Taylor Larrechea

Dr. Gustafson

Math 362 Fourier Analysis

November 4, 2017

Ch. 4.6

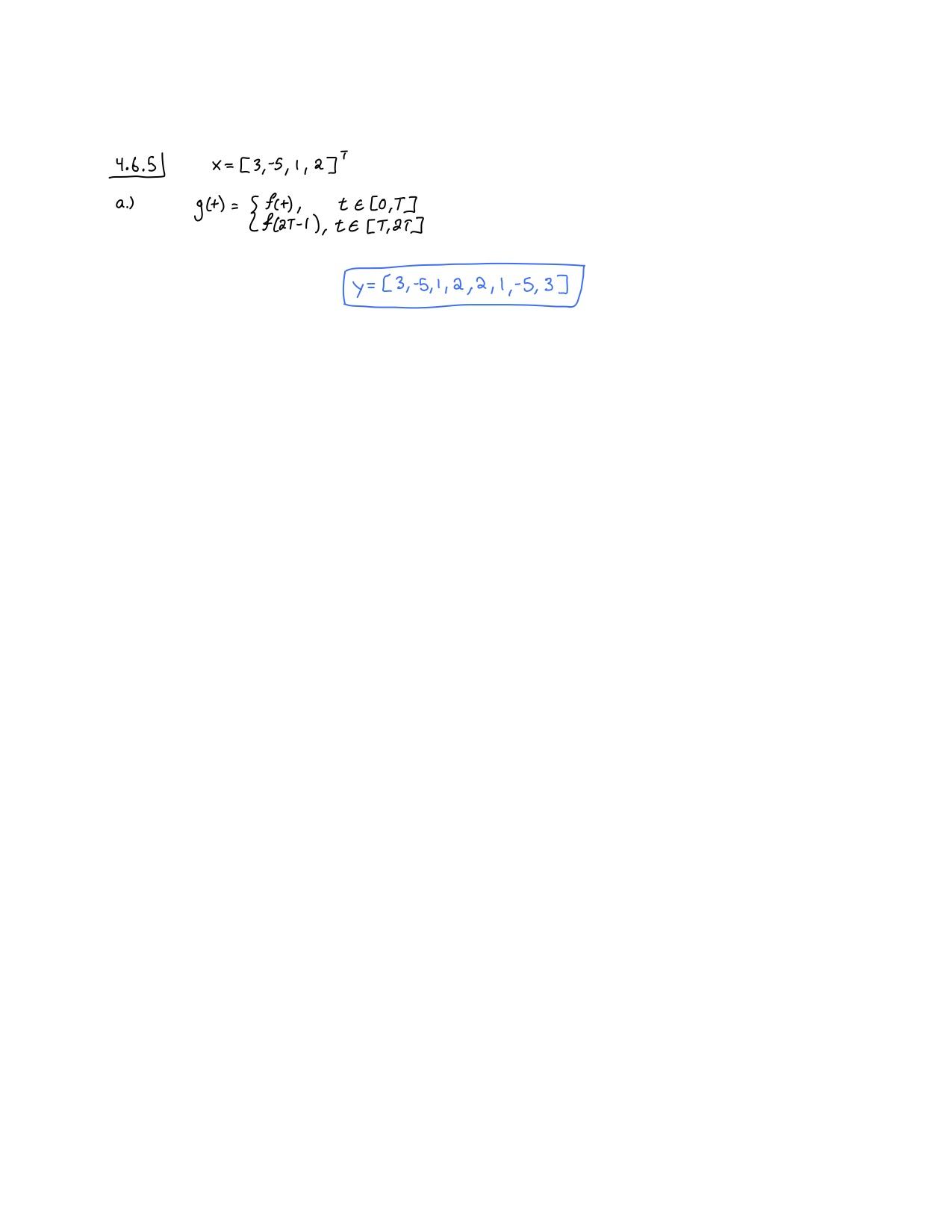
Section 4.6

4.6.5

For the vectors **x** specified, do the following.

1. Use hand calculations to determine the symmetric extension **y** of **x**.
2. Use the MATLAB program EvenSymPlot(x) to determine the symmetric extension **y** of **x** and to plot **x** and **y**.

a.)



b.)

|  |  |
| --- | --- |
| Input Commands | Output (Plot if Applicable) |
| >> x=[3,-5,1,2]';  >> EvenSymPlot(x)  y =  3 -5 1 2 2 1 -5 3 |  |

4.6.11

For the functions on given, adapt the MATLAB m-file EvenSawPlot(N) for to plot the sampled version **f** and its symmetric extension **g**. Show all MATLAB commands used.

|  |  |
| --- | --- |
| Input Commands | Output (Plot if Applicable) |
| >> EvenSawPloti(8) |  |

4.6.24

For the vectors **x** specified, do the following. Show all steps, work, and MATLAB commands used, using Example 4.6.3 as a guide.

(b) Compute the DCT II values of **x** using the m-file DCTII(x)

(d) Compute the DCT II values of **x** using the m-file DCTIImatrix(x)

(g) Compute the inverse DCT II values of **x** using the m-file DCTIImatrix(x).

b.)

|  |  |
| --- | --- |
| Input Commands | Output (Plot if Applicable) |
| >> x=[3,-5,1,2]';  >> DCTII(x) | c2 =  0.5000  -0.9703  4.5000  4.1903 |

d.)

|  |  |
| --- | --- |
| Input Commands | Output (Plot if Applicable) |
| >> x=[3,-5,1,2]';  >> DCTIImatrix(x) | c2 =  0.5000  -0.9703  4.5000  4.1903  x2 =  3.0000  -5.0000  1.0000  2.0000 |

g.)

|  |  |
| --- | --- |
| Input Commands | Output (Plot if Applicable) |
| >> x=[3,-5,1,2]';  >> DCTIImatrix(x) | c2 =  0.5000  -0.9703  4.5000  4.1903  x2 =  3.0000  -5.0000  1.0000  2.0000 |

4.6.36

In the following exercise, use the MATLAB programs DCTIIsound and FFTthreshSpec to perform block thresholding on the sound waves indicated. Use block size and a 95% threshold level to produce the requested results, as in Example 4.6.6. Be sure to show the MATLAB commands used.

1. For the sound wave indicated, plot the before and after time domain and spectrogram graphs for both the DCT II and FFT block thresholding methods. Experiment with different values of ymax before you settle on a particular graph to include here.
2. Briefly compare and contrast what can be seen in the DCT II and FFT spectrograms. Does the DCT II thresholding preserve more of the higher frequencies than the FFT spectrogram? Either way, relate the visual information to what is heard when the thresholded sound waves are played back.
3. Of the DCT II and FFT and block thresholding methods, which one produces a sound wave that sounds more like the original sound wave? Briefly compare, contrast and describe what you hear.

a.)

|  |  |
| --- | --- |
| Input Commands | Output (Plot if Applicable) |
| >> [z,sr]=audioread('meantweet.wav');  >> DCTIIsound(z,sr,256,95,1100)  Percent\_Reduction =  'The percent reduction is 94.921875.'  Compression\_Ratio =  'The compression ratio is 256 to 13, or 19.692308 to 1.'  >> [z,sr]=audioread('meantweet.wav');  >> FFTthreshSpec(z,sr,256,95,1100)  Percent\_Reduction =  'The percent reduction is 94.921875.'  Compression\_Ratio =  'The compression ratio is 256 to 13, or 19.692308 to 1.' |  |

b.)

When playing back the two different methods of block thresholding, the DCTIIsound does a better job of preserving the original sound. The FFTthreshspec sounds very scratchy compared to the DCTIIsound and this is telling in the fact that it doesn’t preserve the higher frequencies as well as the DCTIIsound m-file does. If the FFTthreshspec were able to preserve more of the higher frequencies, it wouldn’t sound as scratchy and it would sound a lot better, similar to the DCTIIsound. This can be observed on the spectrogram plots as well.

c.)

As mentioned in part (b) above, the DCTIIsound sounds more like the original sound wave than the FFTthreshspec. This is largely in part due to the DCTIIsound preserving more of the higher frequencies terms than the FFTthreshspec.