
Subject: Re: Curve Fitting Question

Posted by [korpela](#) on Tue, 14 Sep 1993 19:27:56 GMT

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

In article <1993Sep14.100111.1@aurora.alaska.edu> ftdwh@aurora.alaska.edu writes:

>
> The question I have is how to properly use the curvefit routine to fit a
> function that I am defining. It is as follows:
>
> $mclat = A1 + A2 \cdot \cos(mlt + A3) + A4 \cdot \cos(2 \cdot mlt + 2 \cdot A5) + A6 \cdot \cos(3 \cdot mlt + 3 \cdot A7)$
>
> where mclat is the magnetic co-latitude, mlt is an angular representation of
> the magnetic local time. (This is a Fourier fit)
>
> I have in my data set the mclat and mlt, but I want to find the coefficients
> A1-A7. Can I do this using the fitting routine in IDL? If I can, what are the
> steps I need to follow? (As with most manuals they seem to be written for
> somebody who already knows what they are doing. Along that train of thought
> can any body recommend a book that might help those of us not fully
> knowledgeable in IDL)
>

In order to use the curvefit routine you need to make your own procedure
that returns mclat and if required the partial derivatives
of mclat with respect to the coefficients A1-7. (a(0:6) in idl)

You can find out about a lot of this stuff by reading the actual curvefit
routine. (generally in the \$IDL_DIR/lib/userlib directory)

Here's an example below. Extract the following into two files then
.run mclatfunct. You should then be able to

```
a=[a0,a1,a2,a3,a4,a5,a6] ;first guess of parameters
plot,mltdata,mclatdata ; these are the data you are trying to fit to
mclatfit=curvefit(mltdata,mclatdata,weight,a,sigmaa,FUNCTION_NAME= "mclatfunct")
```

That's all there is to it. I haven't compiled the stuff below, so
beware syntax errors. Depending upon how screwy this function is,
you may need to change the value of dg. Your first guess can also
be important. If you don't like to see how it's doing you can remove
the plot and oplot statements.

```
; file mclat.pro -----
; E. Korpela 9/14/93
Function mclat,mlt,a,pder

; calculate function
```

```

mclat1=a(0)+a(1)*cos(mlt+a(2))+a(3)*cos(2.*mlt+2.*a(4))+a(5) *cos(3.*mlt+3*a(6))
; if necessary calculate partial derrivitives
dg=1.e-5
; dg may need to be larger or smaller or may need different values for each
; coefficient. You'll have to experiment.
if n_params() eq 3 then begin
    szx=size(mlt)
    pder=fltarr(szx(1),7)
    for i=0,6 do begin
        da=a*0.
        da(i)=dg
        pder(*,i)=(mclat(mlt,a+da)-mclat1)/da(i)
    endfor
endif
return,mclat1
end
; end mclat.pro -----

; file mclatfunct.pro -----
; E. Korpela 9/14/93
pro mclatfunct,x,a,y,pder
; who's bright idea at RSI was it to call these procedures functions?

case n_params() of
    3:  y=mclat(x,a)
    4:  y=mclat(x,a,pder)
else:  print,'Function call screwup!!!'
endcase
print,'Evaluating Function'
print,'a= ',a
if !d.name ne 'PS' then oplot,x,y
return
end
; end mclatfunct.pro -----

```

--

Eric Korpela		The two most common things in the
korpela@ssl.berkeley.edu		universe are Hydrogen and stupidity.
		-Harlan Ellison