EE263 Homework 3 Solutions
Fall 2023

3.250. Color perception. Human color perception is based on the responses of three different types of color light receptors, called *cones*. The three types of cones have different spectral-response characteristics, and are called L, M, and, S because they respond mainly to long, medium, and short wavelengths, respectively. In this problem we will divide the visible spectrum into 20 bands, and model the cones' responses as follows:

$$L_{\text{cone}} = \sum_{i=1}^{20} l_i p_i, \qquad M_{\text{cone}} = \sum_{i=1}^{20} m_i p_i, \qquad S_{\text{cone}} = \sum_{i=1}^{20} s_i p_i,$$

where p_i is the incident power in the ith wavelength band, and l_i, m_i and s_i are nonnegative constants that describe the spectral responses of the different cones. The perceived color is a complex function of the three cone responses, *i.e.*, the vector $(L_{\text{cone}}, M_{\text{cone}}, S_{\text{cone}})$, with different cone response vectors perceived as different colors. (Actual color perception is a bit more complicated than this, but the basic idea is right.)

a) *Metamers.* When are two light spectra, p and $\tilde{p}$, visually indistinguishable? (Visually identical lights with different spectral power compositions are called *metamers*.)

b) *Visual color matching.* In a color matching problem, an observer is shown a test light, and is asked to change the intensities of three primary lights until the sum of the primary lights looks like the test light. In other words, the observer is asked the find a spectrum of the form

$$p_{\text{match}} = a_1 u + a_2 v + a_3 w,$$

where u, v, w are the spectra of the primary lights, and a_i are the intensities to be found, that is visually indistinguishable from a given test light spectrum p_{test}. Can this always be done? Discuss briefly.

c) *Visual matching with phosphors.* A computer monitor has three phosphors, R, G, and B. It is desired to adjust the phosphor intensities to create a color that looks like a reference test light. Find weights that achieve the match or explain why no such weights exist. The data for this problem is in `color_perception_data.json`, which contains the vectors `wavelength`, `B_phosphor`, `G_phosphor`, `R_phosphor`, `L_coefficients`, `M_coefficients`, `S_coefficients`, and `test_light`.

d) *Effects of illumination.* An object's surface can be characterized by its reflectance (*i.e.*, the fraction of light it reflects) for each band of wavelengths. If the object is illuminated with a light spectrum characterized by I_i, and the reflectance of the object is r_i (which is between 0 and 1), then the reflected light spectrum is given by $I_i r_i$, where $i = 1, \dots, 20$ denotes the wavelength band. Now consider two objects illuminated (at different times) by two different light sources, say an incandescent bulb and sunlight. Sally argues that if the two objects look identical when illuminated by a tungsten bulb, then they will look identical when illuminated by sunlight. Beth disagrees: she says that two objects can appear identical when illuminated by a tungsten bulb, but look different when lit by sunlight. Who is right? If Sally is right, explain why. If Beth is right give an example of two objects that appear identical under one light source and different under another. You can use the vectors `sunlight` and `tungsten` defined in the data file as the light sources.

Remark. Spectra, intensities, and reflectances are all nonnegative quantities, which the material of EE263 doesn't address. So just ignore this while doing this problem. These issues can be handled using the material of EE364a, however.

Solution.

a) Let

$$A = \begin{bmatrix} l_1 & l_2 & l_3 & \cdots & l_{20} \\ m_1 & m_2 & m_3 & \cdots & m_{20} \\ s_1 & s_2 & s_3 & \cdots & s_{20} \end{bmatrix}.$$

Now suppose that $c = Ap$ is the cone response to the spectrum p and $\tilde{c} = A\tilde{p}$ is the cone response to spectrum $\tilde{p}$. If the spectra are indistinguishable, then $c = \tilde{c}$ and $Ap = A\tilde{p}$. Solving the last expression for zero gives $A(p - \tilde{p}) = 0$. In other words, p and $\tilde{p}$ are metamers if $(p - \tilde{p}) \in \nu llspace(A)$.

b) In symbols, the problem asks if it is always possible to find nonnegative a_1, a_2, and a_3 such that

$$\begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = Ap_{\text{test}} = A \begin{bmatrix} u & v & w \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}.$$

Let $P = \begin{bmatrix} u & v & w \end{bmatrix}$ and let $B = AP$. If B is invertible, then

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = B^{-1} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix}.$$

However, B is not necessarily invertible. For example, if $\text{rank}(A) < 3$ or $\text{rank}(P) < 3$ then B will be singular. Physically, A is full rank if the L, M, and S cone responses are linearly independent, which they are. The matrix P is full rank if and only if the spectra of the primary lights are independent. Even if both A and P are full rank, B could still be singular. Primary lights that generate an invertible B are called *visually independent*. If B is invertible, a_1, a_2, and a_3 exist that satisfy

$$Ap_{\text{test}} = A \begin{bmatrix} u & v & w \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}.$$

One or more of the a_i may be negative in which case in the experimental setup described, no match would be possible. However, in a more complicated experimental setup that allows the primary lights to be combined either with each other or with p_{test}, a match is always possible if B is invertible. In this case, if $a_i < 0$, the ith light should be mixed with p_{test} instead of the other primary lights. For example, suppose $a_1 < 0$, $a_2, a_3 \geq 0$ and $b_1 = -a_1$, then

$$A(b_1 u + p_{\text{test}}) = A(a_2 v + a_3 w),$$

and each spectrum has a nonnegative weight.

c) Weights can be found as described above. The R, G, and B phosphors should be weighted by 0.4226, 0.0987, and 0.5286 respectively.

d) Beth is right. Let r and $\tilde{r}$ be the reflectances of two objects and let p and $\tilde{p}$ be two spectra. Let A be defined as before. Then, the objects will look identical under p if

$$A \underbrace{\begin{bmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & r_{20} \end{bmatrix}}_{R} p = A \underbrace{\begin{bmatrix} \tilde{r}_1 & 0 & \cdots & 0 \\ 0 & \tilde{r}_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \tilde{r}_{20} \end{bmatrix}}_{\tilde{R}} p.$$

This is equivalent to saying $(R - \tilde{R})p \in \nu llspace(A)$. The objects will look different under $\tilde{p}$ if, additionally, $AR\tilde{p} \neq A\tilde{R}\tilde{p}$ which means that $(R - \tilde{R})\tilde{p} \notin \nu llspace(A)$. The following code shows how to find reflectances r_1 and r_2 for two objects such that the objects will have the same color under tungsten light and will have different colors under sunlight.

Here is Julia code solving this problem.

```
using LinearAlgebra
include("readclassjson.jl");
data = readclassjson("color_perception_data.json");
L_coefficients = data["L_coefficients"];
M_coefficients = data["M_coefficients"];
S_coefficients = data["S_coefficients"];
R_phosphor = data["R_phosphor"];
G_phosphor = data["G_phosphor"];
B_phosphor = data["B_phosphor"];
test_light = data["test_light"];
tungsten = data["tungsten"];
sunlight = data["sunlight"];

# PART C

A = [L_coefficients'; M_coefficients'; S_coefficients'];
P = [R_phosphor G_phosphor B_phosphor]
B = A*P
@show weights = B\(A*test_light)

# PART D

tung_L = L_coefficients .* tungsten
tung_M = M_coefficients .* tungsten
tung_S = S_coefficients .* tungsten
tungsten_LMS = [transpose(tung_L); transpose(tung_M); transpose(tung_S)]
nullspace_LMS = nullspace(tungsten_LMS)[:,1];
```

```
r1 = [0; 0.2; 0.3; 0.7; 0.7; 0.8; 0.8; 0.2; 0.9; 0.8;
0.2; 0.8; 0.9; 0.2; 0.8; 0.3; 0.8; 0.7; 0.2; 0.4];
r2 = r1 - nullspace_LMS;
t1 = zeros(20);
t2 = zeros(20);
for i in 1:20
t1[i] = r1[i]*tungsten[i];
t2[i] = r2[i]*tungsten[i];
end
@show color1_tungsten = A*t1
@show color2_tungsten = A*t2

s1 = zeros(20);
s2 = zeros(20);
for i in 1:20
s1[i] = r1[i]*sunlight[i];
s2[i] = r2[i]*sunlight[i];
end
@show color1_sunlight = A*s1
@show color2_sunlight = A*s2
```

3.300. Orthogonal complement of a subspace. If $\mathcal{V}$ is a subspace of $\mathbb{R}^n$ we define $\mathcal{V}^\perp$ as the set of vectors orthogonal to every element in $\mathcal{V}$, *i.e.*,

$$\mathcal{V}^\perp = \{\ x \mid \langle x, y\rangle = 0,\ \forall y \in \mathcal{V}\ \}.$$

a) Verify that $\mathcal{V}^\perp$ is a subspace of $\mathbb{R}^n$.

b) Suppose $\mathcal{V}$ is described as the span of some vectors $v_1, v_2, \ldots, v_r$. Express $\mathcal{V}$ and $\mathcal{V}^\perp$ in terms of the matrix $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_r \end{bmatrix} \in \mathbb{R}^{n\times r}$ using common terms (range, nullspace, transpose, etc.)

c) Show that every $x \in \mathbb{R}^n$ can be expressed uniquely as $x = v + v^\perp$ where $v \in \mathcal{V}$, $v^\perp \in \mathcal{V}^\perp$. *Hint:* let v be the projection of x on $\mathcal{V}$.

d) Show that $\dim \mathcal{V}^\perp + \dim \mathcal{V} = n$.

e) Show that $\mathcal{V} \subseteq \mathcal{U}$ implies $\mathcal{U}^\perp \subseteq \mathcal{V}^\perp$.

Solution.

a) We do not need to check all properties of a vector space to hold for $\mathcal{V}^\perp$, since many of them hold only because $\mathcal{V}^\perp \subseteq \mathbb{R}^n$ and the vector sum and scalar product definitions over $\mathcal{V}^\perp$ and $\mathbb{R}^n$ are the same. We only need to verify the following properties:

- $0 \in \mathcal{V}^\perp$.
- $\forall x_1, x_2 \in \mathcal{V}^\perp :\ x_1 + x_2 \in \mathcal{V}^\perp$.
- $\forall \alpha \in \mathbb{R},\ \forall x \in \mathcal{V}^\perp :\ \alpha x \in \mathcal{V}^\perp$.

The first property comes from the fact that $\langle 0, y\rangle = 0$ for all $y \in \mathbb{R}^n$ and therefore $0 \in \mathcal{V}^\perp$. To verify the second property, we pick two arbitrary elements x_1 and x_2 in $\mathcal{V}^\perp$ and show that $x_1 + x_2 \in \mathcal{V}^\perp$. Let y be any vector in $\mathbb{R}^n$. We have

$$\begin{aligned} \langle x_1 + x_2, y\rangle &= \langle x_1, y\rangle + \langle x_2, y\rangle \\ &= 0 + 0 \qquad\qquad (\text{since } x_1 \in \mathcal{V}^\perp \text{ and } x_2 \in \mathcal{V}^\perp) \\ &= 0, \end{aligned}$$

and therefore $x_1 + x_2 \in \mathcal{V}^\perp$. Finally, we show that if $\alpha \in \mathbb{R}$ and $x \in \mathcal{V}^\perp$ then $\alpha x \in \mathcal{V}^\perp$. If $y \in \mathbb{R}^n$ is arbitrary

$$\begin{aligned} \langle \alpha x, y\rangle &= \alpha \langle x, y\rangle \\ &= \alpha \cdot 0 \qquad\qquad (\text{since } x \in \mathcal{V}^\perp) \\ &= 0, \end{aligned}$$

which by definition of $\mathcal{V}^\perp$, proves that $\alpha x \in \mathcal{V}^\perp$ and we are done.

b) Expressing $\mathcal{V}$ in terms of the matrix V is easy. The span of vectors $v_1, v_2, \ldots, v_r$ is simply all linear combinations of the columns of V and therefore $\mathcal{V} = \text{range}(V)$. To express $\mathcal{V}^\perp$ in terms of V we use the trivial fact that $x \in \mathcal{V}^\perp$ if and only if $x \perp v_i$ for $i = 1, \ldots, r$. (If

$x \perp v_i$ then x is orthogonal to any linear combination of the v_i's and hence any element in $\mathcal{V}^\perp$. If $x \in \mathcal{V}^\perp$ then x is specially orthogonal to the vectors $v_i \in \mathcal{V}^\perp$ for $i = 1, \dots, r$.) Therefore $x \in \mathcal{V}^\perp$ if and only if $v_i^\mathsf{T} x = 0$ for $i = 1, \dots, r$. In other words, using matrix notation, $x \in \mathcal{V}^\perp$ if and only if

$$\begin{bmatrix} v_1^\mathsf{T} \\ v_2^\mathsf{T} \\ \vdots \\ v_r^\mathsf{T} \end{bmatrix} x = 0$$

or $V^\mathsf{T} x = 0$. Therefore $\mathcal{V}^\perp = \mathrm{null}(V^\mathsf{T})$.

c) Suppose that $w_1, w_2, \dots, w_k$ is an orthonormal basis for $\mathcal{V}$. Consider the projection of x on $\mathcal{V}$, *i.e.*,

$$v := (w_1^\mathsf{T} x) w_1 + (w_2^\mathsf{T} x) w_2 + \cdots + (w_k^\mathsf{T} x) w_k.$$

Clearly, $v \in \mathcal{V}$ because it is a linear combination of the basis vectors w_i. Now we show that $x - v$ (projection error) is an element in $\mathcal{V}^\perp$. To do this we only have to verify that $x - v \perp w_i$ or $w_i^\mathsf{T}(x - v) = 0$ for $i = 1, \dots, k$. This is easy because

$$\begin{aligned} w_i^\mathsf{T}(x - v) &= w_i^\mathsf{T} x - w_i^\mathsf{T} v \\ &= w_i^\mathsf{T} x - (w_i^\mathsf{T} x) w_i^\mathsf{T} w_i \qquad && \text{since } w_i^\mathsf{T} w_j = 0 \text{ for } i \neq j \\ &= 0 && \text{since } w_i^\mathsf{T} w_i = 1 \end{aligned}$$

Now that $x - v \in \mathcal{V}^\perp$, define $v^\perp \in \mathcal{V}^\perp$ as $v^\perp = x - v$ so $x = v + v^\perp$ with $v \in \mathcal{V}$ and $v^\perp \in \mathcal{V}^\perp$. Now we show that the decomposition $x = v + v^\perp$ is unique. Suppose that there are two ways to express x as the sum of elements in $\mathcal{V}$ and $\mathcal{V}^\perp$, *i.e.*, $x = v_1 + v_1^\perp$ and $x = v_2 + v_2^\perp$ where $v_1, v_2 \in \mathcal{V}$ and $v_1^\perp, v_2^\perp \in \mathcal{V}^\perp$. Therefore $v_1 + v_1^\perp = v_2 + v_2^\perp$ or $v_1 - v_2 = v_1^\perp - v_2^\perp$. But $v_1 - v_2 \in \mathcal{V}$ (because $v_1, v_2 \in \mathcal{V}$) and $v_1^\perp - v_2^\perp \in \mathcal{V}^\perp$ (because $v_1^\perp, v_2^\perp \in \mathcal{V}^\perp$), and by definition of $\mathcal{V}^\perp$ we should have $(v_1 - v_2) \perp (v_1^\perp - v_2^\perp)$ or $(v_1 - v_2)^\mathsf{T}(v_1^\perp - v_2^\perp) = 0$. Now since $v_1 - v_2 = v_1^\perp - v_2^\perp$ this implies that

$$(v_1 - v_2)^\mathsf{T}(v_1 - v_2) = \|v_1 - v_2\|^2 = 0$$

and

$$(v_1^\perp - v_2^\perp)^\mathsf{T}(v_1^\perp - v_2^\perp) = \|v_1^\perp - v_2^\perp\|^2 = 0$$

so $v_1 = v_2$ and $v_1^\perp = v_2^\perp$ or the decomposition is *unique*.

d) This follows from the previous part. In part (c) we showed that any vector in $\mathbb{R}^n$ can be expressed as the sum of two elements in $\mathcal{V}$ and $\mathcal{V}^\perp$. Therefore, if $\{w_i\}_{i=1}^k$ is a basis for $\mathcal{V}$ and $\{u_i\}_{i=1}^l$ is a basis for $\mathcal{V}^\perp$, for arbitrary $x \in \mathbb{R}^n$ the scalars α_i and β_i exist such that

$$x = \sum_{i=1}^k \alpha_i w_i + \sum_{i=1}^l \beta_i u_i$$

or the set of vectors $\{w_1, \dots, w_k, u_1, \dots, u_l\}$ span $\mathbb{R}^n$. In fact, the vectors w_i for $i = 1, \dots, k$ are orthogonal to the vectors u_i for $i = 1, \dots, l$ by the definition of $\mathcal{V}^\perp$ and are therefore independent. Since the set of vectors $\{w_1, \dots, w_k, u_1, \dots, u_l\}$ span $\mathbb{R}^n$ and $w_1, \dots, w_k, u_1, \dots, u_l$ are independent we get

$$\dim \mathcal{V} + \dim \mathcal{V}^\perp = k + l = n.$$

e) To show that $\mathcal{U}^\perp \subseteq \mathcal{V}^\perp$ we take an arbitrary element $x \in \mathcal{U}^\perp$ and prove that $x \in \mathcal{V}^\perp$. Since $x \in \mathcal{U}^\perp$ then $x \perp y$ for all $y \in \mathcal{U}$. But $\mathcal{V} \subseteq \mathcal{U}$ so we also have $x \perp y$ for all $y \in \mathcal{V}$. By definition of $\mathcal{V}^\perp$, this is nothing but to state that $x \in \mathcal{V}^\perp$ and we are done.

3.430. Single sensor failure detection and identification. We have $y = Ax$, where $A \in \mathbb{R}^{m\times n}$ is known, and $x \in \mathbb{R}^n$ is to be found. Unfortunately, up to one sensor may have failed (but you don't know which one has failed, or even whether any has failed). You are given $\tilde{y}$ and not y, where $\tilde{y}$ is the same as y in all entries except, possibly, one (say, the kth entry). If all sensors are operating correctly, we have $y = \tilde{y}$. If the kth sensor fails, we have $\tilde{y}_i = y_i$ for all $i \neq k$.

The file `one_bad_sensor.json`, available on the course web site, defines A and $\tilde{y}$ (as `A` and `ytilde`). Determine which sensor has failed (or if no sensors have failed). You must explain your method, and submit your code.

For this exercise, you can use the matlab code `rank([F g])==rank(F)` to check if $g \in \text{range}(F)$. (We will see later a much better way to check if $g \in \text{range}(F)$.)

Solution. Let $y^{(i)}$ be the measurement vector y with the ith entry removed. Likewise, let $A^{(i)}$ be the measurement matrix with the ith row of A removed. This corresponds to the system without the ith sensor.

If the ith sensor is faulty, we will almost surely have $y \notin \text{range}(A)$ (unless the sensor failure happens to give the same response y_i as that predicted by A, which is highly unlikely). However, once we remove its faulty measurement, we will certainly have $y^{(i)} \in \text{range}(A^{(i)})$.

To test if a vector z is in $\text{range}(C)$, we can use matlab and compare `rank([C z]) == rank(C)`. If they are equal, $z \in \text{range}(C)$. Otherwise `rank([C z]) == rank(C) + 1`. To find a faulty sensor, we remove one row of A at a time, and use the above test.

The following matlab code solves the problem

```
using LinearAlgebra
include("readclassjson.jl")

data = readclassjson("one_bad_sensor.json")
A = data["A"]
ytilde = data["ytilde"]

if rank(A) == rank(hcat(A, ytilde))
    println("No sensor fault")
else
    m, n = size(A)
    for i in 1:m
        noti = 1:m .!= i
        if rank(A[noti, :]) == rank(hcat(A[noti, :], ytilde[noti]))
            println("Fault in sensor $i")
        end
    end
end
```

The 11th sensor is faulty.

3.680. Coin collector robot. Consider a robot with unit mass which can move in a frictionless two dimensional plane. The robot has a constant unit speed in the y direction (towards north), and it is designed such that we can only apply force in the x direction. We will apply a force at time t given by f_j for $2j-2 \leq t < 2j$ where $j = 1, \ldots, n$, so that the applied force is constant over time intervals of length 2. The robot is at the origin at time $t = 0$ with zero velocity in the x direction.

There are $2n$ coins in the plane and the goal is to design a sequence of input forces for the robot to collect the maximum possible number of coins. The robot is designed such that it can collect the ith coin only if it exactly passes through the location of the coin (x_i, y_i). In this problem, we assume that $y_i = i$.

a) Find the coordinates of the robot at time t, where t is a positive integer. Your answer should be a function of t and the vector of input forces $f \in \mathbf{R}^n$.

b) Given a sequence of k coins $(x_1, y_1), \ldots, (x_{2n}, y_{2n})$, describe a method to find whether the robot can collect them.

c) For the data provided in `robot_coin_collector.json`, show that the robot cannot collect all the coins.

d) Suppose that there is an arrangement of the coins such that it is not possible for the robot to collect all the coins. Suggest a way to check if the robot can collect all but one of the coins.

e) Run your method on data in `robot_coin_collector.json` and report which coin cannot be collected. Report the input that results in collecting $2n-1$ coins. Plot the location of the coins and the location of the robot at integer times.

Solution.

a) The second coordinate at time t is simply equal to t.

Consider $A \in \mathbb{R}^{2n \times n}$ such that

$$A_{ij} = \begin{cases} 1 & j = \lfloor \frac{i+1}{2} \rfloor \\ 0 & \text{Otherwise.} \end{cases}$$

Then we will have $Af = [f_1, f_1, f_2, \cdots, f_n]$. Similar to the mass/force example, the first coordinate at time t will be equal to $b_t^T Af$ where

$$b_t = [t - \frac{1}{2}, \cdots, \frac{1}{2}, 0, \cdots, 0]^T.$$

b) According to part a, the only possible time to collect the ith coin is at time $t = y_i = i$. Define l_i to be the first coordinate of the location of the robot at time $t = i$. From part a, we see that

$$l_i = b_t^T Af.$$

Let $B \in \mathbb{R}^{n \times n}$ be a matrix whose ith column is b_i and define $C = B^T A$. Then we will have $l = Cf$.

Hence, we see that the necessary and sufficient condition to collect all the coins is that $x \in \text{range}(C)$. This can be simply examined with `rank([C x])==rank(C)`.

c) The code to solve parts c,e can be find at the bottom.

d) In part b, we saw that $l = Cf$. We know that there exists a sequence of input forces f such that all but one of the $2n$ equations are satisfied, but we don't know which one.

Let $x^{(i)}$ be the location vector x with the ith entry removed. Likewise, let $C^{(i)}$ be the transition matrix with the ith row of C removed. If we can collect all coins but the ith one, then we will certainly have $x^{(i)} \in \text{range}(C^{(i)})$. We will loop over the coins and see whether it's possible to collect all coins but one.

e) The following code solves the problem:

```
using LinearAlgebra
using Plots

include("readclassjson.jl")
data = readclassjson("robot_coin_collector.json")

n = data["n"]
x = data["x"]

BT = zeros(2*n, 2*n)

for i = 1:2*n
    BT[i,1:i] = i-1/2:-1:1/2
end

A = zeros(2*n,n)
for i=1:n
    A[2*i-1, i], A[2*i, i] = 1, 1
end
C = BT*A

if rank([C x])==rank(C)
    println("All coins can be collected!")
else
    println("All coins cannot be collected!")
end

for i=1:2*n
    xt = x[1:end .!= i]
    Ct = C[1:end .!= i,:]
    if rank([Ct xt])==rank(Ct)
        println("The robot can collect all coins but ", i, "th, ")
        print("and the input will be: \n")
        global input = round.(Ct\xt, digits=4)
        display(input)
```

```
    end
end

plot(C*input, 1:2*n, color = "red", label = "Robot Trajectory")
scatter!(x, 1:2*n, color = "blue", label = "Coin Locations")
```

We see that all coins but the 7^{th} can be collected and the associated input will be

$$f = [1.0000, -4.0000, 7.0000, -10.0000, 20.0000, -35.0000].$$

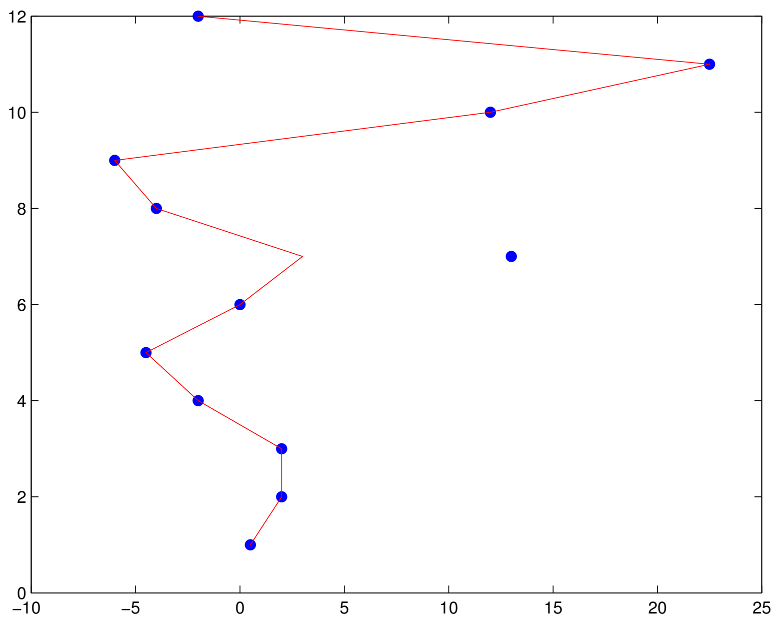

Figure 1: Location of the coins and the trajectory of the robot

4.630. Groups of equivalent statements. In the list below there are 11 statements about two square matrices A and B in $\mathbb{R}^{n\times n}$.

a) $\text{range}(B) \subseteq \text{range}(A)$.

b) there exists a matrix $Y \in \mathbb{R}^{n\times n}$ such that $B = YA$.

c) $AB = 0$.

d) $BA = 0$.

e) $\text{rank}(\begin{bmatrix} A & B \end{bmatrix}) = \text{rank}(A)$.

f) $\text{range}(A) \perp \text{null}(B^{\mathsf{T}})$.

g) $\text{rank}(\begin{bmatrix} A \\ B \end{bmatrix}) = \text{rank}(A)$.

h) $\text{range}(A) \subseteq \text{null}(B)$.

i) there exists a matrix $Z \in \mathbb{R}^{n\times n}$ such that $B = AZ$.

j) $\text{rank}(\begin{bmatrix} A & B \end{bmatrix}) = \text{rank}(B)$.

k) $\text{null}(A) \subseteq \text{null}(B)$.

Your job is to collect them into (the largest possible) groups of equivalent statements. Two statements are equivalent if each one implies the other. For example, the statement 'A is onto' is equivalent to '$\text{null}(A) = \{0\}$' (when A is square, which we assume here), because every square matrix that is onto has zero nullspace, and vice versa. Two statements are not equivalent if there exist (real) square matrices A and B for which one holds, but the other does not. A group of statements is equivalent if any pair of statements in the group is equivalent.

We want *just* your answer, which will consist of lists of mutually equivalent statements; we do not need any justification.

Put your answer in the following specific form. List each group of equivalent statements on a line, in (alphabetic) order. Each new line should start with the first letter not listed above. For example, you might give your answer as

a, c, d, h
b, i
e
f, g, j, k.

This means you believe that statements a, c, d, and h are equivalent; statements b and i are equivalent; and statements f, g, j, and k are equivalent. You also believe that the first group of statements is not equivalent to the second, or the third, and so on.

Solution. Let b_i be the ith column of B.

$$\begin{aligned} \text{range}(B) \subseteq \text{range}(A) &\Leftrightarrow \text{every column of } B \text{ is in the range of } A \\ &\Leftrightarrow \text{there exists a vector } z_i \text{ such that } b_i = Az_i \\ &\Leftrightarrow \text{there exists a matrix } Z \in \mathbb{R}^{n\times n} \text{ such that } B = AZ \\ &\Leftrightarrow \text{rank}(\begin{bmatrix} A & B \end{bmatrix}) = \text{rank}(A). \end{aligned} \tag{1}$$

This shows that statements a, e and i are equivalent.

$$\begin{aligned} \text{null}(A) \subseteq \text{null}(B) &\Leftrightarrow \text{null}(A)^\perp \supseteq \text{null}(B)^\perp \\ &\Leftrightarrow \text{range}(B^\mathsf{T}) \subseteq \text{range}(A^\mathsf{T}) \\ &\Leftrightarrow \text{there exists a matrix } \tilde{Y} \in \mathbb{R}^{n\times n} \text{ such that } B^\mathsf{T} = A^\mathsf{T}\tilde{Y} \\ &\Leftrightarrow \text{there exists a matrix } Y \in \mathbb{R}^{n\times n} \text{ such that } B = YA \\ &\Leftrightarrow \text{rank}(\begin{bmatrix} A^\mathsf{T} & B^\mathsf{T} \end{bmatrix}) = \text{rank}(A^\mathsf{T}) \\ &\Leftrightarrow \text{rank}(\begin{bmatrix} A \\ B \end{bmatrix}) = \text{rank}(A). \end{aligned} \tag{2}$$

This shows that statements b, g and k are equivalent.

$$\begin{aligned} \text{range}(A) \subseteq \text{null}(B) &\Leftrightarrow \text{for all } z \in \mathbb{R}^n,\ B(Az) = 0 \\ &\Leftrightarrow BA = 0. \end{aligned} \tag{3}$$

This shows that statements d and h are equivalent.

$$\begin{aligned} \text{range}(A) \perp \text{null}(B^\mathsf{T}) &\Leftrightarrow \text{range}(A) \subseteq \text{null}(B^\mathsf{T})^\perp \\ &\Leftrightarrow \text{range}(A) \subseteq \text{range}(B) \\ &\Leftrightarrow \text{rank}(\begin{bmatrix} A & B \end{bmatrix}) = \text{rank}(B). \end{aligned} \tag{4}$$

This shows that statements f and j are equivalent.

None of these groups of statements is equivalent to any other, or to c. This is demonstrated by the following counterexamples.

Take

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Since $AB = 0$ but $BA \neq 0$, then group (3) and statement c are not equivalent. Furthermore since

$$\text{rank}(\begin{bmatrix} A \\ B \end{bmatrix}) = \text{rank}(A) = \text{rank}(B) = 1$$

but $\text{rank}([A \quad B]) = 2$, groups (2) and (1) are not equivalent. Groups (2) and (4) are not either.

When $A = B \neq 0$, $\text{null}(A) = \text{null}(B)$ but $AB = BA = A^2 \neq 0$. Hence groups (2) and (3) are not equivalent. Group (2) and statement c are not equivalent either.

Take

$$A = I, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Since $\text{rank}([AB]) = \text{rank}(A) = 2$ but $\text{rank}(B) = 1$, groups (1) and (4) are not equivalent. Furthermore since $BA \neq 0$ groups (1) and (3) are not equivalent. Since $AB \neq 0$, group (1) and statement c aren't either.

In a similar fashion, taking

$$A = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad B = I,$$

shows that groups (3) and (4) are not equivalent and that statement c and group (4) aren't either.

Thus, the final answer is

a, e, i

b, g, k

c

d, h

f, j.