Subject: Why doesn't PERROR make any sense in MPFIT?
Posted by SirCed on Wed, 08 Feb 2017 16:22:11 GMT
View Forum Message <> Reply to Message

Can anybody explain to me what the difference is between how PERROR is calculated in MPFIT, and how Matlab's curvefit produces its 1-sigma errors? Based on the documentation, both use the Levenberg-Marquardt method, and the errors are based on the covariance matrices.

But, Matlab gives reasonable error values, even if all the data is given a weight of 1, or silly weights. By reasonable, I mean that plotting functions with any given fitted parameter + 2sigma gives a curve that passes the edge of the data cluster at some point. See examples:
https://ch.mathworks.com/help/curvefit/confidence-and-predic tion-bounds.html

MPFIT on the other hand often gives very unreasonable error values. For example, where the 2-sigma uncertainty is larger than the fitted parameter, even though this parameter is clearly positive and was bounded to be positive. Or else they become 2 orders of magnitude below the fitted parameter.
I have tried error arrays which were constant but equal to an estimated 1-sigma error in the measurement method. I have tried variable weighted functions. I've set all the weights to 1 and used this equation:

```
DOF     = N_ELEMENTS(X) - N_ELEMENTS(PARMS) ; deg of freedom
PCERROR = PERROR * SQRT(BESTNORM / DOF)   ; scaled uncertainties
```

But in all cases, I get error values that *just don't make sense* That do not behave consistently. And I don't understand why.

Does matlab use a more robust uncertainty search method? Does anybody have suggestions how to get meaningful uncertainty values out of MPFIT?

many thanks!
Cedric