Thanks everyone for your suggestions, I ended up writing a function to
do a windowed brute force search because I wanted the continuous not
discrete maximum of the function. Sometimes it's unfortunate not being
a mathemetician, but thanks Craig for your analytical solution, it
will work for this function!

Kate Morgan
Research Technician
Dept. of Medical Imaging
University of Toronto

Paul van Delst <paul.vandelst@noaa.gov> wrote in message
news:<3EF225E6.33D91065@noaa.gov>...
> Kate wrote:
>>
>> If I have a function definition, say it is
>>
>> FUNCTION MYGAMMA, X, P
>>   RETURN, P[0]*(X^P[1])*EXP(-1*X/P[2])
>> END
>>
>> Where P holds my fit parameters that are already determined.
>>
>> If I want to find a local maximum in a certain range is it possible to
>> do this with an existing written IDL function?
>
> How about something like:
>
>   maxY = MAX(mygamma(X,P))
>
> ?
>
> You already have your P's and your X's define the range in which the maximum is sought. If
> you need the X-value of the maximum, then
>
>   maxY = MAX(mygamma(X,P), maxXLoc)
>   maxX = X[maxXLoc]
>
> If you don't want to carry around a bunch of X-vectors, then why not do something like:
>
>   FUNCTION FINDMAX, Xr, P, UserFunction

```
>     X1 = DOUBLE(Xr[0])  ; The beginning of the range
>     X2 = DOUBLE(Xr[1])  ; The end of the range
>     dX = DOUBLE(Xr[2])  ; The resolution within the range
>
>     ; Compute the ordinate vector
>     nX = LONG( (X2-X1)/dX ) + 1.5d0 )
>     X = ( DINDGEN( nX ) / DOUBLE( nX-1 ) ) * ( X2-X1 ) + X1
>
>     ; Find the maximum value of your user function
>     Y = CALL_FUNCTION( UserFunction, X, DOUBLE(P) )
>     maxY = MAX( Y, maxXLoc )
>     maxX = X[ maxXLoc ]
>
>     ; Return the coordinate of the local max
>     RETURN, [ maxX, maxY ]
>
>   END
>
> where the Xr input defines the function range [x1,x2] and the resolution at which the
> function is calculated (Sort of like a loop triplet.)
>
> Otherwise maybe you could use something like the FX_ROOT function to find the roots of the
> first derivative of your function (via DERIV or your own calculated derivative) and use
> the second derivative to determine if it's a max or min.
>
> paulv
```