
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 ; This value is superceded by the RELSTEP value.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
