Alejandra Slowsky
Math 104A

HW 7

1. I am using the fft package in Matlab so the DFT coefficients are defined as: $c_k = \sum_{j=0}^{N-1} f_j e^{-i2\pi kj/N}$ (1). for $j=0, \ldots, N-1$

2. The inverse DFT is computed by the matlab fft package as follows: $f_j = \frac{1}{N}\sum_{k=0}^{N-1} c_k e^{i2\pi kj/N}$ for $j=0,\ldots,N-1$ (5)

3. Assume that y_j is a real number for $j=0,\ldots,N-1$. WTS $c_{N-k} = \bar{c}_k$ & c_0 is real where the bar denotes the conjugate.
We know that $c_0 = \sum_{j=0}^{N-1} y_j$, which is real since it is the sum of real numbers. Note that:

$$c_{N-k} = \sum_{j=0}^{N-1} y_j e^{-i2\pi(N-k)j/N} = \sum_{j=0}^{N-1} y_j (e^{-i2\pi j} \cdot e^{i2\pi jk/N})$$

$$= \sum_{j=0}^{N-1} y_j e^{i2\pi jk/N} = \bar{c}_k \text{ since}$$

$$e^{-i2\pi j} = \cos(2\pi j) - i\sin(2\pi j) = 1 \ \forall j \in \mathbb{Z} \quad \square$$

4. • $$a_k = \frac{2}{N}\sum_{j=0}^{N-1} f_j \cos(kx_j) = \frac{2}{N}\sum_{j=0}^{N-1} f_j\left(\frac{e^{i2\pi kj/N} + e^{-i2\pi kj/N}}{2}\right)$$

$$= \frac{C_k + \overline{C}_k}{N} \quad \text{where } C_k = \sum_{j=0}^{N-1} f_j e^{-i2\pi kj/N},$$

$$\overline{C}_k = \sum_{j=0}^{N-1} f_j e^{i2\pi kj/N}, \ \& \ x_j = \frac{2\pi j}{N} \text{ for } j = 0, \ldots, N-1$$

• $$b_k = \frac{2}{N}\sum_{j=0}^{N-1} f_j \sin(kx_j) = \frac{2}{N}\sum_{j=0}^{N-1} f_j\left(\frac{e^{i2\pi kj/N} - e^{-i2\pi kj/N}}{2i}\right)$$

$$= \frac{\overline{C}_k - C_k}{Ni}$$

```
% HW 7, Math 104A
% Alejandro Stawsky

% Problem #5

% periodic array for interpolating
f = [6, 10.242640687119284, 2, -2.585786437626905, 2, 1.757359312880716, -6, -5.414213562373095
```

```
% C contains the complex coefficients
C = fft(f);
C
```

```
C =
    8.0000 + 0.0000i   8.0000 -16.0000i  12.0000 -20.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
```

```
% A contains the real a_k Coefficients
A= [];
for k=1:(length(f)/2 + 1)
    A = [A,(C(k)+conj(C(k)))./length(f)];
end

A
```

```
A =
    2.0000    2.0000    3.0000    0.0000    0.0000
```

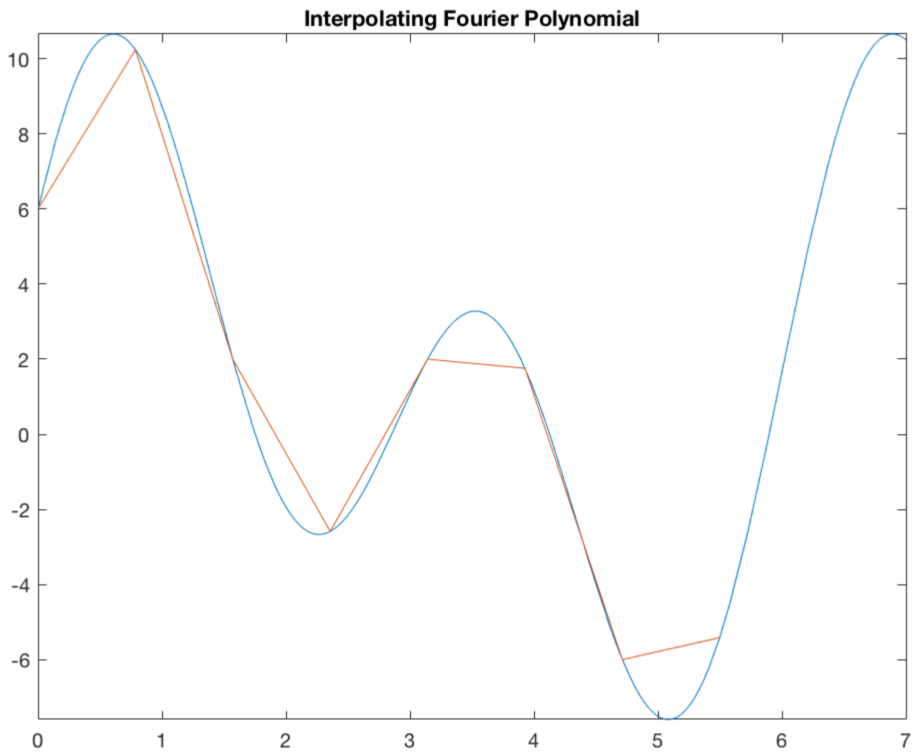
```
% B contains the real b_k Coefficients
B = [];
for k=1:(length(f)/2)
    B = [B,(C(k)-conj(C(k)))./(-1i.*length(f))];
end

B
```

```
B =
         0    4.0000    5.0000         0
```

```
% L(x) is the middle loop inside the interpolating poly
syms L(x)
L(x) = 0;
for i=1:3
    L(x) = L(x) + A(i+1).*cos((i).*x)+B(i+1).*sin((i).*x);
end

% s(x) is the final interpolating fourier polynomial
syms s(x)
s(x) = A(1)./2 + L(x) + (A(4)./2).*cos((length(f)./2).*x);
disp(s(x))
```

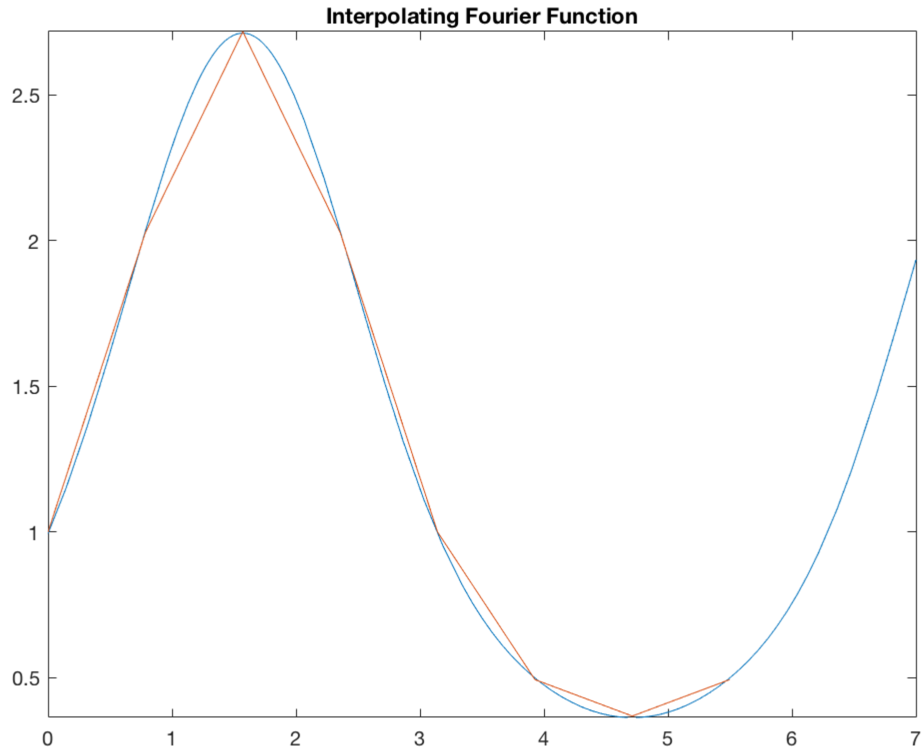
$$3\cos(2x) + \frac{7\cos(3x)}{9007199254740992} + \frac{7\cos(4x)}{18014398509481984} + 5\sin(2x) + 2\cos(x) + 4\sin(x) + 1$$

```
% plotting the interpolating polynomial
fplot(s(x),[0,7])
hold on
plot(X,f)
title('Interpolating Fourier Polynomial')
hold off
```

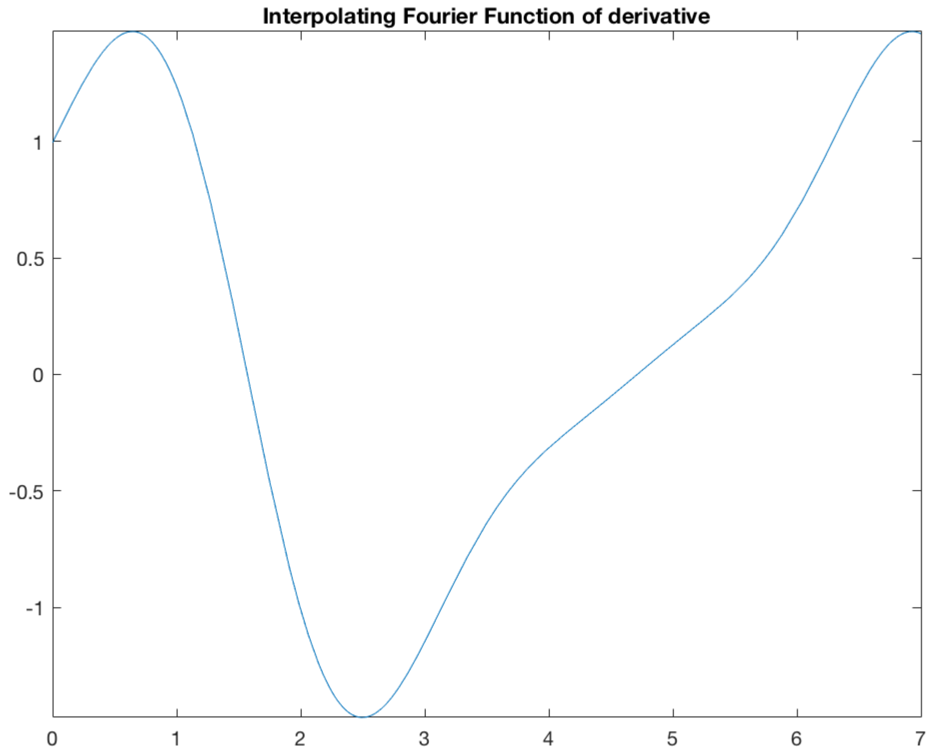



```
% Problem #6

% X is the array of the equidistant nodes
X=[];
for j=0:7
    X=[X,(j.*2.*pi)./8];
end

% G contains the values of the function at the nodes
G = [];
for j=1:8
    G = [G,exp(sin(X(j)))];
end
```

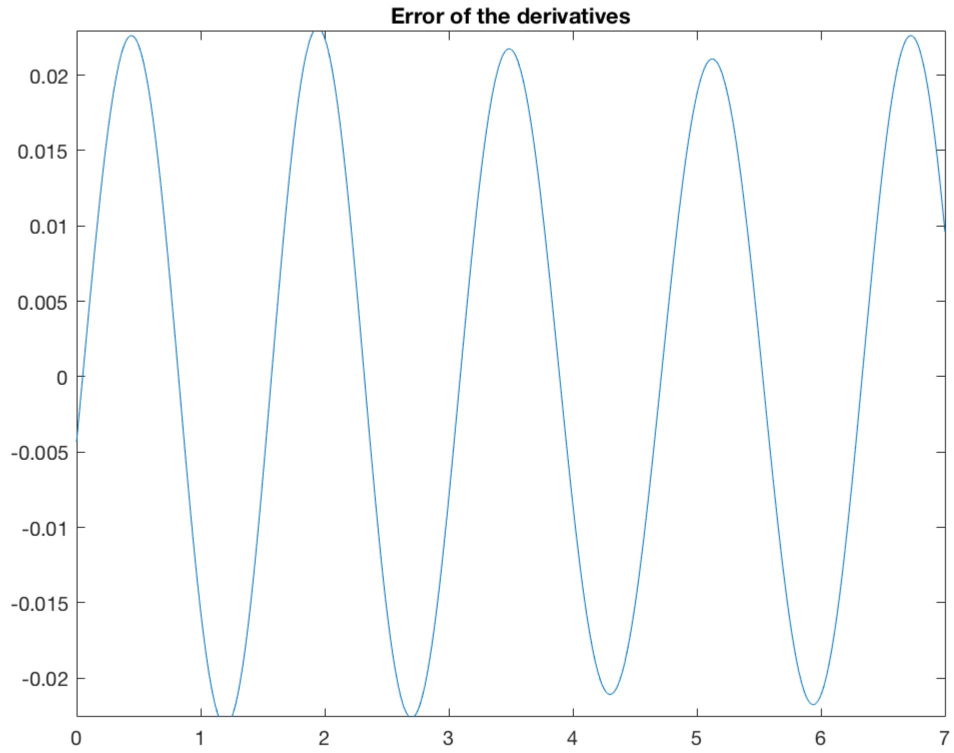
```matlab
% C1 contains the complex coefficients
C1 = fft(G);

% A1 contains all the a_k real coefficients
A1 = [];
for k=1:(length(G)/2 + 1)
    A1 = [A1,(C1(k)+conj(C1(k)))./length(G)];
end

% B1 contains all the b_k real coefficients
B1 = [];
for k=1:(length(G)/2)
    B1 = [B1,(C1(k)-conj(C1(k)))./(-1i.*length(G))];
end

% L(x) is the middle loop inside the interpolating poly
syms L(x)
L(x) = 0;
for i=1:(length(G)/2 - 1)
    L(x) = L(x) + A1(i+1).*cos((i).*x)+B1(i+1).*sin((i).*x);
end

% s(x) is the final interpolating fourier polynomial
syms s(x)
s(x) = A1(1)./2 + L(x) + (A1(length(G)/2)./2).*cos((length(G)./2).*x);

% plotting the Interpolating Fourier Function
figure(1)
fplot(s(x),[0,7])
hold on
plot(X,G)
title('Interpolating Fourier Function')
hold off
```




```
% p(x) is the derivative of interpolating polynomial
syms p(x)
p(x) = diff(s(x));

% plotting the derivative of the interpolating polynomial p(x)
figure(2)
fplot(p(x),[0,7])
title('Interpolating Fourier Function of derivative')
```




```
% g(x) is the function that we are approximating and l(x) is the error
syms g(x) l(x)
g(x) = exp(sin(x));
l(x) = (p(x) - diff(g(x)));

% plotting the error of the derivatives
figure(3)
fplot(l(x),[0,7])
title('Error of the derivatives')
```




```
% Problem #7

% a. Look at complexderivcoeffs

% b.

C4 = complexderivcoeffs(8)
```

```
C4 =
   0.0000 + 0.9841i   2.2576 - 0.6372i   1.0662 - 1.6710i   0.2859 - 1.2294i   0.0857 - 1.0106i  -0.0857 - 1.0106i
```

```

% A1 contains all the a_k real coefficients
A1 = [];
for k=1:5
    A1 = [A1,(C4(k)+conj(C4(k)))./5];
end

% B1 contains all the b_k real coefficients
B1 = [];
for k=1:4
    B1 = [B1,(C4(k)-conj(C4(k)))./(-1i.*5)];
end

% L(x) is the middle loop inside the interpolating poly
```

```
syms L(x)
L(x) = 0;
for i=1:3
    L(x) = L(x) + A1(i+1).*cos((i).*x)+B1(i+1).*sin((i).*x);
end

% s(x) is the final interpolating fourier polynomial
syms r(x)
r(x) = A1(1)./2 + L(x) + (A1(4)./2).*cos(4.*x);

% checking it with the actual values of derivative of function
syms g(x) q(x)
g(x) = exp(sin(x));
q(x) = diff(g(x));

Error8 = q(x) - r(x);
```

```
function F = complexderivcoeffs(N)

% X is the array of the equidistant nodes
X=[];
for j=0:N-1
    X=[X,(j.*2.*pi)./N];
end

syms f(x) g(x)
f(x) = exp(sin(x));
g(x) = diff(exp(sin(x)));

% C contains the complex coefficients
C=[];
for j=1:N
    C = [C,fft(exp(sin(X(j))))];
end

% C1 contains the derivatives of the complex coefficients
C1 = [];

% split the sum into two sums because of indices

for k=(-N/2):0
    C1 = [C1, (1i*k)*C(k+N)];
end
for k=1:N/2
    C1 = [C1, (1i*k)*C(k)];
end

% F contains the approximated value points of the derivative of f(x)
F = ifft(C1);

end
```