CSC 222: Programming Project # 3: Exploring the Fourier Transform

Instructor: V. Paúl Pauca

due date: Tuesday March 31

---

1 Exploring the roots of unity

The Nth roots of unity are all the complex numbers z such that $z^N = 1$. Let $\omega_N = \exp(\imath 2\pi/N)$, where $\imath = \sqrt{-1}$. The Nth roots of unity are the powers:

$$\omega_N^0, \omega_N^1, \omega_N^2, \ldots, \omega_N^{N-1}$$

For this assignment you will need to write code to verify or disprove, via output produced by your code, the various assertions about the Nth roots of unity specified below.

1. *The Nth roots of unity lie in the complex unit circle.*

   Hint: Write a program that draws the Nth roots of unity for some given $N > 0$. You can use whatever programming language you want: Matlab, Python, Java, C++, etc. You will need to use some kind of data structure to store complex numbers. It could as simple as an N-by-2 array of doubles, where the each row is the (real, imaginary) pair. If you are using Java you can use the static methods from this simple class for drawing into a canvas:

   `csweb.cs.wfu.edu/~pauca/csc111/StdDraw.java`

2. *For any even $N > 0$, the plus-minus Nth root pairs are $(\omega_N^0, \omega_N^{N-1})$, $(\omega_N^1, \omega_N^{N-2})$, ..., $(\omega_N^{N/2-1}, \omega_N^{N/2+1})$.*

3. *The 16th roots of unity can be calculated from the 32th roots of unity. The 8th roots of unity can be calculated from the 16th roots of unity and so on.*

2 Exploring the Discrete Fourier Transform

The Discrete Fourier Transform (DFT) enables us to obtain the representation of any discrete signal or one-dimensional function in frequency space and vice versa. Specifically, for a given discrete signal $g = (g_0, g_1, \ldots, g_{N-1})$, its representation in frequency space is given by the complex coefficients (forward DFT):

$$G(\jmath) = \sum_{\imath=0}^{N-1} g(\imath) \exp(-\imath 2\pi i j/N), \quad \text{for } \jmath = 0, 1, \ldots, N-1. \tag{1}$$

Similarly, given a signal in frequency space, represented by coefficients $\{G(\jmath)\}$ for $\jmath = 0, 1, \ldots, N-1$, its representation in real space is re-obtained as (inverse DFT):

$$g(i) = \frac{1}{N} \sum_{\jmath=0}^{N-1} G(\jmath) \exp(\imath 2\pi i j/N), \quad \text{for } i = 0, 1, \ldots, N-1. \tag{2}$$

Matrix representation of the DFT. The forward DFT operation can be expressed as a matrix-vector product $\hat{\mathbf{g}} = \mathbf{F}\mathbf{g}$ of a $N \times N$ matrix:

$$\mathbf{F} = [\mathbf{F}_{ij}] = [\exp(-\imath 2\pi i j/N)]$$

with a $N \times 1$ vector $\mathbf{g} = [g(i)]$. The general approach to compute $\hat{\mathbf{g}} = \mathbf{F}\mathbf{g}$ requires $O(N^2)$ operations (*why?*).

The inverse of F. The inverse DFT operation can also be expressed as a matrix-vector product $\mathbf{g} = \mathbf{F}^{-1}\hat{\mathbf{g}}$. In general, finding the inverse of a matrix requires $O(N^3)$ operations. The beauty of the DFT is that the inverse of $\mathbf{F}$ is very easy to obtain:

$$\mathbf{F}^{-1} = \frac{1}{N}[\exp(\imath 2\pi ij/N)]$$

1. Write a program that creates the DFT matrices $\mathbf{F}$ and $\mathbf{F}^{-1}$ for a given (even) value of N. Use your program to print $\mathbf{F}$ to the screen for $N = 8, 4$, and 2. Also print the corresponding inverses $\mathbf{F}^{-1}$. Verify numerically or through a mathematical proof that $\mathbf{F}\mathbf{F}^{-1} = \mathbf{I}$ (the identity matrix of size $N \times N$).

2. Download the following file: `csweb.cs.wfu.edu/~pauca/csc222/1Dsignal.txt`
   This file contains a 1D signal $\mathbf{g} = (g_0, g_1, \cdots, g_{1023})$ (of length 1024). Plot this signal to see how it looks. Take the forward DFT of $\mathbf{g}$ by 1) forming a 1024×1024 matrix $\mathbf{F}$ and 2) computing the matrix vector product $\hat{\mathbf{g}} = \mathbf{F}\mathbf{g}$. The resulting vector $\hat{\mathbf{g}}$ contains the representation of $\mathbf{g}$ in frequency space. Plot the magnitude of $\hat{\mathbf{g}}$.

3. Verify that you can get back the original signal by transforming $\hat{\mathbf{g}}$ back into real space using the inverse DFT: $\mathbf{y} = \mathbf{F}^{-1}\hat{\mathbf{g}}$. If your code is correct, then the real part of $\mathbf{y}$ should be numerically identical to $\mathbf{g}$.

4. (bonus) *Filtering.* Play around with $\hat{\mathbf{g}}$ by zeroing out elements in specific ranges. For example, if

   $$\hat{\mathbf{g}} = [\hat{g}_0, \hat{g}_1, \hat{g}_2, \hat{g}_3, \hat{g}_4, \hat{g}_5, \hat{g}_6, \hat{g}_7],$$

   What would happen to my original signal if I zero out $\hat{g}_2, \hat{g}_3, \hat{g}_4, \hat{g}_5$?

What to turn in. You must turn a report containing the following items:

1. **Exploring the roots of unity:** *Are the assertions made about the roots of unity true?* Use plots or output of your code to support your answer.

2. **Exploring the discrete Fourier transform:**
   (a) Show the output produced by your code for Fourier matrices of size $N = 8, 4, 2$.
   (b) Show the plot of the magnitude of $\hat{\mathbf{g}}$.
   (c) Show a plot contrasting the original signal $\mathbf{g}$ with the signal you obtained by the inverse DFT of $\hat{\mathbf{g}}$.

Grading rubric

- A range: answers to the above questions and assertions are correct and well verified, report is well presented, and comments and results show great understanding of the DFT.
- B range: answers to the above questions and assertions are correct and mostly well verified, report is pretty well presented and organized.
- C range: answers to the above questions and assertions are correct but not very well verified. The report is not quite well written or organized.
- D range: answers to the above questions and assertions are incorrect and poorly verified. The report is poor.
- F: fails to turn in assignment.