
Subject: Re: FFT+inverse FFT

Posted by [pgrigis](#) on Mon, 06 Dec 2010 15:28:17 GMT

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

On Dec 5, 3:05 pm, burton449 <burton...@gmail.com> wrote:

> On Dec 5, 2:06 pm, David Fanning <n...@dfanning.com> wrote:

>

>

>

>> Natalya Lyskova writes:

>>> Hey! I'm a beginner at IDL and have problem with FFT. I'm trying to

>>> perform 2d-FFT but the code doesn't work properly even on test images.

>>> So I create an image, make the Fourier transform, then the inverse

>>> Fourier transform and finally I expect to get the initial image. But

>>> the resulting image is the initial one, reversed with respect to the

>>> center.

>

>>> My code:

>>> nx=256L

>>> x1=findgen(nx)-nx/2.0+1.0

>>> x2=findgen(nx)-nx/2.0+1.0

>

>>> ytest=fltarr(nx,nx)

>>> for i=0l,nx-1 do begin

>>> for j=0l,nx-1 do begin

>>> if (x1[i] le 20.0 and x1[i] ge 0.0 and x2[j] le 20.0 and x2[j] ge

>>> 0.0) then begin

>>> ytest[i,j]=1.0

>>> endif

>>> endfor

>>> endfor

>

>>> ; So the initial image is a square in the right upper corner

>

>>> FFTtest=FFT(ytest)

>>> sh_FFTtest=SHIFT(FFTtest,nx/2.0-1.0,nx/2.0-1.0)

>>> inv_test=FFT(FFTtest,-1)

>

>>> ;The result is the square in the LEFT LOWER corner.

>

>>> I would be very grateful for comments/advice/solutions

>

>> I think you need to read the on-line help for the FFT function. :-)

>

>> Your code should look like this:

>

>> FFTtest = FFT(ytest, -1)

>> inv_test = Real_Part(FFT(FFTtest, 1))

>
>> Now ytest and inv_test are essentially the same.
>
>> Cheers,
>
>> David
>
>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.
>> Coyote's Guide to IDL Programming:<http://www.dfanning.com/>
>> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
>
> Hello,
> Im working with the FFT also but not directly in IDL, I work in ENVI.
> I wonder how ENVI or IDL can perform a FFT on a rectangular image?
> Theorically the image must be a square of dimension of coefficient of
> 2. (512x512, 1024 x1024 etc...) So what is the process that make
> possible to do a FFT on a rectangular image? Other Image Processing
> software like PCI Geomatica cant do that.

Taking a DFT (Discrete Fourier Transformation) of an array is possible for any array size. There is an algorithm called FFT (Fast Fourier Transformation)

that happens to be very efficient if the size is in the form 2^N for some N.

However, modern incarnations of FFT can deal with other sizes too, albeit less

efficiently (the smaller the factors in the prime decomposition of the size, the better).

Please don't let what a particular software does or fail to do be your guide to what is possible or not (from a mathematical standpoint). If you

want to learn a bit more about the FFT, read for instance the chapter about it on the numerical recipes book.

Ciao,
Paolo

>
> Thank you,
> Max
