
Subject: Re: I need a bit of help....Convol and functions
Posted by [James Kuyper](#) on Mon, 02 Oct 2006 14:58:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

D.Kochman@gmail.com wrote:

...
> 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
> there a shortcut with a sigma type function built in?

Normally, TOTAL() would do the job of the Sigma symbol, but the fact that you have to perform a function call to perform the convolution gets in the way of that approach. Loops are indeed the way to go, in this case.

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

CONVOL cannot convolve functions as functions, it can only calculate a numerical approximation to the result of the convolution. That means you have to represent the functions you are convolving, for CONVOL, as arrays of function values at specific points. You'll want to use one of the functions as the kernel for the convolution; I'd recommend using whichever one drops off fastest with X. For your function, that depends upon P(2), P(4), n, and the range of X values that you're convolving over.

Fourier space won't allow you to get around this problem, unless you can perform the calculations analytically in that space. Otherwise, all you've done is change from using an array which represents the value of the function at specific points, to using an array which represents the value of the function's fourier transform at specific frequencies.

Subject: Re: I need a bit of help....Convol and functions
Posted by [JD Smith](#) on Mon, 02 Oct 2006 17:14:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Sun, 01 Oct 2006 03:42:34 -0700, D.Kochman@gmail.com wrote:

> 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:
>
> $\text{Sum}[(-1)^n \cos(n \cdot P(1) \cdot X) \cdot \exp(-(2n)^2 \cdot P(2) \cdot X), n \rightarrow 0 \text{ to } 10] \text{ convol}$
> $\exp[-X/P(4)]$
>
> **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 there 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 infinitely
> helpful.

For the sum, you can use a total over an array:

```

nx=n_elements(X)
n=rebin(transpose(lindgen(11)),nx,11)
func=total((-1.)^n*cos(n*P[1]*X)*exp(-(2.*n)^2*P[2]*X),2)*exp(-X/P[4])

```

Are you sure it's really a convolution? When the kernel size is the same as the thing its convolving, I almost always think "multiplication" not "convolution". Otherwise, it has no scale. If it occurred over some fixed scale X_0 , then a convolution would be more

understandable.

For speed, you could cache the "n" array above to avoid having to recreate it, assuming nx doesn't change.

JD

Subject: Re: I need a bit of help....Convol and functions
Posted by D.Kochman@gmail.com on Mon, 02 Oct 2006 18:36:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

>
> Are you sure it's really a convolution? When the kernel size is the
> same as the thing its convolving, I almost always think
> "multiplication" not "convolution". Otherwise, it has no scale. If
> it occurred over some fixed scale X0, then a convolution would be more
> understandable.

First of all thanks very much for your guys's help. Much appreciated.
I'm enjoying learning IDL, but at the same time I have to take a
functional approach about it.

So I'm 99% sure its convolution:

we have a 3 state system

lets just call them 1,2,3

Independant of state 3, 2 relaxes to 1 with one particular rate, t, and
is described by $\exp[-x/t]$

2 relaxes to 3 in the other way, which has a time dependant evolution -
the rate is not consistent with time, and is instead described by that
sum. We are only measuring what observed rate of 2 to 1.

My opinion is if 2 to 3 rate was time independant, than it would be
multiplication.

Thats how I feel anyways, but I'm by no means completely stubborn. ;)

Once again, thanks for the help.

Subject: Re: I need a bit of help....Convol and functions
Posted by [James Kuyper](#) on Mon, 02 Oct 2006 20:00:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

JD Smith wrote:

> On Sun, 01 Oct 2006 03:42:34 -0700, D.Kochman@gmail.com wrote:

>

>> 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:

>>

>> $\text{Sum}[(-1)^n \cos(n \cdot P(1) \cdot X) \cdot \exp(-(2n)^2 \cdot P(2) \cdot X), n \rightarrow 0 \text{ to } 10] \text{ convol}$
>> $\exp[-X/P(4)]$

>>

>> **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 there 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 infinitely
>> helpful.

>

> For the sum, you can use a total over an array:

>

> `nx=n_elements(X)`

> `n=rebin(transpose(lindgen(11)),nx,11)`

> `func=total((-1.)^n*cos(n*P[1]*X)*exp(-(2.*n)^2*P[2]*X),2)*exp(-X/P[4])`

>

- > Are you sure it's really a convolution? When the kernel size is the
- > same as the thing its convolving, I almost always think
- > "multiplication" not "convolution".

What you've written is indeed the multiplication, and is therefore not a correct response to his request. The true mathematical convolution of $f(x)$ and $g(x)$ is $y(x) = \text{the integral over } z \text{ from } -\infty \text{ to } +\infty \text{ of } f(x-z)*g(z) \, dz$. It's only possible to approximate the mathematical convolution numerically when you can approximate the infinite integral with acceptable accuracy by integrating over a finite range. That's why the numerical convolutions you've seen have involved small kernels.

Since the relevant integral covers arbitrarily large negative values of x , the exponential terms in the functions he wants to convolve worry me. I'm not sure the integral converges, though it's been more than a decade since the last time I did integrals like that. If it does converge, I suspect it can be convolved analytically for each value of n separately.

Subject: Re: I need a bit of help....Convol and functions
Posted by [JD Smith](#) on Mon, 02 Oct 2006 21:59:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Mon, 02 Oct 2006 13:00:17 -0700, kuyper wrote:

- > JD Smith wrote:
- >> On Sun, 01 Oct 2006 03:42:34 -0700, D.Kochman@gmail.com wrote:
- >>
- >>> 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 there 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
>>> infinitely helpful.
>>
>> For the sum, you can use a total over an array:
>>
>> nx=n_elements(X)
>> n=rebin(transpose(lindgen(11)),nx,11)
>> func=total((-1.)^n*cos(n*P[1]*X)*exp(-(2.*n)^2*P[2]*X),2)*exp(-X/P[4])
>>
>> Are you sure it's really a convolution? When the kernel size is the same
>> as the thing its convolving, I almost always think "multiplication" not
>> "convolution".
>
> What you've written is indeed the multiplication, and is therefore not a
> correct response to his request. The true mathematical convolution of f(x)
> and g(x) is  $y(x) = \int_{-\infty}^{+\infty} f(x-z)g(z) dz$ . It's only possible to approximate the mathematical
> convolution numerically when you can approximate the infinite integral
> with acceptable accuracy by integrating over a finite range. That's why
> the numerical convolutions you've seen have involved small kernels.

```

Yes it is doing multiplication, as noted. I've found that many people use the terminology "convolve x with y" when they mean "multiply x by y". Sounds like that's not the case here, but anyway...

JD

Subject: Re: I need a bit of help....Convol and functions

Posted by D.Kochman@gmail.com on Tue, 03 Oct 2006 20:42:39 GMT

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

Hey you guys, once again thanks for the help. I'm still having fun learning IDL. ;)

So heres my next problem is finding out how this was implemented so I can replicate the procedure in order to do my own convolution. Pretty much what this following function is is a multiple convolution. It inputs a system response file (confun) which is 1024 data points.

```
FUNCTION func, P, X=x, Y=y, ERR=err, CONFUN=confun, $
    funcvals=f, binsizefactor=bsf

fshift=floor(-P(4)/bsf)
rshift=-P(4)/bsf-fshift
fconfun=shift(confun,-fshift)
sconfun=(1-rshift)*fconfun+rshift*SHIFT(fconfun,-1)
sumex=P(0)*exp( -X / P(1) )+P(2)*exp( -X / P(3) )+P(7)*exp( -X / P(8) )
    blank=dblarr(1024)
    consum = convol([blank,sumex],sconfun,total(sconfun),CENTER=0)
    consumex=consum[1024:]*]
scatter= P(6)*sconfun
f=P(5)+consumex+scatter
err=sqrt(f)
return,(y-f)/(err)
end
```

So my big question is what is going on in convol? If I'm guessing correctly it seems like this is creating an array [blank,sumex] in essence to force an array upon sumex - but its a function. Are specific data points being used in sumex (ie x=0->1024)? I thought convol only took two parameters, which would mean sconfun would be the kernel, but I can't see that thats an array either? What is then the purpose of total(sconfun)?

I've managed to figure out just about everything else in this program from the creation of the datastructures, and the widgets and so on. That stuff is easy - and easily changable. I'm kind of lost with this one.

-Dave

Subject: Re: I need a bit of help....Convol and functions

Posted by news.qwest.net on Tue, 03 Oct 2006 23:04:28 GMT

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

<D.Kochman@gmail.com> wrote in message
news:1159908159.612660.56200@i3g2000cwc.googlegroups.com...

> consum = convol([blank,sumex],sconfun,total(sconfun),CENTER=0)

> So my big question is what is going on in convol? If I'm guessing
> correctly it seems like this is creating an array [blank,sumex] in
> essence to force an array upon sumex - but its a function.

[blank,sumex] is a one dimensional array (assuming X is) where the
actual "time series" (sumex) is being zero padded for 1024 points.

sconfun is the kernel

total(sconfun) is a scaling factor, ie the entire result gets divided by
this number

One big thing you should know is that convol() can do either a convolution,
or a correlation.

If the keyword "center" is set to one it does the correlation, and if it is
zero

then you get a convolution (or vice versa) and there may be the need to
reverse
the kernel beforehand.

Cheers,
bob

Subject: Re: I need a bit of help....Convol and functions
Posted by [James Kuyper](#) on Tue, 03 Oct 2006 23:17:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

D.Kochman@gmail.com wrote:

> Hey you guys, once again thanks for the help. I'm still having fun
> learning IDL. ;)

>

> So heres my next problem is finding out how this was implemented so I
> can replicate the procedure in order to do my own convolution. Pretty
> much what this following function is is a multiple convolution. It
> inputs a system response file (confun) which is 1024 data points.

>

>

> FUNCTION func, P, X=x, Y=y, ERR=err, CONFUN=confun, \$
> funcvals=f,binsizefactor=bsf


```

>
> fshift=floor(-P(4)/bsf)
> rshift=-P(4)/bsf-fshift
> fconfun=shift(confun,-fshift)
> sconfun=(1-rshift)*fconfun+rshift*SHIFT(fconfun,-1)
> sumex=P(0)*exp( -X / P(1) )+P(2)*exp( -X / P(3) )+P(7)*exp( -X / P(8) )

```

I would strongly recommend use of the square bracket notation, such as `P[2]`, for array indices. The parenthesis notation, while legal, is easily confused with a function call syntax. The results when you mistype an array name get very confusing.

```

> blank=dblarr(1024)
> consum = convol([blank,sumex],sconfun,total(sconfun),CENTER=0)
> consumex=consum[1024:]*]
> scatter= P(6)*sconfun
> f=P(5)+consumex+scatter
> err=sqrt(f)
> return,(y-f)/(err)
> end
>
> So my big question is what is going on in convol?

```

I'd recommend going to the online help for `CONVOL()`. It's very explicit about what precisely is going on.

```

> If I'm guessing
> correctly it seems like this is creating an array [blank,sumex] in
> essence to force an array upon sumex - but its a function.

```

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".

```

> Are
> specific data points being used in sumex (ie x=0->1024)?

```

Yes. Every single data point in `sumex` is being used to create an element in the temporary array which is created by the expression `[blank,sumex]`. Every single element in that temporary array contributes to the calculation of the `consum` array returned by `CONVOL()`.

```

> ... I thought
> convol only took two parameters, which would mean sconfun would be the
> kernel, but I can't see that thats an array either?

```

sconfun should be an array, in order for this code to work correctly. It will be an array, if confun is an array. In other words, it's up to the caller of func() to make sure that it is.

> ... What is then the
> purpose of total(sconfun)?

The third argument to CONVOL() is an optional scale factor, which defaults to 1. The convolution is implemented as a sum which is divided by that factor. It's not very important for floating point values: CONVOL(array, kernel, S) is pretty much equivalent to CONVOL(array/S, kernel) or CONVOL(array, kernel/S) or CONVOL(array,kernel)/S; which one you use is your choice. However, for integer and byte arrays,

CONVOL(array, [[1,1],[1,1]], 4)

is faster than

CONVOL(array, [[0.25, 0.25],[0.25, 0.25]])

because it avoids the floating point conversion, and it is coded in such a way as to protected against the overflow problems that would occur if you tried

CONVOL(array, [[1,1],[1,1]])/4

Note: the /NORMALIZE option sets the scale factor to TOTAL(ABS(kernel)), which is often the most appropriate value, and is probably more appropriate than the value of TOTAL(sconfun) used in the code you're looking at. It is also recommended if you use the /INVALID or /NAN options, since it will scale the results only by the total over the kernel elements actually used.

Subject: Re: I need a bit of help....Convol and functions
Posted by D.Kochman@gmail.com on Wed, 04 Oct 2006 12:59:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

> 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, $f(x)=x$

if it was [(1,1),(2,2),(3,3)....] that to me is an array.

I know it has to be an array though.....I'll just stare at it some more.

The reason I ask is related to my original question, and that is pretty much that I have to redefine sumex, with my original monstrosity of an equation. Which approximately is $(\cos(x)*\exp(-x)) \text{ conv } \exp(-x/t)$. I wanted to know how the original author of the software made sumex an array, so I can apply the same principles to my equations and not do the convolution analytically, but rather discretely. Its tough jumping into a new language with the source code having no commenting, and trying to pick it all apart.

I will re look at the documentation for convolve, and twiddle around with things to see how they work, and how I can manipulate them I can do.

As always, I'm grateful for whatever help I can get here. This is going to be a long year of IDL learning, and I did myself the necessary disservice of jumping in on the deep-end. =P Its been a long while too since I did any programming (8 years or so), so I'm relearning all the jargon. Slowly but surely its all coming back. Nothing really has the power to make me feel like such an idiot as programming. However, in the end when everything works its a very satisfying feeling.
