
Subject: convolve mystery

Posted by on Wed, 06 Nov 2013 15:13:49 GMT

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

I found something surprising (to me) with the convolve() IDL function. There is something strange about how it does its Fourier wrap-around of an image from one side of the array to the other.

Here is a simple example. First define a simple image where half is unity and half is zero:

```
sz = 10
im = [replicate(1., sz/2), replicate(0., sz/2)] # replicate(1., sz)
print, 'Original:'
print, im[* , sz/2], format = '(f5.2)'
```

This gives the output:

Original:

```
1.00
1.00
1.00
1.00
1.00
0.00
0.00
0.00
0.00
0.00
```

Then define a point spread function and do the convolution:

```
psf1 = [[1., 1., 1.], [1., 5., 1.], [1., 1., 1.]]
psf1 = psf1/total(psf1)
imc1 = convolve(im, psf1)
print, 'With convolve:'
print, imc1[* , sz/2], format = '(f5.2)'
```

The output I get is:

With convolve:

```
0.77
1.00
1.00
1.00
0.77
0.23
-0.00
0.00
```

-0.00
-0.00

See how the wrap-around reduced the 1.00 in the first pixel to 0.75 but the last pixel does not get the corresponding increase?

Whereas if I do the equivalent operation explicitly with FFT, I do get the expected 0.23 in the last pixel:

```
psf2 = fltarr(sz, sz)
psf2[sz/2-1:sz/2+1, sz/2-1:sz/2+1] = psf1*sz*sz
psf2 = shift(psf2, sz/2, sz/2)
imc2 = float(fft(fft(im)*fft(psf2), /inv))
print, 'With fft:'
print, imc2[* , sz/2], format = '(f5.2)'
```

With fft:

0.77
1.00
1.00
1.00
0.77
0.23
-0.00
-0.00
-0.00
0.23

I've looked at the code in <http://www.astro.washington.edu/docs/idl/cgi-bin/getpro/library21.html?CONVOLVE> and as far as I can see (due to various options the code is not entirely straight forward to read), the fft convolution has no reason to do give any different result from what I do explicitly with fft.

Does anybody know what is going on?

Subject: Re: convolve mystery

Posted by [Helder Marchetto](#) on Wed, 06 Nov 2013 15:27:48 GMT

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

On Wednesday, November 6, 2013 4:13:49 PM UTC+1, Mats Löfdahl wrote:

> I found something surprising (to me) with the convolve() IDL function. There is something strange about how it does its Fourier wrap-around of an image from one side of the array to the other.

>

>

>

> Here is a simple example. First define a simple image where half is unity and half is zero:

```

>
>
>
> sz = 10
>
> im = [replicate(1., sz/2), replicate(0., sz/2)] # replicate(1., sz)
>
> print, 'Original:'
>
> print, im[*, sz/2], format = '(f5.2)'
>
>
>
> This gives the output:
>
>
> Original:
>
> 1.00
>
> 1.00
>
> 1.00
>
> 1.00
>
> 1.00
>
> 1.00
>
> 0.00
>
> 0.00
>
> 0.00
>
> 0.00
>
> 0.00
>
>
>
>
>
> Then define a point spread function and do the convolution:
>
>
>
> psf1 = [[1., 1., 1.], [1., 5., 1.], [1., 1., 1.]]

```

```

>
> psf1 = psf1/total(psf1)
>
> imc1 = convolve(im, psf1)
>
> print, 'With convolve:'
>
> print,imc1[*,'sz/2'], format = '(f5.2)'
>
>
>
> The output I get is:
>
>
>
> With convolve:
>
> 0.77
>
> 1.00
>
> 1.00
>
> 1.00
>
> 0.77
>
> 0.23
>
> -0.00
>
> 0.00
>
> -0.00
>
> -0.00
>
>
>
> See how the wrap-around reduced the 1.00 in the first pixel to 0.75 but the last pixel does not
get the corresponding increase?
>
>
>
> Whereas if I do the equivalent operation explicitly with FFT, I do get the expected 0.23 in the
last pixel:
>
>

```

```

>
> psf2 = fltarr(sz, sz)
>
> psf2[sz/2-1:sz/2+1, sz/2-1:sz/2+1] = psf1*sz*sz
>
> psf2 = shift(psf2, sz/2, sz/2)
>
> imc2 = float(fft(fft(im)*fft(psf2), /inv))
>
> print, 'With fft:'
>
> print,imc2[:,sz/2], format = '(f5.2)'
>
>
>
> With fft:
>
> 0.77
>
> 1.00
>
> 1.00
>
> 1.00
>
> 0.77
>
> 0.23
>
> -0.00
>
> -0.00
>
> -0.00
>
> 0.23
>
>
>
> I've looked at the code in http://www.astro.washington.edu/docs/idl/cgi-bin/getpro/library21.html?CONVOLVE and as far as I can see (due to various options the code is not entirely straight forward to read), the fft convolution has no reason to do give any different result from what I do explicitly with fft.
>
>
>
> Does anybody know what is going on?

```

Dunno,
but if you try this, you get what you expected:

```
imc2 = CONVOL_FFT(im, psf1, /NO_PADDING)
print,imc2[* ,sz/2], format = '(f5.2)'
0.77
1.00
1.00
1.00
0.77
0.23
-0.00
-0.00
-0.00
0.23
```

but of course you need at least IDL 8.1.
With padding you get the "wrong" result.

Cheers,
h

Subject: Re: convolve mystery
Posted by on Wed, 06 Nov 2013 15:53:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Den onsdagen den 6:e november 2013 kl. 16:27:48 UTC+1 skrev Helder:

```
>
> Dunno,
> but if you try this, you get what you expected:
>
> imc2 = CONVOL_FFT(im, psf1, /NO_PADDING)
> print,imc2[* ,sz/2], format = '(f5.2)'
> 0.77
> 1.00
> 1.00
> 1.00
> 0.77
> 0.23
> -0.00
> -0.00
> -0.00
> 0.23
>
> but of course you need at least IDL 8.1.
> With padding you get the "wrong" result.
```

Yes, but I see no sign of padding being used in convolve().

/Mats

Subject: Re: convolve mystery

Posted by [David Fanning](#) on Wed, 06 Nov 2013 16:03:06 GMT

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

Helder writes:

```
>
> On Wednesday, November 6, 2013 4:13:49 PM UTC+1, Mats Löfdahl wrote:
>> I found something surprising (to me) with the convolve() IDL function. There is something
>> strange about how it does its Fourier wrap-around of an image from one side of the array to the
>> other.
>>
>>
>> Here is a simple example. First define a simple image where half is unity and half is zero:
>>
>>
>>
>> sz = 10
>>
>> im = [replicate(1., sz/2), replicate(0., sz/2)] # replicate(1., sz)
>>
>> print, 'Original:'
>>
>> print, im[*,sz/2], format = '(f5.2)'
>>
>>
>>
>> This gives the output:
>>
>>
>>
>> Original:
>>
>> 1.00
>>
>> 1.00
>>
>> 1.00
>>
>> 1.00
>>
>> 1.00
```

```

>>
>> 0.00
>>
>> 0.00
>>
>> 0.00
>>
>> 0.00
>>
>> 0.00
>>
>>
>>
>>
>> Then define a point spread function and do the convolution:
>>
>>
>>
>> psf1 = [[1., 1., 1.], [1., 5., 1.], [1., 1., 1.]]
>>
>> psf1 = psf1/total(psf1)
>>
>> imc1 = convolve(im, psf1)
>>
>> print, 'With convolve:'
>>
>> print,imc1[* ,sz/2], format = '(f5.2)'
>>
>>
>>
>> The output I get is:
>>
>>
>>
>> With convolve:
>>
>> 0.77
>>
>> 1.00
>>
>> 1.00
>>
>> 1.00
>>
>> 0.77
>>
>> 0.23

```



```

>>
>> -0.00
>>
>> 0.00
>>
>> -0.00
>>
>> -0.00
>>
>>
>>
>> See how the wrap-around reduced the 1.00 in the first pixel to 0.75 but the last pixel does not
get the corresponding increase?
>>
>>
>>
>> Whereas if I do the equivalent operation explicitly with FFT, I do get the expected 0.23 in the
last pixel:
>>
>>
>>
>> psf2 = fltarr(sz, sz)
>>
>> psf2[sz/2-1:sz/2+1, sz/2-1:sz/2+1] = psf1*sz*sz
>>
>> psf2 = shift(psf2, sz/2, sz/2)
>>
>> imc2 = float(fft(fft(im)*fft(psf2), /inv))
>>
>> print, 'With fft:'
>>
>> print,imc2[:,sz/2], format = '(f5.2)'
>>
>>
>>
>> With fft:
>>
>> 0.77
>>
>> 1.00
>>
>> 1.00
>>
>> 1.00
>>
>> 0.77
>>
>> 0.23

```

```

>>
>> -0.00
>>
>> -0.00
>>
>> -0.00
>>
>> 0.23
>>
>>
>>
>> I've looked at the code in http://www.astro.washington.edu/docs/idl/cgi-bin/getpro/library21.html?CONVOLVE and as far as I can see (due to various options the code is not entirely straight forward to read), the fft convolution has no reason to do give any different result from what I do explicitly with fft.
>>
>>
>>
>> Does anybody know what is going on?
>
> Dunno,
> but if you try this, you get what you expected:
>
> imc2 = CONVOL_FFT(im, psf1, /NO_PADDING)
> print,imc2[* ,sz/2], format = '(f5.2)'
> 0.77
> 1.00
> 1.00
> 1.00
> 0.77
> 0.23
> -0.00
> -0.00
> -0.00
> 0.23
>
> but of course you need at least IDL 8.1.
> With padding you get the "wrong" result.

```

You get the result you expect if you use the IDL routine CONVOL (instead of the CONVOLVE you are using, and set the EDGE_WRAP keyword:

```

psf1 = [[1., 1., 1.], [1., 5., 1.], [1., 1., 1.]]
psf1 = psf1/total(psf1)
imc1 = convol(im, psf1, /edge_wrap)
print, 'With convol:'
print,imc1[* ,sz/2], format = '(f5.2)'
END

```

```
IDL> .go
% Compiled module: $MAIN$.
With convol:
0.77
1.00
1.00
1.00
0.77
0.23
0.00
0.00
0.00
0.23
```

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: convolve mystery
Posted by on Wed, 06 Nov 2013 16:14:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Den onsdagen den 6:e november 2013 kl. 17:03:06 UTC+1 skrev David Fanning:

> Helder writes:

>

>> Dunno,

>> but if you try this, you get what you expected:

>

>> `imc2 = CONVOL_FFT(im, psf1, /NO_PADDING)`

>

>

> You get the result you expect if you use the IDL routine CONVOL (instead
> of the CONVOLVE you are using, and set the EDGE_WRAP keyword:

Yes, but I already know how to get the expected result. The `fft(fft()*fft(),/inv)` method works fine for what I want to do. I merely wonder what's up with the `convolve()` function.

Subject: Re: convolve mystery

Posted by [wlandsman](#) on Wed, 06 Nov 2013 16:54:34 GMT

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

CONVOLVE() is user-written procedure in the IDL Astronomy Library. The version you are looking at is quite old -- the current version is in

<http://idlastro.gsfc.nasa.gov/ftp/pro/image/convolve.pro>

The documentation for the current version says that "the image is padded with zeros so that a large PSF does not overlap one edge of the image with the opposite edge of the image." So I'd say that CONVOLVE is giving the right answer -- or closer to what one would get with a true convolution. It also matches CONVOLVE_FFT() without the /NO_PADDING keyword.

-Wayne

P.S. Is it possible that you are using a newer version of CONVOLVE, and not the version on the Web page?

On Wednesday, November 6, 2013 10:13:49 AM UTC-5, Mats Löfdahl wrote:

> I found something surprising (to me) with the convolve() IDL function. There is something strange about how it does its Fourier wrap-around of an image from one side of the array to the other.

>
>
>
> Here is a simple example. First define a simple image where half is unity and half is zero:

>
>
>
> sz = 10
>
> im = [replicate(1., sz/2), replicate(0., sz/2)] # replicate(1., sz)
>
> print, 'Original:'
>
> print, im[*, sz/2], format = '(f5.2)'

>
>
>
> This gives the output:

>
>
>
> Original:
>
> 1.00
>
> 1.00
>

```

> 1.00
>
> 1.00
>
> 1.00
>
> 0.00
>
> 0.00
>
> 0.00
>
> 0.00
>
> 0.00
>
>
>
>
> Then define a point spread function and do the convolution:
>
>
>
> psf1 = [[1., 1., 1.], [1., 5., 1.], [1., 1., 1.]]
>
> psf1 = psf1/total(psf1)
>
> imc1 = convolve(im, psf1)
>
> print, 'With convolve:'
>
> print,imc1[* ,sz/2], format = '(f5.2)'
>
>
>
> The output I get is:
>
>
>
> With convolve:
>
> 0.77
>
> 1.00
>
> 1.00
>
>

```

```

> 1.00
>
> 0.77
>
> 0.23
>
> -0.00
>
> 0.00
>
> -0.00
>
> -0.00
>
>
>
> See how the wrap-around reduced the 1.00 in the first pixel to 0.75 but the last pixel does not
get the corresponding increase?
>
>
>
> Whereas if I do the equivalent operation explicitly with FFT, I do get the expected 0.23 in the
last pixel:
>
>
>
> psf2 = fltarr(sz, sz)
>
> psf2[sz/2-1:sz/2+1, sz/2-1:sz/2+1] = psf1*sz*sz
>
> psf2 = shift(psf2, sz/2, sz/2)
>
> imc2 = float(fft(fft(im)*fft(psf2), /inv))
>
> print, 'With fft:'
>
> print,imc2[* ,sz/2], format = '(f5.2)'
>
>
>
> With fft:
>
> 0.77
>
> 1.00
>
> 1.00
>

```

> 1.00
>
> 0.77
>
> 0.23
>
> -0.00
>
> -0.00
>
> -0.00
>
> 0.23
>
>
>
> I've looked at the code in <http://www.astro.washington.edu/docs/idl/cgi-bin/getpro/library21.html?CONVOLVE> and as far as I can see (due to various options the code is not entirely straight forward to read), the fft convolution has no reason to do give any different result from what I do explicitly with fft.
>
>
>
> Does anybody know what is going on?

Subject: Re: convolve mystery

Posted by on Wed, 06 Nov 2013 20:00:21 GMT

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

On 2013-11-06 17:54, wlandsman wrote:

> CONVOLVE() is user-written procedure in the IDL Astronomy Library. The version you are looking at is quite old -- the current version is in

>

> <http://idlastro.gsfc.nasa.gov/ftp/pro/image/convolve.pro>

>

> The documentation for the current version says that "the image is padded with zeros so that a large PSF does not overlap one edge of the image with the opposite edge of the image." So I'd say that CONVOLVE is giving the right answer -- or closer to what one would get with a true convolution. It also matches CONVOLVE_FFT() without the /NO_PADDING keyword.

OK, so there is padding. I'm not sure zero padding is what you want to do as a default. For people who don't know when they should pad and/or apodize an image it is probably better to pad with the values in the outermost pixels of the image in each direction. If the image has a bias, zero padding will cause a discontinuity that wasn't there in the input. And if the image has a gradient, you need to pad with different values left and right (or up and down, depending on the direction of the

gradient).

In my case, I was working with simulated images of the solar limb, taking care to "pad" the image myself in the sense that I had enough empty space on one side and enough solar disk on the other, for the wrap-around not to influence the area next to the limb that I'm interested in. So no biggie, I was just surprised that I didn't get symmetric artifacts.

> P.S. Is it possible that you are using a newer version of CONVOLVE, and not the version on the Web page?

Probably, I just thought the one at washington.edu was current. That site often appears near the top when I google for idl programs. Not that that proves anything...

Thanks!

Subject: Re: convolve mystery
Posted by [David Fanning](#) on Wed, 06 Nov 2013 20:20:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mats Löfdahl writes:

> Probably, I just thought the one at washington.edu was current. That
> site often appears near the top when I google for idl programs. Not that
> that proves anything...

That page was last updated in 1999. Things have, uh, changed in the intervening time. Although, in my experience, astronomers are often the last group to get the memo. Something about distribution in reverse alphabetical order, I'm told. ;-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: convolve mystery

Posted by on Wed, 06 Nov 2013 21:22:53 GMT

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

On 2013-11-06 21:20, David Fanning wrote:

> Mats Löfdahl writes:

>

>> Probably, I just thought the one at washington.edu was current. That

>> site often appears near the top when I google for idl programs. Not that

>> that proves anything...

>

> That page was last updated in 1999. Things have, uh, changed in the

> intervening time. Although, in my experience, astronomers are often the

> last group to get the memo. Something about distribution in reverse

> alphabetical order, I'm told. ;-)

With our time scales, who cares about a decade or two?
