EECS 16A Designing Information Devices and Systems I
Fall 2020 Midterm 1

Midterm 1 Solution

1. **HONOR CODE**

   If you have not already done so, please copy the following statements into the box provided for the honor code on your answer sheet, and sign your name.

   I will respect my classmates and the integrity of this exam by following this honor code. I affirm:

   - *I have read the instructions for this exam. I understand them and will follow them.*
   - *All of the work submitted here is my original work.*
   - *I did not reference any sources other than my one reference cheat sheet.*
   - *I did not collaborate with any other human being on this exam.*

2. (a) **Tell us about something you did in the last year that you are proud of.** *All answers will be awarded full credit; you can be brief.* **(2 Points)**

   (b) **Name someone that inspires you.** *All answers will be awarded full credit; you can be brief.* **(2 Points)**

3. Sailing for climate change (22 points)

You are a member of Greta Thunberg's crew that embarks on a new transatlantic boat journey, advocating for climate change awareness and environmentally friendly travel.

(a) (2 points) In the environmentally friendly mode, the ship travels using only the wind and sail; as a result you have to move in direction of the wind.

In this mode, the next location of the ship is given by $\vec{\ell}_{next} = \mathbf{A} \cdot \vec{\ell}_{curr}$, where $\vec{\ell}_{curr} = \begin{bmatrix} 2 \\ 5 \\ 0 \end{bmatrix}$ is the current location of the ship. $\mathbf{A}$ is given as:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Find your next location vector $\vec{\ell}_{next}$ after you apply the steering matrix A once to $\vec{\ell}_{curr}$.

Solution:

Your next location can be calculated by multiplying the steering matrix with your current location so:

$$\vec{\ell}_{next} = \mathbf{A} \cdot \vec{\ell}_{curr} = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 5 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \cdot 2 + 2 \cdot 5 + 0 \cdot 0 \\ 3 \cdot 2 + 6 \cdot 5 + 0 \cdot 0 \\ 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 \end{bmatrix} = \begin{bmatrix} 12 \\ 36 \\ 0 \end{bmatrix}$$

(b) (2 points) **What is the dimension of the columnspace of A? Explain your answer and/or show your work.**

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Solution: Observe that the columns of the steering matrix $\mathbf{A}$ are linearly dependent since $\text{col}_2 = 2 \times \text{col}_1$. This means that the column space is equal to: $C(\mathbf{A}) = \text{span}\left\{ \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix} \right\}$. We can disregard the last column as it is a column of zeros.

Therefore, the column space of $\mathbf{A}$ has dimension 1, and forms a line.

(c) (6 points) A plane is monitoring the journey from a high altitude and wants to come closer to Greta's ship at a certain meeting point, $\vec{\ell}_{meet}$.

The plane's current location is described by $\vec{\ell}_{plane} = \begin{bmatrix} 0 \\ 20 \\ 100 \end{bmatrix}$, and the goal is to hover at the meeting point described by $\vec{\ell}_{meet} = \begin{bmatrix} 20 \\ 20 \\ 50 \end{bmatrix}$. You have been asked to design a steering matrix $\mathbf{B}$ for the plane so that

$$\vec{\ell}_{meet} = \mathbf{B} \cdot \vec{\ell}_{plane}.$$

Some of the entries of the matrix $\mathbf{B}$ are given for you below; however, **you must find the elements** p, q **and** r. Please draw a box around your answers for p, q and r.

$$\mathbf{B} = \begin{bmatrix} 10 & p & 1 \\ 20 & 6 & q \\ 7 & r & -1 \end{bmatrix}.$$

Solution:

The system of equations we need to solve is:

$$\mathbf{B} \cdot \vec{\ell}_{plane} = \vec{\ell}_{meet}$$

$$\begin{bmatrix} 10 & p & 1 \\ 20 & 6 & q \\ 7 & r & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 20 \\ 100 \end{bmatrix} = \begin{bmatrix} 20 \\ 20 \\ 50 \end{bmatrix}$$

The system of equations we can write from the above matrix formulation is:

$$\begin{cases} 10 \cdot 0 + p \cdot 20 + 1 \cdot 100 & = 20 \\ 20 \cdot 0 + 6 \cdot 20 + q \cdot 100 & = 20 \\ 7 \cdot 0 + r \cdot 20 + (-1) \cdot 100 & = 50 \end{cases}$$

Solving this system, we arrive at the following solution: $p = -4, \quad q = -1, \quad r = 7.5.$

(d) (4 points) **For this subpart and the following subparts, you can assume that your ship travels only in 2D.**

Your current location is given by $\vec{\ell}_{curr} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$. Steering matrix $\mathbf{C}$ was multiplied once with your previous location vector, $\vec{\ell}_{prev}$, to bring you to $\vec{\ell}_{curr}$, so you know $\vec{\ell}_{curr} = \mathbf{C} \cdot \vec{\ell}_{prev}$. The matrix $\mathbf{C}$ is:

$$\mathbf{C} = \begin{bmatrix} 1 & 1 \\ 3 & 6 \end{bmatrix}.$$

Find $\vec{\ell}_{prev}$, your previous location vector.

Solution:

There are two acceptable solutions for this answer: 1. to calculate the inverse first, and then multiply the inverse matrix with the current state vector; or 2. to formulate the problem as $\mathbf{C} \cdot \vec{\ell}_{prev} = \vec{\ell}_{curr}$ and row reduce to calculate $\vec{\ell}_{prev}$.

For the first approach, we need to calculate the inverse matrix as follows:

$$\vec{\ell}_{prev} = \mathbf{C}^{-1} \vec{\ell}_{curr}.$$

Using Gaussian Elimination we can get:

$$\left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 3 & 6 & 0 & 1 \end{array}\right]$$
$$\rightarrow \left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 0 & 3 & -3 & 1 \end{array}\right] R_2 \leftarrow R_2 - 3R_1$$
$$\rightarrow \left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 0 & 1 & -1 & \frac{1}{3} \end{array}\right] R_2 \leftarrow \frac{1}{3}R_2$$
$$\rightarrow \left[\begin{array}{cc|cc} 1 & 0 & 2 & -\frac{1}{3} \\ 0 & 1 & -1 & \frac{1}{3} \end{array}\right] R_1 \leftarrow R_1 - R_2$$

So we will have that:

$$\vec{\ell}_{prev} = \begin{bmatrix} 2 & -\frac{1}{3} \\ -1 & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

In the second approach, we augment our matrix accordingly and row reduce:

$$\left[\begin{array}{cc|c} 1 & 1 & 2 \\ 3 & 6 & 3 \end{array}\right]$$
$$\rightarrow \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 3 & -3 \end{array}\right] R_2 \leftarrow R_2 - 3R_1$$
$$\rightarrow \left[\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & -1 \end{array}\right] R_2 \leftarrow \frac{1}{3}R_2$$
$$\rightarrow \left[\begin{array}{cc|c} 1 & 0 & 3 \\ 0 & 1 & -1 \end{array}\right] R_1 \leftarrow R_1 - R_2$$

Again, we see that the previous location vector is:

$$\vec{\ell}_{prev} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$$

(e) (4 points) On the distant horizon, you see three other ships approaching Greta's ship. The current locations of the three ships are given by the column vectors: $\begin{bmatrix} 5 \\ 9 \end{bmatrix}$, $\begin{bmatrix} 4 \\ 12 \end{bmatrix}$, $\begin{bmatrix} 3 \\ 4 \end{bmatrix}$. You stack the ships' locations together to make the matrix $\mathbf{D} = \begin{bmatrix} 5 & 4 & 3 \\ 9 & 12 & 4 \end{bmatrix}$.

The winds driving these ships transform each ship location by the matrix $\mathbf{E} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

Instead of computing their final locations one-by-one, you want one matrix that gives the locations of the three ships, where the two rows of this matrix will contain the x- and y-coordinates of the ships, respectively.

Set this problem up as a matrix-matrix multiplication and find the matrix representing the final locations of all three ships.

Solution:

The 2×3 matrix corresponding to the ships' current location vectors is constructed as a matrix $\mathbf{D}$:

$$\mathbf{D} = \begin{bmatrix} 5 & 4 & 3 \\ 9 & 12 & 4 \end{bmatrix}$$

Note that students may put the column vectors together in varying order; as long as the order matches the ordering of the final location vectors in the answer, full credit should be awarded.

The matrix-matrix product is calculated as follows:

$$\begin{aligned} \mathbf{E} \cdot \mathbf{D} &= \begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} 5 & 4 & 3 \\ 9 & 12 & 4 \end{bmatrix} \\ &= \begin{bmatrix} 5a+9b & 4a+12b & 3a+4b \\ 5c+9d & 4c+12d & 3c+4d \end{bmatrix} \end{aligned}$$

(f) (4 points) Your current location vector is now given as $\vec{\ell}_{curr} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$. Your would like to steer Greta's ship to $\vec{\ell}_{final} = \begin{bmatrix} 0 \\ -10 \end{bmatrix}$. You notice that the ship has several built-in 2×2 steering matrices which you can apply in any order to your current location vector:

- $\mathbf{T}_\theta$, which rotates your current location vector by 90° counterclockwise;
- $\mathbf{T_2}$, which multiplies your current location vector by a factor of 2;
- $\mathbf{T_x}$, which reflects your current location vector across the x-axis;
- $\mathbf{T_y}$, which reflects your current location vector across the y-axis.

You would like to only use these built-in steering matrices to direct Greta's ship to its final goal.

Which of the following options would successfully get you from $\vec{\ell}_{curr}$ to $\vec{\ell}_{final}$? Only one option is correct.

No partial credit will be awarded if any incorrect options are selected. You do not need to show any work for this problem.

(i) $\vec{\ell}_{final} = \mathbf{T}_\theta \mathbf{T_2} \mathbf{T_x} \vec{\ell}_{curr}$

(ii) $\vec{\ell}_{final} = \mathbf{T_x} \mathbf{T_2} \mathbf{T}_\theta \vec{\ell}_{curr}$

(iii) $\vec{\ell}_{final} = \mathbf{T_y} \mathbf{T_2} \mathbf{T_x} \vec{\ell}_{curr}$

(iv) $\vec{\ell}_{final} = \mathbf{T_y} \mathbf{T_x} \mathbf{T_2} \vec{\ell}_{curr}$

Solution:

The only correct choice among the options provided is **option (ii)**. Applying $\mathbf{T}_\theta$ brings $\vec{\ell}_{curr}$ to $[0 \quad 5]^\top$; applying $\mathbf{T_2}$ next brings us to $[0 \quad 10]^\top$. Finally, applying $\mathbf{T_x}$ last brings us to $\vec{\ell}_{final} = [0 \quad -10]^\top$.

Option (i) transforms $\vec{\ell}_{curr}$ to $[5 \quad 0]^\top$ by $\mathbf{T_x}$; transforms to $[10 \quad 0]^\top$ by $\mathbf{T_2}$, then to $\vec{\ell}_{final} = [0 \quad 10]^\top$ by $\mathbf{T}_\theta$.

Option (iii) transforms $\vec{\ell}_{curr}$ to $[5 \quad 0]^\top$ by $\mathbf{T_x}$; transforms to $[10 \quad 0]^\top$ by $\mathbf{T_2}$, then to $\vec{\ell}_{final} = [-10 \quad 0]^\top$ by $\mathbf{T_y}$.

Option (iv) transforms $\vec{\ell}_{curr}$ to $[10 \quad 0]^\top$ by $\mathbf{T_2}$; transforms to $[10 \quad 0]^\top$ by $\mathbf{T_x}$, then to $\vec{\ell}_{final} = [-10 \quad 0]^\top$ by $\mathbf{T_y}$.

4. Mixing Paints (21 points)

Over quarantine, you've really gotten into painting, but you are running low on paints.You would like to mix the paints you have to make different colors.

Notation: **Every** color can be represented as a length-3 vector, $\vec{c} = \begin{bmatrix} c_r \\ c_y \\ c_b \end{bmatrix}$, where c_r, c_y, and c_b represent the number of bottles of red, yellow, and blue paint, respectively, to make the color.

(a) (5 points) You want to make the color brown, $\vec{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, using a linear combination of turquoise, magenta, and peach paints (the only ones you have left). We are given the following information about paint compositions:

- 1 bottle of turquoise is made by combining 0 bottles red, 0.4 bottles yellow, and 0.6 bottles blue.
- 1 bottle of magenta is made by combining 0.5 bottles red, 0 bottles yellow, and 0.5 bottles blue.
- 1 bottle of peach is made by combining 0.5 bottles red, 0.4 bottles yellow, and 0.1 bottles blue.

You would like to find the number of bottles of turquoise (x_t), magenta (x_m), and peach (x_p) to mix to make brown. Formulate this problem as a matrix-vector equation. You do not need to solve the equation.

Solution:

$$\mathbf{A}\vec{x} = \vec{b}$$

$$\begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.4 & 0 & 0.4 \\ 0.6 & 0.5 & 0.1 \end{bmatrix} \begin{bmatrix} x_t \\ x_m \\ x_p \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

(b) (5 points) Your friend wants to make the color brown but has a different set of colors. She also sets up a system of equations to make brown and ends up with $\mathbf{A}\vec{x} = \vec{b}$ where

$$\mathbf{A} = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{bmatrix}, \vec{b} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Use Gaussian Elimination to solve for $\vec{x}$, which represents the mixture of paints your friend must combine to make brown. Show your work.

Solution: Our system of equations is given by:

$$\begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0 \end{bmatrix} \vec{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

which we can express in augmented matrix form as:

$$\left[\begin{array}{ccc|c} 0 & 0.5 & 0.5 & 1 \\ 0.5 & 0 & 0.5 & 1 \\ 0.5 & 0.5 & 0 & 1 \end{array}\right]$$

For simplicity, we multiply all rows by 2 to remove the fractions and then proceed with Gaussian Elimination:

$$\begin{aligned}
&\left[\begin{array}{ccc|c} 0 & 1 & 1 & 2 \\ 1 & 0 & 1 & 2 \\ 1 & 1 & 0 & 2 \end{array}\right] \\
\to &\left[\begin{array}{ccc|c} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 2 \\ 1 & 1 & 0 & 2 \end{array}\right] \text{swap } \mathrm{R}_1, \mathrm{R}_2 \\
\to &\left[\begin{array}{ccc|c} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & -1 & 0 \end{array}\right] R_3 \leftarrow R_3 - R_1 \\
\to &\left[\begin{array}{ccc|c} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & -2 & -2 \end{array}\right] R_3 \leftarrow R_3 - R_2 \\
\to &\left[\begin{array}{ccc|c} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 1 \end{array}\right] R_3 \leftarrow -\frac{1}{2}R_3 \\
\to &\left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{array}\right] R_1 \leftarrow R_1 - R_3; R_2 \leftarrow R_2 - R_3
\end{aligned}$$

So we arrive at the solution $\vec{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$.

(c) (6 points) You now want to make lots of colors. Instead of solving the system $\mathbf{A}\vec{x} = \vec{c}$ every time we want to make a new color $\vec{c}$, you want to find the inverse matrix $\mathbf{A}^{-1}$. For this subpart, use the following matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix}.$$

Use Gaussian Elimination to determine if $\mathbf{A}^{-1}$ exists. If $\mathbf{A}^{-1}$ exists, give its value.

Solution: We start with the following augmented matrix.

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{I} \end{array}\right] = \left[\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 & 0 & 1 \end{array}\right]$$

$$\rightarrow \left[\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & -1 & 0 & 1 \end{array}\right] R_3 \leftarrow R_3 - R_1$$

$$\rightarrow \left[\begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 & 1 \end{array}\right] R_3 \leftarrow R_3 + R_2$$

$$\rightarrow \left[\begin{array}{ccc|ccc} 1 & 1 & 0 & 2 & -1 & -1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 & 1 \end{array}\right] R_1 \leftarrow R_1 - R_3$$

$$\rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 2 & -2 & -1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 & 1 \end{array}\right] R_1 \leftarrow R_1 - R_2$$

$$\left[\begin{array}{c|c} \mathbf{I} & \mathbf{A}^{-1} \end{array}\right] = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 2 & -2 & -1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 & 1 \end{array}\right]$$

Thus the inverse matrix $\mathbf{A}^{-1}$ exists and is given by:

$$\mathbf{A}^{-1} = \begin{bmatrix} 2 & -2 & -1 \\ 0 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix}$$

(d) (5 points) We would like to determine what colors we can make just given three other colors (purple, orange, gray). These are represented by a matrix $\mathbf{D} \in \mathbb{R}^{3\times3}$, which is invertible, but is otherwise unknown to you. You cannot mix together negative amounts of paint.

Consider the set of all colors

$$\mathbb{S} = \left\{ \vec{c} \in \mathbb{R}^3 \mid \quad \mathbf{D}\vec{x} = \vec{c},\ \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix},\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \right\}.$$

that we can get from mixing together non-negative quantities of the three colors. **Is $\mathbb{S}$ a vector subspace of $\mathbb{R}^3$? Justify your answer.**

Solution:

Brief Argument

We know that for $\vec{c}_1 \in \mathbb{S}$, there exists some $\vec{x}_1$ such that $\vec{c}_1 = \mathbf{D}\vec{x}_1$.

Let's take a second vector $\vec{c}_2 = -\vec{c}_1$. The question we want to answer: for any $\vec{c}_1 \in \mathbb{S}$, is $\vec{c}_2 \in \mathbb{S}$ also?

Assume for now that it is (we will show this assumption is a contradiction.)

Since we're assuming $\vec{c}_2 \in \mathbb{S}$, we can write:

$$\begin{aligned} \vec{c}_2 &= \mathbf{D}\vec{x}_2 \\ -\vec{c}_1 &= \mathbf{D}\vec{x}_2 \\ \vec{c}_1 &= -\mathbf{D}\vec{x}_2 \end{aligned}$$

Note that because $\mathbf{D}$ is invertible, there is only one unique $\vec{x}_2$ that satisfies this. Therefore, we can rewrite the left- and right-hand sides as:

$$\mathbf{D}\vec{x}_1 = \mathbf{D}(-\vec{x}_2)$$

i.e.

$$\vec{x}_2 = -\vec{x}_1$$

is the **only** solution to $-\vec{c}_1 = \mathbf{D}\vec{x}_2$. Note that $\mathbf{D}$ being invertible is crucial – otherwise, $\vec{x}_2 = -\vec{x}_1$ would not be guaranteed to be the only solution for any arbitrary $\vec{c}_1$.

Suppose we have chosen a non-zero $\vec{c}_1$ such that $\vec{x}_1$ has elements $x_1, x_2, x_3 \geq 0$ with at least one of x_1, x_2, x_3 being strictly positive. Then $\vec{x}_2 = -\vec{x}_1$ has at least one element that is strictly negative. But our set requires all $\vec{x}$ vectors to only have non-negative elements. Therefore, we cannot generate $-\vec{c}_1$ using $\vec{x}_2 = -\vec{x}_1$. By the invertibility of D, there are no other possible $\vec{x}$ vectors to generate $-\vec{c}_1$ for any arbitrary $\vec{c}_1 \in \mathbb{S}$.

This exposes a contradiction in our assumption: $\vec{c}_2 = -\vec{c}_1 \notin \mathbb{S}$ for any arbitrary $\vec{c}_1 \in \mathbb{S}$.

For example, if we multiply a non-zero vector $\vec{c}$ that is in our set $\mathbb{S}$ by -1, the resulting vector will not be in $\mathbb{S}$. This means $\mathbb{S}$ is not closed under scalar multiplication, and therefore it is not a subspace.

Detailed Argument

We determine if $\mathbb{S}$ is a vector subspace by checking the properties it must satisfy:

i. $\vec{0} \in \mathbb{S}$

ii. $\vec{c}_1, \vec{c}_2 \in \mathbb{S} \Rightarrow \vec{c}_1 + \vec{c}_2 \in \mathbb{S}$

iii. $\alpha \in \mathbb{R}, \vec{c} \in \mathbb{S} \Rightarrow \alpha\vec{c} \in \mathbb{S}$

The key property that is violated is the third one, i.e. the subspace must be closed under scalar multiplication.

Let's pick a non-zero color vector $\vec{c} \in \mathbb{S}, \vec{c} \neq \vec{0}$. It was produced by a unique combination of purple, orange, and gray; i.e. there exists a unique solution $\vec{x} \in \mathbb{R}^3$ to the system $\mathbf{D}\vec{x} = \vec{c}$ such that x_1, x_2, x_3 are non-negative and at least one of x_1, x_2, x_3 is strictly positive.

The uniqueness of this solution can be shown due to the fact that $\mathbf{D}$ is invertible, and we note that

$$\vec{x} = \mathbf{D}^{-1}\vec{c}.$$

Now, let's try multiplying $\vec{c}$ by a scalar $\alpha \in \mathbb{R}$ and check if the resulting vector $\alpha\vec{C} \in \mathbb{S}$.

If $\alpha\vec{c} \in \mathbb{S}$, that would imply that there exists a unique $\vec{z} \in \mathbb{R}^3$ to the system $\mathbf{D}\vec{z} = \alpha\vec{c}$ such that $z_1, z_2, z_3 \geq 0$.

$$\begin{aligned}
&\mathbf{D}\vec{z} = \alpha\vec{c} \\
&\Rightarrow \vec{z} = \alpha\mathbf{D}^{-1}\vec{c} \\
&\Rightarrow \vec{z} = \alpha\vec{x}
\end{aligned}$$

Elementwise, the final statement implies that:

$$\begin{aligned}
z_1 &= \alpha x_1 \\
z_2 &= \alpha x_2 \\
z_3 &= \alpha x_3
\end{aligned}$$

Without loss of generality, let us assume that $x_1 > 0$. Then choosing $\alpha = -1$ would indicate that $z_1 < 0$, which is a contradiction to $\vec{z}$ being non-negative.

To summarize, we have shown the set $\mathbb{S}$ is not closed under scalar multiplication and therefore it is not a subspace.

5. Basis and Linear Independence (18 points)

Let $\mathbb{V}$ be a subspace of $\mathbb{R}^4$ defined as

$$\mathbb{V} = \left\{ \vec{x} \in \mathbb{R}^4 \;\middle|\; \vec{x} = \begin{bmatrix} \alpha \\ \beta \\ \alpha+\beta \\ \alpha-\beta \end{bmatrix}, \text{ where } \alpha, \beta \in \mathbb{R} \right\}.$$

(a) (4 points) **Find a basis for $\mathbb{V}$ and determine the dimension of $\mathbb{V}$. Show your work.**

Solution:

$\vec{x}$ can be written as:

$$\vec{x} = \alpha \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix}.$$

Therefore, a basis for $\mathbb{V}$ is:

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \\ -1 \end{bmatrix} \right\}.$$

The dimension of $\mathbb{V}$ is 2.

(b) (4 points) Given $\vec{v}_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$ and $\vec{v}_2 = \begin{bmatrix} 2 \\ 5 \\ 3 \end{bmatrix}$, **determine if the set $\{\vec{v}_1, \vec{v}_2\}$ is linearly independent, using Gaussian Elimination.**

Solution: We see that $\vec{v}_2$ is not a scaled version of $\vec{v}_1$, so $\{\vec{v}_1, \vec{v}_2\}$ is linearly independent.

We show this explicitly by finding $\alpha_1, \alpha_2 \in \mathbb{R}$ such that $\alpha_1\vec{v}_1 + \alpha_2\vec{v}_2 = \vec{0}$. If there exists a unique solution, $\alpha_1 = \alpha_2 = 0$, then $\{\vec{v}_1, \vec{v}_2\}$ is linearly independent.

We set up the system of equations and use Gaussian Elimination.

$$\begin{bmatrix} 1 & 2 \\ 2 & 5 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left[\begin{array}{cc|c} 1 & 2 & 0 \\ 2 & 5 & 0 \\ 2 & 3 & 0 \end{array}\right]$$

$$\rightarrow \left[\begin{array}{cc|c} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{array}\right] R_2 \leftarrow R_2 - 2R_1; R_3 \leftarrow R_3 - 2R_1$$

$$\rightarrow \left[\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array}\right] R_1 \leftarrow R_1 - 2R_2; R_3 \leftarrow R_3 + R_2$$

Thus there is a unique solution $\alpha_1 = \alpha_2 = 0$ and $\{\vec{v}_1, \vec{v}_2\}$ is a linearly independent set.

(c) (10 points) **Prove the following statement**: If

$$\{\vec{v}_1, \vec{v}_2, \vec{v}_3, \cdots, \vec{v}_n\}$$

is a linearly independent set, then

$$\{\vec{v}_1, \vec{v}_2 - \vec{v}_1, \vec{v}_3 - \vec{v}_2 - \vec{v}_1, \cdots, \vec{v}_n - \vec{v}_{n-1} - \cdots - \vec{v}_2 - \vec{v}_1\}$$

is a linearly independent set.

Solution:

Proof Setup:

We consider the following vectors $\vec{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \in \mathbb{R}^n$, $\vec{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix} \in \mathbb{R}^n$ in the proof to follow.

We know that $\{\vec{v}_1, \vec{v}_2, \vec{v}_3, \cdots, \vec{v}_n\}$ is a linearly independent set, so we can write:

$$\alpha_1\vec{v}_1 + \alpha_2\vec{v}_2 + \alpha_3\vec{v}_3 + \cdots + \alpha_n\vec{v}_n = 0 \Rightarrow \alpha_1, \alpha_2, \alpha_3, \cdots, \alpha_n = 0 \tag{1}$$

We want to prove that:

$$\beta_1\vec{v}_1 + \beta_2(\vec{v}_2 - \vec{v}_1) + \beta_3(\vec{v}_3 - \vec{v}_2 - \vec{v}_1) + \cdots + \beta_n(\vec{v}_n - \vec{v}_{n-1} - \cdots - \vec{v}_2 - \vec{v}_1) = 0 \Rightarrow \beta_1, \beta_2, \beta_3, \cdots, \beta_n = 0 \tag{2}$$

Proof by Construction:

By rearranging the left-hand side of Equation (2), we get

$$(\beta_1 - \beta_2 - \beta_3 - \cdots - \beta_n)\vec{v}_1 + (\beta_2 - \beta_3 - \cdots - \beta_n)\vec{v}_2 + (\beta_3 - \beta_4 - \cdots - \beta_n)\vec{v}_3 + \cdots + \beta_n\vec{v}_n = 0 \tag{3}$$

We now recall our given fact that $\{\vec{v}_1, \vec{v}_2, \vec{v}_3, \cdots, \vec{v}_n\}$ is a linearly independent set and compare the left-hand side of Equation 1 to Equation 3.

$$\begin{aligned} \alpha_n &= \beta_n = 0 \\ \alpha_{n-1} &= \beta_{n-1} - \beta_n = 0 \\ &\vdots \\ \alpha_2 &= \beta_2 - \beta_3 - \cdots - \beta_n = 0 \\ \alpha_1 &= \beta_1 - \beta_2 - \cdots - \beta_n = 0 \end{aligned}$$

Solving this system of equations with back-substitution, starting with β_n and working towards β_1, we get: $\beta_1 = \beta_2 = \beta_3 = \cdots = \beta_n = 0$

So we have thus shown that:

$$\beta_1\vec{v}_1 + \beta_2(\vec{v}_2 - \vec{v}_1) + \beta_3(\vec{v}_3 - \vec{v}_2 - \vec{v}_1) + \cdots + \beta_n(\vec{v}_n - \vec{v}_{n-1} - \cdots - \vec{v}_2 - \vec{v}_1) = 0 \Rightarrow \beta_1, \beta_2, \beta_3, \cdots, \beta_n = 0$$

Therefore, $\{\vec{v}_1, \vec{v}_2 - \vec{v}_1, \vec{v}_3 - \vec{v}_2 - \vec{v}_1, \cdots, \vec{v}_n - \vec{v}_{n-1} - \cdots - \vec{v}_2 - \vec{v}_1\}$ is a linearly independent set.

Proof by Contradiction:

Suppose $\{\vec{v}_1, \vec{v}_2 - \vec{v}_1, \vec{v}_3 - \vec{v}_2 - \vec{v}_1, \cdots, \vec{v}_n - \vec{v}_{n-1} - \cdots - \vec{v}_2 - \vec{v}_1\}$ were **not** a linearly independent set. That would suggest there exists a vector $\vec{\beta} \in \mathbb{R}^n$ such that:

$$\beta_1\vec{v}_1 + \beta_2(\vec{v}_2 - \vec{v}_1) + \beta_3(\vec{v}_3 - \vec{v}_2 - \vec{v}_1) + \cdots + \beta_n(\vec{v}_n - \vec{v}_{n-1} - \cdots - \vec{v}_2 - \vec{v}_1) = 0$$
$$\vec{\beta} \neq \vec{0}$$

As with the proof by construction, we can rearrange the left-hand side of Equation 2 to get Equation 3. Applying the fact that $\{\vec{v}_1, \vec{v}_2, \vec{v}_3, \cdots, \vec{v}_n\}$ is a linearly independent set, then we arrive at the unique solution $\beta_1 = \beta_2 = \beta_3 = \cdots = \beta_n = 0$ which contradicts our assumption that there exists a solution $\vec{\beta} \neq \vec{0}$.

Therefore, $\{\vec{v}_1, \vec{v}_2 - \vec{v}_1, \vec{v}_3 - \vec{v}_2 - \vec{v}_1, \cdots, \vec{v}_n - \vec{v}_{n-1} - \cdots - \vec{v}_2 - \vec{v}_1\}$ is a linearly independent set.

6. GPT-3 wrote this question... or did it? (33 points)

Your friend, a UC Berkeley undergraduate, is working on a project to use OpenAI's GPT-3 language generation algorithm to write blog posts that mimic human writers. After one of their first posts trends on Hacker News, you apply for access to GPT-3 so you can examine how it works for yourself.

The algorithm works by iterating. The blog post at time step n is represented by a vector $\vec{x}[n]$. We will define the entries of the vector later. The GPT-3 algorithm is represented by the transition matrix $\mathbf{A}$. So the blog post at time $n+1$ is given by:

$$\vec{x}[n+1] = \mathbf{A} \cdot \vec{x}[n].$$

(a) (4 points) You find that the transition matrix $\mathbf{A}$ of one version of the algorithm is described by the flow diagram in **Figure 6.1**.

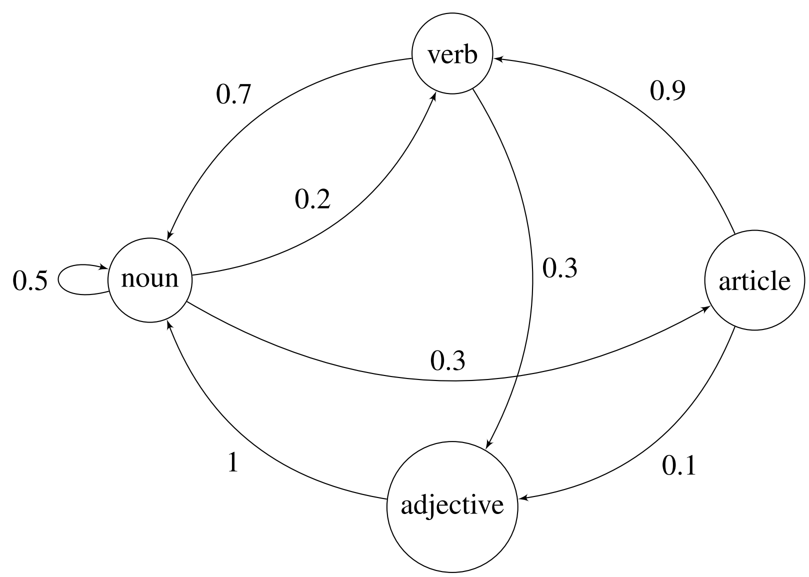

Figure 6.1: A flow diagram to represent how model $\mathbf{A}$ transforms state vector $\vec{x}[n]$.

In this case, each entry of $\vec{x}[n]$ describes the number of nouns, verbs, adjectives and articles in the blog post at time n. The structure of the state vector $\vec{x}[n]$ for $n \geq 0$ is:

$$\vec{x}[n] = \begin{bmatrix} x_{\text{noun}}[n] \\ x_{\text{verb}}[n] \\ x_{\text{adjective}}[n] \\ x_{\text{article}}[n] \end{bmatrix}. \tag{4}$$

What is the A matrix based on the flow diagram in Figure 6.1?

Solution:

Recall that in a transition matrix $\mathbf{A}$, every element $\mathbf{A}[i,j]$ at row i and column j should represent the flow from state j to state i in a timestep i.e. the edge value from state j to state i. In this case, from the state vector structure, **noun** is state 1, **verb** is state 2, **adjective** is state 3, and **article** is state 4.

We write the state equations as follows, keeping in mind that x_1, x_2, x_3, and x_4 indicate the state corresponding to the number of nouns, verbs, adjectives, and articles, respectively:

$$\begin{aligned} x_1[n+1] &= 0.5x_1[n] + 0.7x_2[n] + x_3[n] \\ x_2[n+1] &= 0.2x_1[n] + 0.9x_4[n] \\ x_3[n+1] &= 0.3x_2[n] + 0.1x_4[n] \\ x_4[n+1] &= 0.3x_1[n] \end{aligned}$$

Filling in the elements of $\mathbf{A}$ accordingly gives the following result for the transition matrix of the system above:

$$\mathbf{A} = \begin{bmatrix} 0.5 & 0.7 & 1 & 0 \\ 0.2 & 0 & 0 & 0.9 \\ 0 & 0.3 & 0 & 0.1 \\ 0.3 & 0 & 0 & 0 \end{bmatrix}.$$

(b) (4 points) Now, you are interested in testing out a less complex model that operates on just three parts of speech: nouns, verbs, and adjectives. In this case, the structure of the state vector $\vec{x}[n]$ at each iteration n is:

$$\vec{x}[n] = \begin{bmatrix} x_{\text{noun}}[n] \\ x_{\text{verb}}[n] \\ x_{\text{adjective}}[n] \end{bmatrix}.$$

Your starting article, $\vec{x}[0]$, consists of 4000 nouns, 4000 verbs, and 4000 adjectives, i.e, $\vec{x}[0] = \begin{bmatrix} 4000 \\ 4000 \\ 4000 \end{bmatrix}$.

You want to try out a different 3×3 transition matrix $\mathbf{B}$ to see what happens to your starting article.

$$\mathbf{B} = \begin{bmatrix} 0.25 & 0.25 & 0 \\ 0.25 & 0 & 0.25 \\ 0 & 0.25 & 0.25 \end{bmatrix}.$$

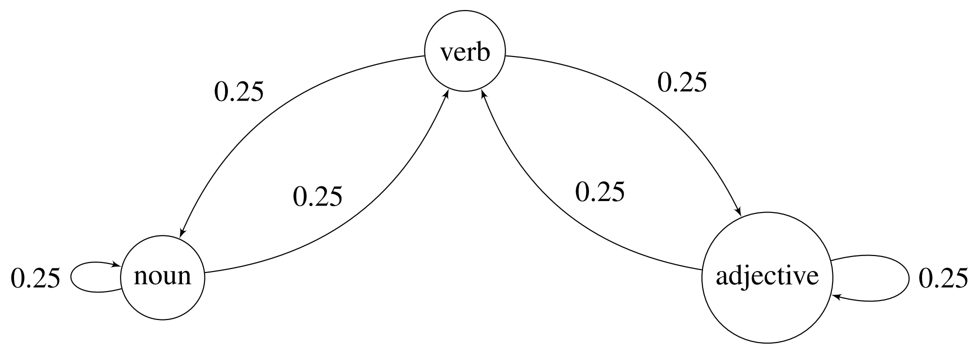

Figure 6.2: A flow diagram to represent how model $\mathbf{B}$ transforms state vector $\vec{x}[n]$.

Compute the next state vector, $\vec{x}[1]$, if we apply matrix B once to the starting state vector, $\vec{x}[0]$. Show your work.

Solution:

Given the information in the problem statement and the model above, we know the following:

$$\mathbf{B} = \begin{bmatrix} 0.25 & 0.25 & 0 \\ 0.25 & 0 & 0.25 \\ 0 & 0.25 & 0.25 \end{bmatrix}, \vec{x}[0] = \begin{bmatrix} 4000 \\ 4000 \\ 4000 \end{bmatrix}$$

Let's apply $\mathbf{B}$ to $\vec{x}[0]$ once, and see what we get:

$$\vec{x}[1] = \mathbf{B} \cdot \vec{x}[0] = \begin{bmatrix} 0.25 & 0.25 & 0 \\ 0.25 & 0 & 0.25 \\ 0 & 0.25 & 0.25 \end{bmatrix} \begin{bmatrix} 4000 \\ 4000 \\ 4000 \end{bmatrix} = \begin{bmatrix} 0.25 \cdot 4000 + 0.25 \cdot 4000 + 0 \\ 0.25 \cdot 4000 + 0 + 0.25 \cdot 4000 \\ 0 + 0.25 \cdot 4000 + 0.25 \cdot 4000 \end{bmatrix} = \begin{bmatrix} 2000 \\ 2000 \\ 2000 \end{bmatrix}$$

Therefore, the resulting article has 2000 nouns, 2000 verbs, and 2000 adjectives.

(c) (2 points) **Is the transition matrix B representative of a conservative system? Why or why not?**

$$\mathbf{B} = \begin{bmatrix} 0.25 & 0.25 & 0 \\ 0.25 & 0 & 0.25 \\ 0 & 0.25 & 0.25 \end{bmatrix}$$

Solution:

The transition matrix $\mathbf{B}$ is not representative of a conservative system. If we sum up the values in any one column of the transition matrix, we notice that the values sum to 0.5, not 1, indicating that generally, words are exiting this system.

(d) (4 points) Now, you are considering testing out an unknown 3×3 model matrix $\mathbf{C}$. You have limited information about $\mathbf{C}$: you know that $\mathbf{C}$ has three eigenvectors $\vec{v}_1$, $\vec{v}_2$, and $\vec{v}_3$, and three corresponding eigenvalues λ_1, λ_2 and λ_3.

You are also told that you can rewrite $\vec{x}[0]$, the starting state vector, as a linear combination of the eigenvectors of $\mathbf{C}$:

$$\vec{x}[0] = -3000\vec{v}_1 + 3000\vec{v}_2 + 7000\vec{v}_3.$$

You are ultimately interested in observing how $\mathbf{C}$ modifies the state vector, $\vec{x}[n]$, at every timestep n. **Write an expression for $\vec{x}[n]$ in terms of n, the eigenvalues of C, and the eigenvectors of C. You do not have to show a lot of work for this problem.** Your answer may also include constants. *Note: Your equation **should not** include any terms involving previous states (in other words, do not use terms such as $n-1$, $n-2$, ...).*

Solution:

According to the GPT-3 algorithm, after n iterations, the start vector $\vec{x}[0]$ will have been left-multiplied by transition matrix $\mathbf{C}$ n times, giving the following explicit relationship between $\vec{x}[0]$ and $\vec{x}[n]$:

$$\vec{x}[n] = \mathbf{C}^n \vec{x}[0]$$

Using the fact that $\mathbf{C}\vec{x}_i = \lambda_i \vec{x}_i \implies \mathbf{C}^n \vec{x}_i = \lambda_i^n \vec{x}_i$:

$$\begin{aligned}\vec{x}[n] &= \mathbf{C}^n \vec{x}[0] \\ &= \mathbf{C}^n(-3000\vec{v}_1 + 3000\vec{v}_2 + 7000\vec{v}_3) \\ &= -3000\lambda_1^n \vec{v}_1 + 3000\lambda_2^n \vec{v}_2 + 7000\lambda_3^n \vec{v}_3\end{aligned}$$

The state vector after the nth iteration of GPT-3 – $\vec{x}[n]$ – in terms of n, $\mathbf{C}$'s eigenvalues, $\mathbf{C}$'s eigenvalues, and constants only is $\vec{x}[n] = -3000\lambda_1^n \vec{v}_1 + 3000\lambda_2^n \vec{v}_2 + 7000\lambda_3^n \vec{v}_3$.

(e) (6 points) Suppose you are now working with a 7000-word blog post consisting entirely of nouns, i.e.,

$$\vec{x}[0] = \begin{bmatrix} x_{\text{noun}}[0] \\ x_{\text{verb}}[0] \\ x_{\text{adjective}}[0] \end{bmatrix} = \begin{bmatrix} 7000 \\ 0 \\ 0 \end{bmatrix}.$$

You'd like to add some variety to the parts of speech used in your article and introduce some verbs and adjectives, too. Someone on the internet suggests using the following matrix $\mathbf{D}$:

$$\mathbf{D} = \begin{bmatrix} 0.3 & 0 & 0.4 \\ 0.4 & 1 & 0.2 \\ 0.3 & 0 & 0.4 \end{bmatrix}.$$

The eigenvalues of $\mathbf{D}$ are $\lambda_1 = 0$, $\lambda_2 = 0.7$, and $\lambda_3 = 1$, with corresponding eigenvectors $\vec{v}_1 = \begin{bmatrix} -\frac{4}{3} \\ \frac{1}{3} \\ 1 \end{bmatrix}$, $\vec{v}_2 = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$, and $\vec{v}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$.

Recall that $\vec{x}[n+1] = \mathbf{D} \cdot \vec{x}[n]$.

What is the steady-state of the system, i.e., $\lim_{n\to\infty} \vec{x}[n]$?

Hint: You do NOT have to express $\vec{x}[0]$ in terms of $\vec{v}_1, \vec{v}_2, \vec{v}_3$ to solve this part. Also, think about if the system is conservative.

Solution:

We will offer three varying solutions to get to the same final answer of $\lim_{n\to\infty} \vec{x}[n] = [0 \quad 7000 \quad 0]^\top$, i.e, 0 nouns, 7000 verbs, 0 adjectives.

i. A shortcut to finding the steady state based on information given in previous parts is that the steady state will be the eigenvector corresponding to the eigenvalue of 1 for transition matrix **C**. In this case, this is:

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

When using this method, the eigenvector will have to be scaled according to the number of words in the article – 7000 – so the final word counts in the post after running GPT-3 would be:

$$\begin{bmatrix} 0 \\ 7000 \\ 0 \end{bmatrix}$$

Note that for this solution, we did not have to use additional information associated with the other two eigenvectors or eigenvalues of **D**; it was sufficent to simply look at the eigenvector corresponding to $\lambda_3 = 1$ and pair it with the fact that matrix **D** is **conservative**, meaning that the total number of words in the system should be exactly the same when comparing the steady state to the start state.

ii. A more rigorous, less intuitive method of computing the steady state vector would be to first decompose the starting state vector into a linear combination of the eigenvectors of matrix **D**. This can be accomplished using our tried-and-true Gaussian Elimination technique, or by writing a system of equations and solving for the unknowns. Here, we show the computation of the weights by solving the system of equations below, where x_1, x_2, and x_3 are the weights on $\vec{v}_1$, $\vec{v}_2$, and $\vec{v}_3$, respectively.

$$\begin{cases} -\frac{4}{3} \cdot x_1 + x_2 = 7000 & (1) \\ \frac{1}{3} \cdot x_1 - 2x_2 + x_3 = 0 & (2) \\ x_1 + x_2 = 0 & (3) \end{cases}$$

Substituting in $x_1 = -x_2$ by using equation (3) into equation (1), we obtain $\frac{7}{3}x_2 = 7000 \implies x_2 = 3000$.

Substituting $x_2 = 3000$ back into equation (3), we obtain $x_1 = -3000$.

Finally substituting $x_1 = -3000$ and $x_2 = 3000$ into equation (2), we obtain $-1000 - 6000 + x_3 = 0 \implies x_3 = 7000$.

Hence, the initial state expressed as a linear combination of the eigenvectors of matrix **D** is:

$$\vec{x}[0] = -3000\vec{v}_1 + 3000\vec{v}_2 + 7000\vec{v}_3$$

Notice that this is the same linear combination as given in part (d). Nifty!

Next, we represent $\vec{x}[n]$ at an arbitrary timestep n as:

$$\begin{aligned}\vec{x}[n] &= \mathbf{D}^n \cdot \vec{x}[0] \\ &= \mathbf{D}^n(-3000\vec{v}_1 + 3000\vec{v}_2 + 7000\vec{v}_3) \\ &= -3000\lambda_1^n\vec{v}_1 + 3000\lambda_2^n\vec{v}_2 + 7000\lambda_3^n\vec{v}_3 \\ &= -3000(0)\vec{v}_1 + 3000(0.7)^n\vec{v}_2 + 7000(1)^n\vec{v}_3\end{aligned}$$

Taking the limit of this expression for $\vec{x}[n]$ as $n \to \infty$, we see that:

$$\begin{aligned}\lim_{n\to\infty}\vec{x}[n] &= \lim_{n\to\infty}[-3000(0)\vec{v}_1 + 3000(0.7)^n\vec{v}_2 + 7000(1)^n\vec{v}_3] \\ &= 7000\vec{v}_3.\end{aligned}$$

The component corresponding to $\lambda_1 = 0$ vanishes immediately, whereas the component corresponding to $\lambda_2 = 0.7$ tends to 0 as $n \to \infty$. This leaves us with a solitary contribution from the term corresponding to $\vec{v}_3$. Therefore, the steady-state vector is:

$$\begin{aligned}\lim_{n\to\infty}\vec{x}[n] &= 7000\vec{v}_3 \\ &= \begin{bmatrix}0\\7000\\0\end{bmatrix}.\end{aligned}$$

iii. We recall that at steady state, $\vec{x}[n+1] = \vec{x}[n]$ or $\mathbf{D}\cdot\vec{x}[n] = \vec{x}[n]$. Substituting for $\mathbf{D}$ and general element placeholders for $\vec{x}[n]$ gives the following:

$$\mathbf{D}\cdot\vec{x}[n] = \vec{x}[n]$$
$$\begin{bmatrix}0.3 & 0 & 0.4\\0.4 & 1 & 0.2\\0.3 & 0 & 0.4\end{bmatrix}\begin{bmatrix}x_{\text{noun}}[n]\\x_{\text{verb}}[n]\\x_{\text{adjective}}[n]\end{bmatrix} = \begin{bmatrix}x_{\text{noun}}[n]\\x_{\text{verb}}[n]\\x_{\text{adjective}}[n]\end{bmatrix}$$
$$\begin{bmatrix}0.3x_{\text{noun}}[n] + 0.4x_{\text{adjective}}[n]\\0.4x_{\text{noun}}[n] + x_{\text{verb}}[n] + 0.2x_{\text{adjective}}[n]\\0.3x_{\text{noun}}[n] + 0.4x_{\text{adjective}}[n]\end{bmatrix} = \begin{bmatrix}x_{\text{noun}}[n]\\x_{\text{verb}}[n]\\x_{\text{adjective}}[n]\end{bmatrix}$$

The above may be written as a system of equations with unknowns $x_{\text{noun}}[n]$, $x_{\text{verb}}[n]$, $x_{\text{adjective}}[n]$.

$$0.3x_{\text{noun}}[n] + 0.4x_{\text{adjective}}[n] = x_{\text{noun}}[n] \to 0.4x_{\text{adjective}}[n] = 0.7x_{\text{noun}}[n] \text{ ... (1)}$$
$$0.4x_{\text{noun}}[n] + x_{\text{verb}}[n] + 0.2x_{\text{adjective}}[n] = x_{\text{verb}}[n] \to 0.4x_{\text{noun}}[n] = -0.2x_{\text{adjective}}[n] \text{ ... (2)}$$
$$0.3x_{\text{noun}}[n] + 0.4x_{\text{adjective}}[n] = x_{\text{adjective}}[n]$$

The relationships in (1) and (2) will only be satisfied if $x_{\text{noun}}[n]$ and $x_{\text{adjective}}[n]$ are both equal to 0. Again, we utilize the fact that $\mathbf{D}$ represents a conservative system – nothing is entering or leaving between iterations because the sum of the elements in each column of the transition matrix is 1. Because $x_{\text{noun}}[n] = x_{\text{adjective}}[n] = 0$, $x_{\text{verb}}[n]$ must be 7000 since we began with 7000 total words (which were nouns).

(f) (5 points) You are still considering new ways to make your starting 7000-word blog post consisting of all nouns (i.e, $\vec{x}[0] = \begin{bmatrix}7000\\0\\0\end{bmatrix}$) more interesting. Your EECS16A TA, Lily, suggests the following transition matrix $\mathbf{E}$, and tells you its eigenvalues are $\lambda_1 = 0.4$, $\lambda_2 = 0.1$, and $\lambda_3 = 1$.

$$\mathbf{E} = \begin{bmatrix} 0.3 & 0.1 & 0.2 \\ 0.4 & 0.8 & 0.4 \\ 0.3 & 0.1 & 0.4 \end{bmatrix}$$

Compute the eigenspace corresponding to the eigenvalue $\lambda_2 = 0.1$ for matrix E. Show your work.

Solution:

We are given in the problem statement that the eigenvalue in question is $\lambda_2 = 0.1$. Therefore, this question is equivalent to asking to solve for the nullspace of $(\mathbf{E} - \lambda_2 \mathbf{I})$:

$$\mathbf{E}\vec{v}_2 = \lambda_2 \vec{v}_2$$
$$(\mathbf{E} - \lambda_2 \mathbf{I})\vec{v}_2 = \vec{0}$$

Setting up for Gaussian Elimination:

$$\left[\begin{array}{ccc|c} 0.2 & 0.1 & 0.2 & 0 \\ 0.4 & 0.7 & 0.4 & 0 \\ 0.3 & 0.1 & 0.3 & 0 \end{array}\right]$$

$$\rightarrow \left[\begin{array}{ccc|c} 1 & 1/2 & 1 & 0 \\ 1 & 7/4 & 1 & 0 \\ 1 & 1/3 & 1 & 0 \end{array}\right] \quad \text{normalizing all rows}$$

$$\rightarrow \left[\begin{array}{ccc|c} 1 & 1/2 & 1 & 0 \\ 0 & 5/4 & 0 & 0 \\ 0 & -1/6 & 0 & 0 \end{array}\right] \quad R_2 \leftarrow R_2 - R_1; \quad R_3 \leftarrow R_3 - R_1$$

$$\rightarrow \left[\begin{array}{ccc|c} 1 & 1/2 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \quad R_3 \leftarrow R_3 + \left(\frac{2}{15}R_2\right); \quad R_2 \leftarrow \left(\frac{4}{5}\right)R_2$$

$$\rightarrow \left[\begin{array}{ccc|c} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] \quad R_1 \leftarrow R_1 - \left(\frac{1}{2}\right)R_2$$

At this point, we can see that x_3 is our one free variable. Parameterizing x_3 with α, we calculate the eigenvector/eigenspace for $\lambda_2 = 0.1$ as:

$$\vec{v}_2 = \alpha \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}, \quad \alpha \in \mathbb{R}$$

(g) (8 points) You are finally considering a model **F** matrix:

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}.$$

The entries of the columns of this matrix add up to 2, i.e.:

$$f_{11} + f_{21} + f_{31} = 2, \qquad f_{12} + f_{22} + f_{32} = 2, \qquad f_{13} + f_{23} + f_{33} = 2.$$

What does this transition matrix do to the **total number of words (i.e. verbs + nouns + adjectives) in each iteration** of the blog post? **Provide a rigorous proof of your answer.** Recall that $\vec{x}[n+1] = \mathbf{F}\cdot\vec{x}[n]$, and that

$$\vec{x}[n] = \begin{bmatrix} x_{\text{noun}}[n] \\ x_{\text{verb}}[n] \\ x_{\text{adjective}}[n] \end{bmatrix}.$$

Solution:

We can begin by seeing what happens to the current state vector $\vec{x}[n]$ at any timestep n upon one application of the $\mathbf{F}$ matrix as given in the problem statement.

$$\begin{aligned} \vec{x}[n+1] &= \mathbf{F}\cdot\vec{x}[n] \\ &= \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \cdot \begin{bmatrix} x_{\text{noun}}[n] \\ x_{\text{verb}}[n] \\ x_{\text{adjective}}[n] \end{bmatrix} \\ &= \begin{bmatrix} f_{11}\cdot x_{\text{noun}}[n] + f_{12}\cdot x_{\text{verb}}[n] + f_{13}\cdot x_{\text{adjective}}[n] \\ f_{21}\cdot x_{\text{noun}}[n] + f_{22}\cdot x_{\text{verb}}[n] + f_{23}\cdot x_{\text{adjective}}[n] \\ f_{31}\cdot x_{\text{noun}}[n] + f_{32}\cdot x_{\text{verb}}[n] + f_{33}\cdot x_{\text{adjective}}[n] \end{bmatrix} \end{aligned}$$

Note that the expression for $\vec{x}[n+1]$ is valid for the current state vector at *any* timestep $n+1$.

Next, we consider the *total number of words* in the article at any timestep $n+1$. We can find the total number of words simply by summing up all of the entries of the $\vec{x}[n+1]$ state vector which was found above:

$$\begin{aligned} \text{total \# of words at } n+1 \quad = \quad & (f_{11}\cdot x_{\text{noun}}[n] + f_{12}\cdot x_{\text{verb}}[n] + f_{13}\cdot x_{\text{adjective}}[n]) + \\ & (f_{21}\cdot x_{\text{noun}}[n] + f_{22}\cdot x_{\text{verb}}[n] + f_{23}\cdot x_{\text{adjective}}[n]) + \\ & (f_{31}\cdot x_{\text{noun}}[n] + f_{32}\cdot x_{\text{verb}}[n] + f_{33}\cdot x_{\text{adjective}}[n]). \end{aligned}$$

If we regroup terms to combine all of the $x_{\text{noun}}[n]$, $x_{\text{verb}}[n]$, and $x_{\text{adjective}}[n]$ respectively, we get:

$$\begin{aligned} \text{total \# of words at } n+1 \quad = \quad & (f_{11}+f_{21}+f_{31})\cdot x_{\text{noun}}[n] + \\ & (f_{12}+f_{22}+f_{32})\cdot x_{\text{verb}}[n] + \\ & (f_{13}+f_{23}+f_{33})\cdot x_{\text{adjective}}[n]. \end{aligned}$$

Applying the properties of each column as given in the question statement, i.e, that the entries in each respective column of $\mathbf{F}$ sum to 2, we can simplify this expression for the total number of words further:

$$\begin{aligned} \text{total \# of words at } n+1 \quad &= 2\cdot x_{\text{noun}}[n] + 2\cdot x_{\text{verb}}[n] + 2\cdot x_{\text{adjective}}[n] \\ &= 2\cdot (x_{\text{noun}}[n] + x_{\text{verb}}[n] + x_{\text{adjective}}[n]). \end{aligned}$$

Recall that the total number of words in the system at timestep n was simply the sum of each of the entries of the state vector $\vec{x}[n]$, i.e, the total number of words at time n is always equal to the term in parenthesis above, $(x_{\text{noun}}[n] + x_{\text{verb}}[n] + x_{\text{adjective}}[n])$.

Hence, at any timestep $n+1$, the transition matrix $\mathbf{F}$ is **doubling the total amount of words in the system from timestep** n.

■

7. Get in the robot, 16A people! (31 points)

You are testing a giant robot that you programmed.

(a) (8 points) You can use a navigation control input $\vec{u} \in \mathbb{R}^3$ to move the robot to different positions $\vec{p}$, as described by:

$$\mathbf{A}\vec{u} = \vec{p}, \quad \text{where} \quad \mathbf{A} = \begin{bmatrix} 1 & 0 & 2 \\ 1 & 1 & 2 \\ 0 & 1 & a \end{bmatrix}.$$

The value of the parameter $a \in \mathbb{R}$ determines the robot's behavior.
You find your giant robot moving only along a plane in space defined by

$$\mathbb{U} = \left\{ \vec{p} \in \mathbb{R}^3 \;\middle|\; \vec{p} = \begin{bmatrix} \alpha \\ \alpha + \beta \\ \beta \end{bmatrix}, \quad \alpha, \beta \in \mathbb{R} \right\}.$$

Find the value of a in the programmed matrix, which caused your robot to get stuck on the plane $\mathbb{U}$. Show your work and justify your answer.

Solution:

Solution 1: Inspection of Matrix A and Using Definition of Linear Independence

In order to solve this problem, we need to find a value of a such that the column space of $\mathbf{A}$, $C(\mathbf{A})$, is represented by the plane $\mathbb{U}$. Since $C(\mathbf{A})$ has a dimension of 2, there are two linearly independent columns in matrix $\mathbf{A}$. We can see that the first two columns are linearly independent and $C(\mathbf{A})$ is defined by their span. So the third column must be linearly dependent on the first and second columns. So we can write:

$$\begin{aligned} \begin{bmatrix} 2 \\ 2 \\ a \end{bmatrix} &= \alpha_1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \\ \implies \begin{bmatrix} 2 \\ 2 \\ a \end{bmatrix} &= \begin{bmatrix} \alpha_1 \\ \alpha_1 + \alpha_2 \\ \alpha_2 \end{bmatrix} \\ \implies \alpha_1 &= 2 \\ \alpha_1 + \alpha_2 &= 2 \\ \alpha_2 &= a \end{aligned}$$

Hence, $\alpha_1 = 2$ and $\alpha_2 = a = 0$.

Solution 2: Setting up a System of Equations and Using Gaussian Elimination

We know if the column space of A is equal to the given subspace, i.e. $C(\mathbf{A}) = \mathbb{U}$, then the following must hold for all $\vec{u} \in \mathbb{R}^3$:

$$\mathbf{A}\vec{u} = \vec{p}$$

$$\begin{bmatrix} 1 & 0 & 2 \\ 1 & 1 & 2 \\ 0 & 1 & a \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \alpha \\ \alpha + \beta \\ \beta \end{bmatrix}$$

We then apply Gaussian Elimination to this system:

$$\left[\begin{array}{ccc|c} 1 & 0 & 2 & \alpha \\ 1 & 1 & 2 & \alpha+\beta \\ 0 & 1 & a & \beta \end{array}\right]$$
$$\rightarrow \left[\begin{array}{ccc|c} 1 & 0 & 2 & \alpha \\ 0 & 1 & 0 & \beta \\ 0 & 1 & a & \beta \end{array}\right] R_2 \leftarrow R_2 - R_1$$
$$\rightarrow \left[\begin{array}{ccc|c} 1 & 0 & 2 & \alpha \\ 0 & 1 & 0 & \beta \\ 0 & 0 & a & 0 \end{array}\right] R_3 \leftarrow R_3 - R_2$$

Reading out the last row, $au_3 = 0$ must hold for all possible values of $u_3 \in \mathbb{R}$. Therefore, $a = 0$ for $C(\mathbf{A}) = \mathbb{U}$.

(b) (6 points) Your robot operates with the same model as in part (a), i.e. $\mathbf{A}\vec{u} = \vec{p}$. We recall $\vec{u}$ is a navigation control input and $\vec{p}$ is the robot's position. However, we lost the documentation about the $\mathbf{A}$ matrix! In other words, **we don't know what A is and it may not be the same matrix as in part (a)**.

You found an interesting phenomenon: two different navigation control inputs, $\vec{u}_1$ and $\vec{u}_2$ move the robot to the same position $\vec{p}$, i.e.

$$\mathbf{A}\vec{u}_1 = \mathbf{A}\vec{u}_2 = \vec{p}, \quad \text{where} \quad \vec{u}_1 = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}, \quad \vec{u}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \text{and } \vec{p} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}.$$

With this information, **find a control input $\vec{u}$ that can return you back to $\vec{p} = \vec{0}$**, where your base of operations is located. Due to some limitations of the robot motors, **you cannot command a zero control input, i.e. $\vec{u} \neq \vec{0}$. Show your work and justify your answer.**

Solution:

Solution 1: Findng a vector in the nullspace directly. We have to find a value of $\vec{u}$ such that:

$$\mathbf{A}\vec{u} = \vec{0}, \vec{u} \neq \vec{0}$$

Also we are given:

$$\mathbf{A}\vec{u}_1 = \vec{p}$$
$$\mathbf{A}\vec{u}_2 = \vec{p}$$

Subtracting the equations from each other we get:

$$\mathbf{A}(\vec{u}_1 - \vec{u}_2) = \vec{p} - \vec{p}$$
$$\mathbf{A}(\vec{u}_1 - \vec{u}_2) = \vec{0}$$
$$\mathbf{A}(\begin{bmatrix}2\\1\\0\end{bmatrix} - \begin{bmatrix}1\\1\\1\end{bmatrix}) = \begin{bmatrix}0\\0\\0\end{bmatrix}$$
$$\mathbf{A}\begin{bmatrix}1\\0\\-1\end{bmatrix} = \begin{bmatrix}0\\0\\0\end{bmatrix}$$

This means that any vector

$$\vec{u} = \alpha\begin{bmatrix}1\\0\\-1\end{bmatrix}, \alpha \in \mathbb{R}, \alpha \neq 0$$

would be a valid control input to take us back to the base.

Solution 2: Finding structure of the matrix A and solving $\mathbf{A}\vec{u} = \vec{0}$.

The strategy here will be to write **A** in terms of its elements and use the given equations to determine the structure of **A**.

Let's express the elements of **A** as:

$$\mathbf{A} = \begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix}.$$

We will now write out our two equations, $\mathbf{A}\vec{u}_1 = \vec{p}$ and $\mathbf{A}\vec{u}_2 = \vec{p}$:

$$\begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix}\begin{bmatrix}2\\1\\0\end{bmatrix} = \begin{bmatrix}2\\3\\1\end{bmatrix},$$
$$\begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix} = \begin{bmatrix}2\\3\\1\end{bmatrix}.$$

Intuitively, we have 9 unknowns in our matrix **A** and a system of 6 equations (2 equations per row of **A**). So we can use this system of equations to gain more knowledge about the structure of **A**:

$$2a_{11} + 1a_{12} + 0a_{13} = 2 \Rightarrow a_{12} = 2 - 2a_{11} \quad (5)$$
$$2a_{21} + 1a_{22} + 0a_{23} = 3 \Rightarrow a_{22} = 3 - 2a_{21} \quad (6)$$
$$2a_{31} + 1a_{32} + 0a_{33} = 1 \Rightarrow a_{32} = 1 - 2a_{31} \quad (7)$$
$$a_{11} + a_{12} + a_{13} = 2 \quad (8)$$
$$a_{21} + a_{22} + a_{23} = 3 \quad (9)$$
$$a_{31} + a_{32} + a_{33} = 1 \quad (10)$$

Substituting the values of a_{12}, a_{22}, a_{32} from Equations (5), (6), and (7) into Equations (8), (9), and (10), respectively, we arrive at the following constraints on the structure of $\mathbf{A}$:

$$\begin{aligned} a_{12} &= 2 - 2a_{11} \\ a_{22} &= 3 - 2a_{21} \\ a_{32} &= 1 - 2a_{31} \\ a_{13} &= a_{11} \\ a_{23} &= a_{21} \\ a_{33} &= a_{31} \end{aligned}$$

We can substitute these expressions back into $\mathbf{A}$. By doing this, we should be able to describe every entry of $\mathbf{A}$ with only 3 variables a_{11}, a_{21}, a_{31}:

$$\mathbf{A} = \begin{bmatrix} a_{11} & 2-2a_{11} & a_{11} \\ a_{21} & 3-2a_{21} & a_{21} \\ a_{31} & 1-2a_{31} & a_{31} \end{bmatrix}.$$

Now all that is left is to solve the expression $\mathbf{A}\vec{u} = \vec{0}$. We can do this using Gaussian Elimination.

$$\left[\begin{array}{ccc|c} a_{11} & 2-2a_{11} & a_{11} & 0 \\ a_{21} & 3-2a_{21} & a_{21} & 0 \\ a_{31} & 1-2a_{31} & a_{31} & 0 \end{array}\right]$$

$$\rightarrow \left[\begin{array}{ccc|c} 1 & \frac{2}{a_{11}}-2 & 1 & 0 \\ a_{21} & 3-2a_{21} & a_{21} & 0 \\ a_{31} & 1-2a_{31} & a_{31} & 0 \end{array}\right] R_1 \leftarrow \frac{1}{a_{11}} R_1$$

$$\rightarrow \left[\begin{array}{ccc|c} 1 & \frac{2}{a_{11}}-2 & 1 & 0 \\ 0 & 3-2\frac{a_{21}}{a_{11}} & 0 & 0 \\ 0 & 1-2\frac{a_{31}}{a_{11}} & 0 & 0 \end{array}\right] R_2 \leftarrow R_2 - a_{21}R_1; R_3 \leftarrow R_3 - a_{31}R_1$$

$$\rightarrow \left[\begin{array}{ccc|c} 1 & \frac{2}{a_{11}}-2 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1-2\frac{a_{31}}{a_{11}} & 0 & 0 \end{array}\right] R_2 \leftarrow \frac{1}{3-2\frac{a_{21}}{a_{11}}} R_2$$

$$\rightarrow \left[\begin{array}{ccc|c} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right] R_1 \leftarrow R_1 - (\frac{2}{a_{11}}-2)R_2; R_3 \leftarrow R_3 - (1-2\frac{a_{31}}{a_{11}})R_2$$

Letting $u_3 = \alpha \in \mathbb{R}$, we arrive at the parametric solution $\vec{u} = \alpha \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$. Recalling that we cannot command a zero control input, we arrive at the following valid set of control inputs to get us to $\vec{p} = \vec{0}$:

$$\vec{u} = \alpha \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}, \alpha \in \mathbb{R}, \alpha \neq 0.$$

This matches the set of valid control inputs found using the first approach.

(c) (5 points) Another challenge in navigation is collision avoidance — you don't want to crash.

You are given the following documentation about the range sensor on Robot A, which is the robot you control. The range sensor can only detect other robots in the following subspace $\mathbb{S}$:

$$\mathbb{S} = \left\{ \vec{p} \in \mathbb{R}^3 \;\middle|\; \vec{p} = \alpha \begin{bmatrix}1\\0\\1\end{bmatrix} + \beta \begin{bmatrix}0\\1\\1\end{bmatrix}, \quad \alpha, \beta \in \mathbb{R} \right\}.$$

You know that Robot B is located at $\vec{b} = \begin{bmatrix}1\\-1\\1\end{bmatrix}$. Will you be able to detect Robot B, i.e., **does $\vec{b}$ belong to the subspace $\mathbb{S}$? Show your work and justify your answer.**

Solution: In order for $\vec{b} \in \mathbb{S}$, there must exist $\alpha, \beta \in \mathbb{R}$ that satisfy the following system of equations:

$$\alpha = 1 \tag{11}$$

$$\beta = -1 \tag{12}$$

$$\alpha + \beta = 1 \tag{13}$$

By adding Equations 11 and 12, we arrive at the following:

$$\alpha + \beta = 0 \tag{14}$$

Finally, we subtract Equation 14 from 13 to get:

$$0 = 1$$

which indicates we have an inconsistent system of equations. Therefore, we cannot detect Robot B as $\vec{b} \notin \mathbb{S}$.

We can also arrive at the same conclusion by using Gaussian elimination to solve the system of equations for the vector of unknowns, $\begin{bmatrix}\alpha\\\beta\end{bmatrix}$.

$$\left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 1 & 1 \end{array}\right]$$

$$\rightarrow \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 1 & 0 \end{array}\right] R_3 \leftarrow R_3 - R_1$$

$$\rightarrow \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{array}\right] R_3 \leftarrow R_3 - R_2$$

We observe the contradictory row corresponding to $0 = 1$ and conclude the system is inconsistent.

(d) (6 points) You are now working on the power source of the robot. The nuclear chain reaction that powers it can be modeled by state transition matrix $\mathbf{Q} \in \mathbb{R}^{3\times 3}$ and the equation:

$$\vec{v}[n+1] = \mathbf{Q}\vec{v}[n],$$

where $\vec{v}[n]$ is the quantity of reactive species at time step n. $\mathbf{Q}$ is:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 10 \end{bmatrix}.$$

You realize that some components of $\vec{v}[n] \to \infty$ as $n \to \infty$.
To stabilize the system you multiply it by a matrix $\mathbf{R}$:

$$\mathbf{R} = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix}.$$

Now the **new system** would be modeled by

$$\vec{v}[n+1] = \mathbf{Q}\mathbf{R}\vec{v}[n].$$

Your job is to pick the entries of $\mathbf{R}$, and **you can only use non-negative values in R, so** $r_1, r_2, r_3 \geq 0$. **What are the largest values you can choose for r_1, r_2, and r_3 to ensure that the absolute values of the components in $\vec{v}[n] \not\to \infty$ as $n \to \infty$? Show your work and justify your answer.**

Solution: We first recall our design constraints that r_1, r_2, r_3 have to be non-negative. So now we need to use the system model and stability constraints to get upper bounds for r_1, r_2, r_3.

In order to ensure our new system is stable (absolute values of the components in $\vec{v}[n] \not\to \infty$ as $n \to \infty$), we need to ensure that the eigenvalues of $\mathbf{QR}$ need to be ≤ 1.

The transition matrix of the new system is given by:

$$\mathbf{QR} = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & 0.5r_2 & 0 \\ 0 & 0 & 10r_3 \end{bmatrix}.$$

Since this matrix is a diagonal matrix, we can immediately read off the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of $\mathbf{QR}$ and add the constraint for stability:

$$\lambda_1 = r_1 \leq 1$$
$$\lambda_2 = 0.5r_2 \leq 1$$
$$\lambda_3 = 10r_3 \leq 1$$

Therefore, we identify the following bounds for r_1, r_2, r_3 for the new system to be stable:

$$r_1 \in [0, 1]$$
$$r_2 \in [0, 2]$$
$$r_3 \in [0, 0.1]$$

The maximum values then are:

$$r_{1,max} = 1$$
$$r_{2,max} = 2$$
$$r_{3,max} = 0.1$$

(e) (6 points) You want to order a new drone to support your robot fleet. The location of the drone is represented by a vector in $\mathbb{R}^3$. You want to program the drone so that it can travel anywhere in the columnspace of an **invertible** matrix $\mathbf{P} \in \mathbb{R}^{3\times 3}$. However, a typo in the commands unfortunately means you end up programming the drone to move in the columnspace of $\mathbf{P}^{-1}$. **Is the columnspace of P the same as the columnspace of $\mathbf{P}^{-1}$? Why or why not?**

Hint: Consider the matrix $\mathbf{P}^{-1}$. *Is* $\mathbf{P}^{-1}$ *is invertible?*

Solution:

In order for $\mathbf{P}$ to be invertible, its 3 columns must be linearly independent. By the same logic, $\mathbf{P}^{-1}$ has 3 linearly independent columns.

We also recall that a set of n linearly independent vectors in $\mathbb{R}^n$ is a spanning set for $\mathbb{R}^n$.

Therefore, the columnspace of $\mathbf{P}$ is all of $\mathbb{R}^3$. Similarly, the columnspace of $\mathbf{P}^{-1}$ is all of $\mathbb{R}^3$.

So the columnspace $\mathbf{P}$ is the same as the columnspace of $\mathbf{P}^{-1}$.