Subject: Re: CURVEFIT.PRO standard deviations?
Posted by thompson on Mon, 13 May 2002 17:46:30 GMT
View Forum Message <> Reply to Message

I tested this with one of my own programs, under IDL/v5.4, and got the
following:

```
LINFIT parameters, sigma, and chi-square :
    -7.67345      2.36052
     11.8136     0.843827
     21290.1
CURVEFIT parameters, sigma, and chi-square :
    -7.67969      2.36077
    0.388235    0.0277367
     925.657
LSTSQR parameters, sigma, and chi-square (w/o):
     -7.6734443       2.3605204
      11.813374      0.84381246
      925.65655
LSTSQR parameters, sigma, and chi-square (with):
     -7.6734443       2.3605204
     0.38828358     0.027734542
      925.65655
```

My chi-squared values agree with those from CURVEFIT. I can replicate
either the LINFIT or CURVEFIT errors depending on how I define the
weights to be applied to the data. The two cases are described below

Case 1: No errors passed

In the LINFIT program, as in my own, the errors are passed in through an
optional keyword. (This is my "w/o" case.) For LINFIT, this is done
through the keyword MEASURE_ERRORS. When this keyword is omitted, the
LINFIT program tries to estimate the errors in the data based on the
reduce chi-squared values. This behavior in LINFIT is clearly discussed
in the documentation:

```
;       Note: if MEASURE_ERRORS is omitted, then you are assuming that the
;             linear fit is the correct model. In this case,
;             SIGMA is multiplied by SQRT(CHISQ/(N-M)), where N is the
;             number of points in X. See section 15.2 of Numerical Recipes
;             in C (Second Edition) for details.
```

Apparently it manages to do this correctly, even though it reports a
completely bogus chi-squared. I believe that someone mentioned that the
chi-squared problem is fixed in v5.5?

Case 2: Error bars passed.

In the CURVEFIT program, the error bars are passed in through the WEIGHTS parameter in the calling sequence. There doesn't seem to be any way to make this optional. When you put in a WEIGHTS array of all ones, you're saying that the error in each point is +/- 1.0. If you put in more realistic weights, you'd get a more realistic error and chi-squared. With realistic weights, you should get a chi-squared of about one.

I was able to replicate your CURVEFIT results in my own program by passing in an array of error bars of unity. The same happens with LINFIT (except for chi-squared), when the MEASURE_ERROR is passed with all ones.

In short, LINFIT returned the correct SIGMAA values, but the wrong CHISQR. CURVEFIT would have returned the correct SIGMAA and CHISQR values if an appropriate WEIGHTS array had been passed.

William Thompson

P.S. Did you really intend to define your errors the way you did? Your errors vary with position, and go to exactly 0 at X=5. I would have expected something like

```
 f0 = (findgen(25)*3.-15) + randomn(seed,25)*noise
```

instead of

```
 f0 = (findgen(25)*3.-15)*(1.+randomn(seed,25)*noise)
```

Ralf Flicker <rflicker@gemini.edu> writes:

> Folks, I'm at wits' end here, don't know what's going on.

> I need to do a chi-square minimization curve fitting of a nonlinear
> parametrized function to noisy data. Using the routine CURVEFIT.PRO
> works admirably for the fitting itself, and the chi-square comes out
> ok, but the values returned for the standard deviations (sigma
> optional argument) on the fitted coefficients just don't make sense
> to me.

> Looking at the source code (idl 5.3) and comparing with the
> Levenberg-Marquardt algorithm in Numerical Recipes (ch 15.5), I am
> still unable to pinpoint the error (whether in my code or in my

> admittedly lacking understanding of chi-square statistics..).

> So here's a code I wrote (FTEST.PRO) for testing CURVEFIT versus a
> routine I know works, LINFIT. It generates a noisy straight line,
> fits a line using both LINFIT and CURVEFIT, and plots both fits
> together with the fits offset by one sigma on the coefficients.

> You can see the discrepancy in the one-sigma lines: can someone tell
> me what's up with the sigma returned from CURVEFIT, and how I can
> make them conform?

> ralf

```
> ;=============================================
> ; straight line
> pro fline,x,a,f,pder
> f = a(0)+x*a(1)
> end

> ;=============================================
> ; Test of CURVEFIT vs. LINFIT
> pro ftest,noise

> ; sample calling sequence : ftest,0.4

>  ; generate noisy line
> f0 = (findgen(25)*3.-15)*(1.+randomn(seed,25)*noise)
> x = findgen(25)
> loadct,4
> plot,x,f0,xstyle=2,psym=-1

>  ; straight-line chi-square fitting with LINFIT.PRO
> a = linfit(x,f0,chisq=csq,sigma=sig)
> fline,x,a,f & oplot,x,f,color=80
> fline,x,[a+sig],f & oplot,x,f,linestyle=2,color=80
> fline,x,[a-sig],f & oplot,x,f,linestyle=2,color=80
> print,'LINFIT parameters, sigma, and chi-square : '
> print,a,sig,csq

>  ; Levenberg-Marquardt chi-square fitting
>  ;of straight line with CURVEFIT.PRO
> a = [0.,1.]
> weights = fltarr(n_elements(x))+1.
>  f=curvefit(x,f0,weights,a,sig,function_name='fline',chisq=cs q,/noderivative)
> oplot,x,f,color=200
> oplot,x,a(0)+sig(0)+x*(a(1)+sig(1)),linestyle=2,color=200
```

```
> oplot,x,a(0)-sig(0)+x*(a(1)-sig(1)),linestyle=2,color=200
> print,'CURVEFIT parameters, sigma, and chi-square : '
> print,a,sig,csq

> end
> ;=============================================

> (recombine as necessary)
```