Subject: MPFIT and initial Guesses
Posted by steve.kaeppler on Mon, 10 Mar 2014 16:32:35 GMT
View Forum Message <> Reply to Message

Hi All-

I have been trying to use the MPfunfit and MPfit function that I downloaded directly off Craig's UW-Madison website for idl. I am attempting to fit an accelerated Maxwellian to electron flux data that was obtained on a sounding rocket. I have gotten the function up and running correctly (I think), and it does produce a solution.

However, I have found that the resulting parameter estimates change depending upon the initial guesses put in. I have tried to implement doing a small grid search on one of the parameters and then using the lowest chi square value from that grid search as the initial guess. I know that Levenberg-Marquardt routines do not converge to a global solution, but I am concerned that there are multiple local solutions that produce similar values of chi-squared.

To that end, I am trying to track down whether I am properly implementing this function.

I have tried to manually play with setting various step sizes in the parameters. Is there a location within the code or a parameter I could set which would allow me to see what the step size is? I am concerned I am either setting the step sizes too large or too small.

A second question, which may be harder to answer, how close does the initial guess need to be to obtain a global minimum or something close? I suspect this has to do with how well defined the problem is.

Any help would be appreciated and please let me know if you would like me to post some code or results I am getting.

Thank you,
Steve

Subject: Re: MPFIT and initial Guesses
Posted by wlandsman on Mon, 10 Mar 2014 17:01:54 GMT
View Forum Message <> Reply to Message

This doesn't directly address your question but I wanted to know if anyone had implemented the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) in IDL before I try to do it myself. Unlike the Levenberg_marquardt algorithm, CMS-ES does find the global minimum. It has been coded in most scientific programming languages (https://www.lri.fr/~hansen/cmaes_inmatlab.html) but not IDL as far as I know.

--Wayne

On Monday, March 10, 2014 12:32:35 PM UTC-4, Steve Kaeppler wrote:
> Hi All-
>
>
>
> I have been trying to use the MPfunfit and MPfit function that I downloaded directly off Craig's UW-Madison website for idl. I am attempting to fit an accelerated Maxwellian to electron flux data that was obtained on a sounding rocket. I have gotten the function up and running correctly (I think), and it does produce a solution.
>
>
>
> However, I have found that the resulting parameter estimates change depending upon the initial guesses put in. I have tried to implement doing a small grid search on one of the parameters and then using the lowest chi square value from that grid search as the initial guess. I know that Levenberg-Marquardt routines do not converge to a global solution, but I am concerned that there are multiple local solutions that produce similar values of chi-squared.
>
>
>
> To that end, I am trying to track down whether I am properly implementing this function.
>
>
>
>
>
> I have tried to manually play with setting various step sizes in the parameters. Is there a location within the code or a parameter I could set which would allow me to see what the step size is? I am concerned I am either setting the step sizes too large or too small.
>
>
>
> A second question, which may be harder to answer, how close does the initial guess need to be to obtain a global minimum or something close? I suspect this has to do with how well defined the problem is.
>
>
>
> Any help would be appreciated and please let me know if you would like me to post some code or results I am getting.
>
>
>
> Thank you,
>
> Steve

Subject: Re: MPFIT and initial Guesses
Posted by steve.kaeppler on Mon, 10 Mar 2014 17:51:37 GMT
View Forum Message <> Reply to Message

I also have to add that I am wondering about the size of the derivative and how large a parameter can be stepped because as I watch the parameter output from the function, it looks like the functions are changing a very small amount. Example: 1.4000000 1000.0000 2842.8501 1000.0000 26.000000

1.4000000	1000.0000	2842.8501	1000.0000	26.000000
1.4000000	1000.0000	2842.8501	1000.0000	26.000000
1.4000000	1000.0000	2842.8501	1000.0000	26.000000
1.4000000	1000.0000	2842.8501	1000.0000	26.000000
1.4000000	999.99999	2842.8501	999.99999	26.000000
1.4000000	1000.0000	2842.8501	1000.0000	26.000000
1.4000000	1000.0000	2842.8501	1000.0000	26.000000
1.3784718	1055.2616	2990.1060	1055.2616	21.457859

Notice how the 999.99999 moved very slightly.

Also, why is there this sudden abrupt shift in the last set of variables.

All I am wondering is whether I can get some of these 'initial steps' in the first iteration of the routine to be larger from the start. How do I do that? Does it involve setting the parameter information differently?

Hopefully this helps to clarify things a bit.

Thanks,
Steve

Subject: Re: MPFIT and initial Guesses
Posted by Phillip Bitzer on Mon, 10 Mar 2014 21:13:43 GMT
View Forum Message <> Reply to Message

You can change the default step size by using the appropriate tags in the parinfo array of structures:

```
;     .STEP - the step size to be used in calculating the numerical
;             derivatives.  If set to zero, then the step size is
;             computed automatically.  Ignored when AUTODERIVATIVE=0.
;             This value is superceded by the RELSTEP value.
;
;     .RELSTEP - the *relative* step size to be used in calculating
;                the numerical derivatives.  This number is the
;                fractional size of the step, compared to the
;                parameter value.  This value supercedes the STEP
;                setting.  If the parameter is zero, then a default
```

; step size is chosen.

What are you using now for parinfo?

Subject: Re: MPFIT and initial Guesses
Posted by steve.kaeppler on Tue, 11 Mar 2014 03:01:23 GMT
View Forum Message <> Reply to Message

On Monday, March 10, 2014 2:13:43 PM UTC-7, Phillip Bitzer wrote:
> You can change the default step size by using the appropriate tags in the parinfo array of structures:
>
>
>
> ; .STEP - the step size to be used in calculating the numerical
>
> ; derivatives. If set to zero, then the step size is
>
> ; computed automatically. Ignored when AUTODERIVATIVE=0.
>
> ; This value is superceded by the RELSTEP value.
>
> ;
>
> ; .RELSTEP - the *relative* step size to be used in calculating
>
> ; the numerical derivatives. This number is the
>
> ; fractional size of the step, compared to the
>
> ; parameter value. This value supercedes the STEP
>
> ; setting. If the parameter is zero, then a default
>
> ; step size is chosen.
>
>
>
> What are you using now for parinfo?

Phil-

I ran a test this afternoon just to convince myself, but using the relstep size of 0.25 and letting it default to determining an auto step size produced exactly the same parameter estimates and chi

squares. It is like the step size doesn't matter.

In terms of the step size I am using it is 0.25 for each of the parameters, where the parameters have these approximate values:
P[0]~1
P[1] ~ 2000
P[2] ~ 1000
P[4]~ 10

I am going to attach the code showing how I am declaring the parameters, maybe I am screwing something up there:

```
; parameter info - to constrain parameters
pi = replicate({fixed:0, limited:[0,0], limits:[0.D,0.D], mpside:0, relstep:0},5)

;n_0
pi[0].limited[0] = 1 ; turn on lower boundary
pi[0].limits[0] = 0.4 ; set lower boundary
pi[0].limited[1] = 1 ; turn on upper boundary
pi[0].limits[1] = 2.5 ; set upper boundary

;V_0
pi[1].limited[0] = 1 ; turn on lower boundary
pi[1].limits[0] = 0. ; set lower boundary
pi[1].limited[1] = 1 ; turn on upper boundary
pi[1].limits[1] =  4000. ; set upper boundary

;E_0
pi[2].limited[0] = 1 ; turn on lower boundary
pi[2].limits[0] = 0. ; set lower boundary
pi[2].limited[1] = 1 ; turn on upper boundary
pi[2].limits[1] = 1800. ; set upper boundary


;kappa
; note if kappa < 1.5 then temperature goes to zero
; kappa < 1.5 would correspond to a 'negative temperature' which doesn't make sense to me
; check that the equation I am using is correct.
pi[3].limited[0] = 1 ; turn on lower boundary
pi[3].limits[0] = 1.5 ; set lower boundary
pi[3].limited[1] = 1 ; turn on upper boundary
pi[3].limits[1] = 50. ; set upper boundary

pi[1].MPSIDE =2
pi[0].mpside=2
pi[2].mpside=2
pi[3].mpside=2
```

pi[0].relstep=0;.25
pi[1].relstep=0;.25
pi[2].relstep=0;.25
pi[3].relstep=0;.25

; keep the plotting fixed, that should NOT change
pi[4].fixed=1.

Again, the routine is reaching a solution, but I can't seem to control the first search very well or at all - there is no effect changing the relative step size. I have also played with changing the step size to a fixed amount.

Steve

Subject: Re: MPFIT and initial Guesses
Posted by Craig Markwardt on Tue, 11 Mar 2014 05:11:39 GMT
View Forum Message <> Reply to Message

On Monday, March 10, 2014 12:32:35 PM UTC-4, Steve Kaeppler wrote:
> However, I have found that the resulting parameter estimates change depending upon the initial guesses put in. I have tried to implement doing a small grid search on one of the parameters and then using the lowest chi square value from that grid search as the initial guess. I know that Levenberg-Marquardt routines do not converge to a global solution, but I am concerned that there are multiple local solutions that produce similar values of chi-squared.

The only way to guarantee the global minimum solution is an exhaustive grid search.

In a complicated chi-square space, choosing appropriate initial parameter values may be the most difficult part of the problem.

> I have tried to manually play with setting various step sizes in the parameters. Is there a location within the code or a parameter I could set which would allow me to see what the step size is? I am concerned I am either setting the step sizes too large or too small.

If you use .STEP or .RELSTEP, then you have manual control over the step size. If you don't set it, then it basically uses a step size of 1D-8 times the parameter value. I.e. the default is .RELSTEP=1D-8 for double precision parameters (1E-4 for single precision).

> A second question, which may be harder to answer, how close does the initial guess need to be to obtain a global minimum or something close? I suspect this has to do with how well defined the problem is.

As you say, L-M solutions are not guaranteed to find a global minimum. So basically, your initial conditions have to be within the "watershed" that includes the global minimum.

> Also, why is there this sudden abrupt shift in the last set of variables.

If you are computing numerical derivatives, MPFIT adjusts each parameter by .STEP / .RELSTEP to estimate the derivative. That is why you are seeing each vary in turn. This is not a search. It's just a mechanistic tallying of derivatives. After that is done, MPFIT computes a direction and then minimizes along that direction, which you could call a search. After minimizing along a line, then MPFIT goes back and recomputes new derivatives again! (and this repeats until convergence)

> All I am wondering is whether I can get some of these 'initial steps' in the > first iteration of the routine to be larger from the start. How do I do
> that? Does it involve setting the parameter information differently?

I think you already know now that you can choose this with .STEP and .RELSTEP, but these will only change how the derivatives are estimated.

Your example gives .RELSTEP=0.25. I think this is going to be way too big (but it is problem dependent). Your choice of .STEP and/or .RELSTEP should provide MPFIT a step size which is big enough to avoid round off error and small enough to capture the smallest variation of the function value.

Just one more comment: the fit will converge faster if you can remove .LIMITS constraints.

Craig
