
Subject: Re: Fitting curves

Posted by [agrap](#)s on Sat, 03 Sep 1994 00:52:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

larkum@optolab.unibe.ch (Matthew Larkum) writes:

> Hi,

> I know this goes around once in a while but I'm really confused about

> how to use the CURVEFIT procedure.

> I would really like to see an example and copy that. Is there some kind

> soul out there who would has an example from beginning to end for the

> dummies like me?

> Thanks heaps,

> Matthew.

> larkum@optolab.unibe.ch

Mathew:

Here is a test I worked up seven years (! my how time flies) ago when I was concerned with how CURVEFIT from VAX VMS v1.0 IDL compared to the Numerical Recipes nonlinear least-squares MRQMIN procedure. I very badly needed to have the covariance matrix output from CURVEFIT in the modeling I was doing, and the standard IDL CURVEFIT didn't have it. I also found that the coefficients weren't quite right- once the chi-square criteria was satisfied, a final call of the function needed to occur with the current coefficients. I calculate the covariance matrix at this final pass.

It was a minor change, but I succeeded in getting the same results as MRQMIN. I don't know if IDL v3.6.1 CURVEfit has evolved, because I've always used this version (and variations, thereof). So this example should help you see the steps, and feel free to use my version of CURVEFIT.

Amara

Amara Graps

Computational Physicist

Intergalactic Reality / NASA-Ames Research Center

agrap@netcom.com

graps@clio.arc.nasa.gov

```

-----cut here-----
,*****
,
PRO FUNCT, X,A,F,PDER
;This function is necessary to run IDL's CURVEFIT procedure
;A. Graps 10-29-87
;
;
F = A(0) + A(1) * X + A(2) * ALOG(X)
;
;Need partial derivatives
PDER = FLTARR(N_ELEMENTS(X),3)
PDER(*,0) = 1
PDER(0,1) = X
PDER(0,2) = ALOG(X)
RETURN
END
,*****
,
PRO CURVEFITTEST,XDATA,YDATA,FIT,A,SIGMAA,COVAR
;PURPOSE: To test IDL's CURVEFIT procedure. All of the parameters
;are OUTPUT.
;Amara Graps 11-3-87
;
;
XDATA=FLTARR(6)
YDATA=FLTARR(6)
;
;Input the values for XDATA and YDATA
;Generate Independent data array
FOR I=0,5 DO XDATA(I) = I+1
;Generate Dependent data array
YDATA(0) = 2.6
YDATA(1) = 2.4
YDATA(2) = 3.0
YDATA(3) = 4.1
YDATA(4) = 5.4
YDATA(5) = 6.6
;
;Set the weights (Here, the weights = 1, i.e. no weighting)
W = REPLICATE(1.,6)
;
;Give the initial estimates for A
A = [.75,1.70,-2.45]
;
;Call the Curvefit Procedure
CURVEFIT,XDATA,YDATA,W,A,SIGMAA,FIT,COVAR
;
;Print results
PRINT, 'Calculated A= ',A

```

```

PRINT, 'Covariance Matrix= '
PRINT, COVAR
PRINT, 'Standard Deviation of Fit Parameters= '
PRINT, SIGMAA

;
;
END
,*****
,

PRO CURVEFIT, X, Y, W, A, SIGMAA, YFIT, COVAR
;+
; NAME:
; CURVEFIT
; PURPOSE:
; Non-linear least squares fit to a function of an
; arbitrary number of parameters.
; Function may be any non-linear function where
; the partial derivatives are known or can be approximated.
; CATEGORY:
; E2 - Curve and Surface Fitting
; CALLING SEQUENCE:
; CURVEFIT,X,Y,W,A,SIGMAA,YFIT,COVAR
; INPUTS:
; X = Row vector of independent variables.
; Y = Row vector of dependent variable, same length as x.
; W = Row vector of weights, same length as x and y.
; For no weighting
; w(i) = 1., instrumental weighting w(i) =
; 1./y(i), etc.
; A = Vector of nterms length containing the initial estimate
; for each parameter. If A is double precision, calculations
; are performed in double precision, otherwise in single prec.
;
;
; OUTPUTS:
; A = Vector of parameters containing fit.
; Function result = YFIT = Vector of calculated
; values.
; Covariance matrix= error of YFIT showing correlations
; Sigmaa = Vector of standard deviations for parameters
; A.
;
;
; COMMON BLOCKS:
; NONE.
; SIDE EFFECTS:
; The function to be fit must be defined and called FUNCT.
; For an example see FUNCT in the IDL User's Library.
; Call to FUNCT is:
; FUNCT,X,A,F,PDER

```

```

; where:
; X = Vector of NPOINT independent variables, input.
; A = Vector of NTERMS function parameters, input.
; F = Vector of NPOINT values of function, y(i) = funct(x(i)), output.
; PDER = Array, (NPOINT, NTERMS), of partial derivatives of funct.
; PDER(I,J) = Derivative of function at ith point with
; respect to jth parameter. Optional output parameter.
; PDER should not be calculated if parameter is not
; supplied in call (Unless you want to waste some time).
; RESTRICTIONS:
; NONE.
; PROCEDURE:
; Copied from "CURFIT", least squares fit to a non-linear
; function, pages 237-239, Bevington, Data Reduction and Error
; Analysis for the Physical Sciences.
;
; "This method is the Gradient-expansion algorithm which
; compines the best features of the gradient search with
; the method of linearizing the fitting function."
;
; Iterations are perform until the chi square changes by
; only 0.1% or until 20 iterations have been performed.
;
; The initial guess of the parameter values should be
; as close to the actual values as possible or the solution
; may not converge.
;
; MODIFICATION HISTORY:
; Written, DMS, RSI, September, 1982.
; Modified ;to output covariance matrix, ALG ,August 1987
;
;-----
ON_ERROR,2 ;RETURN TO CALLER IF ERROR
A = 1.*A ;MAKE PARAMS FLOATING
NTERMS = N_ELEMENTS(A) ;# OF PARAMS.
NFREE = (N_ELEMENTS(Y)<N_ELEMENTS(X))-NTERMS ;Deps of freedom
IF NFREE LE 0 THEN STOP,'Curvefit - not enough points.'
DIAG = INDGEN(NTERMS)*(NTERMS+1) ;SUBSCRIPTS OF DIAGONAL ELEMENTS
;
FOR ITER = 1,20 DO BEGIN ;Iteration loop
;
; EVALUATE ALPHA AND BETA MATRICIES.
;
FUNCT,X,A,YFIT,PDER ;COMPUTE FUNCTION AT A.
BETA = (Y-YFIT)*W # PDER
ALPHA = TRANSPOSE(PDER) # (W # (FLTARR(NTERMS)+1)*PDER)
;
CHISQ1 = TOTAL(W*(Y-YFIT)^2)/NFREE ;PRESENT CHI SQUARED

```

```

;
; INVERT MODIFIED CURVATURE MATRIX TO FIND NEW PARAMETERS.
;
REPEAT BEGIN
  C = SQRT(ALPHA(DIAG) # ALPHA(DIAG))
  ARRAY = ALPHA/C
  ARRAY(DIAG) = (1.+FLAMBDA)
  ARRAY = INVERT(ARRAY)
  B = A+ ARRAY/C # TRANSPOSE(BETA) ;NEW PARAMS
  FUNCT,X,B,YFIT ;EVALUATE FUNCTION
  CHISQR = TOTAL(W*(Y-YFIT)^2)/NFREE ;NEW CHISQR
  FLAMBDA = FLAMBDA*10. ;ASSUME FIT GOT WORSE
  ENDREP UNTIL CHISQR LE CHISQ1
;
  FLAMBDA = FLAMBDA/100. ;DECREASE FLAMBDA BY FACTOR OF 10
  A=B ;SAVE NEW PARAMETER ESTIMATE.
  PRINT,'ITERATION =',ITER,' ,CHISQR =',CHISQR
  PRINT,A
  IF ((CHISQ1-CHISQR)/CHISQ1) LE .001 THEN GOTO,DONE ;Finished?
  ENDFOR ;ITERATION LOOP
;
  PRINT,'CURVEFIT - Failed to converge'
;
DONE: FUNCT,X,A,YFIT,PDER
  ALPHA = TRANSPOSE(PDER) # (W # (FLTARR(NTERMS)+1)*PDER)
  COVAR = INVERT(ALPHA)
  SIGMAA = SQRT(COVAR(DIAG)) ;RETURN SIGMA'S

END
.*****
,

```
