MA2507 Computing Mathematics Laboratory: Week 9

1. Submit PDF files produced by `publish` as attachments to Canvas. (the PDF should include the source codes and the figures)

2. The assignment is due on 5:00pm of March 16 (Wednesday). Late submissions will NOT be marked.

3. Please write down your name and student ID.

Q1 For a function $f(x)$, its Fourier transform can be defined as

$$g(y) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi xy} dx.$$

Use FFT, calculate (approximately) the Fourier transform of

$$f(x) = [\cos(6\pi x) + \sin(7\pi x)] e^{-\pi x^2}.$$

Suggestions: truncate x to $[-L, L)$ for $L = 20$ with $N = 2m = 4096$ points, truncate y to $[-m/(2L), (m-1)/(2L)]$ with also $2m$ points, plot $f(x)$ on $[-2, 2]$ and plot real and imaginary parts of g on $[-7, 7]$.

We need to approximate the integral relation and write it to something like Eq. (5) of lecture note. Define $x_j = jL/m$ for $j = -m, -m+1, ..., 0, 1, ..., m-1$, define $y_k = k/(2L)$ for $k = -m, -m+1, ..., m-1$. Notice that $x_j y_k = jk/(2m) = jk/N$. We can approximate $g(y)$ as

$$g(y_k) = \frac{L}{m} \sum_{j=-m}^{m-1} f(x_j) e^{-i2\pi x_j y_k} = \frac{L}{m} \sum_{j=-m}^{m-1} f(x_j) e^{-i2\pi jk/N}, \quad k = -m, -m+1, ..., m-1.$$

The above is somewhat similar to Eq. (5) of the lecture note, but the ranges for j and k are different. For DFT, we should have indices from 0 to $N-1$. Now, suppose j is negative, we can add N to it and define $j+N$ as the new j. Notice that since $e^{-i2\pi k} = 1$ for any integer k, we have $e^{-i2\pi jk/N} = e^{-i2\pi(j+N)k/N}$. Similarly, for a negative k, we can replace it by $k+N$. This implies that if we can define

$$\tilde{\mathbf{f}} = \begin{bmatrix} \tilde{f}_0 \\ \tilde{f}_1 \\ \tilde{f}_2 \\ \vdots \\ \tilde{f}_{N-1} \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{m-1}) \\ f(x_{-m}) \\ \vdots \\ f(x_{-2}) \\ f(x_{-1}) \end{bmatrix}, \quad \tilde{\mathbf{g}} = \begin{bmatrix} \tilde{g}_0 \\ \tilde{g}_1 \\ \tilde{g}_2 \\ \vdots \\ \tilde{g}_{N-1} \end{bmatrix} = \begin{bmatrix} g(y_0) \\ g(y_1) \\ \vdots \\ g(y_{m-1}) \\ g(y_{-m}) \\ \vdots \\ g(y_{-2}) \\ g(y_{-1}) \end{bmatrix}$$

then

$$\tilde{g}_k = \frac{L}{m} \sum_{j=0}^{N-1} \tilde{f}_j e^{-i2\pi jk/N}, \quad k = 0, 1, ..., N-1.$$

In MATLAB, the program `fft` calculates above without the coefficient L/m. So we just have to multiply L/m. Our program is as follows.

```
% main script for Fourier transform
L = 20;
```

```
m = 2048;
N = 2*m;
x = (L/m)* (-m: m-1);
f = myfn(x);
subplot(1,2,1)
plot(x,f), xlabel('x'), title('f(x)')
xlim([-2 2])      % show plot from x=-2 to x=2
ff = [f(m+1:N), f(1:m)];
gg = fft(ff)*(L/m);
g = [gg(m+1:N), gg(1:m)];
y = (-m : m-1)/(2*L);
subplot(1,2,2)
plot(y, real(g),'b',y, imag(g),'r'), xlabel('y'), title('g(y)')
xlim([-7, 7])

function z = myfn(x);
z = (cos(6*pi*x)+sin(7*pi*x)).* exp(-pi*x.^2);
end
```

The program gives the following figure.