Taylor Larrechea

Dr. Gustafson

Fourier Analysis Math 362

October 7, 2017

Class Prep 8

Section 4.3

Key Concepts – In this section we use complex exponential vectors as our orthogonal expansion vectors, and derive the discrete Fourier (DFT) and inverse discrete Fourier (IDFT) transform matrices. We will compute the discrete Fourier transform and inverse discrete Fourier transform of a signal vector **v** using the DFT and IDFT matrices. We will also use MATLAB’s FFT command to compute the discrete Fourier transform of **v**, and see how aliasing creates a conjugate symmetry in the DFT output.

|  |  |
| --- | --- |
| Input Commands | Output Plots (if Applicable) |
| >> InverseDFT(4)  t =  0  0.2500  0.5000  0.7500  G =  1.0000 + 0.0000i 1.0000 + 0.0000i 1.0000 + 0.0000i 1.0000 + 0.0000i  1.0000 + 0.0000i 0.0000 + 1.0000i -1.0000 + 0.0000i -0.0000 - 1.0000i  1.0000 + 0.0000i -1.0000 + 0.0000i 1.0000 - 0.0000i -1.0000 + 0.0000i  1.0000 + 0.0000i -0.0000 - 1.0000i -1.0000 + 0.0000i 0.0000 + 1.0000i | |
| >> x=[1,2,3,4]'  x =  1  2  3  4  >> F=1/4\*[1,1,1,1;1,-i,-1,i;1,-1,1,-1;1,i,-1,-i]  F =  0.2500 + 0.0000i 0.2500 + 0.0000i 0.2500 + 0.0000i 0.2500 + 0.0000i  0.2500 + 0.0000i 0.0000 - 0.2500i -0.2500 + 0.0000i 0.0000 + 0.2500i  0.2500 + 0.0000i -0.2500 + 0.0000i 0.2500 + 0.0000i -0.2500 + 0.0000i  0.2500 + 0.0000i 0.0000 + 0.2500i -0.2500 + 0.0000i 0.0000 - 0.2500i  >> c=F\*x  c =  2.5000 + 0.0000i  -0.5000 + 0.5000i  -0.5000 + 0.0000i  -0.5000 - 0.5000i | |
| >> G=inv(F)  G =  1.0000 + 0.0000i 1.0000 + 0.0000i 1.0000 + 0.0000i 1.0000 + 0.0000i  1.0000 + 0.0000i 0.0000 + 1.0000i -1.0000 + 0.0000i 0.0000 - 1.0000i  1.0000 + 0.0000i -1.0000 + 0.0000i 1.0000 + 0.0000i -1.0000 + 0.0000i  1.0000 + 0.0000i 0.0000 - 1.0000i -1.0000 + 0.0000i 0.0000 + 1.0000i  >> x=c(1)\*G(:,1)+c(2)\*G(:,2)+c(3)\*G(:,3)+c(4)\*G(:,4)  x =  1  2  3  4 | |
| >> x=[1,2,3,4]'  x =  1  2  3  4  >> c=1/4\*fft(x)  c =  2.5000 + 0.0000i  -0.5000 + 0.5000i  -0.5000 + 0.0000i  -0.5000 - 0.5000i | |
| >> x=[1,2,3,4,5,6,7,8]'  x =  1  2  3  4  5  6  7  8  >> c=1/8\*fft(x)  c =  4.5000 + 0.0000i  -0.5000 + 1.2071i  -0.5000 + 0.5000i  -0.5000 + 0.2071i  -0.5000 + 0.0000i  -0.5000 - 0.2071i  -0.5000 - 0.5000i  -0.5000 - 1.2071i | |
| >> x=audioread('aah.wav');  >> plot(x) |  |

|  |  |
| --- | --- |
| >> plot(x(10000:11900)) |  |
| >> N=length(x)  N =  88238  >> c=(1/N)\*fft(x);  >> c(1:8)  ans =  1.0e-03 \*  -0.0549 + 0.0000i  0.0331 + 0.0187i  -0.0460 + 0.0716i  -0.1118 - 0.0707i  0.2272 + 0.3723i  0.2374 - 0.0936i  0.2482 - 0.4955i  -0.1352 + 0.4921i | |
| >> N=8;  >> t=[0:1/N:(N-1)/N]';  >> f=2\*t-1;  >> plot(f);  >> c=(1/N)\*fft(f)  c =  -0.1250 + 0.0000i  -0.1250 + 0.3018i  -0.1250 + 0.1250i  -0.1250 + 0.0518i  -0.1250 + 0.0000i  -0.1250 - 0.0518i  -0.1250 - 0.1250i  -0.1250 - 0.3018i |  |
| >> a0=real(c(1))  a0 =  -0.1250  >> a=2\*real(c(2:4))  a =  -0.2500  -0.2500  -0.2500  >> b=-2\*imag(c(2:4))  b =  -0.6036  -0.2500  -0.1036 | |
| >> N=64;  >> t=[0:1/N:(N-1)/N]';  >> f=2\*t-1;  >> c=(1/N)\*fft(f);  >> stem(abs(c(1:N/2))) |  |

|  |  |
| --- | --- |
| >> SawtoothFFT(64)  Coeff\_a\_0 =  -0.0156  Coeffs\_ak\_bk =  -0.0312 -0.6361  -0.0312 -0.3173  -0.0313 -0.2107  -0.0312 -0.1571  -0.0312 -0.1248  -0.0312 -0.1030  -0.0312 -0.0873  -0.0312 -0.0754  -0.0312 -0.0661  -0.0313 -0.0585  -0.0312 -0.0521  -0.0312 -0.0468  -0.0312 -0.0421  -0.0312 -0.0381  -0.0312 -0.0345  -0.0312 -0.0312  -0.0313 -0.0283  -0.0313 -0.0256  -0.0312 -0.0232  -0.0312 -0.0209  -0.0313 -0.0187  -0.0312 -0.0167  -0.0312 -0.0148  -0.0312 -0.0129  -0.0313 -0.0112  -0.0312 -0.0095  -0.0312 -0.0078  -0.0312 -0.0062  -0.0312 -0.0046  -0.0312 -0.0031  -0.0312 -0.0015 |  |

Section 4.4

Key Concepts – In this section we revisit the concept of a frequency domain plot, paying particular attention to the connection between frequency index and frequency. This is important when writing MATLAB programs for frequency domain applications. Three frequency domain applications that we will look at include bandpass filters, bandstop filters, and DFT thresholding.

|  |  |
| --- | --- |
| Input Commands | Output Plots (if Applicable) |
| >> c=[8,0,16,0,12,0,4,0,0];  >> stemT(c,1)  >> stemT(c,4) |  |
| >> [x,sr]=audioread('aah.wav');  >> sr  sr =  44100  >> N=length(x)  N =  88238  >> T=N/sr  T =  2.0009 | |
| >> c=1/N\*fft(x);  >> y=abs(c);  >> stemT(y(1:1000),1)  >> stemT(y(1:1000),T) |  |
| >> [x,sr]=audioread('aah.wav');  >> SoundWaveTimeFreq(x,sr,0.5,0.54,60,160) |  |
| >> plotfft(y(1:1000),T)  >> stemT(y(100:200),T) |  |

|  |  |
| --- | --- |
| >> [x,sr]=audioread('aah.wav');  >> N=length(x);  >> z=zeros(1,N);  >> c=1/N\*fft(x);  >> z(100:200)=c(100:200);  >> T=N/sr;  >> y=abs(z);  >> stemT(y(1:1000),T) |  |
| >> z(N-200:N-100)=c(N-200:N-100);  >> f=N\*real(ifft(z));  >> plot(f(1:2291)),axis tight, title('Plot of Bandpassed f') |  |
| >> z=c;  >> z(100:200)=zeros(1,101);  >> z(N-200:N-100)=zeros(1,101);  >> y=abs(z);  >> stemT(y(1:1000),T)  >> f=N\*real(ifft(z));  >> plot(f(1:2291)), axis tight, title('Plot of Bandstopped f') |  |

|  |  |
| --- | --- |
| >> [x,sr]=audioread('aah.wav');  >> BandPass(x,sr,77,50,100,1,500)  >> BandStop(x,sr,77,50,100,1,500) |  |

|  |  |
| --- | --- |
| >> [x,sr]=audioread('aah.wav');  >> FFTsoundthresh(x,sr,95,0.5,0.54,60,160)  Dominant\_frequency =  'The dominant frequency is 77 Hz.'  Percent\_Reduction =  'The percent reduction is 94.999887.'  Compression\_Ratio =  'The compression ratio is 88238 to 4412, or 19.999547 to 1.'  Run\_Time =  'The run time was 0 minutes and 2.700680e-01 seconds.' |  |
| >> FFTsoundthresh(x,sr,98,0.5,0.54,60,160)  Dominant\_frequency =  'The dominant frequency is 77 Hz.'  Percent\_Reduction =  'The percent reduction is 97.999728.'  Compression\_Ratio =  'The compression ratio is 88238 to 1765, or 49.993201 to 1.'  Run\_Time =  'The run time was 0 minutes and 2.761699e-01 seconds.' |  |

|  |  |
| --- | --- |
| >> RawSoundThresh(x,sr,60,0.5,0.54)  Percent\_Reduction =  'The percent reduction is 60.000227.'  Compression\_Ratio =  'The compression ratio is 88223 to 35289, or 2.500014 to 1.'  Run\_Time =  'The run time was 0 minutes and 2.152902e-01 seconds.' |  |