Subject: Re: Convolution, IDL & Numerical Recipes
Posted by JD Smith on Tue, 05 Nov 2002 22:42:59 GMT
View Forum Message <> Reply to Message

On Tue, 05 Nov 2002 06:34:42 -0700, R.G. Stockwell wrote:

> Hector Aceves wrote:
>> "R.G. Stockwell" <sorry@noemail.now> wrote in message
>> news:<3DC28954.7060605@noemail.now>...
>>
>>> Perhaps you want to use the following keywords: Check out the help file
>>> to see the effects the keywords have on how the arrays line up to be
>>> convolved. (Note: you must explicitly set center=0, or else it defaults
>>> to 1)
>>>
>>> z=convol(a,k,center=0,edge_wrap=1)
>>>
>>> a 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 k 1 0 0 0 0 0 0
>>> 0 0
>>>
>>> z 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0
>>>
>>>
>>> Cheers,
>>> bob stockwell
>>
>>
>> Dear Bob...
>>
>> It works well with the kernel [1,0,...] But when I tried the actual
>> examples of Numerical Recipes it did not give me the same results:
>>
>> a=[0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0]
>> k=[0,0,1,1,1,1,0,0,0]
>>
>> z=convol(a,k,center=0,edge_wrap=1)
>> IDL> print,z
>> 0 0 0 0 0 0 0 1 2 3
>> 4 4 3 2 1 0
>> IDL>
>>
>> With Numerical Recipes gives..
>>
>> 0 1 1 1 1 1 0 1 2 3 3 3 2 1 0 0
>>
>> which seems ok!
>
> If by "ok" you mean "completely wrong" then I agree with you. :)

>
> Correllating two "boxcars" gives you a "triangle". Perhaps you typed in
> the wrong "k" in your numrec code?
>
> a=[0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0]
> k=[1,1,1,0,0,0,0,0,1]
>
> z=convol(a,k,center=1,edge_wrap=0,edge_trunc=1)
>
> 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1
>
> 0 1 1 1 1 1 0 1 2 3 3 3 2 1 0 0
>
> Also, keep in mind, as J.D. mentioned, that IDL convol is a correlation
> with center=0, and a convolution with center = 1 (among other things).
>
> You'd probably be better off to write your own 10 line piece of code to
> perform the exact operation you want. Actually, I might even do that,
> but I have a lot of other work to do, so it's gonna be a while.
>
> I'd use an fft to do it, and if you want no edge wrap, just zeropad.

Have a look at the NASA-library's CONVOLVE, which explicitly takes all these IDL-native "features" into account, uses FFT when appropriate, and may save you the trouble of writing one yourself.

JD

---