Fourier Theory

Due March 20, 2023

What you need to get

- `YOU_a4q3q4.ipynb`
- `YOU_a4q5.ipynb`
- `Jinan.jpg`: image file for Q4
- `recording.wav`: WAV audio file for Q5

What to do

1. **[15 marks]** Let $f = [1,\ 3,\ 6,\ 3,\ 9,\ 3,\ 6,\ 3]^{\mathrm{T}}$. Use the FFT algorithm to compute the DFT of f. Show your work in a butterfly diagram. You may **handwrite** this solution only (but still submit it as a PDF).

2. **[20 marks]** Consider the *real, even* vector $f = [f_0, f_1, \ldots, f_{N-1}]$, in which $f_n \in \mathbb{R}$, and $f_n = f_{N-n}$ (analogous to an even function, in which $f(x) = f(-x)$). Given the Discrete Fourier Transform (DFT),

$$F_k = \sum_{n=0}^{N-1} f_n \exp\left(\frac{-2\pi i n k}{N}\right) \quad \text{for } k = 0, \ldots, N-1\ ,$$

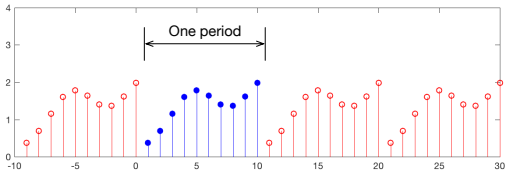
prove that the vector of Fourier coefficients, $F = [F_0, F_1, \ldots, F_{N-1}]$, is also a real, even vector. Be sure to justify your steps.

3. **[15 marks] Discrete Cosine Transform**

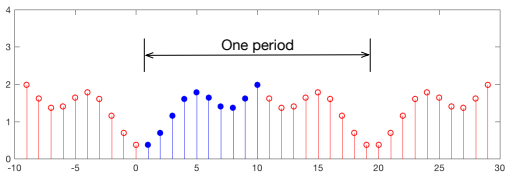
The theorem you (hopefully) proved in question 2 can be summarized as,

Theorem: The DFT of a real, even vector is also real and even.

This property forms the basis of the Discrete Cosine Transform (DCT). The DCT is a special form of the DFT that makes some assumptions about its input. It operates on only real-valued input, and implicitly assumes that the input comes from an even vector. That is, unlike the DFT that implicitly assumes the vector is one period taken from a periodic signal (shown below on the left), the DCT assumes that the vector is *one half of one period* taken from a signal that is periodic **and** even (shown below on the right).



Periodic Extension



Even Extension

A benefit of the DCT is that it produces real-valued Fourier coefficients. The other benefit is that the Fourier coefficients are even, so you only need to store half of them.

As it turns out, the DFT of an even vector g yields real-valued Fourier coefficients if g is real-valued. So, one way to compute the DCT of a real-valued N-vector f is to perform an even extension of f to get g, compute the DFT of g, then extract only the first N Fourier coefficients. This also works in 2-D, 3-D, etc.

Complete the two functions `myDCT` and `myIDCT` (in the notebook `YOU_a4q3q4.ipynb`) that implement the DCT and its inverse for 2-D graylevel images. You should use the supplied functions `EvenExtension` and `IEvenExtension` to do (and undo) the even extensions. See the functions' help files for more details.

4. **[25 marks] Image Compression**

   In this question, you will use the DCT to perform JPEG-like image compression. The notebook `YOU_a4q3q4.ipynb` contains the function `myJPEGCompress`, which takes an image as input and is supposed to generate an output array containing DCT coefficients. The notebook also contains the function `myJPEGDecompress`, which is supposed to do the opposite; takes an array of DCT coefficients and generate an image from it.

   (a) **[10 marks]** Complete the function `myJPEGCompress`. The compression algorithm breaks the input image into tiles of size $T \times T$ (one of the inputs to `myJPEGCompress` is T, the tile size). Note that there might be leftover margins on the right and bottom edge of the image that are not part of the tiling. You can ignore those margins.

   Each tile is compressed by storing only a subset of the tile's DCT coefficients. Hence, the compression method performs lossy compression. Each $T \times T$ image tile is processed like this:

   i. Compute the DCT of the $T \times T$ tile.

   ii. Extract a $D \times D$ sub-array of low-frequency coefficients from the array of DCT coefficients (note that D must be smaller than T).

   iii. Save the $D \times D$ sub-array as the compressed representation of the tile.

   Do this for each tile in the image, and assemble the DCT blocks into another array; this array is your compressed representation of the image. You should use your `myDCT` and/or `myIDCT` to help you with this question.

   (b) **[10 marks]** Complete the function `myJPEGDecompress`. It should take the compressed representation (and T and D) and undo the compression process to produce an approximation to the original image. For each $D \times D$ block,

   i. Embed the $D \times D$ array of DCT coefficients into a $T \times T$ array of zeros.

   ii. Compute the inverse DCT on the array.

   Assemble all the $T \times T$ tiles back into an image. Note that this compressed image might be a bit smaller than the original. Again, use `myDCT` and/or `myIDCT` for this question.

   (c) **[5 marks]** The **compression ratio** is the number of pixels in your original image divided by the number of elements in your compressed representation, often expressed using "x:1" notation.

   For example, suppose your original image is 120×120, and it is broken into tiles of size 10×10. For each tile, only 4×4 DCT coefficients are kept. Hence, the compressed representation would be 48×48. In this case, the compression ratio is 6.25:1, meaning the original image consists of 6.25 times as many numbers as the compressed representation.

   Edit the notebook to demonstrate your compression and decompression functions. Compress and decompress the `Jinan.jpg` image (supplied) using a variety of D-values. Use a tile size of 25 for each. Try to achieve compression ratios close to 4:1, 10:1, and 25:1. Display each of the (de)compressed images. The title of each image should state the compression ratio for that image.

5. **[25 marks]** After unlocking James Carver's smartphone, the police check his calendar app and find an appointment labelled "Dentist: Routine cleaning". But the meeting with the dentist is suspicious because there is another routine cleaning appointment just 2 weeks earlier, and Mr. Carver's dental insurance only covers dental cleaning every 9 months. This prompted the police to install covert listening devices throughout his dentist's clinic. Unfortunately, the recording captured during Mr. Carver's meeting is contaminated by noise from the dental drill. The police would like to know what was said in the recording, but need the forensic specialists to help them isolate the conversation.

   Your job, as a forensic specialist, is to process the recording to isolate the speech, and convey your analysis in a report, as follows.

   The jupyter notebook `YOU_a4q5.ipynb` reads in the recording, constructs some useful variables, and plots the sound in the time domain. Edit the notebook to also do the following:

   (a) Plot the modulus of the Fourier coefficients in zero-shifted format (ie. with the DC in the middle, positive frequencies to the right, and negative frequencies to the left). Note that the variable `omega` is an array of frequencies corresponding to the Fourier coefficients.

   (b) Try to filter out the sound of the dentist's drill. To do this, choose a threshold frequency and set all Fourier coefficients above that frequency to zero. Plot your filtered Fourier coefficients (also in zero-shifted format).

   (c) Reconstruct the sound corresponding to the filtered signal. Plot the filtered sound (in the time domain).

   (d) What did Mr. Carver say? Write a transcript of what he said.

   Ensure that the plots are labelled appropriately, including a title and axis labels.

What to submit

You must submit a series of PDF documents to Crowdmark. Each coding question should be in a PDF export of a jupyter notebook (sometimes several questions in a single notebook). When a proof or manual calculation is requested, you can typeset your solution using LaTeX or Word – **Photographs or scans of handwritten solutions should be legible, otherwise, the TAs might deduct marks.** . These solutions should also be submitted as PDFs.

Finally, upload your Python notebooks on Learn dropbox.