**MATLAB Code**

%Experiment 3 Date: 4th February 2020

%Aim: Computation of DFT, IDFT and Linear & Circular Convolution

%dft.m

function X = dft(x,N)

L = length(x);

if(L<N)

for i=1:N-L

x(L+i) = 0;

i=i+1;

end

elseif(L>N)

for i=1:N

x1(i) = x(i);

end

x = x1;

end

X = zeros(1,N);

for i=1:N

for j=1:N

X(i) = X(i) + x(j)\*exp(-2i\*pi\*(i-1)\*(j-1)/N);

end

end

%idft.m

function x = idft(X,N)

L = length(X);

if(L<N)

for i=1:N-L

X(L+i) = 0;

i=i+1;

end

elseif(L>N)

for i=1:N

x1(i) = X(i);

end

X = x1;

end

x = zeros(1,N);

for i = 1:N

for j = 1:N

x(i) = x(i)+X(j)\*exp(2i\*pi\*(i-1)\*(j-1)/N);

end

end

x = x/N;

%main.m

clc;

clear all;

close all;

N = 200;

x1 = zeros(1,N);

x2 = zeros(1,N);

for n=1:N

x1(n) = 2\*cos(2\*pi\*n/10) + cos(2\*pi\*n/5);

x2(n) = n;

end

figure();

subplot(231);

stem(x1);

title("x1[n] = 2cos(2PIn/10) + cos(2PIn/5)");

xlabel n;

ylabel x[n];

subplot(234);

stem(x2);

title("x2[n] = n");

xlabel n;

ylabel x2[n];

%Finding the N-point DFT

X1 = dft(x1,N);

X2 = dft(x2,N);

subplot(232);

stem(X1);

title ("DFT of x1[n]");

xlabel n;

ylabel X1[n];

subplot(235);

stem(X2)

title ("DFT of x2[n]");

xlabel n;

ylabel X2[n];

%Finding the IDFT of the above

y1 = idft(X1,N);

y2 = idft(X2,N);

subplot(233);

stem(y1);

title ("IDFT of X1[n]");

xlabel n;

ylabel x1[n];

ylim([-2,3]);

subplot(236);

stem(y2);

title ("IDFT of X2[n]");

xlabel n;

ylabel x2[n];

ylim([0,200]);

%Linear and circular convolution using dft and idft

a1 = [1 3 -2 4 7];

h1 = [3 1 21 -3];

n1 = length(a1);

m1 = length(h1);

N1 = n1 + m1 -1;

A1 = dft(a1,N1);

H1 = dft(h1,N1);

B1 = A1.\*H1;

b1 = idft(B1,N1)

if(round(b1) == conv(a1,h1))

disp("Linear Convolution correct")

end

cA1 = dft(a1,n1);

cH1 = dft(h1,n1);

cB1 = cA1.\*cH1;

cb1 = idft(cB1,n1)

if(round(cb1) == round(cconv(a1,h1,n1)))

disp("Circular convolution correct");

end

a2 = zeros(1,10);

h2 = zeros(1,10);

n2 = length(a2);

m2 = length(h2);

for i=1:n2

a2(i) = i;

end

for i=1:m2

h2(i) = power(0.5,i);

end

N2 = n2 + m2 -1;

A2 = dft(a2,N2);

H2 = dft(h2,N2);

B2 = A2.\*H2;

b2 = idft(B2,N2)

if(round(b2,4) == round(conv(a2,h2),4))

disp("Linear Convolution correct")

end

cA2 = dft(a2,n2);

cH2 = dft(h2,n2);

cB2 = cA2.\*cH2;

cb2 = idft(cB2,n2)

if(round(cb2,4) == round(cconv(a2,h2,n2),4))

disp("Circular convolution correct");

end

a3 = zeros(1,20);

h3 = zeros(1,20);

n3 = length(a3);

m3 = length(h3);

for i=1:n3

a3(i) = sin(pi\*i/20);

end

for i=1:m3

h3(i) = power(0.25,i);

end

N3 = n3 + m3 -1;

A3 = dft(a3,N3);

H3 = dft(h3,N3);

B3 = A3.\*H3;

b3 = idft(B3,N3)

if(round(b3,4) == round(conv(a3,h3),4))

disp("Linear Convolution correct")

end

cA3 = dft(a3,n3);

cH3 = dft(h3,n3);

cB3 = cA3.\*cH3;

cb3 = idft(cB3,n3)

if(round(cb3,4) == round(cconv(a3,h3,n3),4))

disp("Circular convolution correct");

end

**RESULT**

b1 =

1.0e+02 \*

Columns 1 through 3

0.0300 - 0.0000i 0.1000 - 0.0000i 0.1800 - 0.0000i

Columns 4 through 6

0.7000 - 0.0000i -0.2600 - 0.0000i 0.9700 - 0.0000i

Columns 7 through 8

1.3500 + 0.0000i -0.2100 + 0.0000i

Linear Convolution correct

cb1 =

1.0e+02 \*

Columns 1 through 3

1.0000 + 0.0000i 1.4500 + 0.0000i -0.0300 - 0.0000i

Columns 4 through 5

0.7000 - 0.0000i -0.2600 - 0.0000i

Circular convolution correct

b2 =

Columns 1 through 3

0.5000 - 0.0000i 1.2500 - 0.0000i 2.1250 - 0.0000i

Columns 4 through 6

3.0625 + 0.0000i 4.0313 - 0.0000i 5.0156 + 0.0000i

Columns 7 through 9

6.0078 - 0.0000i 7.0039 + 0.0000i 8.0020 + 0.0000i

Columns 10 through 12

9.0010 - 0.0000i 4.5000 + 0.0000i 2.2490 + 0.0000i

Columns 13 through 15

1.1230 + 0.0000i 0.5596 + 0.0000i 0.2773 - 0.0000i

Columns 16 through 18

0.1357 + 0.0000i 0.0645 + 0.0000i 0.0283 - 0.0000i

Column 19

0.0098 + 0.0000i

Linear Convolution correct

cb2 =

Columns 1 through 3

5.0000 + 0.0000i 3.4990 + 0.0000i 3.2480 - 0.0000i

Columns 4 through 6

3.6221 - 0.0000i 4.3086 + 0.0000i 5.1514 - 0.0000i

Columns 7 through 9

6.0723 - 0.0000i 7.0322 + 0.0000i 8.0117 + 0.0000i

Column 10

9.0010 - 0.0000i

Circular convolution correct

b3 =

Columns 1 through 3

0.0391 - 0.0000i 0.0870 + 0.0000i 0.1353 - 0.0000i

Columns 4 through 6

0.1808 - 0.0000i 0.2220 - 0.0000i 0.2577 - 0.0000i

Columns 7 through 9

0.2872 - 0.0000i 0.3096 - 0.0000i 0.3243 + 0.0000i

Columns 10 through 12

0.3311 + 0.0000i 0.3297 + 0.0000i 0.3202 - 0.0000i

Columns 13 through 15

0.3028 - 0.0000i 0.2780 + 0.0000i 0.2463 - 0.0000i

Columns 16 through 18

0.2085 - 0.0000i 0.1656 + 0.0000i 0.1187 + 0.0000i

Columns 19 through 21

0.0688 + 0.0000i 0.0172 + 0.0000i 0.0043 + 0.0000i

Columns 22 through 24

0.0011 + 0.0000i 0.0003 + 0.0000i 0.0001 + 0.0000i

Columns 25 through 27

0.0000 + 0.0000i 0.0000 - 0.0000i 0.0000 - 0.0000i

Columns 28 through 30

0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 - 0.0000i

Columns 31 through 33

0.0000 + 0.0000i 0.0000 - 0.0000i 0.0000 + 0.0000i

Columns 34 through 36

0.0000 - 0.0000i 0.0000 - 0.0000i 0.0000 - 0.0000i

Columns 37 through 39

0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i

Linear Convolution correct

cb3 =

Columns 1 through 3

0.0434 - 0.0000i 0.0881 - 0.0000i 0.1355 - 0.0000i

Columns 4 through 6

0.1808 - 0.0000i 0.2220 - 0.0000i 0.2578 - 0.0000i

Columns 7 through 9

0.2872 - 0.0000i 0.3096 + 0.0000i 0.3243 + 0.0000i

Columns 10 through 12

0.3311 + 0.0000i 0.3297 + 0.0000i 0.3202 - 0.0000i

Columns 13 through 15

0.3028 - 0.0000i 0.2780 + 0.0000i 0.2463 + 0.0000i

Columns 16 through 18

0.2085 + 0.0000i 0.1656 + 0.0000i 0.1187 + 0.0000i

Columns 19 through 20

0.0688 + 0.0000i 0.0172 + 0.0000i

Circular convolution correct