1 Representing linear functions as matrix multiplication
Suppose \( f : \mathbb{R}^n \rightarrow \mathbb{R}^m \) is linear. Find a matrix \( A \in \mathbb{R}^{m \times n} \) such that \( f(x) = Ax \) for all \( x \in \mathbb{R}^n \). Is the matrix \( A \) unique? Give a proof or counterexample.

2 Gradients of common functions
Recall that the gradient of a differentiable function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) at a point \( x \in \mathbb{R}^n \) is defined to be the vector

\[
\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}.
\]

The first-order Taylor expansion of \( f \) near \( x \) is given by

\[
\hat{f}_1(z) = f(x) + \nabla f(x)^T(z - x).
\]

This function is affine: that is, a linear function plus a constant offset. If \( z \) is near \( x \), then \( \hat{f}_1(z) \) is very near \( f(z) \). Find the gradients of the following functions.

(a) \( f(x) = a^T x + b \), where \( a \in \mathbb{R}^n \) and \( b \in \mathbb{R} \)

(b) \( f(x) = x^T Ax \), where \( A \in \mathbb{R}^{n \times n} \)

(c) \( f(x) = x^T Ax \), where \( A \in \mathbb{S}^n \)

3 Input/output matrix of a discrete-time linear dynamical system
Consider a discrete-time linear dynamical system:

\[
x(t + 1) = A(t)x(t) + B(t)u(t), \quad y(t) = C(t)x(t) + D(t)u(t).
\]

Find a matrix \( G \) such that

\[
\begin{bmatrix} y(0) \\ \vdots \\ y(T) \end{bmatrix} = G \begin{bmatrix} x(0) \\ u(0) \\ \vdots \\ u(T) \end{bmatrix}.
\]

The matrix \( G \) shows how the sequence of outputs, \( y(0), \ldots, y(T) \), depends on the initial state, \( x(0) \), and the sequence of inputs, \( u(0), \ldots, u(T) \).

4 A mass subject to applied forces
Consider a unit mass subject to a time-varying force \( f(t) \) for \( 0 \leq t \leq n \). Let the initial position and velocity of the mass both be zero. Suppose that the force has the form \( f(t) = x_j \) for \( j - 1 \leq t < j \) and \( j = 1, \ldots, n \). Let \( y_1 \) and \( y_2 \) denote, respectively, the position and velocity of the mass at time \( t = n \).
(a) Find a matrix $A \in \mathbb{R}^{2 \times n}$ such that $y = Ax$.

(b) For $n = 4$, find a sequence of input forces $x_1, \ldots, x_n$ that moves the mass to position 1 with velocity 0 at time $n$.

5 Counting sequences in a language or code
Consider a language or code with alphabet $\mathbb{N}_n = \{1, \ldots, n\}$. A sentence is a finite sequence of symbols, $(k_1, \ldots, k_L)$, where $k_\ell \in \mathbb{N}_n$ for all $\ell = 1, \ldots, L$. A language or code consists of a set of sequences, which we will call the set of allowable sequences. A language is called Markov if the allowed sequences can be described by giving the allowable transitions between consecutive symbols: for each symbol, we give a set of symbols that are allowed to follow that symbol. As a simple example, consider a Markov language with $n = 3$ symbols. Suppose symbol 1 must be followed by 1 or 3; symbol 2 must be followed by 3; and symbol 3 must be followed by 1 or 2. The sentence $(1, 1, 3, 2, 3, 1, 3)$ is allowable (that is, in the language) because all the transitions in this sentence are allowed; the sentence $(1, 1, 3, 2, 3, 1, 2)$ is not allowable (that is, not in the language) because the final transition in this sentence is not allowed. We can describe the allowed transitions using a matrix $A \in \mathbb{R}^{n \times n}$ with

$$A_{ij} = \begin{cases} 1 & \text{symbol } i \text{ is allowed to follow symbol } j, \\ 0 & \text{symbol } i \text{ is not allowed to follow symbol } j. \end{cases}$$

(a) Give an interpretation of $(A^p)_{ij}$ (that is, the $(i, j)$-entry of $A^p$) for $p \in \mathbb{N}$.

(b) Consider a Markov language with $n = 5$ symbols, and the following transition rules:

- 1 must be followed by 2 or 3,
- 2 must be followed by 2 or 5,
- 3 must be followed by 1,
- 4 must be followed by 2, 4 or 5,
- 5 must be followed by 1 or 3.

Find the total number of sentences of length $L = 10$. Compare this to the total number of sequences of length $L$ that can be formed from $n$ symbols (that is, the total number of sentences of length $L$ in a language with $n$ symbols, and no restrictions on the allowable transitions.)

6 Channel equalizer with disturbance rejection
A communication channel is described by the equation

$$y = Ax + v,$$

where

- $x \in \mathbb{R}^n$ is the (unknown) transmitted signal,
- $y \in \mathbb{R}^m$ is the (known) received signal,
• $v \in \mathbb{R}^m$ is the (unknown) disturbance signal, and
• $A \in \mathbb{R}^{m \times n}$ is a (known) matrix describing the channel.

Although the disturbance $v$ is unknown, we do know that $v$ is a linear combination of some (known) disturbance patterns:

$$d_1, \ldots, d_k \in \mathbb{R}^m.$$

We consider linear equalizers for the channel, which have the form $\hat{x} = By$, where $B \in \mathbb{R}^{n \times m}$ is called the equalizer (more precisely, the $B_{ij}$ are the equalizer coefficients). We say that the equalizer $B$ rejects the disturbance pattern $d_i$ if $\hat{x} = x$ for every $x \in \mathbb{R}^n$ when $v = d_i$.

Note that if the equalizer rejects a set of disturbance patterns, then it can reconstruct the transmitted signal exactly when the disturbance $v$ is a linear combination of the disturbance patterns that are rejected.

The following are defined in equalizer_disturbance_rejection_data.m:

• $n$, the size of the transmitted signal,
• $m$, the size of the received signal,
• $k$, the number of disturbance patterns,
• $A$, the $m \times n$ matrix describing the channel, and
• $D$, an $m \times k$ matrix whose columns are the disturbance patterns.

Find an equalizer $B$ that rejects as many disturbance patterns as possible. Give the specific set of disturbance patterns that your equalizer rejects. (You only need to find one equalizer that rejects a set of disturbances of maximum size.) Explain how you know that there is no equalizer that rejects more disturbance patterns than yours does. Include MATLAB code verifying all of your claims. Hint. The function nchoosek may be useful.

7 Digital circuit gate sizing

A digital circuit consists of $n$ logic gates connected by wires. Each gate has one or more inputs, and one output. The output of a gate is connected via wires to the inputs of other gates, or to external circuitry. If the output of gate $i$ is connected to an input of gate $j$, then we say that gate $i$ drives gate $j$, or that gate $j$ is in the fan-out of gate $i$. We describe the topology of the circuit by the fan-out list for each gate; $\text{FO}(i) \subseteq \{1, \ldots, n\}$ denotes the fan-out list of gate $i$. Note that it is possible that $\text{FO}(i) = \emptyset$, which indicates that the output of gate $i$ is not connected to any other gates (presumably the output of gate $i$ is connected to some external circuitry). It is common to order the gates in such a way that each gate only drives gates with higher indices – that is, we have that $\text{FO}(i) \subseteq \{i + 1, \ldots, n\}$. We will assume that the gates are ordered in this way. (Note that this implies that the gate interconnections form a directed acyclic graph.)

To illustrate the notation, consider the following simple digital circuit in figure 1.
Figure 1 – a digital circuit with $n = 4$ gates, each with 2 inputs

The fan-out lists for this circuit are

$$\text{FO}(1) = \{3, 4\}, \quad \text{FO}(2) = \{3\}, \quad \text{FO}(3) = \emptyset, \quad \text{and} \quad \text{FO}(4) = \emptyset.$$ 

The three input signals arriving from the left are called primary inputs, and the three output signals emerging from the right are called primary outputs.

Gate $i$ has a scale factor $x_i \in \mathbb{R}$ ($x_i$ is also sometimes called the size of gate $i$). These scale factors are the design variables in the gate-sizing problem. The gate sizes must satisfy $1 \leq x_i \leq x_{\text{max}}$, where $x_{\text{max}}$ is a given maximum allowed gate scale. The total area of the circuit is

$$A = \sum_{i=1}^{n} a_i x_i,$$

where the $a_i$ are known positive constants. The input capacitance $C_{i}^{\text{in}}$ of gate $i$ is

$$C_{i}^{\text{in}} = \alpha_i x_i,$$

where the $\alpha_i$ are known positive constants. The delay of gate $i$ is

$$d_i = \beta_i + \frac{\gamma_i C_{i}^{\text{load}}}{x_i},$$

where $\beta_i$ and $\gamma_i$ are known positive constants, and $C_{i}^{\text{load}}$ is the load capacitance of gate $i$. Note that the gate delay $d_i$ is always greater than $\beta_i$, which we can think of as the minimum possible delay of gate $i$, achieved in the limit as the gate scale factor becomes large. The load capacitance of gate $i$ is

$$C_{i}^{\text{load}} = C_{i}^{\text{ext}} + \sum_{j \in \text{FO}(i)} C_{j}^{\text{in}},$$

where $C_{i}^{\text{ext}}$ is a known positive constant that accounts for the capacitance of the interconnect wires and external circuitry.

We want to design the circuit so that every gate has delay $T$ for some value of $T > 0$. For a given value of $T$, there may or may not be a feasible design choice (that is, a choice of $x_1, \ldots, x_n$ such that $1 \leq x_i \leq x_{\text{max}}$ and $d_i = T$ for $i = 1, \ldots, n$). For example, we must have $T > \max_i \beta_i$. 

Alex Lemon  
July 2, 2015  
EE263: Introduction to Linear Dynamical Systems  
Homework 1: Page 4 of 5
(a) Explain how to find a design $x^* \in \mathbb{R}^n$ that minimizes $T$ subject to the area constraint $A \leq A^{\text{max}}$, where $A^{\text{max}}$ is a given constant. Your value of $T$ only needs be accurate to two decimal places. You may assume that the fan-out lists, and all constants are known; your job is just to find the scale factors $x_i$. Be sure to explain how you can determine if the design problem is feasible.

(b) Carry out your method on the data given in `gate_sizing_data.m`. The fan-out lists are described by an $n \times n$ matrix $F$, where $F(i,j) = 1$ if $j \in \text{FO}(i)$, and $F(i,j) = 0$ otherwise.

8 Interpolation with rational functions.

Consider a function $f: \mathbb{R} \to \mathbb{R}$ of the form

$$f(x) = \frac{a_0 + a_1 x + \cdots + a_m x^m}{1 + b_1 x + \cdots + b_m x^m},$$

where $a_0, \ldots, a_m$ and $b_1, \ldots, b_m$ are parameters, with either $a_m \neq 0$ or $b_m \neq 0$. Such a function is called a rational function of degree $m$. We are given data points $x_1, \ldots, x_N \in \mathbb{R}$, and $y_1, \ldots, y_N \in \mathbb{R}$, where $y_i = f(x_i)$.

(a) Explain how to find a rational function of smallest degree that is consistent with the data: that is, explain how to find the smallest value of $m$, and corresponding values of $a_0, \ldots, a_m$, and $b_1, \ldots, b_m$ such that $f(x_i) = y_i$ for $i = 1, \ldots, N$.

(b) Carry out your method on the data in `rational_interpolation_data.m`. Report your value of $m$, and the corresponding coefficients $a_0, \ldots, a_m$, and $b_1, \ldots, b_m$. Plot the data and the rational function $f(x)$. Verify that $y_i = f(x_i)$ for $i = 1, \ldots, N$ (possibly with small numerical errors).