Subject: Re: Help Wanted: IDL Math Expert
Posted by J.D. Smith on Tue, 19 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Hi Folks,
>
> I had a friend ask a question about IDL that I didn't know how
> to answer. (Uh, math is not my strength, you understand. :-))
>
> Suppose I have a set of raw data that is described
> theoretically by two parametric equations. And suppose
> I need to fit these two parametric equations to the data
> since there is no mathematical way to convert the x(t)
> and y(t) equations into a y(x) form. Is there a
> curve fitting routine in IDL that can handle parametric
> equations? Or, failing that, has anyone handled something
> like this in IDL and would be willing to give us a little
> help?
>
> Many thanks,
>
> David
>

Presuming I understand you correctly, here is how I'd handle it: You have some data y and x, each of which depend on a parameter t through the equations y=y(t); x=x(t). Neither are analytically invertible.. i.e. you cannot compute:

$$t = x^{-1}(t)[x] \; ; \; y = y(t)$$

where the inverse of x(t) is applied to the data values x. The solution is to numerically compute the inverse of x(t) for each data value x. There's no efficient IDL builtin for doing this, though NEWTON could be used. I'd use zbrent() based on Num. Recipes routine of the same name. It's in the nasa package of IDL routines, available many places. This would be called in the curvefit fitting function. E.g.

fit=curvefit(xdata,ydata,weights, A, /NODERIVATIVE,FUNCTION_NAME='func')

where A=[ax,bx,cx,...,ay,by,cy,...] ... i.e. the parameters from both your functions concatenated together. And now for func:

pro func,X,A,F

```
 common funcblock, Asav, Xsav
 Asav=A & Xsav=X
 tmin=0. & tmax=200. ; substitute min and max possible t's
       ; x(tmin) and x(tmax) must be opposite sign
       ; these values might depend on X, or could
        ; be fixed (valid for each X data value)
 t=zbrent(tmin,tmax,FUNC_NAME='brentfunc')
 F=.... ;compute F=y(t) with t and ay,by,cy,....
 return
end

function brentfunc,t
 common funcblock, Asav, Xsav
 x= ... ;compute x(t) using ax,bx,cx,... from Asav
 return, x-Xsav ;finds t such that x(t)=Xsav (data value)
end
```

This should do it. It's a bit inefficient. If x and t increase together for all relevant t, you could make it more efficient by inverting the xdata to t in a separate step, using the previous value of t found as the new tmin (a starting point) in the call to zbrent for the next X (sorted of course). This should save a few iterations in zbrent (but it's usually really fast), and overcome curvefit inefficiencies. Good luck.

JD
