EECS 16B Designing Information Devices and Systems II
Spring 2019 UC Berkeley

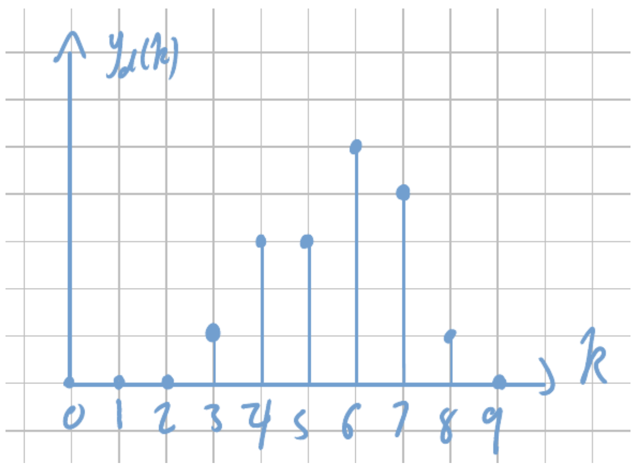
Homework 12

This homework is due on Wednesday, May 1, 2019, at 11:59PM.
Self-grades are due on Saturday, May 4, 2019, at 11:59PM.

1. Piecewise Linear interpolation

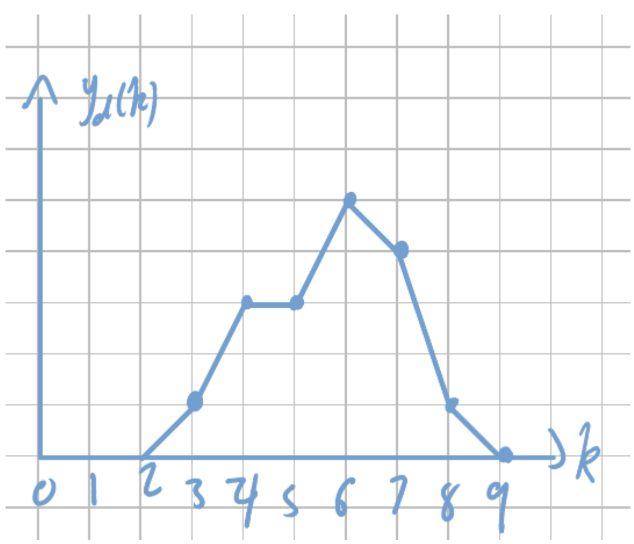
Suppose we have a discrete-time signal $y_d(k)$ that we would like to interpolate.

We will assume that the discrete-time signal is of *finite duration*– that is, the signal "begins" at some time k_1 and "ends" at some time k_2, and we can assume $y_d(k) = 0$ for $k < k_1$ and $k > k_2$. For example, if our discrete-time signal looked like this:



then we would have $k_1 = 3, k_2 = 8$.

One of the simplest ways to interpolate this signal would be to simply connect the points of the discrete-time signal with straight lines. If we were to interpolate the signal above this way, we would get this:



As you can see, the interpolated signal $y(t)$ is a straight line over intervals of the form $[k, k+1]$ for all integers k, although the entire function $y(t)$ is not itself a straight line. For this reason, we call this $y(t)$ the *piecewise linear* (PWL) interpolation of the discrete-time signal $y_d(k)$.

Although we've just described the PWL interpolation in an intuitive and somewhat *ad hoc* way, it turns out that the PWL interpolation can be expressed as a basis function interpolation. **in this problem, you will show how this is true**.

(a) Consider the function $\phi(t)$ defined as,

$$\phi(t) = \begin{cases} 1 - |t|, & t \in [-1, 1] \\ 0, & \text{otherwise} \end{cases} . \quad (1)$$

Sketch $\phi(t-k)$ for some arbitrary integer k. You may choose a specific integer for k in your sketch ($k = 3$ perhaps), or you may keep the sketch entirely in terms of k. The graph in the solution will be drawn in terms of k, so we encourage you to try it that way.

(b) We'll be using the function $\phi(t)$ as the basis function for the PWL interpolation.

We will begin our analysis right at the beginning of the signal. Write the basis function and coefficient that captures the line of $y(t)$ from $t = k_1 - 1$ to $t = k_1$. That is to say, find real number α and integer p such that,

$$y(t) = \alpha\phi(t-p) \text{ for } t \in [k_1 - 1, k_1]$$

(c) Now, consider any integer $k^\star$such that $k_1 < k^\star < k_2$. Over the interval $[k^\star, k^\star + 1]$, the interpolated signal $y(t)$ is a straight line. What is the equation of this line? In other words, **find real numbers m and b such that**

$$y(t) = mt + b, \text{ over the interval } [k^\star, k^\star + 1]. \quad (2)$$

(d) Consider the function

$$g(t) = y_d(k^\star)\phi(t-k^\star) + y_d(k^\star + 1)\phi(t - (k^\star + 1))$$

This function is also a straight line over the interval $[k^\star, k^\star + 1]$. What is the equation of the line over this region? Write it again in the form

$$y(t) = mt + b, \text{ over the interval } [k^\star, k^\star + 1]. \quad (3)$$

This should match your previous answer.

(e) Given what you've shown in the previous parts, we can now express the PWL interpolation of $y_d(k)$ as a sum of shifted ϕ functions. Find the coefficients α_k such that

$$y(t) = \sum_{k=-\infty}^{\infty} \alpha_k \phi(t-k). \quad (4)$$

hint: remember, our goal is to show that the PWL interpolation is a basis function interpolation *with basis function ϕ. In that sense, the α_k have already been chosen (consult your notes and the relavent discussion handout), and you need only verify that they are correct.*

2. Lagrange Interpolation by Polynomials

Given n distinct points and the corresponding sampling of a function $f(x)$, $(x_i, f(x_i))$ for $0 \le i \le n-1$, the Lagrange polynomial interpolation is the polynomial of the least degree that passes through all of the given points.

Given n distinct points and the corresponding evaluations, $(x_i, f(x_i))$ for $0 \le i \le n-1$, the Lagrange polynomial interpolation is the $n-1^{th}$ degree polynomial

$$P(x) = \sum_{i=0}^{i=n-1} f(x_i) L_i(x),$$

where

$$L_i(x) = \prod_{j=0, j\neq i}^{j=n-1} \frac{(x-x_j)}{(x_i-x_j)} = \frac{(x-x_0)}{(x_i-x_0)} \cdots \frac{(x-x_{i-1})}{(x_i-x_{i-1})} \frac{(x-x_{i+1})}{(x_i-x_{i+1})} \cdots \frac{(x-x_{n-1})}{(x_i-x_{n-1})}. \tag{5}$$

Here is an example: for two data points, $(x_0, f(x_0))$ = (0, 4), $(x_1, f(x_1))$ = (-1, -3), we have

$$L_0(x) = \frac{x-x_1}{x_0-x_1} = \frac{x-(-1)}{0-(-1)} = x+1$$

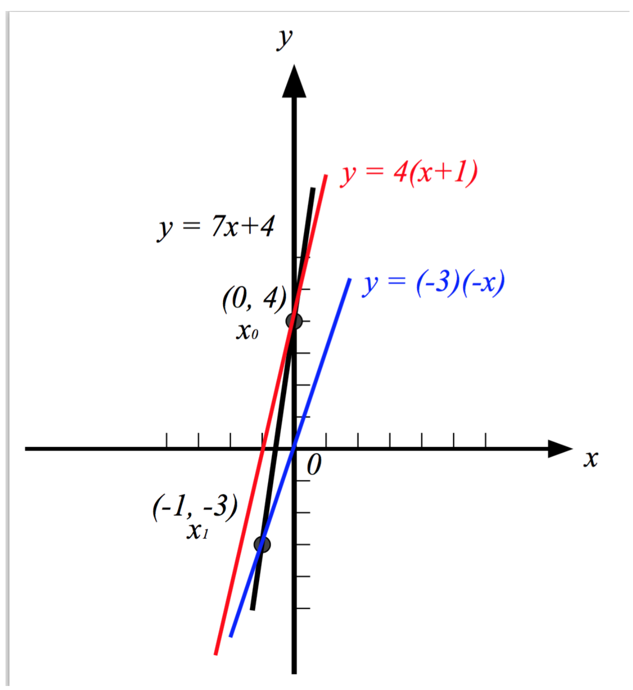
and

$$L_1(x) = \frac{x-x_0}{x_1-x_0} = \frac{x-(0)}{(-1)-(0)} = -x.$$

Then

$$P(x) = f(x_0)L_0(x) + f(x_1)L_1(x) = 4(x+1) + (-3)(-x) = 7x+4.$$

We can sketch those equations on the 2D plane as follows:



In the figure above, the red line is the 0^{th} interpolating polynomial L_0 weighted by the 0^{th} function values $f(x_0)$, $y = f(x_0)L_0 = 4(x+1)$. The blue line is the 1^{st} interpolating polynomial L_1 weighted by the 1^{st} function values $f(x_1)$, $y = f(x_1)L_1 = (-3)(-x) = 3x$. The black line is the interpolated signal, $P(x) = 7x+4$.

(a) Before we find the Lagrange interpolation, let us first use interpolation by global polynomials so we can verify our Lagrange interpolation results. Using the polynomial function basis $\{1,x,x^2,\cdots x^{n-1}\}$, the interpolation problem can be cast into finding the coefficients $a_0,a_1,a_2,\cdots,a_{n-1}$ of the function

$$g(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{n-1}x^{n-1}$$

such that $g(x_i) = f(x_i)$ for n samples of a function $(x_i, f(x_i))$ with $i \in \{0,1,2,...,n-1\}$.

Given three data points, $(2,3)$, $(0,-1)$, and $(-1,-6)$, find a polynomial $g(x) = a_2x^2 + a_1x + a_0$ fitting the three points using global polynomial interpolation. Is this polynomial unique? That is, is it the only second degree polynomial that fits this data?

It is computationally expensive to do this process for large numbers of points, which is why we use the Lagrange interpolation method.

(b) The set of Lagrange polynomials $\{L_i(x)\}$, $i \in \{0,1,2,...,n-1\}$ is a new function basis for the subspace of degree $n-1$ or lower polynomials. **Find the $L_i(x)$ given by Eq. 5 corresponding to the three sample points in (a).** Show your work.

(c) $P(x)$ is the sum of the Lagrange polynomials weighted by the function value at the corresponding points, giving the Lagrange interpolation of the given points. **Find the Lagrange polynomial interpolation $P(x)$ that goes through the three points in (a). Compare the result to the global polynomial interpolation of the same points, which you calculated in (a). Are they different from each other? Why or why not?**

(d) **Plot $P(x)$ and each $f(x_i)L_i(x)$.** You can use a plotting utility (e.g. matplotlib) and or plot by hand.

(e) **Show that $P(x_i) = f(x_i)$ for all x_i.** That is, show that the Lagrange interpolation passes through all given data points. Show this symbolically in the general case, not just for the example above.

3. Roots of Unity

The DFT is a coordinate transformation to a basis made up of roots of unity. In this problem we explore some properties of the roots of unity. An Nth root of unity is a complex number ω satisfying the equation $\omega^N = 1$ (or equivalently $\omega^N - 1 = 0$).

(a) **Show that the polynomial $z^N - 1$ factors as**

$$z^N - 1 = (z-1)\left(\sum_{k=0}^{N-1} z^k\right).$$

(b) **Show that any complex number of the form $\omega_N^k = e^{j\frac{2\pi}{N}k}$ for $k \in \mathbb{Z}$ is an N-th root of unity.** From here on, let $\omega_N = e^{j\frac{2\pi}{N}}$.

(c) For a given integer $N \geq 2$, using the previous questions, **give the complex roots of the polynomial $z \mapsto 1 + z + z^2 + ... + z^{N-1}$.**

(d) **Draw the fifth roots of unity in the complex plane. How many unique fifth roots of unity are there?**

(e) **For $N = 5$, $\omega_5 = e^{j\frac{2\pi}{5}}$. What is another expression for ω_5^{42}?**

(f) **What is the complex conjugate of ω_5? What is the complex conjugate of ω_5^{42}? What is the complex conjugate of ω_5^4?**

(g) **Write the expression for the N basis vectors $\vec{u}_0,\ldots,\vec{u}_{N-1} \in \mathbb{R}^N$ for the DFT of a signal of length N in terms of ω_N^k.**

4. DFT

In order to get practice with calculating the Discrete Fourier Transform (DFT), this problem will have you calculate the DFT for a few variations on a cosine signal.

Consider a sampled signal that is a function of discrete time $x[t]$. We can represent it as a vector of discrete samples over time $\vec{x}$, of length N.

$$\vec{x} = \begin{bmatrix} x[0] & \dots & x[N-1] \end{bmatrix}^T \tag{6}$$

Let $\vec{X} = \begin{bmatrix} X[0] & \dots & X[N-1] \end{bmatrix}^T$ be the signal $\vec{x}$ represented in the frequency domain, that is

$$\vec{X} = F_N \vec{x} \tag{7}$$

and the inverse operation is given by

$$\vec{x} = F_N^{-1} \vec{X} = \frac{1}{N} F_N^* \vec{X} \tag{8}$$

where F_N is the DFT matrix

$$F_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{-j\frac{2\pi}{N}} & e^{-j\frac{2\pi(2)}{N}} & \cdots & e^{-j\frac{2\pi(N-1)}{N}} \\ 1 & e^{-j\frac{2\pi(2)}{N}} & e^{-j\frac{2\pi(4)}{N}} & \cdots & e^{-j\frac{2\pi 2(N-1)}{N}} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & e^{-j\frac{2\pi(N-1)}{N}} & e^{-j\frac{2\pi 2(N-1)}{N}} & \cdots & e^{-j\frac{2\pi(N-1)(N-1)}{N}} \end{bmatrix}. \tag{9}$$

We sometimes call the components of $\vec{X}$ the *DFT coefficients* of the time-domain signal $\vec{x}$. We can think of the components of $\vec{X}$ as weights that represent $\vec{x}$ in the DFT basis.

Hint: Though it is possible to calculate the DFT coefficients $\vec{X}$ by doing matrix multiplication of $\vec{x}$ with F_N, you do not actually have to do this for this problem because we are considering the special case of cosine signals. Consider how you can represent the given signals in terms of the DFT basis in order to calculate the DFT representation.

Note: For this problem, you can use numpy or other calculation tools to evaluate cosines or do matrix multiplication, but you will not get credit if you directly calculate the DFT using a function in numpy. You must show your work to get credit.

(a) We define $x_1[n] = \cos\left(\frac{2\pi}{7}n\right)$ for $N = 7$ samples $n \in \{0, 1, \dots, 6\}$. **Compute the DFT coefficients $\vec{X}_1$ for signal $\vec{x}_1$.**

(b) **Plot the time domain representation of $\vec{x}_1$. Plot the magnitude, $|\vec{X}_1|$, and plot the phase, $\angle\vec{X}_1$, for the DFT representation $\vec{X}_1$.**

(c) We define $x_2[n] = \cos\left(\frac{4\pi}{7}n\right)$ for $N = 7$ samples $n \in \{0, 1, \dots, 6\}$. **Compute the DFT coefficients $\vec{X}_2$ for signal $\vec{x}_2$.**

(d) **Plot the time domain representation of $\vec{x}_2$. Plot the magnitude, $|\vec{X}_2|$, and plot the phase, $\angle\vec{X}_2$, for the DFT representation $\vec{X}_2$.**

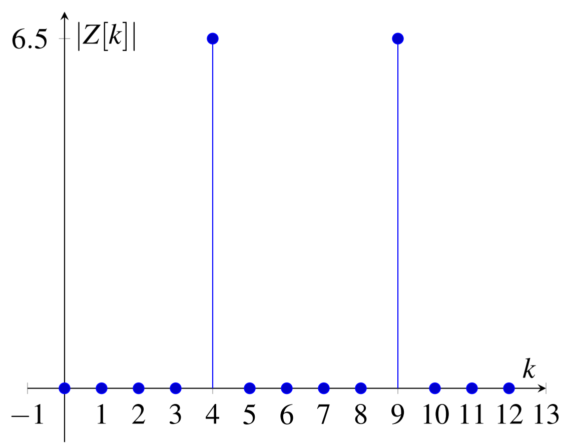
(e) To generalize this result, say we have some $p \in \{1,2,3\}$ which scales the frequency of our signal $\vec{x}_p$, which we define as $x_p[n] = \cos\left(\frac{2\pi}{7}pn\right)$ for $N = 7$ samples $n \in \{0,1,\ldots,6\}$. **Compute the DFT coefficients $\vec{X}_p$ for signal $\vec{x}_p$ in terms of this scalar p.**

(f) Let's see what happens when we have an *even* number of samples. We define $\vec{s} = \begin{bmatrix}1 & 0 & 1 & 0 & 1 & 0\end{bmatrix}^T$, which has $N = 6$ samples. **Compute the DFT coefficients $\vec{S}$ for signal $\vec{s}$.** Hint: Try to represent this signal in the form $x_p[n] = \cos\left(\frac{2\pi}{6}pn\right)$ first.

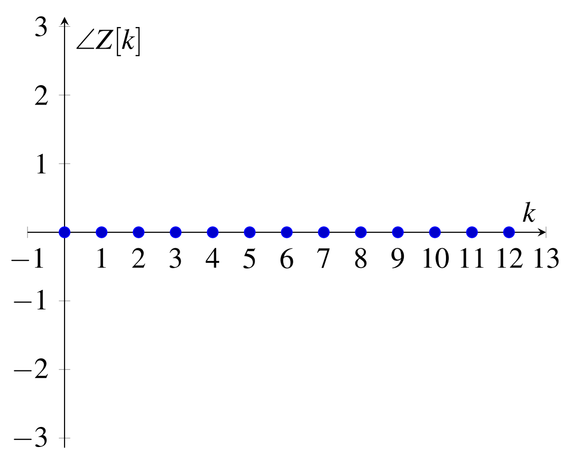
(g) Continuing with an even number of samples, we define a phase-delayed cosine signal $y[n] = \cos\left(\frac{2\pi}{6}n - \pi\right)$ for $N = 6$ samples $n \in \{0,1,\ldots,5\}$. **Compute the DFT coefficients $\vec{Y}$ for signal $\vec{y}$.**

(h) You measure the following DFT representation $\vec{Z}$ of a signal $\vec{z}$ length $N = 13$ samples:

Frequency Domain Magnitude



Frequency Domain Phase



Compute the signal $\vec{z}$ that has the DFT representation given by $\vec{Z}$.

5. DFT Projection

We can think about discrete sequences $x[n]$ with N samples as N-dimensional vectors. In this N-dimensional vector space, the DFT is a coordinate transformation, and the DFT is computed by the projecting these vectors onto a set of DFT basis vectors. The meaning of the coefficients on each of these basis vectors is the "frequency content" of the original signal $x[n]$.

(a) The DFT basis vectors $\vec{u}_k$ are given by $u_k[n] = e^{-j\frac{2\pi}{N}kn}$ for $k = 0, 1, \ldots, N-1$ and $n = 0, 1, \ldots, N-1$. **Write the basis vectors for sequences of length 2**. Use these vectors to **write the transformation matrix F_N that fulfills the change of basis equation $\vec{X} = F_N\vec{x}$** where $\vec{X}$ is the representation of $\vec{x}$ in the frequency domain.

(b) **Plot these basis vectors as discrete sequences of length 2. Plot these basis vectors as vectors in the Cartesian plane $\begin{bmatrix} x[0] & x[1] \end{bmatrix}^\top$.**

(c) **Find the basis vectors for sequences of length 3 and 4. Plot these basis vectors as discrete sequences** (you can plot the real and imaginary components as stems topped with an x instead of a dot). **What is the length (magnitude) of the basis vectors? What property of a discrete sequence does the first basis vector correspond to in every case?**

6. Write Your Own Question And Provide a Thorough Solution.

Writing your own problems is a very important way to really learn material. The famous "Bloom's Taxonomy" that lists the levels of learning is: Remember, Understand, Apply, Analyze, Evaluate, and Create. Using what you know to create is the top level. We rarely ask you any homework questions about the lowest level of straight-up remembering, expecting you to be able to do that yourself (e.g. making flashcards). But we don't want the same to be true about the highest level. As a practical matter, having some practice at trying to create problems helps you study for exams much better than simply counting on solving existing practice problems. This is because thinking about how to create an interesting problem forces you to really look at the material from the perspective of those who are going to create the exams. Besides, this is fun. If you want to make a boring problem, go ahead. That is your prerogative. But it is more fun to really engage with the material, discover something interesting, and then come up with a problem that walks others down a journey that lets them share your discovery. You don't have to achieve this every week. But unless you try every week, it probably won't ever happen.

7. Homework Process and Study Group

Citing sources and collaborators are an important part of life, including being a student! We also want to understand what resources you find helpful and how much time homework is taking, so we can change things in the future if possible.

(a) **What sources (if any) did you use as you worked through the homework?**

(b) **Who did you work on this homework with?** List names and student ID's. (In case of homework party, you can also just describe the group.)

(c) **How did you work on this homework?** (For example, *I first worked by myself for 2 hours, but got stuck on problem 3, so I went to office hours. Then I went to homework party for a few hours, where I finished the homework.*)

(d) **Roughly how many total hours did you work on this homework?**

Contributors:

- Siddharth Iyer.
- Alex Devonport.
- John Maidens.

- Geoffrey Négiar.
- Yen-Sheng Ho.
- Harrison Wang.
- Regina Eckert.
- Justin Yim.

