Subject: negative return values after FFT
Posted by adisn123 on Wed, 26 Jul 2006 21:01:57 GMT

View Forum Message <> Reply to Message

Hi,

I did FFT from spacial domain to frequency domain on an image of about
500 x 500 pixel size.

IDL> ft = FFT(image, -1)

After filtering job, it was inversly fourier tranformed back using
IDL> inverse = FFT(ft, 1)

When I printed "inverse", the values were complex numbers.

1. Aren't they supposed to real numbers since I tranformed back to
spcial domain?

When I only get real numbers, using
IDL> print, float(FFT(ft,1))
There were some negative values in the array (quite a lot).

For my understanding, the inversely fourier tranformed values should
represent the pixel values corresponding to individual pixel
coordinates in 500 x 500 size.
How do I interpret those negative pixel values?

Thanks.

Subject: Re: negative return values after FFT
Posted by adisn123 on Thu, 27 Jul 2006 19:53:00 GMT

View Forum Message <> Reply to Message

The returned (inversely fourier transformed) values are in a complex
number format, but
I realized that those imaginary parts are very small, almost close to
zero with ~10^-8 floating
point.

My array goes such as the following
h(-f) = (h(f))* after fourier transforming from spicial to frequency
domain.
So, the inversely FFT seems giving real values with almost zero values
of imaginary part since
when I plot it either only with real values or the whole values
including imaginary, those

looked the same.

I have another question related to the returned values.

How do I interpret the "negative" spacial pixel values after inverse
FFT?

kuyper@wizard.net wrote:
> edward.s.meinel@aero.org wrote:
>> FFT(*, *) can take REAL input and return a COMPLEX result; however, a
>> COMPLEX input always returns a COMPLEX result. To get a REAL result you
>> need to do:
>>
>> inverse = REAL(ABS(FFT(ft, 1)))
>>
>> Ed
>>
>> PS. The one-line solution: inverse =
>> REAL(ABS(FFT(FILTERING_JOB(FFT(image, -1)), 1)))
>
> OK - that's a different way of interpreting the message. I was
> assuming, when he said that result was complex, that he wasn't
> referring to the data type of the result, but to it's value: in other
> words, that he was saying that the imaginary parts of the resulting
> array had significantly non-zero magnitudes. With real-valued images,
> and a properly defined filter, that shouldn't happen.
>
> To the original poster (Google shortens your e-mail address to
> 'adisn...@yahoo.com', so I have no idea what I should call you):
> Are you merely saying that the data type of the result was complex, or
> are you making the stronger statement that the values in that result
> had signficantly non-zero imaginary components?

Subject: Re: negative return values after FFT
Posted by James Kuyper on Fri, 28 Jul 2006 12:15:19 GMT

View Forum Message <> Reply to Message

adisn123@yahoo.com wrote:
> The returned (inversely fourier transformed) values are in a complex
> number format, but
> I realized that those imaginary parts are very small, almost close to
> zero with ~10^-8 floating
> point.

OK, that's a normal consequence of the fact that all floating point
mathematics have a certain inherent inaccuracy. Values that

mathematically should be exactly 0 come out numerically as "almost" 0;
it's unfortunately unavoidable. In that case extracting the real
component and ignoring the imaginary components is the appropriate
solution.

...
> I have another question related to the returned values.
>
> How do I interpret the "negative" spacial pixel values after inverse
> FFT?

If your unfiltered image frequently goes close to zero, filtiering it
is likely to cause it to sometimes go negative. That's because each
component in the frequency domain represents a function in the spatial
domain that oscilates between positive and negative values. No matter
how you change the value of a frequency component, either by increasing
it or by decreasing it, you'll be increasing the image in some
locations, and decreasing it somewhere else. If you're unlucky enough,
the places where it decreases the image brightness might be places
where the brightness is already so low that the changes made by the
filter make it go negative.

If you're sure your filter implements what you want it to implement,
I'd recommend treating the negative pixels as zeros. However, if you
ever decide to rebin the data to a lower resolution, use the original
values, including the negatives - don't replace the negative values
with zeros until after re-binning, because otherwise you'll be creating
a systematic bias, making the darkest parts of your image slightly
brighter than they should be.

Subject: Re: negative return values after FFT
Posted by adisn123 on Fri, 28 Jul 2006 19:52:37 GMT

View Forum Message <> Reply to Message

Thanks. It makes sense more now.

kuyper@wizard.net wrote:
> adisn123@yahoo.com wrote:
>> The returned (inversely fourier transformed) values are in a complex
>> number format, but
>> I realized that those imaginary parts are very small, almost close to
>> zero with ~10^-8 floating
>> point.
>
> OK, that's a normal consequence of the fact that all floating point
> mathematics have a certain inherent inaccuracy. Values that

> mathematically should be exactly 0 come out numerically as "almost" 0;
> it's unfortunately unavoidable. In that case extracting the real
> component and ignoring the imaginary components is the appropriate
> solution.
>
> ...
>> I have another question related to the returned values.
>>
>> How do I interpret the "negative" spacial pixel values after inverse
>> FFT?
>
> If your unfiltered image frequently goes close to zero, filtiering it
> is likely to cause it to sometimes go negative. That's because each
> component in the frequency domain represents a function in the spatial
> domain that oscilates between positive and negative values. No matter
> how you change the value of a frequency component, either by increasing
> it or by decreasing it, you'll be increasing the image in some
> locations, and decreasing it somewhere else. If you're unlucky enough,
> the places where it decreases the image brightness might be places
> where the brightness is already so low that the changes made by the
> filter make it go negative.
>
> If you're sure your filter implements what you want it to implement,
> I'd recommend treating the negative pixels as zeros. However, if you
> ever decide to rebin the data to a lower resolution, use the original
> values, including the negatives - don't replace the negative values
> with zeros until after re-binning, because otherwise you'll be creating
> a systematic bias, making the darkest parts of your image slightly
> brighter than they should be.

Subject: Re: negative return values after FFT
Posted by news.qwest.net on Fri, 28 Jul 2006 22:46:03 GMT

View Forum Message <> Reply to Message

<adisn123@yahoo.com> wrote in message
news:1154029980.397254.232310@h48g2000cwc.googlegroups.com.. .

> How do I interpret the "negative" spacial pixel values after inverse
> FFT?

Did you filter away the DC level? How does the mean of the
original compare with the mean of the filtered image.

What type of filter is it (low pass? high pass?).
I would be suprised to see a low pass filter extend the span of the data.
In fact, I would be surprised to see any filter extend the span of the data
by any appreciable amount.

Cheers,
bob

PS it seems like you solved the problem with "complex" values. I would have mentioned to be sure your filter is "symmetric" about the origin in 2d fourier
space, in order to ensure a real valued result.