Lab 2: The discrete Fourier transform (DFT)

Problem 1 (DFT implementation):

test the correctness of DFT function by checking that the output coincides
with that of Matlab's fft function for a short test signal (1:4, for instance).

```
Y =

  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 - 0.0000i  -2.0000 - 2.0000i

Elapsed time is 0.000645 seconds.

ans =

  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 + 0.0000i  -2.0000 - 2.0000i

Elapsed time is 0.000707 seconds.
```

DFT implementation against Matlab's fft function for an input signal with a length of 10000 sampling points.

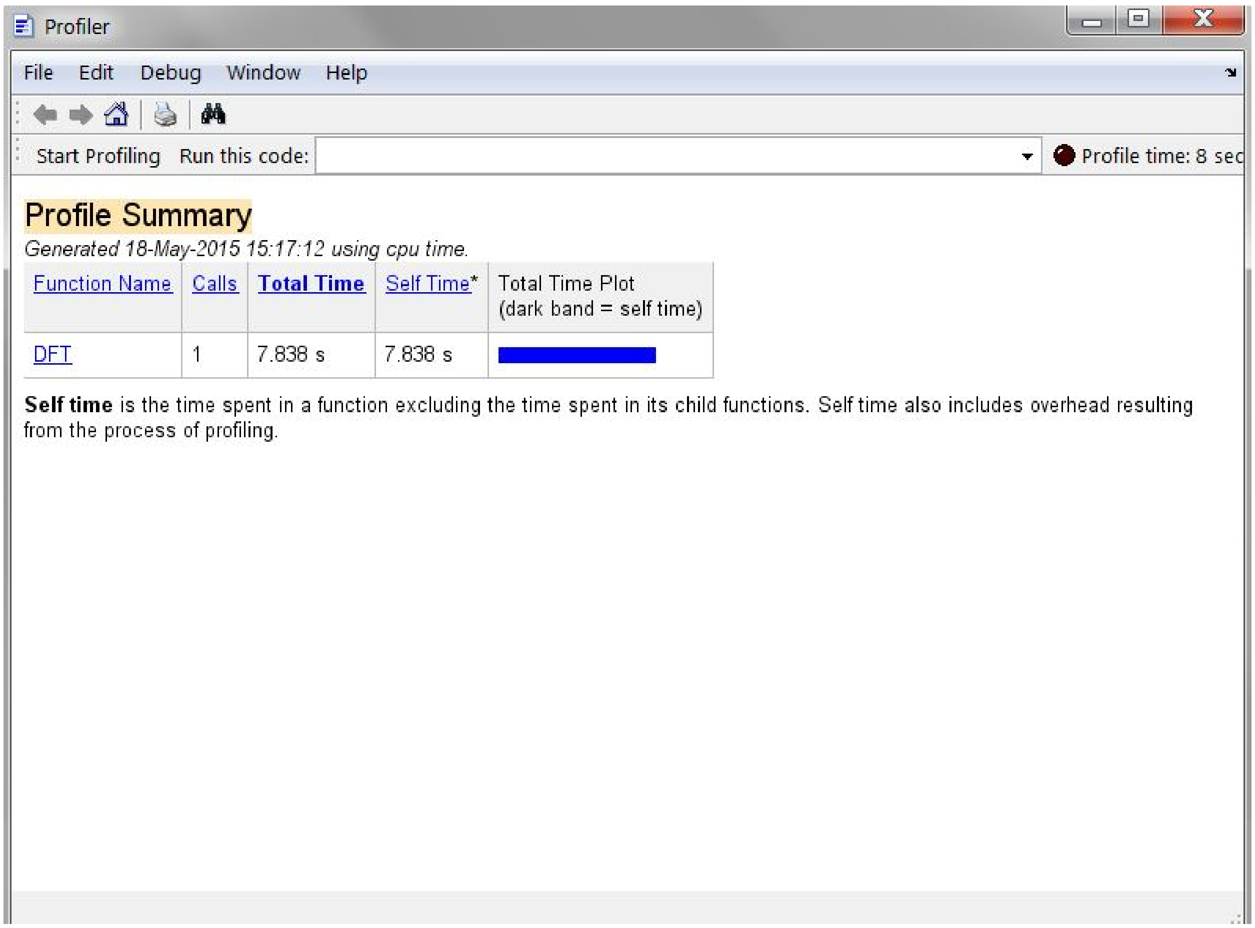
```
%input array
x = 0:1:10000;
tic;
profile on;
Y = DFT(x); %passing thearray to function of DFT
profile viewer
toc;

tic;
fftshift(fft(x));%using matlab built in function to check
%fft(x)
toc;
```

```
Elapsed time is 7.847971 seconds.
Elapsed time is 0.001609 seconds.
```

DFT function

```
function [ X ] = DFT( input )
%step1------------
%making the u1 * k . n
N=length(input);


n=0:N-1;
k=0:N-1;
%k=-N/2:N/2-1%k for negative values

y=n';
% The result is a 2D array such that
u1=y*k;




%step 2 --------------

% the columns vary as k varies and rows vary as n varies
u2=exp(-1i*u1*2*pi/N);

%step 3 -----------------------------


xMat=input'*ones(1,N);
%step4----------

argMat=u2.*xMat;
%step 5 ----------------------
% We take the sum in row direction ( n direction)
X=sum(argMat,1);


end
```

use the profiler by adding the following code
profile on;
... code to be profiled;
profile viewer;



Problem 2 (Spectrum of a rectangular signal):

(a) Theoretical considerations:

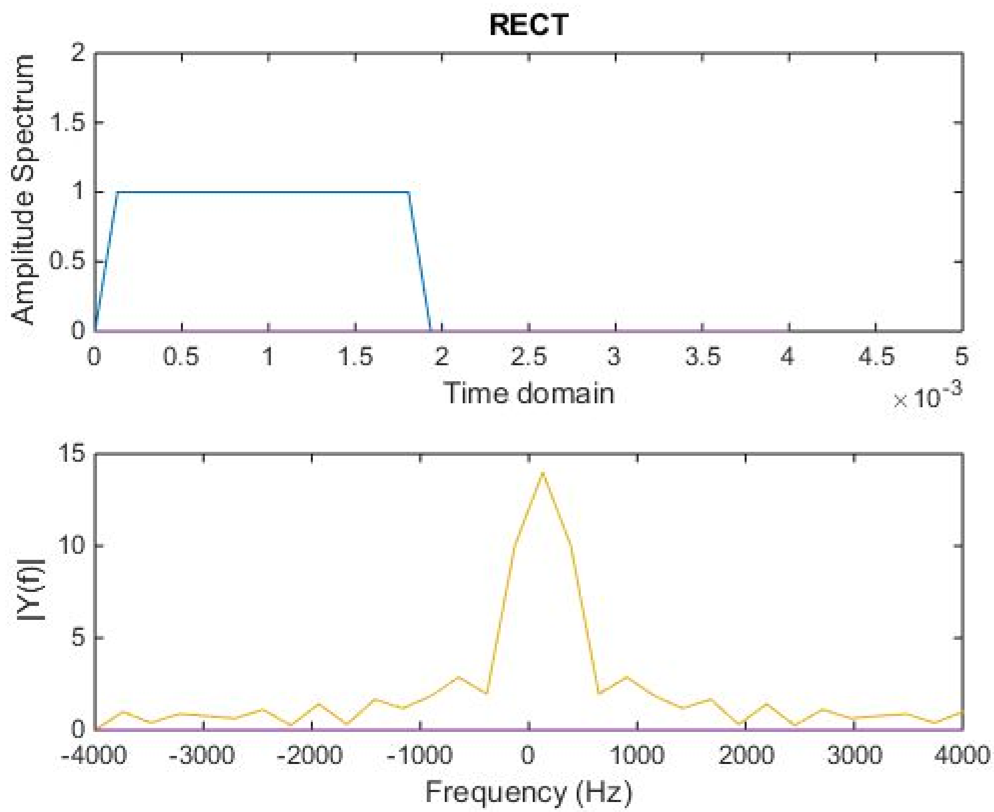
(i) Calculate the Fourier transform of x(t) and sketch the amplitude spectrum (using pen and paper).

2) a) i) it is a rect function shifted by 1 ms, so

$$X(j\omega) = 2\,S_i(\omega) \cdot e^{-j\omega}$$

```
%%Problem 2 (Spectrum of a rectangular signal):
fs = 8000;
To = 0.004;
Ts = 1/fs;
N = 2^nextpow2(To/Ts);% Next power of 2 from length of y[n]


y = zeros(N);
for k = 1:N
    y(k+1) = rect((k+1)*Ts);
end

x = linspace(0,To,N);
subplot(2,1,1);
plot(x,y);
title('RECT');
xlim([0, 5/1000]);
ylim([0,2]);
xlabel('Time domain');
ylabel('Amplitude Spectrum');
%--------------------------------------------------------------------
subplot(2,1,2);
f = fs/2 * linspace(-1,1,N);%generates N points. The spacing between the points is (1-(-1))/(N-1)

toShift =DFT (y(:, 1)');
plot(f , abs(fftshift(toShift)));%using my own DFT function

% toShift = fft(y);
% plot(f , abs(fftshift(toShift)));%using a built in fft function
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')
```

```
function outp = rect( x )

max = 1/500;% 2 ms
    if((x < max) && (x > 0))%between 0 and 2 ms is 1
        outp = 1;
    else
        outp = 0;% other is 0
    end ;
end
```



(iv) How does the spectrum change when the sampling frequency f_s is increased?

by increasing the zeroes (increase fs) to have a better resolution.

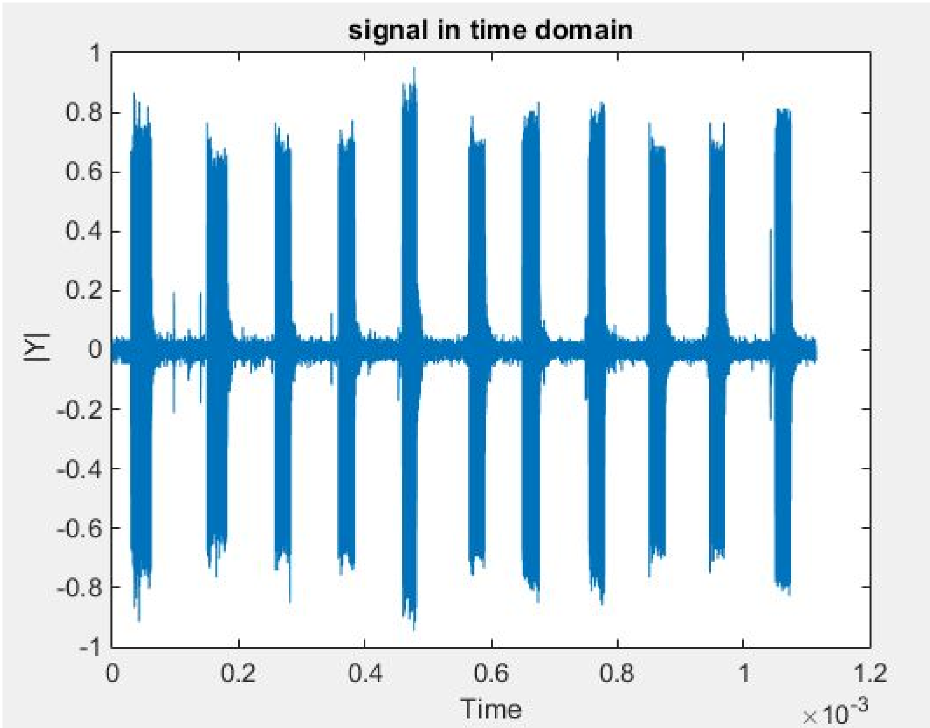
Problem 3 (Dual tones):

(b) Matlab function wavread and use it to load the file touchtone1.wav and Play the signal with the function soundsc.

```
%[y, Fs] = wavread(filename) returns the sample rate (Fs) in Hertz used to encode the data in the file.
[y, Fs] = wavread('touchtone1.wav');
soundsc(y, Fs);%soundsc(y,Fs) sends audio signal y to the speaker at sample rate Fs.
```

(c) Plot the signal in the time domain.

```
Ts = 1/Fs;
N = length(y)-1;
To = N * Ts;
n=0:Ts:To;
plot(n.*Ts, y);
title('signal in time domain');
xlabel('Time');
ylabel('|Y|');
```



(d) Plot the amplitude spectrum over the entire frequency range.

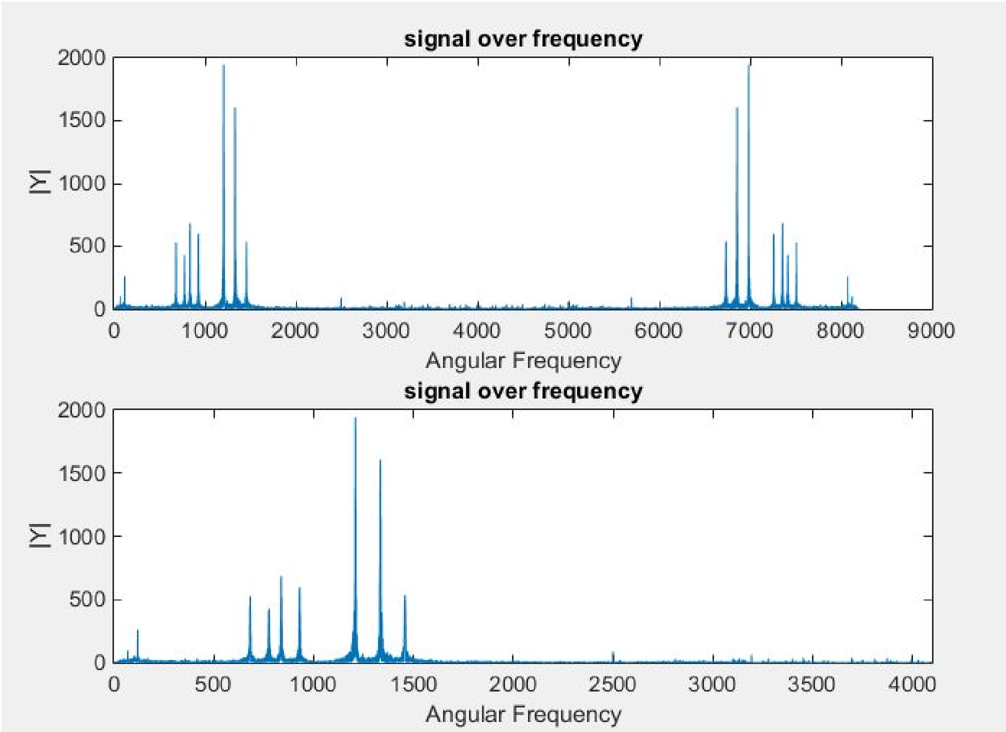
```
f0=1/To;
X=0:f0:Fs;
subplot(2, 1, 1);
plot(X, abs(fft(y)));
title('signal over frequency');
ylabel('|Y|');
xlabel('Angular Frequency');
```

Can you explain the peaks at large frequencies? **Symmetry of the fft around ws/2**

plot where the displayed frequencies are restricted to a range relevant for dual tones

```
subplot(2, 1, 2);

plot(X, abs(fft(y)));
xlim ([0 4096]);
title('signal over frequency');
xlabel('Angular Frequency');
ylabel('|Y|');
```

Do you see the expected frequencies? **Yes.**



(e) Write a Matlab function

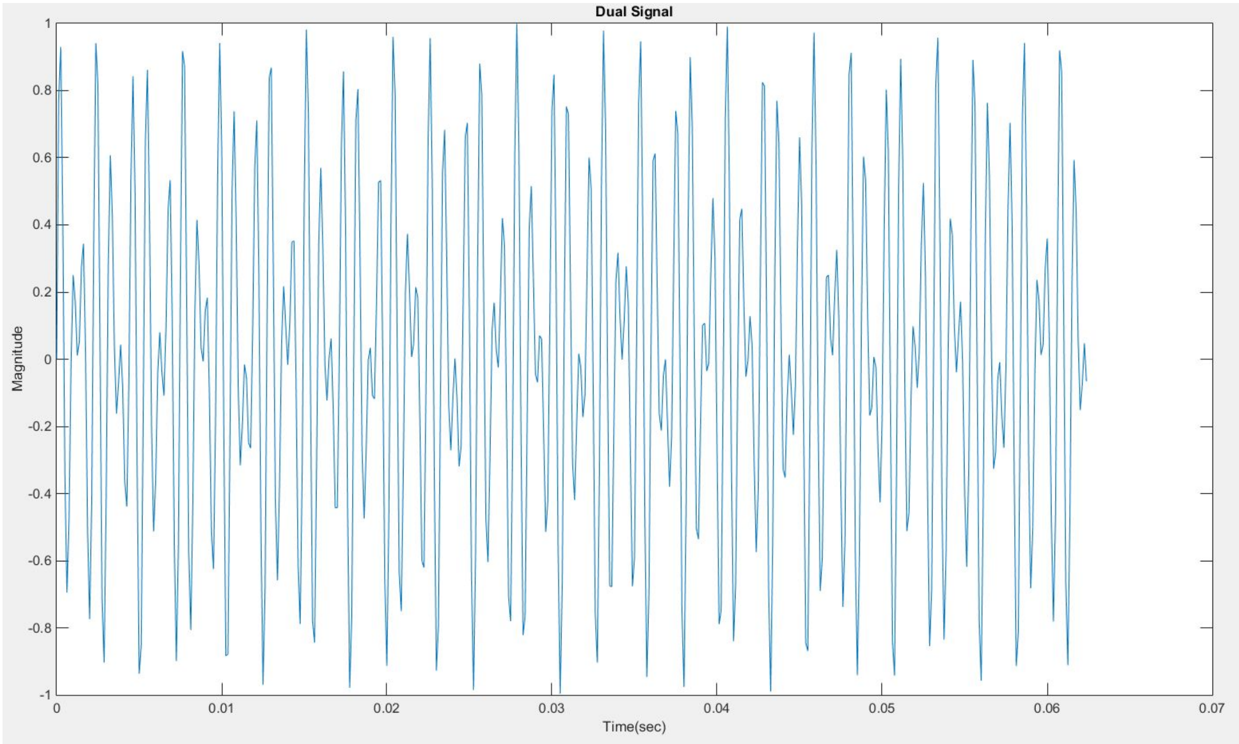
```
function [ tones ] = dial_tones()
Fs=8000;
key = input('Please enter your number:');
key = num2str(key);
num = length(key);
tones = zeros(1,num*Fs);
time = 0: 1/Fs:(num -1/Fs);
for x = 1:num
    tones((1+Fs*(x-1)):(Fs/2 +Fs*(x-1))) = generate_tones(str2num(key(x)));
end
```

```
function  [tone] = generate_tones( key )

Fs = 8000;
t = 0:1/Fs:(1/2-1/Fs);
key=0;
num=length(key);
    for k=1:num;
        value =key(k);
            switch value
                case{1,2,3}
                    row = 697;
                case{4,5,6}
                    row = 770;
                case{7,8,9}
                    row = 852;
                otherwise
                    row = 942;
            end
            switch value
                case{1,4,7}
                    column = 1209;
                case{2,3,8,0}
                    column = 1336;
                otherwise
                    column = 1477;
            end
     tone = 0.5*(sin(2*pi*row*t)+sin(2*pi*column*t));
    sound(tone,Fs);
    end

end
```

plot the signal

```
sound(tones,Fs);
plot(time,tones);
plot(time(1:500),tones(1:500));
title('Dual Signal');
xlabel('Time(sec)');
ylabel('Magnitude');
end
```

for number 45


Dual Signal
Magnitude
Time(sec)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07

.