Subject: MPFITFUN Problem
Posted by Sean[1] on Fri, 22 Jul 2005 21:55:32 GMT
View Forum Message <> Reply to Message

Hello

I am having a problem with MPFITFUN. The code executes okay, but one of the parameters that I put into the program does not ever change... I know this is alot of text, but here's the output I receive on the screen....

```
IDL> imodfit
Iter      1   CHI-SQUARE =       16.926937        DOF = 3
   P(0) =          0.0250000
   P(1) =            75.0000
Iter      2   CHI-SQUARE =       16.857964        DOF = 3
   P(0) =          0.0250000
   P(1) =            76.9480
Iter      3   CHI-SQUARE =       16.855946        DOF = 3
   P(0) =          0.0250000
   P(1) =            77.0617
Iter      4   CHI-SQUARE =       16.855669        DOF = 3
   P(0) =          0.0250000
   P(1) =            77.0102
Iter      5   CHI-SQUARE =       16.855644        DOF = 3
   P(0) =          0.0250000
   P(1) =            77.0369
Iter      6   CHI-SQUARE =       16.854858        DOF = 3
   P(0) =          0.0250000
   P(1) =            77.0236
Iter      7   CHI-SQUARE =       16.854755        DOF = 3
   P(0) =          0.0250000
   P(1) =            77.0235
Iter      7   CHI-SQUARE =       16.854755        DOF = 3
   P(0) =          0.0250000
   P(1) =            77.0235
% Program caused arithmetic error: Floating underflow
IDL>
```

As you can see, the parameter P(0) does not change. ...If I change the order of the parameters, or the value of P(0), that value is still unaffected! I have included the code I use below. As is evident, P(0) IS used in the user-supplied function 'eval_pp2fvsimod', and it does affect the output of that function. Any thoughts... anyone?

Thanks,
Sean

```
PRO imodfit

;Take the imod test data and fit to it to try and find the modulation
;conversion
;restore, '~/windows/CLHCalibrations/imodtest/imod123.sav'

pp2ftestshort = [0.000411136,  0.000483014,  0.000574490,  0.000626735,
 0.000630612]
imodtestshort = [.5, 1., 1.5, 2, 2.5]

;need to scale the pp2f values to rcalb = 10.5. ...value used was 21
pp2ftestshort = pp2ftestshort * 21. / 10.5

weights =  .2*pp2ftestshort
vmrguess = 75.
omegaguess = .025d              ;guess for the conversion, cm^-1 / mA
params = [omegaguess, vmrguess]
result = mpfitfun('EVAL_pp2fvsimod', imodtestshort, pp2ftestshort,
weights, params, yfit=fitval, perr=perr)


END

function eval_pp2fvsimod, x,a

pmod = a[0] * x * 29979.
ihm = 2                      ;ihm=2 means second harmonic
sr = 1.                     ;scan range, in cm-1
res = 10.                    ;resolution, in MHz
npts = sr * 29979.246 / res     ;number of points in frequency and
transmission arrays
f1 = 7306.252                ;start frequency of scan, in cm-1
dlt = res / 29979.246
frq = f1 + findgen(npts)*dlt    ;frequency array, in cm -1
pressure = 312. / 1013.

make_line, pressure, a[1], frq, trans=t3
t3 = reform(t3)

pp2freturn = fltarr(n_elements(x))
FOR i = 0, n_elements(x)-1 DO begin
   der2f, frq, t3, npts, res, ihm, pmod[i], t2, pratio, zlobe
   pp2freturn[i] = pratio
ENDFOR

return, pp2freturn
```

END

Subject: Re: MPFITFUN problem
Posted by Craig Markwardt on Tue, 28 Aug 2012 14:54:25 GMT
View Forum Message <> Reply to Message

On Tuesday, August 28, 2012 7:43:11 AM UTC-4, Rui She wrote:
> Hi,
>
> I met a problem, when I used MPFITFUN.pro to fit a base of data using a user function, the procedure gave a error as:
>
> % MPFIT: Error detected while calling mpfitfun_eval:
>
> % MPFIT: Array dimensions must be greater than 0.
>
> % MPFIT: Error condition detected. Returning to MAIN level.
>
> % MPFITFUN: Error detected while calling mpfitfun_eval: Array dimensions must be greater than 0.
>
>
>
> I believe my user function was right, and I met this problem sometimes(not all data, error occurred about every 50 case), what's more, if I changed the start value of the parameters, it usually got to normal. So I think this may be a hidden bug of the MPFIT.
>
> Or someone has good idea?

I am 99% certain the error is happening inside of your user function. It's probably at the point where you create an array with fltarr() or dblarr() or make_array().

Here's a well-kept secret. You can call MPFITFUN() with the /NOCATCH keyword set. When NOCATCH is set, MPFITFUN will not try to trap any errors, and IDL will break at the location of the error. Then start debugging.

Craig