Lab 3：FFT

|  |  |
| --- | --- |
| Author | Name： 崔俞崧 赵奕帆 Student ID: 11811305 11812011 |
| Introduction  Lab results & Analysis：  3.5  3.8  3.9      3.10    We use the to calculate the coefficient. In this formular, we need to calculate the multiplications from n=0 to n=N-1 and then multiply this with 1/N. So there are N+1 multiplications in total.  We need to add these N numbers, so need N-1 add operations.        Because the flops is dropped, so here we use the etime function to measure the running time.        The number of operations required by fft is much less than dtfs. As for the larger N, the difference is more obvious.    Ny = N. In each period, the y[n] has N times multiplication and N-1 addition. Ny=N, so in each period the number of y[n] is N. And there are N^2 multiplication and N\*(N-1) addition. So in total it needs O(N^2) operations, which means that Ny = N.      F40c = 1.5892e-06      f80c = 3.4036e-06      f40f = 8.0091e-07    We can find that the y[n] of part(e) and part (g) are almost the same.      F80f = 9.875e-07  We can find that the y[n] of part(f) and part (h) are almost the same.    f40c/f40f = 2.3845  f80c/f80f = 3.3183  FFT is more efficient for every N. When N > 80, I will choose FFT. Because when N>80, the ratio of fNc/fNf will grow larger and the time spend of FFT will be less than dtfs.  代码：  3.10：  b, c  x1 = (0.9).^[0:7];  x2 = (0.9).^[0:31];  x3 = (0.9).^[0:63];  x4 = (0.9).^[0:127];  x5 = (0.9).^[0:255];  t1 = clock;  X1 = dtfs(x1,0);  dtfstime1 = etime(clock,t1);  t2 = clock;  X2 = dtfs(x2,0);  dtfstime2 = etime(clock,t2);  t3 = clock;  X3 = dtfs(x3,0);  dtfstime3 = etime(clock,t3);  t4 = clock;  X4 = dtfs(x4,0);  dtfstime4 = etime(clock,t4);  t5 = clock;  X5 = dtfs(x5,0);  dtfstime5 = etime(clock,t5);  dtfscomps = [dtfstime1,dtfstime2,dtfstime3,dtfstime4,dtfstime5];  t1 = clock;  X1 = fft(x1,0);  dtfstime1 = etime(clock,t1);  t2 = clock;  X2 = fft(x2,0);  dtfstime2 = etime(clock,t2);  t3 = clock;  X3 = fft(x3,0);  dtfstime3 = etime(clock,t3);  t4 = clock;  X4 = fft(x4,0);  dtfstime4 = etime(clock,t4);  t5 = clock;  X5 = fft(x5,0);  dtfstime5 = etime(clock,t5);  fftcomps = [dtfstime1,dtfstime2,dtfstime3,dtfstime4,dtfstime5];  x = [8 32 64 128 256];  loglog(x,dtfscomps);  hold on  loglog(x,fftcomps);  legend('dtfscomps','fftcomps');  xlabel('N');  e.  x = 0.9.^[0 : 39];  h = 0.5.^[0 : 39];  y = conv([x x],h);  stem(0:39,y(1:40));  xlabel('n');  ft1 = conv(x,h);  f40c = timeit(ft1);  f.  x = 0.9.^[0 : 79];  h = 0.5.^[0 : 79];  y = conv([x x],h);  stem(0:79,y(1:80));  xlabel('n');  ft1 = conv(x,h);  f40c = timeit(ft2);  g.  x = 0.9.^[0 : 39];  h = 0.5.^[0 : 39];  a\_k = (1/40)\*fft(x);  h\_k = (1/40)\*fft(h);  b\_k = 40\*a\_k.\*h\_k;  y = 40 \* ifft(b\_k);  stem(0:39,y);  xlabel('n');  ylabel('y[n]');  figure;  stem(0:39,y,'r-o');  hold on;  stem(0:39,y1(1:40),'b-x');  xlabel('n');  ylabel('y[n]');  ft3 = 40 \* ifft(b\_k);  f40f = timeit(ft3);  h.  x = 0.9.^[0 : 79];  h = 0.5.^[0 : 79];  a\_k = (1/80)\*fft(x);  h\_k = (1/80)\*fft(h);  b\_k = 80\*a\_k.\*h\_k;  y1 = 80 \* ifft(b\_k);  stem(0:79,y1);  figure;  stem(0:79,y1,'r-o');  hold on;  stem(0:79,yf(1:80),'b-x');  xlabel('n');  ylabel('y[n]');  ft4 = 80 \* ifft(b\_k);  f80f = timeit(ft4);  Note: Please indicate meaning of the symbols in all expressions. Please indicate the coordinate and unit in all figures. | |
| Experience  You can write your experience with this project. Any comment and suggestion on this course are also very welcome. | |
| Score |  |