
Subject: Re: mpfit and linear equality constraints

Posted by on Tue, 31 Jan 2012 10:03:04 GMT

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

Den måndagen den 30:e januari 2012 kl. 22:56:44 UTC+1 skrev Craig Markwardt:

> Greetings--

>

> On Jan 30, 11:32 am, mats.l...@gmail.com wrote:

>> I want to do non-linear model fitting with linear equality constraints and I just want to check that I've understood the mpfit "tied" keyword functionality correctly. Am I correct in believing it does not use the theory for linear equality constraints at all, it just removes the tied parameters from the problem and calculates them afterwards from the non-tied parameters?

>

> You are correct that MPFIT does the former, not the latter.

I'm getting confused by my own negations so I'll stick with the notion that I'm correct... :o)

>> By "theory for linear equality constraints" I'm referring the procedure ...

> ...

>> If I'm right, I'd be interested in pointers to IDL code implementing this kind of processing. Particularly if it does it and still uses mpfit for the actual fitting once the constraints part is out of the way.

>

> I think you can accomplish some of what you want to do if you impose
> constraints as additional equations in the problem. I outlined how
> one can do this in my 2009 talk at IDL Visualize. [*] You need to
> think of your N data points as N equations (to be solved in a least
> squared sense). Adding more constraints is just adding more equations
> to the list.

>

> Basically, if you have N data points and M constraints, your user
> function needs to compute N+M residual values.
> The first N residuals are just the standard (data-model) residuals.
> The following M residuals are the result of computing your constraint
> equations. MPFIT doesn't care which is which.

>

> The M constraints should be expressed as $f(p) = 0$, (where p =
> parameters). For example if your constraint is $P[0] \text{ EQ } 4 * P[1]$, it
> needs to be re-expressed as $P[0] - 4 * P[1] \text{ EQ } 0$. MPFIT just solves
> systems of equations where $\{f_i(p) = 0\}$, so if you can express your
> constraints in the same form it will do the trick. You may need to re-
> scale the constraint equations so that their residuals are comparable
> to the N data constraints.

>

> MPFIT solves the equation by orthogonal decomposition, so adding
> constraint equations to the problem should achieve what you want to
> do.

Thanks! I read your slides and I think I get some of it. That is, it seems to me that I would be able to do it with a reasonable effort.

But does it really do what I'm trying to accomplish? I want those constraints to be fulfilled exactly. If I were to add my constraints equations naively, for sure they would only be solved in the same approximate sense as the rest of the equations. If there is something in the method you describe that forces them to be solved exactly, I don't quite see it.

I guess one could force the approximation of the constraints to be arbitrarily good by setting the tolerances for those equations as very small. But above you say that I should rescale the constraints equations so their residuals are comparable to the data constraints, so this does not seem to be what you mean.

/Mats
