Subject: Re: IDL FFT vs MATLAB FFT
Posted by Paul Van Delst[1] on Tue, 08 Oct 2002 17:56:56 GMT
View Forum Message <> Reply to Message

Randall Skelton wrote:
>
> Argh.... it is a bad day when I need to sort out both Matlab's and IDL's
> FFT functions...

Yes, but once you do you'll sleep better.... :o)

> Assuming that, IDL's FFT function uses a one-sided format, and divides
> by N on the forward transform and MATLAB's FFT function uses a
> one-sided format and divides by N on the inverse transform, I still am
> having difficulty comparing the FFT results from each.
>
> In Matlab:
>
>>> fft(eye(4))
>
> ans =
> 1.0000 1.0000 1.0000 1.0000
> 1.0000 0 - 1.0000i -1.0000 0 + 1.0000i
> 1.0000 -1.0000 1.0000 -1.0000
> 1.0000 0 + 1.0000i -1.0000 0 - 1.0000i
>
> IDL> print, 4*fft(identity(4),/double)
> (1.0000000,0.0000000) (0.0000000, 0.0000000) (0.0000000,0.0000000) (0.0000000,
0.0000000)
> (0.0000000,0.0000000) (0.0000000,-3.0628711e-17) (0.0000000,0.0000000)
(1.0000000,3.0628711e-17)
> (0.0000000,0.0000000) (0.0000000, 0.0000000) (1.0000000,0.0000000) (0.0000000,
0.0000000)
> (0.0000000,0.0000000) (1.0000000,-3.0628711e-17) (0.0000000,0.0000000)
(0.0000000,3.0628711e-17)
>
> Can someone please explain how and why these are different?

Weird. Here's wot I did:

```
IDL> x=identity(4)
IDL> print, x
     1.00000      0.00000      0.00000      0.00000
     0.00000      1.00000      0.00000      0.00000
     0.00000      0.00000      1.00000      0.00000
     0.00000      0.00000      0.00000      1.00000
IDL> xi=complexarr(4,4)
IDL> for i=0,3 do xi[*,i]=fft(x[*,i],/double)
```

IDL> print, 4.0*xi
(1.00000, 0.00000)(1.00000, 0.00000)(1.00000, 0.00000)(1.00000, 0.00000)
(1.00000, 0.00000)(-6.12574e-17, -1.00000)(-1.00000, -0.00000)(-6.12574e-17, 1.00000)
(1.00000, 0.00000)(-1.00000, -0.00000)(1.00000, 0.00000)(-1.00000, 0.00000)
(1.00000, 0.00000)(6.12574e-17, 1.00000)(-1.00000, -0.00000)(6.12574e-17, -1.00000)

Looks the same as the matlab one now (assuming 6.12574e-17 == 0.0). There are no words in the IDL fft docs about what is done when the input array is not a vector.

I think the matlab result is the correct one (based on my image processing class from 1987...ehem.)

cheers,

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7274
Fax:(301)763-8545

Subject: Re: IDL FFT vs MATLAB FFT
Posted by Don J Lindler on Tue, 08 Oct 2002 18:01:29 GMT
View Forum Message <> Reply to Message

> Argh.... it is a bad day when I need to sort out both Matlab's and IDL's
> FFT functions...
>
> Assuming that, IDL's FFT function uses a one-sided format, and divides
> by N on the forward transform and MATLAB's FFT function uses a
> one-sided format and divides by N on the inverse transform, I still am
> having difficulty comparing the FFT results from each.
>
> In Matlab:
>
>>> fft(eye(4))
>
> ans =
> 1.0000 1.0000 1.0000 1.0000
> 1.0000 0 - 1.0000i -1.0000 0 + 1.0000i
> 1.0000 -1.0000 1.0000 -1.0000

>   1.0000   0 + 1.0000i -1.0000   0 - 1.0000i
>
>   IDL> print, 4*fft(identity(4),/double)
>   (1.0000000,0.0000000) (0.0000000,   0.0000000) (0.0000000,0.0000000)
(0.0000000,   0.0000000)
>   (0.0000000,0.0000000) (0.0000000,-3.0628711e-17) (0.0000000,0.0000000)
(1.0000000,3.0628711e-17)
>   (0.0000000,0.0000000) (0.0000000,   0.0000000) (1.0000000,0.0000000)
(0.0000000,   0.0000000)
>   (0.0000000,0.0000000) (1.0000000,-3.0628711e-17) (0.0000000,0.0000000)
(0.0000000,3.0628711e-17)
>
> Can someone please explain how and why these are different?
>

fft in matlab only does a 1-D transform. Use fft2.

Don
