be omitted if results of ERRMAT AND CHISQ are not needed by calling program; CHISQ has been printed out by LSQFT, in any case).

```
COMMON/NFUN/YA(300), YASIG(300), TEST,  
      MAXCYC, N, DAMP(10), PSIG(10)  
      /FFUN/NPT, XA(300), PARTIT(20), P(10),  
      F(300), D(300, 10)  
      /OUT/ERRMAT(50, 10), CHISQ
```

where

YA(I) ($I = 1, \dots, NPT$) are the values of the dependent variable at each of the NPT data points.

YASIG(I) ($I = 1, \dots, NPT$) are the errors assigned to the corresponding

YA (I)

TEST is factor in convergence criterion (see above)

MAXCYC is maximum number of iterations (see above)

N is number of parameters to be varied

DAMP contains damping factors (see above)

DAMP(I) multiplies the increment at each iteration of P(I). Normally all DAMP(I) are one; this is signalled to LSQFT by FLAG = 0.0 without the necessity of setting all DAMP(I) = 1.0. Damping factors less than 1.0 might cause some problems to converge which might not otherwise, but will slow convergence for normal problems.

PSIG is set by LSQFT to the values of parameter errors according to $PSIG(J) = M_{jj}^{-1}$; $J = 1, \dots, N$.

NPT is number of data points

XA(I) ($I = 1, \dots, NPT$) are the values of independent variables.

PARTIT(K) ($K = 2I - 1, 2I$) must be set by the FUNCT subroutine to contain the parameter title of the Ith parameter (two six-character alphameric words).

P(I) ($I = 1, \dots, N$) must be given by the calling program as initial guesses at parameter values. LSQFT returns in these locations the corresponding correct least square fit values.

F(I) ($I = 1, \dots, NPT$) must be set by FUNCT to correspond to $F(\bar{p}^0, x_I)$ using the values of \bar{p}^0 given by P and x_I given by XA in common block FFUN.

D(I, M) ($I = 1, \dots, NPT$; $M = 1, \dots, N$) must be set by FUNCT to correspond to D_{IM}^0 using \bar{p}^0 given by p and x_I given by XA in common block FFUN.

ERRMAT (I, J) (I, J = 1, ..., N) is the error matrix M_{IJ}^{-1} after calling of LSQFT.

CHISQ is set by LSQFT to the value of chi-squared for the least-squares fit.

Of the above common variables, the following must be set by the calling program, and are unaltered by LSQFT: YA, YASIG, TEST, MAXCYC, N, NPT and XA. The array DAMP must be set by the calling program only if FLAG \neq 0.0. (If FLAG = 0.0, DAMP(I) is set to 1.0 for all I by LSQFT.) The array P must be set by the calling program (to initial guesses for parameter values) and is altered by LSQFT (to final least square fit values). The arrays ERRMAT and PSIG, and CHISQ, are given values by LSQFT. F and D are for internal use of LSQFT and FUNCT.

Notes on Use

Probably the best way for FUNCT to set PARTIT in FORTRAN language is to define another array, say PRTIT, in a DATA statement, and there set PARTIT (I) = PRTIT (I) for I = 1, ... 2N.

A 7094 binary deck and a FORTRAN deck of LSQFT are available, and listings of a sample job.

A modification of the method for use in fitting pulse-height data to a Gaussian exists (see Steve Murray).