**Exercise 1 & 2**

N = 6;

n = 0:N-1;

x\_n = cos(n \* pi / 3);

Ck = DTFS(x\_n, N);

[pow\_xn, pow\_Ck] = parsevals(x\_n, Ck);

disp('DTFS Coefficients:');

disp(Ck);

disp('Power in time domain:');

disp(pow\_xn);

disp('Power in frequency domain:');

disp(pow\_Ck);

subplot(3,1,1);

stem(n, x\_n);

title('x(n)');

grid on;

subplot(3,1,2);

stem(0:N-1, abs(Ck));

title('Magnitude Spectrum |C\_k|');

grid on;

subplot(3,1,3);

stem(0:N-1, abs(Ck).^2);

title('Power Spectrum |C\_k|^2');

grid on;

function Ck = DTFS(x, N)

Ck = zeros(1, N);

for k = 1:N

Ck(k) = (1/N) \* sum(x .\* exp(-1j \* 2 \* pi \* (k-1) \* (0:N-1) / N));

end

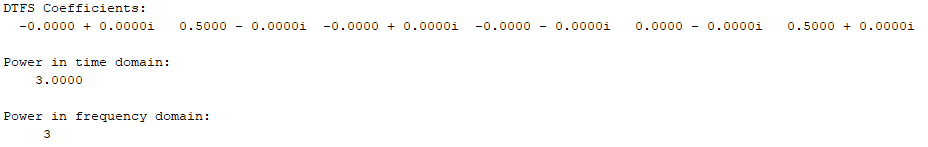
end

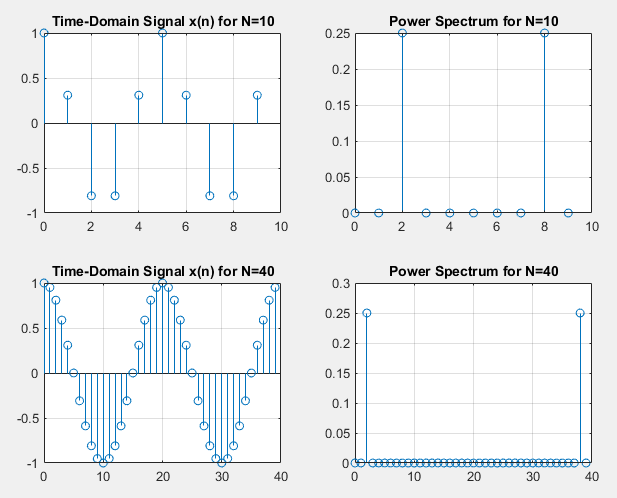
function [pow\_xn, pow\_Ck] = parsevals(xn, Ck)

pow\_xn = sum(abs(xn).^2);

pow\_Ck = length(xn) \* sum(abs(Ck).^2);

end



**Exercise 3**

L = 2;

A = 1;

% Case 1: N = 10

N1 = 10;

n1 = 0:N1-1;

x\_n1 = A \* cos((2 \* pi \* L / N1) \* n1);

% Case 2: N = 40

N2 = 40;

n2 = 0:N2-1;

x\_n2 = A \* cos((2 \* pi \* L / N2) \* n2);

Ck1 = DTFS(x\_n1, N1);

Ck2 = DTFS(x\_n2, N2);

[pow\_xn1, pow\_Ck1] = parsevals(x\_n1, Ck1);

[pow\_xn2, pow\_Ck2] = parsevals(x\_n2, Ck2);

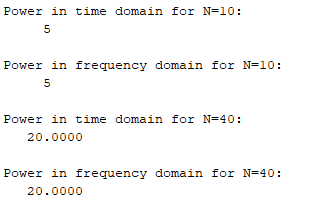
disp('Power in time domain for N=10:'); disp(pow\_xn1);

disp('Power in frequency domain for N=10:'); disp(pow\_Ck1);

disp('Power in time domain for N=40:'); disp(pow\_xn2);

disp('Power in frequency domain for N=40:'); disp(pow\_Ck2);

% Case 1: N = 10

subplot(2,2,1);

stem(n1, x\_n1);

title('Time-Domain Signal x(n) for N=10');

grid on;

subplot(2,2,2);

stem(0:N1-1, abs(Ck1).^2);

title('Power Spectrum for N=10');

grid on;

% Case 2: N = 40

subplot(2,2,3);

stem(n2, x\_n2);

title('Time-Domain Signal x(n) for N=40');

grid on;

subplot(2,2,4);

stem(0:N2-1, abs(Ck2).^2);

title('Power Spectrum for N=40');

grid on;

function Ck = DTFS(x, N)

Ck = zeros(1, N);

for k = 1:N

Ck(k) = (1/N) \* sum(x .\* exp(-1j \* 2 \* pi \* (k-1) \* (0:N-1) / N));

end

end

function [pow\_xn, pow\_Ck] = parsevals(xn, Ck)

pow\_xn = sum(abs(xn).^2);

pow\_Ck = length(xn) \* sum(abs(Ck).^2);

end

**Exercise 4**

function y = IDTFS(Ck, N)

y = zeros(1, N);

for n = 1:N

sum\_val = 0;

for k = 1:N

sum\_val = sum\_val + Ck(k) \* exp(1j \* 2 \* pi \* (k-1) \* (n-1) / N);

end

y(n) = sum\_val;

end

end

**Exercise 5**

N1 = 10; % Case 1

N2 = 40; % Case 2

Ck1 = fftshift(fft([1 0 0 0 0 1 0 0 0 0]) / N1);

Ck2 = fftshift(fft([1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0] / N2));

x\_reconstructed\_1 = IDTFS(Ck1, N1);

x\_reconstructed\_2 = IDTFS(Ck2, N2);

subplot(2,1,1);

stem(0:N1-1, real(x\_reconstructed\_1));

title('Reconstructed x(n) for N=10');

grid on;

subplot(2,1,2);

stem(0:N2-1, real(x\_reconstructed\_2));

title('Reconstructed x(n) for N=40');

grid on;

function y = IDTFS(Ck, N)

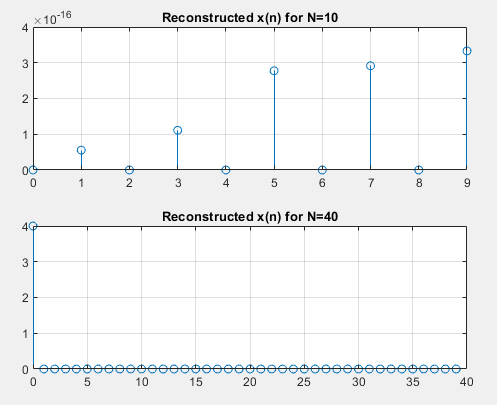
n = 0:N-1;

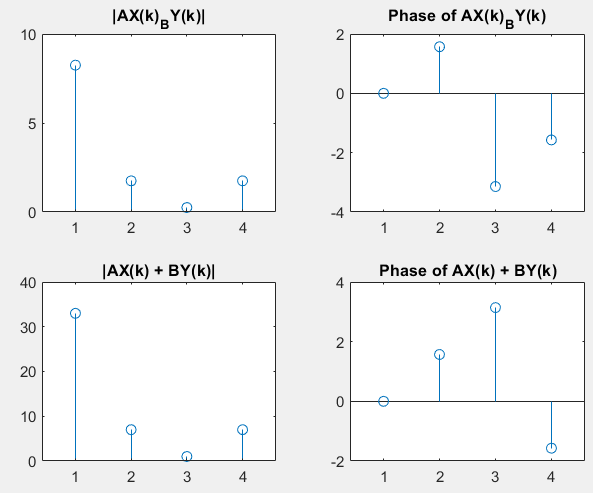
k = (-floor(N/2)):(ceil(N/2)-1);

WN = exp(1j \* 2 \* pi / N);

y = real(sum(Ck .\* (WN .^ (k' \* n)), 1));

end



**Exercise 6**

N = 4;

xn = [1, 1, 1, 0];

yn = [2, 1, 2, 4];

A = 2;

B = 3;

Xk = DTFS(xn, N);

Yk = DTFS1(yn, N);

Axn\_Byn = A \* xn + B \* yn;

Axk\_Byk = fft(Axn\_Byn);

AXk\_BYk = A \* Xk + B \* Yk;

figure;

subplot(2,2,1);

stem(abs(AXk\_BYk));title('|AX(k)\_BY(k)|');

subplot(2,2,2);

stem(angle(AXk\_BYk));title('Phase of AX(k)\_BY(k)');

subplot(2,2,3);

stem(abs(Axk\_Byk));title('|AX(k) + BY(k)|');

subplot(2,2,4);

stem(angle(Axk\_Byk));title('Phase of AX(k) + BY(k)');

function Xk = DTFS(xn, N)

Xk = zeros(1, N);

for k = 1:N

Xk(k) = (1/N) \* sum(xn .\* exp(-1j \* 2 \* pi \* (k-1) \* (0:N-1) / N));

end

end

function Yk = DTFS1(yn, N)

Yk = zeros(1, N);

for k = 1:N

Yk(k) = (1/N) \* sum(yn .\* exp(-1j \* 2 \* pi \* (k-1) \* (0:N-1) / N));

end

end

**Exercise 7**

xn = [3 2 1 0];

N = length(xn);

n0 = 1;

Xk = DTFS(xn,N);

x\_shifted = circshift(xn, -n0);

X\_shifted\_k = fft(x\_shifted);

k = 0:N-1;

X\_k\_theoretical = Xk .\* exp(-1j \* 2 \* pi \* k \* n0 / N);

subplot(2,2,1);

stem(k, abs(X\_shifted\_k));title('Magnitude of Shifted');

subplot(2,2,2);

stem(k, abs(X\_k\_theoretical));title('Magnitude of Theoretical');

subplot(2,2,3);

stem(k, angle(X\_shifted\_k));title('Phase of Shifted');

subplot(2,2,4);

stem(k, angle(X\_k\_theoretical));title('Phase of Theoretical');

function Xk = DTFS(xn, N)

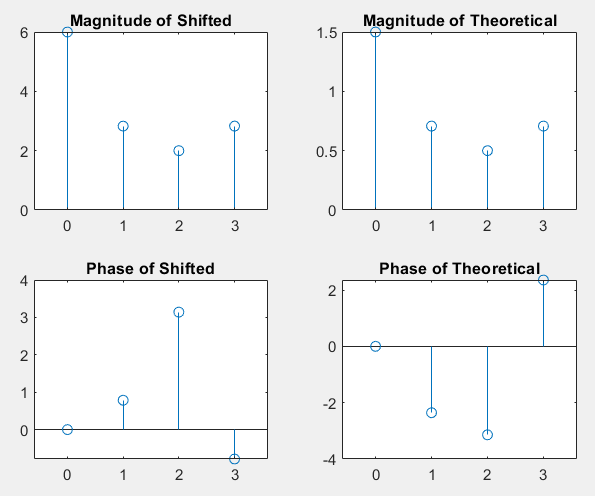
Xk = zeros(1, N);

for k = 1:N

Xk(k) = (1/N) \* sum(xn .\* exp(-1j \* 2 \* pi \* (k-1) \* (0:N-1) / N));

end

end



**Post Lab**

xn = [3 2 1 0];

N = length(xn);

k0 = 1;

Xk = fft(xn);

n = 0:N-1;

x\_freq\_shifted = xn .\* exp(1j \* 2 \* pi \* k0 \* n / N);

X\_shifted\_k = fft(x\_freq\_shifted);

X\_k\_theoretical = circshift(Xk, -k0);

subplot(2,2,1);

stem(0:N-1, abs(X\_shifted\_k));title('Magnitude of Frequency Shifted');

subplot(2,2,2);

stem(0:N-1, abs(X\_k\_theoretical));title('Magnitude of Frequency Theoretical');

subplot(2,2,3);

stem(0:N-1, angle(X\_shifted\_k));title('Phase of Frequency Shifted');

subplot(2,2,4);

stem(0:N-1, angle(X\_k\_theoretical));title('Phase of Frequency Theoretical');

