Homework 01

Due: January 17, 9PM
Point total: 60

Instructions:

- Submit your PDF and .py file to Blackboard by the due date and time. Please do not zip your files together, as this interferes with Blackboard's preview functionality. Always show all your work, and for full credit, you must use the method that the problem instructs you to use (unless none is mentioned). Handwritten or typeset solutions are both acceptable, but unreadable submissions will be penalized. You may discuss problems with other students, but you may not write up solutions together, copy solutions from a common whiteboard, or otherwise share your written work or code. Do not use code or language that is copied from the Internet or other students; attribute the ideas *and* rephrase in your own words.

Problem 1 (15 points)

For each system of equations, write the corresponding augmented matrix. Then solve for the solution *using Gauss-Jordan elimination*, describing what you are doing with each step (for example, "row1 = row1 - 2*row2"). To avoid rewriting the same numbers over and over, you can perform multiple steps at a time as long as each step affects a different row. If there is no solution, say so; you can stop as soon as this fact is apparent from Gauss-Jordan elimination. If there are multiple solutions, give equations describing the solutions in terms of z.

i. $x + y + z = 3$
$2x + y - z = 8$
$-x - y + z = -3$

ii. $x + y = -1$
$-y + z = 2$
$x + z = 1$

iii. $w + y + z = 1$
$w + x + y + z = 7$
$y + z = 4$
$w + x = 4$

Problem 2 (8 points)

For each system of equations in problem 1, determine whether the corresponding homogeneous matrix is singular or not, and explain how we know.

Problem 3 (10 points)

Suppose we have a small online game with a userbase consisting of three kinds of user experience levels: "hardcore," "intermediate," and "just starting." We can represent the counts of these three kinds of users with the letters h, i, and j. From month to month, we estimate that the "just starting" users are growing 20 percent, but that every "hardcore" player causes a "just starting" player to leave. Otherwise, users tend to either stop playing or graduate to the next level of user.

Our modeling department suggests the following linear equations to describe this population's change from month t to month $t+1$:

$$\begin{aligned} h_{t+1} &= 0.95h_t + 0.1i_t \\ i_{t+1} &= 0.8i_t + 0.2j_t \\ j_{t+1} &= -h_t + 1.2j_t \end{aligned}$$

i. Write this linear system as a matrix. (Columns 1, 2, and 3 should correspond to h, i, and j, respectively, and the order of the rows should match the order of the equations.)

ii. Recently, the intermediate players mostly fled after the game changed, leaving 10000 hardcore players, 10000 just starting players, and only 10 intermediate players. Use matrix multiplication to predict the populations one month later.

iii. Perform one more matrix multiplication. What could be done to the transformation from one month to the next to make this prediction more realistic? Is what you suggest a linear operation? (Multiple answers are possible here.)

Problem 4 (12 points)

A common problem in the field of information retrieval is to try to identify documents that are similar to each other. If we want to use methods that are related to linear algebra, but the documents are in English, then we need to first convert those documents (composed of words) to vectors (composed of numbers). Whatever method we use should cause similar documents to turn into vectors that point in similar directions.

One way to achieve this is to have each word correspond to a different dimension of a vector. A document that contains that word could contain a 1 for that dimension, while a document that does not contain the word would contain a zero for that dimension. In the simpler versions of these methods, word order doesn't matter; just the presence or absence of the word does.

For simplicity, let's assume our product reviews on a website just consist of the words "good" (dimension 1), "works" (dimension 2), "terrible" (dimension 3), "car" (dimension 4), and "vacuum" (dimension 5). Thus, "good car" would be the vector $\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ and "terrible vacuum" would be the vector $\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$.

Using this system, and the formulas for the dot product, find the angle between each pair of sentences. Be exact if possible (if using radians, give any nonzero answer in terms of π) and assume angles are between 0 and π inclusive.

i. "good car works" vs "terrible vacuum"

ii. "good car" vs "car works"

iii. "terrible vacuum" vs "vacuum works"

Problem 5 (15 points)

For this part, you'll learn to use some Python. The instructions for both the tutorial and the graded part of Problem 5 are in Jupyter notebook files, which are a mix of instructions and executable code.

First, install the Anaconda distribution of Python 3, available at https://www.anaconda.com/distribution/. If you already have a Python install, you might want to do this anyway so that we're all on the same page.

Next, open up a Terminal window (Mac) or Command Prompt (Windows) and type

```
jupyter notebook
```

From there, open up the tutorial part of the assignment (`hw01_program_tutorial.ipynb`), and walk through it to explore the parts of Python that we'll be using. After that, you can open the other notebook (`hw01_program.ipynb`), which contains the actual programming assignment (about repeatedly multiplying matrices).

For this assignment, turn problems 1-4 in to Blackboard in a single PDF, and also turn in problem 5 as a separate .py file. The tutorial doesn't have anything you need to turn in, but is recommended regardless.