Subject: Re: using mpfit
Posted by Craig Markwardt on Mon, 09 Nov 1998 08:00:00 GMT
View Forum Message <> Reply to Message

nospam@ll.mit.edu (Joseph Scott Stuart) writes:
>
>
> I'm using Craig Markwardt's excellent fitting routine mpfit, and I'm
> getting strange behavior that I don't understand. It calls the
> function I'm fitting 48 times to do one fitting iteration, but each
> time that it calls it, it passes in the exact same parameters. Then
> mpfit exits with the message about the angle between fvec and the
> jacobian. The fitted parameter values are unchanged from the input
> parameters.
>
> I have specified, via the params structure, that a couple of the
> parameters are to be held fixed, but I've checked the params structure
> as it is going in, and it looks OK to me, only the first few
> parameters have the fixed keyword equal to 1. I don't understand why,
> when I print out the parameters from within my function, they are
> unchanged all 48 times that mpfit calls the routine.
>

Any fitting routine must compute derivatives of the model function
with respect to each parameter. MPFIT computes derivatives
automatically by performing a finite difference approximation.
It's quite a simple routine:

derivative = (f(p+h) - f(p)) / h

h is a finite step in the parameter value p, and is chosen *by
default* to be very small. It does this one time for each of the
parameters in your fit.

Therefore, it may *look* like MPFIT is calling your function with the
same parameters each time, but I would bet that they are actually
slightly different. [Also, check to be sure that both your
parameters and data are the same type, preferably DOUBLE].

If your fitting function is not-so-well behaved, or if it is not
sensitive to small changes in the parameter values, then it is
possible that MPFIT can't compute a proper automatic derivative. Of
course the fitting will fail. Incidentally, the MPFIT code retries
several times in one iteration before failing. That's probably why
your function is called so many times.

OK, don't worry, there is a simple solution. You can *manually*
choose the step size of any parameter using the PARINFO structure. If

the STEP tag is greater than zero, then it is used to compute the
derivative. Here's an example:

```
pi = replicate({value:0.D, fixed:0, limited:[0,0], $
                limits:[0.D,0.D], step:0.D}, 4)  ;; Assumes 4 parameters
pi(0).step = 0.01  ;; Makes 0.01 the derivative step size for parameter 0
;; All other parameters use default step size
```

The step size should be small enough to capture the relevant scale of
interest in your problem, whatever that may be.

Good luck,

Craig

P.S. The parameter itself can still vary on finer scales than the
step size. STEP is only used to compute the derivative.

--

 --- --------------
Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@astrog.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 --- --------------