Subject: Re: MPFITFUN .TIED
Posted by Craig Markwardt on Fri, 25 Aug 2017 14:20:20 GMT
View Forum Message <> Reply to Message

On Wednesday, August 23, 2017 at 3:09:31 PM UTC-4, Charlie Roberts wrote:
> On Mon, 21 Aug 2017 12:18:12 +0200, Markus Schmassmann
> <...@leibniz-kis.de> wrote:
>
>> On 08/18/2017 02:47 PM, ...@gmail.com wrote:
>>> I'm having some issues using fitting constraints using the MPFITFUN
>>> package in IDL.
>>>
>>> Basically, I have several parameters that I'm fitting, some of which
>>> are constrained to a single parameter (P[0]) as a factor of that
>>> parameter. As such, I set parinfo[*].tied so that, after printing,
>>> they read as follows: '0.662104 * P[0]' '0.245035 * P[0]' ...
>>>
>>> After fitting I read out the parameters, and even though P[0] has an
>>> appropriate value, it appears the constrained parameters have
>>> obtained the values (in this case) 0.66210400 0.24503500 (and P[0] is
>>> not 1., it is somewhere around 475 for my case) As such, the
>>> constraint does not seem to work for me. It just seems to return the
>>> factor with which I wanted to multiply P[0], but does not actually
>>> multiply it.
>>>
>>> Does anyone have an idea how to resolve this, or why it does not seem
>>> to work? Am I somehow using the wrong syntax for the .TIED keyword
>>> etc.? Any help is welcome. Many thanks!
>>
>> I haven't used MPFITFUN before, but from reading the code and your
>> problem description I have a few guesses what could have gone wrong.
>>
>> Any chance you have the TIED assigned to the wrong parameters?
>> e.g. parinfo[0].tied='0.662104 * P[0]'
>>
>> Any chance you use in the TIED definition a parameter with a higher number?
>> e.g. parinfo[1].tied='0.245035 * P[2]'
>>
>> Any chance that the result is correct except for the tied parameters?
>> If you use the other parameters to calculate the tied ones and then use
>> the forward function are you at a minimum?
>> You might have to get the derivatives (numerical or analytical) to verify.
>>
>> If none of that helps, write here a complete minimal working example of
>> the problem. Hopefully Craig has time to look into it.
>>
>> Good Luck, Markus
>

> As I kill as posts from google, I do not have the original message.
> But, in spite of this I have a question!
>
> If the fit parameters are related in some fashion, why not eliminate
> the 'dependent' ones all together and reduce the number of parameters
> to be fit? For exaample if it is known that
>
> p[7] = alfa*p[3]
>
> why leave p[7] as a constrained parameter, when it can be removed
> all together from the problem.
>
> Reducing the degrees of freedom decreased the amount of time needed
> to find the optimum and can improve accuracy as well.
>
> Scaling the variables to make all parameters lie roughly in the same
> numercial range is another thing to do ..... but it is a rather
> different kettle of fish than what I am suggesting.
>
> Do I have the correct picture of what is going on?

You are right that it is possible to recast a problem by re-writing the user model function to remove the constrained parameters.

HOWEVER, in practice, we often have more generic user functions that are usable across a wide class of conditions and use cases. Rather than re-writing our function every time for a new case, we can re-use the existing one by freezing or tying the parameters we don't care about.

Also, in practice, we often do exploratory type work where it is useful to have the generic user model which has lots of knobs to turn, but explore if a particular problem is sensitive or not to a given parameter. If we are testing for sensitivity to a parameter, it's not practical to rewrite the user function just to remove the parameter we are testing.

Also, from the point of view of statistics, tied parameters are treated as fixed parameters. Internally, MPFIT does not consider .FIXED or .TIED parameters as actual fitted variables. Fixed or tied variables are simply not part of the least squares set of fitted variables.

Craig
