EECS 16A Spring 2023 Designing Information Devices and Systems I Exam Prep 4A

1. Information Storage (Spring 2022 Midterm 1 Question 3)

(a) Your team plans to build a database that stores information as vectors $\vec{v}_s \in \mathbb{R}^3$. Due to system constraints, Ayush, an engineer on your team, mentions that it'll be easiest to store these vectors as linear combinations of

$$\vec{w}_1 = \begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix}, \vec{w}_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \vec{w}_3 = \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix}$$

For each of the following vectors $\vec{v}_i$, state if it can be written as a linear combination of $\vec{w}_1, \vec{w}_2, \vec{w}_3$. If so, find the coefficients. If not, explain why.

$$\vec{v}_1 = \begin{bmatrix} 0 \\ 4 \\ 4 \end{bmatrix}, \vec{v}_2 = \begin{bmatrix} 0 \\ -2 \\ -2 \end{bmatrix}, \vec{v}_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

(b) Consider a matrix $\mathrm{M} \in \mathbb{R}^{3\times 3}$ formed by the column vectors $\vec{v}_1, \vec{v}_2, \vec{v}_3$ from part (a). What is the rank of M?

(c) To manipulate the vectors in your database, you can multiply them by $\mathrm{M} \in \mathbb{R}^{3\times 3}$. Suppose $\vec{v_j}$ is some vector in the database and

$$\mathrm{M} = \begin{bmatrix} 3 & -1 & 5 \\ 0 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix}.$$

We generate $\vec{v_{new}} = \mathrm{M}\vec{v_j}$. Can Dahlia find a new matrix P that reverses this operation such that $\mathrm{P}\vec{v_{new}} = \vec{v_j}$? If so, find this new matrix. If not, why not?

(d) Now given a new matrix

$$\mathrm{M}_{\text{new}} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 1 \\ 0 & 2 & \alpha \end{bmatrix}$$

what value of α makes M_{new} have a rank of 2?

2. A Problem N(o)-body Can Solve (Spring 2021 Midterm 1 Question 7)

An N-body simulation is a method of modeling the interactions between a set of particles, and it is commonly implemented in an astrophysics context to study the movements of celestial bodies and galactic formation under the constraints of gravitational forces. In each timestep, the core algorithm iterates through particle pairs to calculate the force on each particle and update its current position.

As part of your work in a research lab, you are developing an efficient N-body simulation for the solar system that exploits computationally fast operations on matrices to speed up runtime (good thing you're taking EECS16A!). You represent each body as a vector in 3D space:

$$\vec{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

You calculate that the position of one particular body – Earth – is updated in the following way during every timestep:

- $x[t+1] = 0.5x[t] + 0.7y[t] + 0.3z[t]$
- $y[t+1] = 0.6y[t] + 0.1z[t]$
- $z[t+1] = 0.3x[t] + 0.2y[t] + z[t]$

(a) After one timestep, at time $t = 1$, Earth is located at $\begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$. You want to calculate the position of Earth at $t = 0$. **Formulate this problem as a matrix-vector equation in the form $A\vec{x} = \vec{b}$.** You do not need to solve for Earth's position.

(b) You have determined that Neptune's position in each timestep is updated according to the following matrix:

$$\mathbf{N} = \begin{bmatrix} 1 & 0.2 & 0 \\ 0 & -0.2 & 0.1 \\ -1 & 0 & 0.1 \end{bmatrix}$$

You let the simulation run in the background for a while, and at $t = n$, Neptune is located at $\begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix}$. You then realize that you've forgotten to record position data since you started! **Is it possible to recover Neptune's position uniquely at $t = n - 1$?** If it is, **use Gaussian elimination to find the inverse of $\mathbf{N}$, $\mathbf{N}^{-1}$.**

(c) Additionally, you have the following matrix for Pluto:

$$\begin{bmatrix} 0.1 & 0 & 0.1 \\ 0.1 & 0.1 & 0.2 \\ 0.2 & 0.1 & 0.3 \end{bmatrix}$$

If Pluto is positioned at some unspecified $\begin{bmatrix} x[0] \\ y[0] \\ z[0] \end{bmatrix}$ at $t = 0$, **are there any points in $\mathbb{R}^3$ space that you cannot reach at at $t = 1$? If so, what is the subspace that Pluto can be located in?**

Note: You do not have to provide rigorous justification.

(d) After running your simulation repeatedly, you notice that with the current update matrices you have entered, Venus and Mars are all moving within the same 2D orbital plane. You refer back to your calculations, but you notice there is a smudge obscuring one element of matrix **M**:

$$\mathbf{V} = \begin{bmatrix} 0.3 & 0.4 & 0.1 \\ 0 & 0.7 & 0.7 \\ 0.7 & 1.1 & 0.4 \end{bmatrix}, \mathbf{M} = \begin{bmatrix} 0.4 & 0.6 & 0.2 \\ -1.4 & 0 & 1.4 \\ 0.6 & m_{32} & 0.8 \end{bmatrix}$$

V and **M** are the update matrices for Venus and Mars respectively. **Fill in the missing matrix element (denoted by "m_{32}") in a way that would explain the behavior of these 2 planets.**

