Discrete Fourier Transform

Bin Fu
fubin1991@outlook.com
Rm 418, CYM Physics BLG

Course website: http://www.physics.hku.hk/~phys4150/

Discrete Fourier Transform

A signal has this form:

$$f(x) = 0.5\sin(30\pi x) + 2\sin(80\pi x)$$

Compute the Fourier transform of this function in domain $[0 - 4\pi]$ and compare your result with the internal function (FFT).

Discrete Fourier Transform

```
% Discrete Fourier Transform
N=1023;
x=linspace(0,4*pi,N+1);
fx=0.5*sin(2*pi*15*x)+2*sin(2*pi*40*x);
a=exp((2*pi*1i)/(N+1));

A0=zeros(1,N+1); A=zeros(N+1,N+1);
for ii=1:N+1;
    A0(1,ii)=a^(ii-1);
end

for ii=1:N+1
    A(ii,:)=A0.^(ii-1);
end
gx=inv(A)*(fx.');

gxx=(1/(N+1))*fft(fx); % the defination are different
% gx=abs(gx);
% gxx=abs(gxx);
figure
plot(x,fx)
figure
plot(x,gx)
hold on
plot(x,gxx)
hold off
```

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^N \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \cdots & \alpha^{2N} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^N & \alpha^{2N} & \alpha^{3N} & \cdots & \alpha^{NN} \end{pmatrix}$$

The functions `Y = fft(x)` and `y = ifft(X)` implement the transform and inverse transform pair given for vectors of length N by:

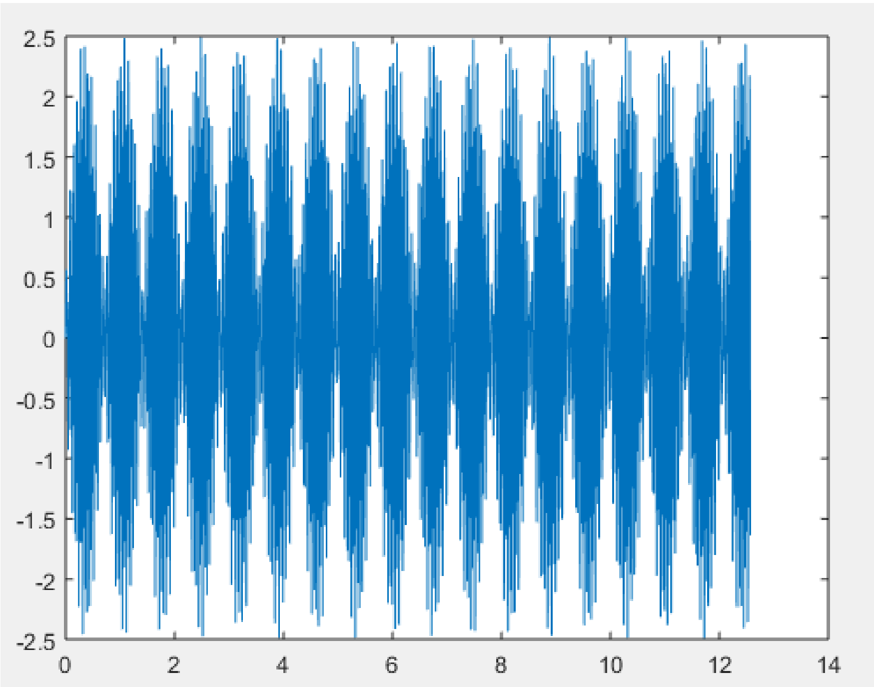
$$X(k) = \sum_{j=1}^{N} x(j)\omega_N^{(j-1)(k-1)}$$

$$x(j) = (1/N)\sum_{k=1}^{N} X(k)\omega_N^{-(j-1)(k-1)}$$

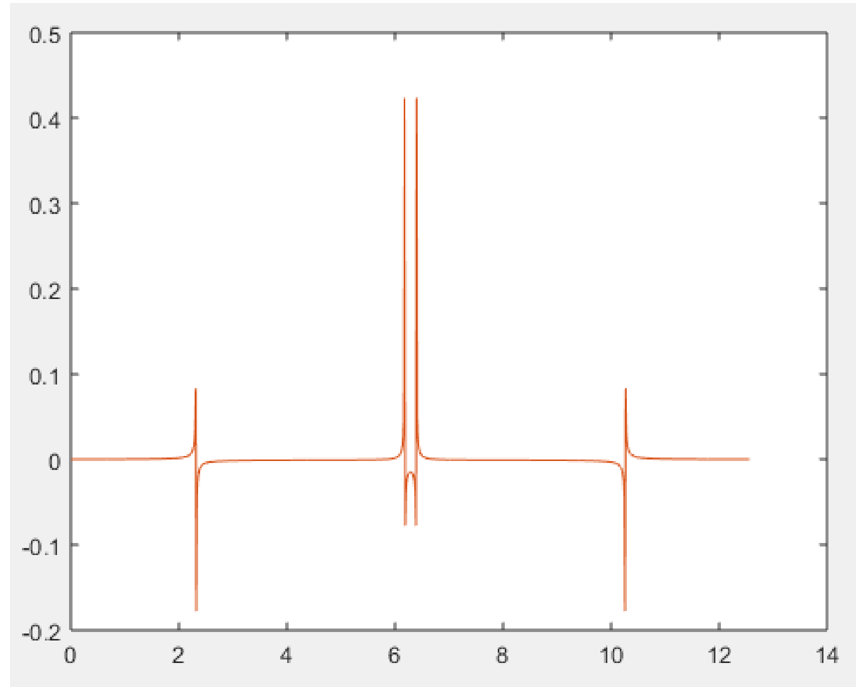
where

$$\omega_N = e^{(-2\pi i)/N}$$

Discrete Fourier Transform

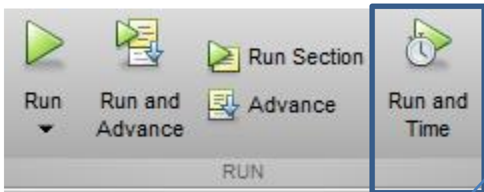


F(x)



G(x)

Discrete Fourier Transform



Profile Summary

Generated 23-Feb-2017 19:00:08 using cpu time.

Function Name	Calls	Total Time	Self Tir
DFT	1	19.110 s	19.087
newplot	3	0.017 s	0.010 s
cla	2	0.006 s	0.001 s
newplot>ObserveAxesNextPlot	3	0.006 s	0.000 s
hold	2	0.005 s	0.003 s
graphics\private\clo	2	0.004 s	0.004 s
axescheck	2	0.002 s	0.002 s
linspace	1	0.001 s	0.001 s
newplot>ObserveFigureNextPlot	3	0.001 s	0.001 s
graphics\private\claNotify	2	0.001 s	0.001 s
ishold	3	0 s	0.000 s

Function listing

Color highlight code according to time

time	calls	line	
	1	2	N=4095;
< 0.01	1	3	x=linspace(0,4*pi,N+1);
< 0.01	1	4	fx=0.5*sin(2*pi*15*x)+2*sin(2*pi*40*x);
	1	5	a=exp((2*pi*1i)/(N+1));
		6	
0.06	1	7	A0=zeros(1,N+1); A=zeros(N+1,N+1);
	1	8	for ii=1:N+1;
< 0.01	4096	9	A0(1,ii)=a^(ii-1);
	4096	10	end
		11	
	1	12	for ii=1:N+1
3.77	4096	13	A(ii,:)=A0.^(ii-1);
< 0.01	4096	14	end
15.15	1	15	gx=inv(A)*(fx.');
		16	
< 0.01	1	17	gxx=(1/(N+1))*fft(fx); % the defination are different
		18	% gx=abs(gx);
		19	% gxx=abs(gxx);
0.05	1	20	figure
0.01	1	21	plot(x,fx)
0.05	1	22	figure
< 0.01	1	23	plot(x,gx)
< 0.01	1	24	hold on
< 0.01	1	25	plot(x,gxx)
< 0.01	1	26	hold off

```
K>> tic; inv(A); toc
Elapsed time is 15.028603 seconds.
```

Discrete Fourier Transform

cumprod

Cumulative product

Syntax

```
B = cumprod(A)
B = cumprod(A,dim)
B = cumprod(___,direction)
```

Description

`B = cumprod(A)` returns the cumulative product of `A` starting at the beginning of the first array dimension in `A` whose size does not equal 1.

- If `A` is a vector, then `cumprod(A)` returns a vector containing the cumulative product of the elements of `A`.
- If `A` is a matrix, then `cumprod(A)` returns a matrix containing the cumulative products for each column of `A`.
- If `A` is a multidimensional array, then `cumprod(A)` acts along the first nonsingleton dimension.

`B = cumprod(A,dim)` returns the cumulative product along dimension `dim`. For example, if `A` is a matrix, then `cumprod(A,2)` returns the cumulative product of each row.

Discrete Fourier Transform

```
for ii=1:N+1
    A(ii,:)=A0.^(ii-1);
end
```

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^N \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \cdots & \alpha^{2N} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^N & \alpha^{2N} & \alpha^{3N} & \cdots & \alpha^{NN} \end{pmatrix}$$

$$\boldsymbol{B} = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^N \\ 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^N \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^N \end{pmatrix} \xrightarrow{\textbf{cumprod(B,1)}} A = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^N \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \cdots & \alpha^{2N} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^N & \alpha^{2N} & \alpha^{3N} & \cdots & \alpha^{NN} \end{pmatrix}$$

Discrete Fourier Transform

$$B = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^N \\ 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^N \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^N \end{pmatrix} \xrightarrow{\text{cumprod(B,1)}} A = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^N \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \cdots & \alpha^{2N} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^N & \alpha^{2N} & \alpha^{3N} & \cdots & \alpha^{NN} \end{pmatrix}$$

```
tic
B=ones(N+1,1)*A0;
B(1,:)=ones(1,N+1);
A1=cumprod(B,1);
toc
```

```
tic
for ii=1:N+1
    A2(ii,:)=A0.^(ii-1);
end
toc
```

```
max(max(abs(A1-A2)))
```

```
>> DFT
Elapsed time is 0.253981 seconds.
Elapsed time is 4.156657 seconds.

ans =

   3.0987e-13
```

THANKS FOR ATTENTION !