Subject: I need a bit of help....Convol and functions
Posted by D.Kochman@gmail.com on Sun, 01 Oct 2006 10:42:34 GMT
View Forum Message <> Reply to Message

So, I'm fairly new to IDL (and an organic chemist so programming is not my forte), but I'm chunking my way through it. I'm currently in the process of modifying a program that fits exponential decays given the impulse response function and the decay curve.

I have to remodel it to fit another much more complex function than an exponential decay, however with a similar number of parameters. I've already fixed the GUI, and changed all the references to the widgets, along with adjusting the appropriate arrays. It now compiles after many hours of debugging and displays itself appropriately with a dummy function with the appropriate amount of parameters.

I'm stuck with now implementing the function itself, any help on the implementation will be *highly* appreciated.

The function is an infinite sum convolved with an exponential decay. I've done modeling with the sum, and it converges fairly rapidly, and I can limit it to 10 terms or so and still get accuracy to 6 decimal places.

Approximately it is:

Sum[(-1)^n*cos(n*P(1)*X)*exp(-(2n)^2*P(2)*X), n ->0 to 10] convol exp[-X/P(4)]

whew

anyways, I've been working through the documentation on convol, and I find it a bit cryptic. I have very few clues how to implement this function in code. I'm guessing the first portion (the sum portion) needs to be recursively defined in a for loop. Is this the case, or is their a shortcut with a sigma type function built in?

However, how do I easily convolve the two functions if they are functions and not arrays? Should I just go to fourier space?

Thanks for any help. I don't expect anyone to code this for me, just a gentle (or violent) shove in the appropriate direction will be infinately helpful.

Subject: Re: I need a bit of help....Convol and functions
Posted by James Kuyper on Wed, 04 Oct 2006 13:43:33 GMT

View Forum Message <> Reply to Message

D.Kochman@gmail.com wrote:
>> Yes, func() needs to create an array for CONVOL() to convolve. However,
>> sumex should already be an array, in order for this code to work as
>> intended, and this code does nothing to change any aspect of sumex. I'm
>> not sure I understand what you mean by the comment "but its a
>> function".
>
>
> Thanks for the help, slowly but surely I'm starting to get it. What I
> meant by "but its a function" is I just don't see how sumex is an
> array. If I were to put
>
> sumex = X
>
> that to me makes sumex a function, namely, f(x)=x

No, in IDL, the statement sumex=x would make the variable named sumex
refer to an object which is an exact copy of x. If X is an array, then
sumex will be an array of the same type and shape. In the actual code
you gave us,

sumex=P(0)*exp(-X / P(1))+P(2)*exp(-X / P(3))+P(7)*exp(-X / P(8))

dividing X by P(1) creates a new array with the same shape as X.
Passing that array to exp() produces a result with the same shape as X,
and multiplying it by P(0) also produces an array in the same shape.
The same applies to each of the terms in that sum, and adding all three
terms together also produces an array of the same shape.

Subject: Re: I need a bit of help....Convol and functions
Posted by D.Kochman@gmail.com on Wed, 04 Oct 2006 13:53:49 GMT
View Forum Message <> Reply to Message

kuyper@wizard.net wrote:
> D.Kochman@gmail.com wrote:
>>> Yes, func() needs to create an array for CONVOL() to convolve. However,
>>> sumex should already be an array, in order for this code to work as
>>> intended, and this code does nothing to change any aspect of sumex. I'm
>>> not sure I understand what you mean by the comment "but its a
>>> function".
>>
>>
>> Thanks for the help, slowly but surely I'm starting to get it. What I
>> meant by "but its a function" is I just don't see how sumex is an
>> array. If I were to put

>>
>> sumex = X
>>
>> that to me makes sumex a function, namely, f(x)=x
>
> No, in IDL, the statement sumex=x would make the variable named sumex
> refer to an object which is an exact copy of x. If X is an array, then
> sumex will be an array of the same type and shape. In the actual code
> you gave us,
>
> sumex=P(0)*exp(-X / P(1))+P(2)*exp(-X / P(3))+P(7)*exp(-X / P(8))
>
> dividing X by P(1) creates a new array with the same shape as X.
> Passing that array to exp() produces a result with the same shape as X,
> and multiplying it by P(0) also produces an array in the same shape.
> The same applies to each of the terms in that sum, and adding all three
> terms together also produces an array of the same shape.

Aha! Therein lies my problem. You sir are a god amongst men. =)
Thanks a billion.

OK, thanks for all the help, I do understand now. WOOHOO! Sometimes
you have to realize how far you have to come in order to really
understand something. I know I don't really understand IDL, but I'm
slowly progressing. The deceptively simple syntax of IDL hides some
surprises.
