Subject: MPfit question
Posted by Wox on Tue, 30 Sep 2008 14:38:20 GMT
View Forum Message <> Reply to Message

Hi all,

I'm using the wonderfull mpfit routine from Craig Markwardt. Today I
was having problems with convergence and I was looking in the code
what caused it. I was using parameter limits and as far as I can tell,
the parameter incrementation is done like this:

xnew = x + alpha * dx

where dx are the parameter increments and alpha = 1, except when xnew
would exceed the limits, in which case alpha < 1. So if one parameter
is close to the limit, this effects the increments for all other
parameters. If alpha is too small, the fit stops.

Why isn't dx adapted, so that pegged parameters won't effect the
convergence for other parameters?

Wout

Subject: Re: MPfit question
Posted by Craig Markwardt on Tue, 30 Sep 2008 15:49:46 GMT
View Forum Message <> Reply to Message

Wox <nomail@hotmail.com> writes:

> Hi all,
>
> I'm using the wonderfull mpfit routine from Craig Markwardt. Today I
> was having problems with convergence and I was looking in the code
> what caused it. I was using parameter limits and as far as I can tell,
> the parameter incrementation is done like this:
>
> xnew = x + alpha * dx
>
> where dx are the parameter increments and alpha = 1, except when xnew
> would exceed the limits, in which case alpha < 1. So if one parameter
> is close to the limit, this effects the increments for all other
> parameters. If alpha is too small, the fit stops.
>
> Why isn't dx adapted, so that pegged parameters won't effect the
> convergence for other parameters?

Hello, thanks for your message.

The original MINPACK fortran code did not have parameter limits, so limits are my own bolt-on.

If you look at the code, the value of ALPHA is adjusted so that, at the next iteration, a parameter will exactly touch its boundary, within a small tolerance. At that point, the parameter will be considered fixed, and will no longer enter into the calculation of the value of ALPHA. [*] Thus, the step *is* adaptive, it just doesn't happen in a single iteration.

If your constraints are pathalogical enough that the fitter is ping-ponging between two constraints continuously, I don't have much to say other than to offer my pity. It's not an easy problem to solve.

Craig

[*] - the convergence criteria are also pro-rated because ALPHA is smaller than 1.

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: cbmarkwardt+usenet@gmail.com
---------- ----------

Subject: Re: MPfit question
Posted by Wox on Wed, 01 Oct 2008 08:31:54 GMT
View Forum Message <> Reply to Message

On 30 Sep 2008 11:49:46 -0400, Craig Markwardt
<craigmnet@REMOVEcow.physics.wisc.edu> wrote:

> If you look at the code, the value of ALPHA is adjusted so that, at
> the next iteration, a parameter will exactly touch its boundary,
> within a small tolerance. At that point, the parameter will be
> considered fixed, and will no longer enter into the calculation of the
> value of ALPHA. [*] Thus, the step *is* adaptive, it just doesn't
> happen in a single iteration.

I'm sorry, but I don't see how it does this. ALPHA is adjusted and immediatly used (see below). In the next iteration, the increments are calculated again by mpfit_lmpar and used again to calculate ALPHA, whether the param was at the limit in the previous iteration or not.

The thing is, my problem is solved when I adjust the increments themselves and leave ALPHA=1. I was just wondering whether I introduce

some errors by doing this.

```
INNER_LOOP:
; mpfit_lmpar:
; solve A.wa1=B and sqrt(par).D.wa1=0
; where wa1 is the parameter increment
...
; When some param goes outside boundary, make alpha smaller so it
touches the boundary:

dwa1 = abs(wa1) GT MACHEP0
whl = where(dwa1 AND qllim AND (x + wa1 LT llim), lct)
if lct GT 0 then $
        alpha = min([alpha, (llim[whl]-x[whl])/wa1[whl]])
whu = where(dwa1 AND qulim AND (x + wa1 GT ulim), uct)
if uct GT 0 then $
        alpha = min([alpha, (ulim[whu]-x[whu])/wa1[whu]])

; Apply increment!
xnew = x + alpha * wa1

; When the step puts us on a boundary, make sure it is exact
...

; Convergence tests + succesfull inner loop test
...
goto INNER_LOOP
```

Subject: Re: MPfit question
Posted by Craig Markwardt on Wed, 01 Oct 2008 17:16:10 GMT
View Forum Message <> Reply to Message

Wox <nomail@hotmail.com> writes:

> On 30 Sep 2008 11:49:46 -0400, Craig Markwardt
> <craigmnet@REMOVEcow.physics.wisc.edu> wrote:
>
>> If you look at the code, the value of ALPHA is adjusted so that, at
>> the next iteration, a parameter will exactly touch its boundary,
>> within a small tolerance. At that point, the parameter will be
>> considered fixed, and will no longer enter into the calculation of the
>> value of ALPHA. [*] Thus, the step *is* adaptive, it just doesn't
>> happen in a single iteration.
>
> I'm sorry, but I don't see how it does this. ALPHA is adjusted and
> immediatly used (see below). In the next iteration, the increments are

> calculated again by mpfit_lmpar and used again to calculate ALPHA,
> whether the param was at the limit in the previous iteration or not.

That is not correct. Please search for 'zeroing the derivatives of pegged parameters'. Once a parameter is pegged at a boundary in the previous iteration, it no longer contributes to the congugate gradiate solution because its derivatives have been zeroed.

> The thing is, my problem is solved when I adjust the increments
> themselves and leave ALPHA=1. I was just wondering whether I introduce
> some errors by doing this.

Probably your best bet is to see which convergence criterium is satisfied when ALPHA < 1, and go from there.

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: cbmarkwardt+usenet@gmail.com
---------- ----------

Subject: Re: MPfit question
Posted by Wox on Thu, 02 Oct 2008 11:39:19 GMT
View Forum Message <> Reply to Message

On 01 Oct 2008 13:16:10 -0400, Craig Markwardt
<craigmnet@REMOVEcow.physics.wisc.edu> wrote:

>
> Wox <nomail@hotmail.com> writes:
>
>> On 30 Sep 2008 11:49:46 -0400, Craig Markwardt
>> <craigmnet@REMOVEcow.physics.wisc.edu> wrote:
>>
>>> If you look at the code, the value of ALPHA is adjusted so that, at
>>> the next iteration, a parameter will exactly touch its boundary,
>>> within a small tolerance. At that point, the parameter will be
>>> considered fixed, and will no longer enter into the calculation of the
>>> value of ALPHA. [*] Thus, the step *is* adaptive, it just doesn't
>>> happen in a single iteration.
>>
>> I'm sorry, but I don't see how it does this. ALPHA is adjusted and
>> immediatly used (see below). In the next iteration, the increments are
>> calculated again by mpfit_lmpar and used again to calculate ALPHA,
>> whether the param was at the limit in the previous iteration or not.
>

> That is not correct. Please search for 'zeroing the derivatives of
> pegged parameters'. Once a parameter is pegged at a boundary in the
> previous iteration, it no longer contributes to the congugate gradiate
> solution because its derivatives have been zeroed.

Ah yes, I see now. Thanks for your comments. I must say it's quite an impressive and complicated piece of code. I'm having a hard time understanding the convergence criteria and the calculation of the LM-step. Can you recommend some reference where I can find more details? I sure have some books explaining NLLS-refinement but they don't seem to mention some of the criteria I see in your code. Also the calculation of the LM-step is done differently each time I open a new book :-).

Subject: Re: MPfit question
Posted by Craig Markwardt on Thu, 02 Oct 2008 15:59:50 GMT
View Forum Message <> Reply to Message

Wox <nomail@hotmail.com> writes:
>
> Ah yes, I see now. Thanks for your comments. I must say it's quite an
> impressive and complicated piece of code. I'm having a hard time
> understanding the convergence criteria and the calculation of the
> LM-step. Can you recommend some reference where I can find more
> details? I sure have some books explaining NLLS-refinement but they
> don't seem to mention some of the criteria I see in your code. Also
> the calculation of the LM-step is done differently each time I open a
> new book :-).

Hi, the code is based on MINPACK-1, so the convergence criteria come from that library. The references section of mpfit.pro contains two citations by one of the primary authors of MINPACK. Unfortunately they are hard to find. :-(

I am suspecting that for your problem, MPFIT is claiming the problem converged when it didn't, if the "actual" change in chi^2 is much smaller than "predicted". The only real way I can see this happening is if you started a parameter *very close* to a boundary but not exactly on the boundary. The fix for that should be obvious.

Craig

--
 -- --------------
Craig B. Markwardt, Ph.D.   EMAIL: cbmarkwardt+usenet@gmail.com
 -- --------------

Subject: Re: MPFIT question
Posted by Craig Markwardt on Wed, 14 Jan 2009 06:41:17 GMT
View Forum Message <> Reply to Message

On Jan 13, 2:33 pm, "j.coe...@gmail.com" <j.coe...@gmail.com> wrote:
> I'm fitting data to a gamma variate function using Craig Markwardt's
> MPFIT. This has been working great except for a chronic error message
> that occurs once every 50 fits or so:
>
> MPFIT: Error detected while calling mpfitfun_eval:
> MPFIT: Array dimensions must be greater than 0.
> MPFIT: Error condition detected. Returning to MAIN level.
> MPFITFUN: Error detected while calling mpfitfun_eval: Array dimensions
> must be greater than 0.
> Attempt to subscript P with <INT (1)> is out of range.
>
> My five parameters are getting lost. The parameters are first passed
> to MPFITFUN via the parinfo structure because some are constrained.
> Then the parameters are passed along by MPFITFUN to the user-supplied
> model function as a double array, p. When the error occurs, the p
> array has shrunk from five doubles to just one NaN, as you can see
> from the abbreviated output reproduced at the end of this post. The
> subscripting error happens when the user-supplied model function tries
> to subscript the suddenly nonexistent second element of p, which is
> supposed to have five parameters/elements (and had five elements at
> all previous iterations). This can happen during any MPFIT iteration,
> but usually around iteration 4.
>
> Does anyone know what's going on? I checked to make sure that there
> are no NaN values in the data, and that my gamma variate model
> function is not producing any NaN values at any iteration. I have the
> latest version of the MPFIT library. I've just been catching the
> error and fitting the problematic data with IDL's routine, but MPFIT
> does a much better job when it works for me. Hopefully someone else
> who has encountered this issue knows what I am doing wrong. Thanks.
>
> Iter 1 CHI-SQUARE = 9182.7891 DOF = 353
> P DOUBLE = Array[5]
> .
> .
> Iter 2 CHI-SQUARE = 6448.4258 DOF = 353
> P DOUBLE = Array[5]
> .
> .
> Iter 3 CHI-SQUARE = 8402.1122 DOF = 353
> P DOUBLE = Array[5]
> .
> .
> Iter 4 CHI-SQUARE = 1564.3159 DOF = 353

> P DOUBLE = Array[5]
> .
> .
> MPFIT: Error detected while calling mpfitfun_eval:
> MPFIT: Array dimensions must be greater than 0.
> MPFIT: Error condition detected. Returning to MAIN level.
> P DOUBLE = NaN
> Attempt to subscript P with <INT (1)> is out of range.

Greetings--

I don't believe MPFIT should change the number of elements in the parameter array P.

My first guess is that your user-function is redefining the P array. Try doing a HELP on P at both the beginning and the end of your user function to see if that's true.

Another possibility is to set !EXCEPT=2 to see if IDL will indicate where the numerical exceptions first start to occur.

Finally, nothing beats good ol' stepping through line by line until you find the culprit.

Craig

Subject: Re: MPFIT question
Posted by j.coenia@gmail.com on Wed, 14 Jan 2009 15:54:24 GMT
View Forum Message <> Reply to Message

Thank you for the reply. I also assume it is my user function. I did put a help comment at the very beginning -- that's what's generating the p description in the output text in my first post. Also, the user function is very simple, and I just can't see where it could be changing p, especially since the "Help, p" statement is the very first line. I'll step through line by line and if I can't track the problem down, I'll send you a reproducible case. Thanks again.

Subject: Re: MPFIT question
Posted by Craig Markwardt on Wed, 14 Jan 2009 16:23:44 GMT
View Forum Message <> Reply to Message

On Jan 14, 10:54 am, "j.coe...@gmail.com" <j.coe...@gmail.com> wrote:
> Thank you for the reply. I also assume it is my user function. I did
> put a help comment at the very beginning -- that's what's generating

> the p description in the output text in my first post. Also, the user
> function is very simple, and I just can't see where it could be
> changing p, especially since the "Help, p" statement is the very first
> line. I'll step through line by line and if I can't track the problem
> down, I'll send you a reproducible case. Thanks again.

Right, but remember I also suggested putting a HELP statement at the end of your function as well!

Craig

Subject: Re: MPFIT question
Posted by j.coenia@gmail.com on Wed, 14 Jan 2009 21:37:13 GMT
View Forum Message <> Reply to Message

The user function has help statements at its beginning and end now, and I tried the other suggestions, but I'm still stumped. I'm doing something foolish.

Below is the problem code. Most of it is just the 353-element data array. Without the data, it's only two short functions, 20 lines of code or so total. I hope it's not bad netiquette to post so many lines. I can't figure out why these data crash the program. I can fit most other data acquired in this manner -- I have fit hundreds of thousands of these curves with this method, but the error crops up once every 50 cases or so (usually for data that don't fit the model very well, but it still shouldn't crash?).

The data are being fit to a gamma variate function. Two things I'm not sure about: (1) I'm passing t0 as a fitting parameter, and I'm not sure if this is acceptable. (2) Worse, I don't have a good idea of how to estimate error, so I'm just using the standard deviation of the baseline (pre-arrival) curve, which is probably wrong. Suggestions?

The code crashes after just two MPFIT iterations. Just type 'MPFIT_problem' to run.

Thanks for any help.

```
function gamma_variate, x, p

print, 'p start...'
help, p

baseline = p[0]
```

```
t0 = p[1]
tmax = p[2]  ; time of peak signal
smax = p[3]  ; signal max in mVolts
alpha = p[4]

 ; shift time, apply gamma variate only to data after t0

wh = Where(x GE t0, ct)
t = (x - t0)[wh]
tmax = tmax - t0

 ; gamma variate function

s = baseline + (smax) * (tmax^(-alpha)) * exp(alpha) * (t^alpha) *  exp
((-alpha)  * t / tmax)

 ; add the pre-t0 data (baseline)

n = n_elements(x)
pre = make_array(n - ct, Value = baseline)
s = [pre, s]

print, 'p end...'
help, p

return, s

 end;---------------------------------------------------- --


pro mpfit_problem

!except = 2

data = [ $
 64.1682 , $
 66.3804 , $
 73.4243 , $
 64.1682 , $
 64.1682 , $
 55.9551 , $
 47.0046 , $
 52.2084 , $
 48.6855 , $
 52.2084 , $
 57.9162 , $
 75.9137 , $
 64.1682 , $
```

75.9137 , $
86.6244 , $
89.4994 , $
89.4994 , $
101.845 , $
105.153 , $
89.4994 , $
95.4993 , $
92.457 , $
105.153 , $
89.4994 , $
101.845 , $
92.457 , $
86.6244 , $
83.8301 , $
89.4994 , $
92.457 , $
92.457 , $
95.4993 , $
95.4993 , $
95.4993 , $
89.4994 , $
89.4994 , $
75.9137 , $
71.0068 , $
78.4766 , $
57.9162 , $
64.1682 , $
59.9377 , $
52.2084 , $
45.3754 , $
39.3496 , $
39.3496 , $
48.6855 , $
62.0212 , $
73.4243 , $
50.4197 , $
73.4243 , $
66.3804 , $
66.3804 , $
68.6594 , $
68.6594 , $
86.6244 , $
57.9162 , $
89.4994 , $
73.4243 , $
73.4243 , $
78.4766 , $

75.9137 , $
83.8301 , $
95.4993 , $
86.6244 , $
73.4243 , $
81.1148 , $
86.6244 , $
89.4994 , $
40.785 , $
54.053 , $
68.6594 , $
78.4766 , $
73.4243 , $
95.4993 , $
83.8301 , $
81.1148 , $
95.4993 , $
83.8301 , $
75.9137 , $
57.9162 , $
68.6594 , $
83.8301 , $
78.4766 , $
75.9137 , $
68.6594 , $
78.4766 , $
75.9137 , $
83.8301 , $
75.9137 , $
64.1682 , $
71.0068 , $
78.4766 , $
73.4243 , $
64.1682 , $
92.457 , $
68.6594 , $
73.4243 , $
75.9137 , $
62.0212 , $
73.4243 , $
55.9551 , $
57.9162 , $
62.0212 , $
73.4243 , $
75.9137 , $
75.9137 , $
92.457 , $
83.8301 , $

112.047 , $
98.6279 , $
123.118 , $
92.457 , $
105.153 , $
92.457 , $
75.9137 , $
86.6244 , $
92.457 , $
73.4243 , $
78.4766 , $
66.3804 , $
75.9137 , $
119.327 , $
127.011 , $
105.153 , $
73.4243 , $
54.053 , $
73.4243 , $
75.9137 , $
66.3804 , $
86.6244 , $
68.6594 , $
78.4766 , $
75.9137 , $
78.4766 , $
81.1148 , $
127.011 , $
152.651 , $
95.4993 , $
83.8301 , $
119.327 , $
50.4197 , $
68.6594 , $
95.4993 , $
75.9137 , $
131.009 , $
54.053 , $
108.553 , $
62.0212 , $
83.8301 , $
57.9162 , $
152.651 , $
119.327 , $
119.327 , $
54.053 , $
73.4243 , $
86.6244 , $

89.4994 , $
105.153 , $
73.4243 , $
78.4766 , $
112.047 , $
50.4197 , $
89.4994 , $
92.457 , $
30.5028 , $
55.9551 , $
119.327 , $
131.009 , $
68.6594 , $
71.0068 , $
81.1148 , $
112.047 , $
73.4243 , $
64.1682 , $
52.2084 , $
123.118 , $
127.011 , $
75.9137 , $
112.047 , $
101.845 , $
95.4993 , $
105.153 , $
55.9551 , $
135.114 , $
83.8301 , $
95.4993 , $
54.053 , $
36.6134 , $
42.2669 , $
71.0068 , $
68.6594 , $
95.4993 , $
115.638 , $
57.9162 , $
148.095 , $
48.6855 , $
75.9137 , $
95.4993 , $
127.011 , $
187.992 , $
112.047 , $
172.084 , $
71.0068 , $
92.457 , $

135.114 , $
148.095 , $
36.6134 , $
89.4994 , $
105.153 , $
68.6594 , $
119.327 , $
27.2923 , $
55.9551 , $
95.4993 , $
83.8301 , $
152.651 , $
43.7966 , $
62.0212 , $
112.047 , $
123.118 , $
135.114 , $
105.153 , $
32.8277 , $
62.0212 , $
39.3496 , $
47.0046 , $
92.457 , $
108.553 , $
86.6244 , $
152.651 , $
55.9551 , $
119.327 , $
52.2084 , $
75.9137 , $
71.0068 , $
35.3102 , $
43.7966 , $
119.327 , $
112.047 , $
127.011 , $
127.011 , $
62.0212 , $
54.053 , $
177.256 , $
311.315 , $
123.118 , $
101.845 , $
92.457 , $
152.651 , $
127.011 , $
101.845 , $
68.6594 , $

162.121 , $
172.084 , $
167.04 , $
152.651 , $
108.553 , $
172.084 , $
143.655 , $
152.651 , $
162.121 , $
279.383 , $
162.121 , $
123.118 , $
34.0487 , $
62.0212 , $
83.8301 , $
177.256 , $
172.084 , $
157.326 , $
73.4243 , $
135.114 , $
92.457 , $
59.9377 , $
131.009 , $
42.2669 , $
71.0068 , $
78.4766 , $
135.114 , $
78.4766 , $
143.655 , $
112.047 , $
78.4766 , $
105.153 , $
66.3804 , $
73.4243 , $
89.4994 , $
143.655 , $
75.9137 , $
152.651 , $
89.4994 , $
98.6279 , $
95.4993 , $
66.3804 , $
24.3893 , $
86.6244 , $
57.9162 , $
119.327 , $
135.114 , $
75.9137 , $

40.785 , $
112.047 , $
47.0046 , $
89.4994 , $
92.457 , $
95.4993 , $
89.4994 , $
50.4197 , $
81.1148 , $
57.9162 , $
45.3754 , $
66.3804 , $
73.4243 , $
59.9377 , $
73.4243 , $
105.153 , $
78.4766 , $
64.1682 , $
54.053 , $
62.0212 , $
39.3496 , $
47.0046 , $
89.4994 , $
28.327 , $
81.1148 , $
43.7966 , $
68.6594 , $
48.6855 , $
54.053 , $
40.785 , $
35.3102 , $
42.2669 , $
75.9137 , $
64.1682 , $
64.1682 , $
39.3496 , $
95.4993 , $
78.4766 , $
101.845 , $
64.1682 , $
68.6594 , $
48.6855 , $
59.9377 , $
66.3804 , $
83.8301 , $
40.785 , $
39.3496 , $
78.4766 , $

```
 127.011 , $
 172.084 , $
 52.2084 , $
 119.327 , $
 37.9595 , $
 66.3804 , $
 108.553 , $
 86.6244 , $
 83.8301  $
 ]

 ; calculate times @ 30Hz

n = n_elements(data)
t = findgen(n)/30.0

 ; define and contrain parameters

parinfo = replicate({value:0.D, fixed:0, limited:[0,0], $
       limits:[0.D,0], mpmaxstep: 0.0D}, 5)

 ; baseline
parinfo(0).fixed = 0b
parinfo(0).limited = [1,1]
parinfo(0).limits = [0, 3675.36]
parinfo(0).value = 84.551201

 ; t0
t0 = 4.0333333
parinfo(1).fixed = 0b
parinfo(1).limited(0) = 1
parinfo(1).limits(0)  = 0.D
parinfo(1).value = 4.0333333

 ; tmax
parinfo(2).fixed = 0b
parinfo(2).limited = [1,1]
parinfo(2).limits = [t0, n/30.0 - 0.5] ; 0.5 secs before end
parinfo(2).value =   8.5333338

 ; max
parinfo(3).limited  = [1,1]
parinfo(3).limits = [0, 3675.36]
parinfo(3).value = 127.96773

 ; alpha
parinfo(4).fixed = 0b
parinfo(4).limited = [1,1]
```

```
parinfo(4).limits = [0.5, 12]
parinfo(4).value = 1.0

 ; calculate std deviation of signal up to t0
 ; probably not the best way to estimate error...

err = stddev(data[0:t0*30]) > 1.0

 ; plot the signal

window, /free
plot, t, data, $
 Title = 'Signal', $
 XTitle = 'Time (s)', $
 YTitle = 'mVolts'

 ; fit

parms = mpfitfun('gamma_variate', t, data, replicate(err, n), $
  YFit = fit, $
  Bestnorm = chisq, $
  /NaN, $
  Parinfo = parinfo, $
  Quiet = 0 $
  )

 ; plot fit

oplot, t, gamma_variate(t, parms), Color = FSC_Color('red')

 end;---------------------------------------------------- --
```

Subject: Re: MPFIT question
Posted by pgrigis on Wed, 14 Jan 2009 22:08:48 GMT
View Forum Message <> Reply to Message

j.coenia@gmail.com wrote:
> The user function has help statements at its beginning and end now,
> and I tried the other suggestions, but I'm still stumped. I'm doing
> something foolish.
>
> Below is the problem code. Most of it is just the 353-element data
> array. Without the data, it's only two short functions, 20 lines of
> code or so total. I hope it's not bad netiquette to post so many
> lines. I can't figure out why these data crash the program. I can
> fit most other data acquired in this manner -- I have fit hundreds of
> thousands of these curves with this method, but the error crops up

> once every 50 cases or so (usually for data that don't fit the model
> very well, but it still shouldn't crash?).
>
> The data are being fit to a gamma variate function. Two things I'm
> not sure about: (1) I'm passing t0 as a fitting parameter, and I'm not
> sure if this is acceptable. (2) Worse, I don't have a good idea of
> how to estimate error, so I'm just using the standard deviation of the
> baseline (pre-arrival) curve, which is probably wrong. Suggestions?
>
> The code crashes after just two MPFIT iterations. Just type
> 'MPFIT_problem' to run.
>
> Thanks for any help.
>
> ----------
>
> function gamma_variate, x, p
>
> print, 'p start...'
> help, p
>
> baseline = p[0]
> t0 = p[1]
> tmax = p[2] ; time of peak signal
> smax = p[3] ; signal max in mVolts
> alpha = p[4]
>
> ; shift time, apply gamma variate only to data after t0
>
> wh = Where(x GE t0, ct)

I think that you may have a problem here when ct EQ 0 and wh EQ -1.
(that is, all x<t0).
Ciao,
Paolo
> t = (x - t0)[wh]
> tmax = tmax - t0
>
> ; gamma variate function
>
> s = baseline + (smax) * (tmax^(-alpha)) * exp(alpha) * (t^alpha) * exp
> ((-alpha) * t / tmax)
>
> ; add the pre-t0 data (baseline)
>
> n = n_elements(x)
> pre = make_array(n - ct, Value = baseline)
> s = [pre, s]

```
>
>  print, 'p end...'
>  help, p
>
>  return, s
>
>   end;---------------------------------------------------- --
>
>
>  pro mpfit_problem
>
>  !except = 2
>
>  data = [ $
>   64.1682 , $
>   66.3804 , $
>   73.4243 , $
>   64.1682 , $
>   64.1682 , $
>   55.9551 , $
>   47.0046 , $
>   52.2084 , $
>   48.6855 , $
>   52.2084 , $
>   57.9162 , $
>   75.9137 , $
>   64.1682 , $
>   75.9137 , $
>   86.6244 , $
>   89.4994 , $
>   89.4994 , $
>   101.845 , $
>   105.153 , $
>   89.4994 , $
>   95.4993 , $
>   92.457 , $
>   105.153 , $
>   89.4994 , $
>   101.845 , $
>   92.457 , $
>   86.6244 , $
>   83.8301 , $
>   89.4994 , $
>   92.457 , $
>   92.457 , $
>   95.4993 , $
>   95.4993 , $
>   95.4993 , $
```

> 89.4994 , $
> 89.4994 , $
> 75.9137 , $
> 71.0068 , $
> 78.4766 , $
> 57.9162 , $
> 64.1682 , $
> 59.9377 , $
> 52.2084 , $
> 45.3754 , $
> 39.3496 , $
> 39.3496 , $
> 48.6855 , $
> 62.0212 , $
> 73.4243 , $
> 50.4197 , $
> 73.4243 , $
> 66.3804 , $
> 66.3804 , $
> 68.6594 , $
> 68.6594 , $
> 86.6244 , $
> 57.9162 , $
> 89.4994 , $
> 73.4243 , $
> 73.4243 , $
> 78.4766 , $
> 75.9137 , $
> 83.8301 , $
> 95.4993 , $
> 86.6244 , $
> 73.4243 , $
> 81.1148 , $
> 86.6244 , $
> 89.4994 , $
> 40.785 , $
> 54.053 , $
> 68.6594 , $
> 78.4766 , $
> 73.4243 , $
> 95.4993 , $
> 83.8301 , $
> 81.1148 , $
> 95.4993 , $
> 83.8301 , $
> 75.9137 , $
> 57.9162 , $
> 68.6594 , $

```
>  83.8301 , $
>  78.4766 , $
>  75.9137 , $
>  68.6594 , $
>  78.4766 , $
>  75.9137 , $
>  83.8301 , $
>  75.9137 , $
>  64.1682 , $
>  71.0068 , $
>  78.4766 , $
>  73.4243 , $
>  64.1682 , $
>  92.457 , $
>  68.6594 , $
>  73.4243 , $
>  75.9137 , $
>  62.0212 , $
>  73.4243 , $
>  55.9551 , $
>  57.9162 , $
>  62.0212 , $
>  73.4243 , $
>  75.9137 , $
>  75.9137 , $
>  92.457 , $
>  83.8301 , $
>  112.047 , $
>  98.6279 , $
>  123.118 , $
>  92.457 , $
>  105.153 , $
>  92.457 , $
>  75.9137 , $
>  86.6244 , $
>  92.457 , $
>  73.4243 , $
>  78.4766 , $
>  66.3804 , $
>  75.9137 , $
>  119.327 , $
>  127.011 , $
>  105.153 , $
>  73.4243 , $
>  54.053 , $
>  73.4243 , $
>  75.9137 , $
>  66.3804 , $
```

> 86.6244 , $
> 68.6594 , $
> 78.4766 , $
> 75.9137 , $
> 78.4766 , $
> 81.1148 , $
> 127.011 , $
> 152.651 , $
> 95.4993 , $
> 83.8301 , $
> 119.327 , $
> 50.4197 , $
> 68.6594 , $
> 95.4993 , $
> 75.9137 , $
> 131.009 , $
> 54.053 , $
> 108.553 , $
> 62.0212 , $
> 83.8301 , $
> 57.9162 , $
> 152.651 , $
> 119.327 , $
> 119.327 , $
> 54.053 , $
> 73.4243 , $
> 86.6244 , $
> 89.4994 , $
> 105.153 , $
> 73.4243 , $
> 78.4766 , $
> 112.047 , $
> 50.4197 , $
> 89.4994 , $
> 92.457 , $
> 30.5028 , $
> 55.9551 , $
> 119.327 , $
> 131.009 , $
> 68.6594 , $
> 71.0068 , $
> 81.1148 , $
> 112.047 , $
> 73.4243 , $
> 64.1682 , $
> 52.2084 , $
> 123.118 , $
> 127.011 , $

```
>  75.9137 , $
>  112.047 , $
>  101.845 , $
>  95.4993 , $
>  105.153 , $
>  55.9551 , $
>  135.114 , $
>  83.8301 , $
>  95.4993 , $
>  54.053 , $
>  36.6134 , $
>  42.2669 , $
>  71.0068 , $
>  68.6594 , $
>  95.4993 , $
>  115.638 , $
>  57.9162 , $
>  148.095 , $
>  48.6855 , $
>  75.9137 , $
>  95.4993 , $
>  127.011 , $
>  187.992 , $
>  112.047 , $
>  172.084 , $
>  71.0068 , $
>  92.457 , $
>  135.114 , $
>  148.095 , $
>  36.6134 , $
>  89.4994 , $
>  105.153 , $
>  68.6594 , $
>  119.327 , $
>  27.2923 , $
>  55.9551 , $
>  95.4993 , $
>  83.8301 , $
>  152.651 , $
>  43.7966 , $
>  62.0212 , $
>  112.047 , $
>  123.118 , $
>  135.114 , $
>  105.153 , $
>  32.8277 , $
>  62.0212 , $
>  39.3496 , $
```

```
>  47.0046 , $
>  92.457 , $
>  108.553 , $
>  86.6244 , $
>  152.651 , $
>  55.9551 , $
>  119.327 , $
>  52.2084 , $
>  75.9137 , $
>  71.0068 , $
>  35.3102 , $
>  43.7966 , $
>  119.327 , $
>  112.047 , $
>  127.011 , $
>  127.011 , $
>  62.0212 , $
>  54.053 , $
>  177.256 , $
>  311.315 , $
>  123.118 , $
>  101.845 , $
>  92.457 , $
>  152.651 , $
>  127.011 , $
>  101.845 , $
>  68.6594 , $
>  162.121 , $
>  172.084 , $
>  167.04 , $
>  152.651 , $
>  108.553 , $
>  172.084 , $
>  143.655 , $
>  152.651 , $
>  162.121 , $
>  279.383 , $
>  162.121 , $
>  123.118 , $
>  34.0487 , $
>  62.0212 , $
>  83.8301 , $
>  177.256 , $
>  172.084 , $
>  157.326 , $
>  73.4243 , $
>  135.114 , $
>  92.457 , $
```

```
>  59.9377 , $
>  131.009 , $
>  42.2669 , $
>  71.0068 , $
>  78.4766 , $
>  135.114 , $
>  78.4766 , $
>  143.655 , $
>  112.047 , $
>  78.4766 , $
>  105.153 , $
>  66.3804 , $
>  73.4243 , $
>  89.4994 , $
>  143.655 , $
>  75.9137 , $
>  152.651 , $
>  89.4994 , $
>  98.6279 , $
>  95.4993 , $
>  66.3804 , $
>  24.3893 , $
>  86.6244 , $
>  57.9162 , $
>  119.327 , $
>  135.114 , $
>  75.9137 , $
>  40.785 , $
>  112.047 , $
>  47.0046 , $
>  89.4994 , $
>  92.457 , $
>  95.4993 , $
>  89.4994 , $
>  50.4197 , $
>  81.1148 , $
>  57.9162 , $
>  45.3754 , $
>  66.3804 , $
>  73.4243 , $
>  59.9377 , $
>  73.4243 , $
>  105.153 , $
>  78.4766 , $
>  64.1682 , $
>  54.053 , $
>  62.0212 , $
>  39.3496 , $
```

```
>   47.0046 , $
>   89.4994 , $
>   28.327 , $
>   81.1148 , $
>   43.7966 , $
>   68.6594 , $
>   48.6855 , $
>   54.053 , $
>   40.785 , $
>   35.3102 , $
>   42.2669 , $
>   75.9137 , $
>   64.1682 , $
>   64.1682 , $
>   39.3496 , $
>   95.4993 , $
>   78.4766 , $
>   101.845 , $
>   64.1682 , $
>   68.6594 , $
>   48.6855 , $
>   59.9377 , $
>   66.3804 , $
>   83.8301 , $
>   40.785 , $
>   39.3496 , $
>   78.4766 , $
>   127.011 , $
>   172.084 , $
>   52.2084 , $
>   119.327 , $
>   37.9595 , $
>   66.3804 , $
>   108.553 , $
>   86.6244 , $
>   83.8301  $
>   ]
>
>   ; calculate times @ 30Hz
>
>  n = n_elements(data)
>  t = findgen(n)/30.0
>
>   ; define and contrain parameters
>
>  parinfo = replicate({value:0.D, fixed:0, limited:[0,0], $
>         limits:[0.D,0], mpmaxstep: 0.0D}, 5)
>
```

```
>   ; baseline
>  parinfo(0).fixed = 0b
>  parinfo(0).limited = [1,1]
>  parinfo(0).limits = [0, 3675.36]
>  parinfo(0).value = 84.551201
>
>   ; t0
>  t0 = 4.0333333
>  parinfo(1).fixed = 0b
>  parinfo(1).limited(0) = 1
>  parinfo(1).limits(0)  = 0.D
>  parinfo(1).value = 4.0333333
>
>   ; tmax
>  parinfo(2).fixed = 0b
>  parinfo(2).limited = [1,1]
>  parinfo(2).limits = [t0, n/30.0 - 0.5] ; 0.5 secs before end
>  parinfo(2).value =   8.5333338
>
>   ; max
>  parinfo(3).limited  = [1,1]
>  parinfo(3).limits = [0, 3675.36]
>  parinfo(3).value = 127.96773
>
>   ; alpha
>  parinfo(4).fixed = 0b
>  parinfo(4).limited = [1,1]
>  parinfo(4).limits = [0.5, 12]
>  parinfo(4).value = 1.0
>
>   ; calculate std deviation of signal up to t0
>   ; probably not the best way to estimate error...
>
>  err = stddev(data[0:t0*30]) > 1.0
>
>   ; plot the signal
>
>  window, /free
>  plot, t, data, $
>   Title = 'Signal', $
>   XTitle = 'Time (s)', $
>   YTitle = 'mVolts'
>
>   ; fit
>
>  parms = mpfitfun('gamma_variate', t, data, replicate(err, n), $
>    YFit = fit, $
>    Bestnorm = chisq, $
```

> /NaN, $
> Parinfo = parinfo, $
> Quiet = 0 $
>)
>
> ; plot fit
>
> oplot, t, gamma_variate(t, parms), Color = FSC_Color('red')
>
> end;-- --

Subject: Re: MPFIT question
Posted by Craig Markwardt on Thu, 15 Jan 2009 07:52:17 GMT
View Forum Message <> Reply to Message

On Jan 14, 4:37 pm, "j.coe...@gmail.com" <j.coe...@gmail.com> wrote:
> The user function has help statements at its beginning and end now,
> and I tried the other suggestions, but I'm still stumped. I'm doing
> something foolish.
>
> Below is the problem code. Most of it is just the 353-element data
> array. Without the data, it's only two short functions, 20 lines of
> code or so total. I hope it's not bad netiquette to post so many
> lines. I can't figure out why these data crash the program. I can
> fit most other data acquired in this manner -- I have fit hundreds of
> thousands of these curves with this method, but the error crops up
> once every 50 cases or so (usually for data that don't fit the model
> very well, but it still shouldn't crash?).
>
> The data are being fit to a gamma variate function. Two things I'm
> not sure about: (1) I'm passing t0 as a fitting parameter, and I'm not
> sure if this is acceptable. (2) Worse, I don't have a good idea of
> how to estimate error, so I'm just using the standard deviation of the
> baseline (pre-arrival) curve, which is probably wrong. Suggestions?
>
> The code crashes after just two MPFIT iterations. Just type
> 'MPFIT_problem' to run.
>
> Thanks for any help.
>
> ----------
>
> function gamma_variate, x, p
>
> print, 'p start...'
> help, p
>

```
>  baseline = p[0]
>  t0 = p[1]
>  tmax = p[2]            ; time of peak signal
>  smax = p[3]            ; signal max in mVolts
>  alpha = p[4]
>
>         ; shift time, apply gamma variate only to data after t0
>
>  wh = Where(x GE t0, ct)
>  t = (x - t0)[wh]
>  tmax = tmax - t0
>
>         ; gamma variate function
>
>  s = baseline + (smax) * (tmax^(-alpha)) * exp(alpha) * (t^alpha) *  exp
>  ((-alpha)  * t / tmax)
>
>         ; add the pre-t0 data (baseline)
>
>  n = n_elements(x)
>  pre = make_array(n - ct, Value = baseline)
>  s = [pre, s]
```

Thanks for your complete example, very useful!

Your function is crashing at the MAKE_ARRAY stage. This is because N EQ CT, so you are asking for an empty array.

After the error occurs, MPFITFUN() returns !NaN, in addition to error status keywords. Since you don't trap the error codes, execution proceeds to the next statement which attempts to evaluate GAMMA_VARIATE (T, !NaN).

So the lessons learned are:
  * validity checking before MAKE_ARRAY
  * error checking after MPFITFUN returns

Good luck!
Craig

Subject: Re: MPFIT question
Posted by j.coenia@gmail.com on Thu, 15 Jan 2009 16:17:56 GMT
View Forum Message <> Reply to Message

Thanks both of you. I knew it would be something easy for someone. I was thinking that the lower constraint on t0 in parinfo was good enough to ensure that the WHERE statement found at least some times

above t0, so that the found count (ct) would never be 0 as Paolo describes... unfortunately t0 can go all the way down to 0, in which case ct = n, which leads to the problem Craig describes: there is no pre-arrival baseline to prepend to the gamma variate if t0 is 0! Simply changing the GE to GT in the WHERE statement is a quick fix, but I'll implement better validity checking and MPFIT error trapping (which I didn't really understand until Craig explained that the initial MAKE_ARRAY error is being intercepted).

So my code was wrong, but there's nothing theoretically wrong with trying to fit t0 here, or any piecewise continuous function, using Levenberg-Marquardt? And my method for guessing error is not specious? Sorry for the naive questions -- it takes a few days to run all these fits, so I don't want to do everything wrong from the start.

Subject: Re: MPFIT question
Posted by Craig Markwardt on Sat, 17 Jan 2009 06:10:30 GMT
View Forum Message <> Reply to Message

On Jan 15, 11:17 am, "j.coe...@gmail.com" <j.coe...@gmail.com> wrote:

> So my code was wrong, but there's nothing theoretically wrong with
> trying to fit t0 here, or any piecewise continuous function, using
> Levenberg-Marquardt?

There's definitely no requirement that the model function is smooth.

Craig
