**Fast Fourier Transform**

**Exercise 1**

x1 = [0 2 4 6];

x2 = [1 3 5 7];

N = 8;

k = (0:N/2-1);

WNk = exp(-1j \* 2 \* pi \* k / N);

G1 = fft(x1);

G2 = fft(x2);

X = [G1 + WNk .\* G2, G1 - WNk .\* G2];

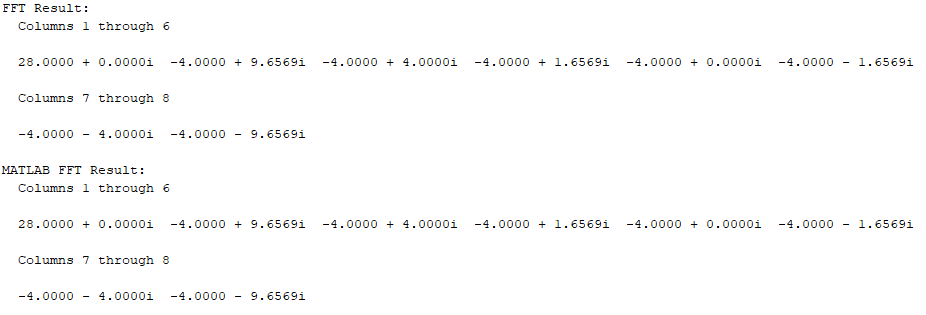
disp('FFT Result:');

disp(X);

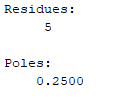
X\_matlab = fft([0 1 2 3 4 5 6 7]);

disp('MATLAB FFT Result:');

disp(X\_matlab);



**Exercise 2**

function y = fft\_2pt(x)

if length(x) ~= 2

error('Input must be of length 2');

end

y = [1 1; 1 -1] \* x(:);

end

**Z-Transform**

**Exercise 1**

num = [5];

den = [1, -1/4];

zplane(num, den);

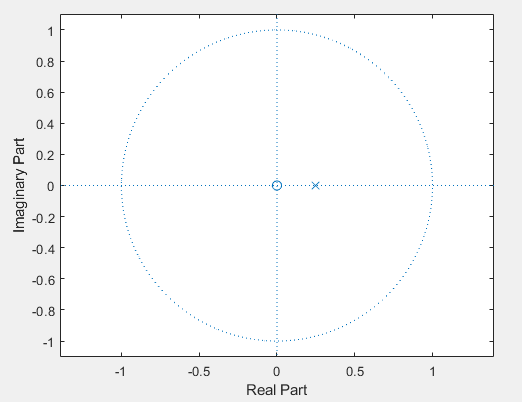
[r, p, c] = residue(num, den);

disp('Residues:');

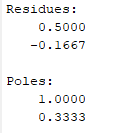
disp(r);

disp('Poles:');

disp(p);



**Exercise 2**

num = [1,0];

den = [3,-4,1];

[r, p, c] = residue(num, den);

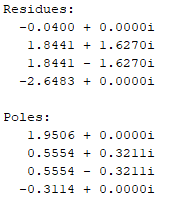
disp('Residues:');

disp(r);

disp('Poles:');

disp(p);

**Exercise 3**

**1)**

num = [1 -1 -4 4];

den = [1 -11/4 13/8 0 -1/4];

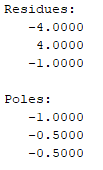
[r, p, c] = residue(num, den);

disp('Residues:');

disp(r);

disp('Poles:');

disp(p);

**2)**

num = [0 0 1 0];

den = [1 2 1.25 0.25];

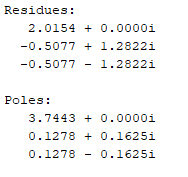
[r, p, c] = residue(num, den);

disp('Residues:');

disp(r);

disp('Poles:');

disp(p);

**3)**

num = [1 -3 4 1];

den = [1 -4 1 -0.16];

[r, p, c] = residue(num, den);

disp('Residues:');

disp(r);

disp('Poles:');

disp(p);