
Subject: Re: IDL 5.5, 2D FFT indexing confusion.

Posted by [Benjamin Hornberger](#) on Tue, 19 Jul 2005 14:29:46 GMT

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

Pitufa wrote:

> Hi,
>
> I have been trying to generate an real even function in fourier space
> that I can INVERSE FFT in order to get a function which has no
> imaginary part. I have no problems when the function is a vector, but I
> get an imaginary part when it is a two dimensional array.
>
> Below is the test program that shows my problem. Am I defining wrong
> the variables 'centre' or 'nshift'?
>
> I would be really grateful if someone could let me know what I am doing
> wrong.
>
> Thanks!
>
> Pitufa.

I believe your TEST array is not even in the sense which is required for the FT to be real. I think it has to be even (symmetric) in each dimension separately, and not point-symmetric about the center. Please correct me if I'm wrong. E.g., the following TEST array gives a real FT:

```
test = sqrt((rebin(f, 100, 100))^2+$  
(rebin(reform(f, 1, 100), 100, 100))^2)
```

which is basically the same as a shifted DIST array.

With respect to centers and shifts, I prefer to shift everything by $n/2$ when going from real to Fourier space and $-n/2$ when going back:

```
fourier = shift(fft(real), n/2)  
real = fft(shift(fourier, -n/2), /inverse)
```

In this case, the zero frequency is right in the center for an odd number of pixels and just right/above the center for an even number of pixels, and the frequency array goes from -Nyquist to +(1 index below Nyquist). For even number of pixels, you don't even have to worry about shifting plus or minus any more.

As a side note, you can easily vectorize the calculation of your PHI array:

```
nx2d = rebin(indgen(nx)-cx, nx, ny)
```

```
ny2d = rebin(reform(indgen(ny)-cy, 1, ny), nx, ny)
phi = atan(ny2d, nx2d)
```

where nx, ny are the numbers of pixels and cx, cy the centers.

Benjamin

```
>
> pro test_index
>
> npix = 100
> centre = npix/2.d - 1.d
> nshift = npix/2 + 1
>
> ;1d example:
>
> f = abs(findgen(npix) - centre)
> ifft = fft(shift(f,nshift),1,/double)
>
> print,'1D: Imaginary maximum:', Max(Abs(Imaginary(ifft)))
> print,'1D: Real maximum   : ', Max(Abs(Double(ifft)))
>
> ;2d example:
>
> ;angle of position vector w.r.t x axis:
>
> phi = dblarr(NPIX,NPIX)
> FOR X = 0, NPIX-1 DO FOR Y = 0, NPIX-1 DO $
> PHI[X,Y] = ATAN(Y*1.D - centre,X*1.D - centre)
>
> TEST = sin(2.d*phi)
> IFFTTEST = FFT(SHIFT(TEST,nshift,nshift),1,/DOUBLE)
>
> print,'2D: Imaginary maximum: ', $
> MAX(ABS(imaginary(IFFTTEST))),mean(ABS(imaginary(IFFTTEST)))
> print,'2D: Real maximum   : ',MAX(ABS(double(IFFTTEST)))
>
> end
>
```
