

## EE263 Final Exam Solutions, December 2023

- This is a 24-hour take-home exam. Please turn it in on Gradescope. Be aware that you *must* turn it in within 24 hours of downloading it. After that, Gradescope will not let you turn it in and we cannot accept it.
- You may use any books, notes, or computer programs. You may not discuss the exam or course material with others, or work in a group.
- The exam should not be discussed at all, with anyone, until 12/13 after everyone has taken it.
- If you have a question, please submit a private question on Ed, or email the staff mailing list. We have tried very hard to make the exam unambiguous and clear, so unless there is a mistake on the exam we're unlikely to say much.
- We expect your solutions to be legible, neat, and clear. Do not hand in your rough notes, and please try to simplify your solutions as much as you can. We will deduct points from solutions that are technically correct, but much more complicated than they need to be.
- Please check your email during the exam, just in case we need to send out a clarification or other announcement.
- Start each question on a new page, and make sure to correctly assign pages to problems in gradescope.
- When a problem involves some computation (say, using Julia), we do not want just the final answers. We want a clear discussion and justification of exactly what you did as well as the final numerical result.
- **You must turn in your code.** Include the code in your pdf submission.
- In the portion of your solutions where you explain the mathematical approach, you *cannot* refer to Julia operators, such as the backslash operator. (You can, of course, refer to inverses of matrices, or any other standard mathematical constructs.)
- Some of the problems require you to download data or other files. These files can be found at the URL

<https://ee263.stanford.edu/grandfinale23>

- Good luck!

**1. Hovercraft racing.** You've just entered into the world's first Hovercraft Grand Prix! This high-stakes competition will put the controllability of your homemade hovercraft to the test. The primary goal of your hovercraft is to intercept **proximity targets** laid out in a track. The targets must be hit at particular times (so it's not really a race, it's a test of accuracy.) Your controller can apply forces in the  $x$  and  $y$  directions.

More specifically, you can apply piecewise constant force inputs  $u(t) = u_d(k)$  for  $k \leq t < k + 1$  where  $k \in \mathbb{Z}^+$  and  $u(t) \in \mathbb{R}^2$ . The state of the system is

$$x(t) = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix}$$

where  $q(t) \in \mathbb{R}^2$  is the position of the hovercraft at time  $t$ . The starting line of the race is at  $(0, 0)$  and the hovercraft must be stationary until the race begins. The hovercraft has mass 1.

- a) Find matrices  $A$  and  $B$  that model the hovercraft's dynamics exactly as a discrete-time linear dynamical system of the form

$$x(k + 1) = Ax(k) + Bu_d(k)$$

- b) You are given the  $m = 4$  target positions  $p_1 = (1, 1)$ ,  $p_2 = (0, 2)$ ,  $p_3 = (-1, 1)$  and  $p_4 = (0, 0)$ , and target times  $t_1 = 10$ ,  $t_2 = 40$ ,  $t_3 = 50$ , and  $t_4 = 60$ . Let  $t_{\max} = 65$ . Find the sequence of forces  $u_d(0), \dots, u_d(t_{\max})$  that minimizes

$$\sum_{k=0}^{t_{\max}} \|u_d(k)\|^2$$

such that the hovercraft passes through each of the target positions at the corresponding target time. Plot the trajectory of the hovercraft in  $\mathbb{R}^2$ . Plot the components of the input force vs. time.

- c) New regulations have been issued this year that now limit the total thrust. We would like to trade off

$$J_1 = \sum_{i=1}^m \|q(t_i) - p_i\|^2$$

and

$$J_2 = \sum_{k=0}^{t_{\max}} \|u_d(k)\|^2$$

While we may not be able to pass through each target exactly, we still get points for getting as close as possible to each one. Describe a method to do this. Plot the optimal trade-off curve of  $J_2$  versus  $J_1$ .

- d) Find the controller inputs that minimize  $J_1$  while achieving the total thrust limit

$$J_2 \leq 0.005$$

Plot the trajectory of the hovercraft in  $\mathbb{R}^2$ . Plot the components of the input force vs. time.

**Solution.**

- a) The dynamics of this system are described by  $\ddot{q} = u$  which leads to the continuous system equations of

$$A_{cont} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_{cont} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Then, by discretizing the system and letting  $h = 1$ , we get

$$A = e^{hA_{cont}} = I + hA_{cont} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \left( \int_0^h e^{\tau A} d\tau \right) B_{cont} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- b)

$$q(t) = y(t) = \tilde{A}_t u_d$$

where

$$\tilde{A}_t = [A^{t-1}B \quad A^{t-2}B \quad \dots \quad 0]$$

Then,

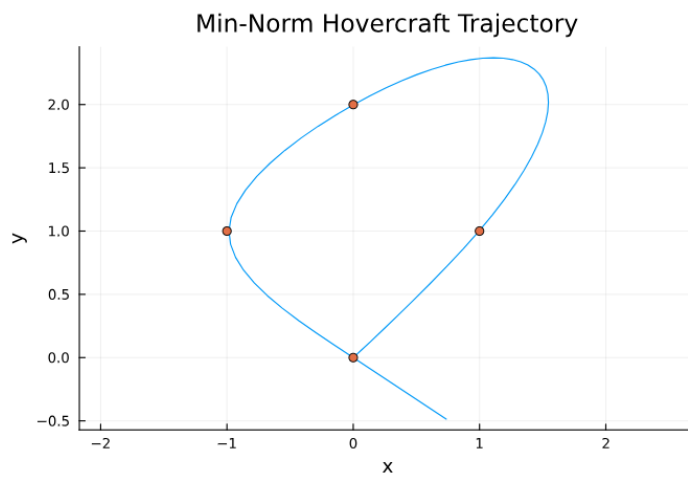
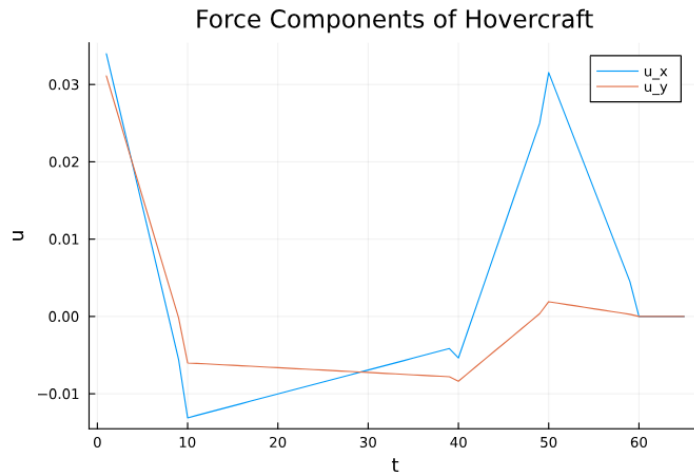
$$\begin{bmatrix} y(t_1) \\ y(t_2) \\ y(t_3) \\ y(t_4) \end{bmatrix} = \tilde{A} u_d$$

where

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{t_1} \\ \tilde{A}_{t_2} \\ \tilde{A}_{t_3} \\ \tilde{A}_{t_4} \end{bmatrix}$$

Then, the minimization problem becomes

$$\begin{aligned} & \min \|u_d\|^2 \text{ s.t. } y_{des} = \tilde{A}u_d \\ \implies & u_d = \tilde{A}^T (\tilde{A}\tilde{A}^T)^{-1} y_{des} = \tilde{A}^\dagger y_{des} \end{aligned}$$



c)

$$J_1 = \|\tilde{A}u_d - y_{des}\|^2$$

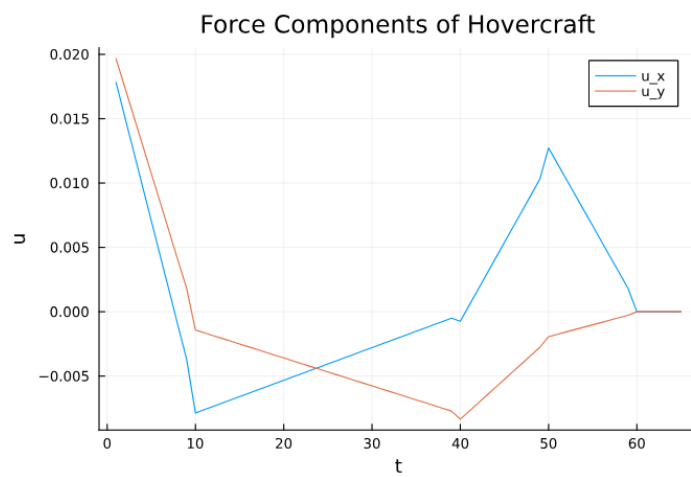
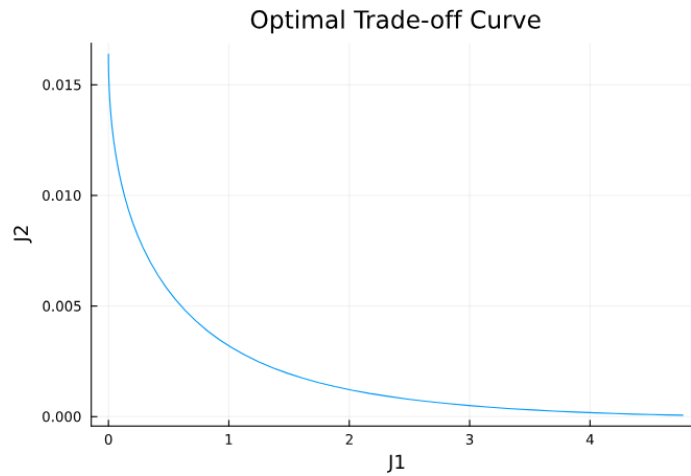
$$J_2 = \|u_d\|^2$$

$$\min J_1 + \mu J_2 \implies u_{seq} = \left(\tilde{A}^T \tilde{A} + \mu I\right)^{-1} \tilde{A}^T y_{des}$$

d) See Julia code below:

```
import Pkg
using LinearAlgebra
Pkg.add("RecipesBase")
import RecipesBase: plot
Pkg.add("Plots")

# Problem 1
```



# part b

```

A_tilde = zeros(8,130)
t1 = 10
t2 = 40
t3 = 50
t4 = 60
tmax = 65
A = [1 0 1 0;0 1 0 1;0 0 1 0;0 0 0 1]
B = [0.5 0;0 0.5;1 0;0 1]
C = [1 0 0 0;0 1 0 0]

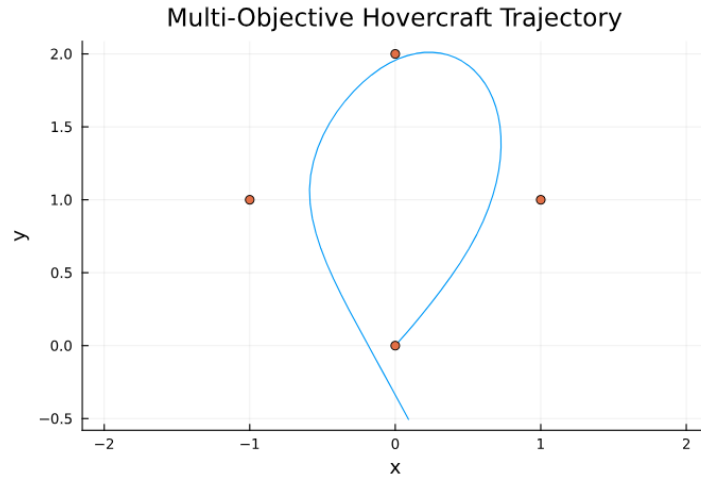
```

# Constructing Toeplitz Rows

```

for i = 1:tmax
    if i < t1
        A_tilde[1:2,2*(i-1)+1:2*i] = C*A^(t1-i)*B
    end
end

```



```

end
if i<t2
    A_tilde[3:4,2*(i-1)+1:2*i] = C*A^(t2-i)*B
end
if i<t3
    A_tilde[5:6,2*(i-1)+1:2*i] = C*A^(t3-i)*B
end
if i<t4
    A_tilde[7:8,2*(i-1)+1:2*i] = C*A^(t4-i)*B
end
end

end

# Solving using min-norm solution
y_des = [1;1;0;2;-1;1;0;0]
u_seq = pinv(A_tilde)*y_des

# Solving for position at each time step
xs = zeros(4,tmax+1)
for k = 1:tmax
    xs[:,k+1] = A*xs[:,k]+B*u_seq[2*(k-1)+1:2*k]
end

# Plotting
using Plots
plot(1:tmax,u_seq[1:2:end],title="Force Components of
    Hovercraft",label="u_x")
display(plot!(1:tmax,u_seq[2:2:end],label="u_y",xlabel="t"
    ,ylabel="u"))
plot(xs[1,:],xs[2:],aspect_ratio=:equal,xlabel="x"

```

```

        ,ylabel="y")
plot!(y_des[1:2:end],y_des[2:2:end],seriestype=:scatter,
      legend=false,title="Min-Norm Hovercraft Trajectory")

# Problem 1
# part c

p = LinRange(-2,4,100)
mu = 10 .^p
J1 = zeros(length(mu),1)
J2 = zeros(length(mu),1)
for i=1:length(mu)
    u_seq = inv(A_tilde'*A_tilde+mu[i]*I)*A_tilde'*y_des
    J1[i] = (norm(A_tilde*u_seq-y_des))^2
    J2[i] = (norm(u_seq))^2
end
plot(J1,J2,title="Optimal Trade-off Curve",xlabel="J1"
     ,ylabel="J2",legend=false)

# Problem 1
# part d

mu = 166
u_seq = inv(A_tilde'*A_tilde+mu*I)*A_tilde'*y_des
J1 = (norm(A_tilde*u_seq-y_des))^2
J2 = (norm(u_seq))^2
print(String("J2 = $J2"))
xs = zeros(4,tmax+1)
for k = 1:tmax
    xs[:,k+1] = A*xs[:,k]+B*u_seq[2*(k-1)+1:2*k]
end
using Plots
plot(1:tmax,u_seq[1:2:end],title="Force Components of
    Hovercraft",label="u_x")
display(plot!(1:tmax,u_seq[2:2:end],label="u_y",xlabel="t"
             ,ylabel="u"))
plot(xs[1,:],xs[2,:],aspect_ratio=:equal,xlabel="x"
     ,ylabel="y")
plot!(y_des[1:2:end],y_des[2:2:end],seriestype=:scatter
     ,legend=false,title="Multi-Objective Hovercraft Trajectory")

```

**2. Heat flow.** We consider here an example of a simple model for heat flow. We have a directed graph with  $m$  edges and  $n$  nodes. The incidence matrix  $A \in \mathbb{R}^{n \times m}$  is defined

by

$$A_{ij} = \begin{cases} 1 & \text{if edge } j \text{ points to node } i \\ -1 & \text{if edge } j \text{ starts at node } i \\ 0 & \text{otherwise} \end{cases}$$

An edge cannot start and end at the same node. There can be at most one edge joining any two nodes. The temperature at node  $i$  at time  $t$  is  $x_i(t)$ . Suppose we add heat to node  $i$  at rate  $s_i(t)$ . Let  $f_k(t)$  be the amount of heat flowing through edge  $k$  at time  $t$ . Then we have

$$\dot{x}(t) = s(t) + Af(t)$$

Now the rate of heat flow on an edge is proportional to the temperature difference at its endpoints, so we have

$$Kf = -A^T x$$

where  $K \in \mathbb{R}$  is a constant. Then  $1/K$  is the *thermal conductivity* of the edge. For convenience, in this question we will assume  $K = 1$ . This model is a linear dynamical system of the form

$$\dot{x}(t) = Qx(t) + s$$

where  $Q = -AA^T$ .

- a) Show that for any graph  $Q$  has at least one zero eigenvalue. What is the corresponding eigenvector?
- b) We will first consider the unforced system

$$\dot{x}(t) = Qx(t)$$

What are the equilibrium solutions of this system? Give a brief one-sentence interpretation.

- c) A matrix  $P \in \mathbb{R}^{n \times n}$  is called *off-diagonal nonnegative* if

$$P_{ij} \geq 0 \quad \text{for } i \neq j$$

Show that  $Q$  is off-diagonal nonnegative.

- d) We would like our model to only produce nonnegative temperatures. Consider a general linear dynamical system

$$\dot{x}(t) = Px(t)$$

For every node  $i$  we would like the quantity  $x_i(t)$  to be nonnegative for all time. That is, if  $x_i(0) \geq 0$  for all  $i$ , then

$$x_i(t) \geq 0 \quad \text{for all } i \text{ and all } t \geq 0$$

Show that this implies that  $P$  is off-diagonal nonnegative.



- e) Recall that a matrix  $P$  is called *Hurwitz* if all of its eigenvalues have negative real parts, and it is called *stable* if  $e^{tP} \rightarrow 0$  as  $t \rightarrow \infty$ . A matrix is Hurwitz if and only if it is stable.

Suppose  $P$  has the property that

$$e^{tP}x(0) \rightarrow 0 \quad \text{for all } x(0) \geq 0$$

where  $x(0) \geq 0$  means that all elements of  $x(0)$  are nonnegative. Can we conclude that  $P$  is Hurwitz? If you think the answer is no, give an example, otherwise give a proof.

- f) Suppose now that the source  $s(t)$  is constant, so that the dynamics become

$$\dot{x}(t) = Qx(t) + s$$

where  $s \in \mathbb{R}^n$  is a vector which does not depend on time. Show that

$$Qx(t) = (e^{tQ} - I)s + Qe^{tQ}x(0)$$

### Solution.

- a)  $A$  is an incidence matrix, the elements of each column sums to 0. This means that for  $A^T$  each row sums to 0  $\implies A^T \mathbf{1} = 0$   
 $Q\mathbf{1} = -AA^T \mathbf{1} = -A0 = 0$   
 $Q$  has 0 as an eigenvalue and the vector  $\mathbf{1}$  is the corresponding eigenvector.
- b) At equilibrium all the nodes are at a constant temperature and there is no heat transfer. At equilibrium we require  $Qx(t) = 0$ . This means that  $x(t) \in \text{null}(Q)$ . In part a we identified that  $\mathbf{1}$  is one such vector. Any vector of all constants would be an equilibrium solution.
- c) Let us consider each element of  $Q_{ij}$ ;  $Q_{ij} = -\sum_{k=1}^m A_{ik}A_{jk}$  where  $k$  is an edge. There are three cases to consider (1) edge  $k$  starts at  $i$  and points to  $j \implies A_{ik} = -1$  and  $A_{jk} = 1$ , (2) edge  $k$  starts at  $j$  and points to  $i \implies A_{ik} = 1$  and  $A_{jk} = -1$  or (3) edge  $k$  does not connect  $i$  and  $j$ . In all three cases the product  $A_{ik}A_{jk}$  is non-positive. Hence,  $Q_{ij}$  is non-negative.
- d) For the system:  $\dot{x} = Px(t), x(t) = e^{tP}x(0)$   
 If  $x_i(0) \geq 0$  and  $x_i(t) \geq 0$ ,  $\implies e^{tP} \geq 0$   
 Let us define  $P^* = P + \alpha I$  where  $\alpha = |\min(\text{diagonal entries of } P)|$   
 $\implies P = P^* - \alpha I$   
 $e^{tP} = e^{tP^*} e^{-t\alpha}$   
 Since both  $e^{tP^*}$  and  $e^{-t\alpha}$  are non-negative,  $\implies e^{tP}$  is non-negative.  
 Based on the expansion of the exponential,  $P^*$  must have all non-negative entries.  $P$  and  $P^*$  share the same off-diagonal entries. Hence,  $P$  is off-diagonal non-negative.

e) We can express  $x(0)$  as  $\sum_{i=1}^n \alpha_i e_i$ . Since  $e^{tP} x(0) \rightarrow 0$ ,  $e^{tP} e_i \rightarrow 0$  for all  $i$ . This implies that each column of  $(e^{tP})_i \rightarrow 0$ . We can conclude that  $e^{tP} \rightarrow 0$  and hence  $P$  is Hurwitz.

f) The solution to this system is:

$$x(t) = e^{tQ} x(0) + \int_0^t e^{(t-\tau)Q} I s d\tau$$

$$Qx(t) = Qe^{tQ} x(0) + Q \int_0^t e^{(t-\tau)Q} I s d\tau$$

$$\begin{aligned} \int_0^t e^{(t-\tau)Q} d\tau &= \int_0^t I + (t-\tau)Q + \frac{(t-\tau)^2 Q^2}{2!} + \frac{(t-\tau)^3 Q^3}{3!} + \dots d\tau \\ &= \tau - \frac{(t-\tau)^2 Q}{2!} - \frac{(t-\tau)^3 Q^2}{3!} - \dots \Big|_0^t \\ &= t + \frac{(t)^2 Q}{2!} + \frac{(t)^3 Q^2}{3!} + \dots \end{aligned}$$

Hence,

$$\begin{aligned} Q \int_0^t e^{(t-\tau)Q} d\tau &= tQ + \frac{(t)^2 Q^2}{2!} + \frac{(t)^3 Q^3}{3!} + \dots \\ &= e^{tQ} - I \end{aligned}$$

Substituting this back we obtain:

$$Qx(t) = Qe^{tQ} x(0) + (e^{tQ} - I)s$$

**3. Some true/false questions.** For each statement, give a proof if it is true, or a counterexample if it is false.

- Suppose  $A \in \mathbb{R}^{n \times n}$  is invertible, and let  $A = QR$  be its full QR factorization with  $Q$  orthogonal and  $R$  upper triangular. Let  $B = RQ$ . If  $A$  is symmetric, then  $B$  is symmetric.
- Suppose  $A \in \mathbb{R}^{n \times n}$  is invertible, and let  $A = QR$  be its full QR factorization with  $Q$  orthogonal and  $R$  upper triangular. Let  $B = RQ$ . If  $A$  is upper triangular, then  $B$  is upper triangular.
- Suppose  $A \in \mathbb{R}^{n \times n}$  is symmetric with eigenvalues  $\lambda_1 \geq \dots \geq \lambda_n$ . Of all vectors  $x$  with  $\|x\| = 1$ , the  $x$  that maximizes  $(x^T A x)^2$  is the eigenvector corresponding to the largest eigenvalue  $\lambda_1$ .

d) Suppose  $A \in \mathbb{R}^{n \times n}$  is symmetric with eigenvalues  $\lambda_1 \geq \dots \geq \lambda_n$ . Of all vectors  $x$  with  $\|x\| = 1$ , the  $x$  that minimizes  $(x^T Ax)^2$  is the eigenvector corresponding to the smallest eigenvalue  $\lambda_n$ .

e) For any matrices  $A, B, C$  we have

$$\text{range}(ABC) \subseteq \text{range}(A)$$

f)  $\|Ax\| \leq \|A\|_F \|x\|$  for all matrices  $A \in \mathbb{R}^{m \times n}$  and vectors  $x \in \mathbb{R}^n$ . Here  $\|A\|_F$  denotes the Frobenius norm.

g) If  $A \in \mathbb{R}^{n \times n}$  has all eigenvalues  $\lambda$  satisfying  $|\lambda| < 1$ , then  $\|A^2\| \leq \|A\|$ .

**Solution.** Here is the solution.

a) **True:** Suppose  $A$  is symmetric. Since  $A = QR$ ,  $R = Q^T A$ . Therefore,  $B = RQ = Q^T A Q$ . Then,  $B^T = (Q^T A Q)^T = Q^T A^T (Q^T)^T = Q^T A Q$ . Therefore,  $B$  is also symmetric.

b) **True:** Suppose  $A$  is upper triangular. Then, its QR decomposition is given by  $A = IA$ , since  $I$  is orthogonal and  $A$  is upper triangular, and the QR decomposition is unique for nonsingular matrices (up to the sign). Therefore,  $B = AI = A$  is also upper triangular.

c) **False:** Consider the symmetric matrix  $A = \text{diag}(3, 0, -4)$ . The largest eigenvalue is  $\lambda_1 = 3$  with eigenvector  $e_1$ . However,  $(x^T Ax)^2$  is maximized by the third eigenvector  $e_3$  associated with  $\lambda_3 = -4$ , since  $(-4)^2 > 3^2$ .

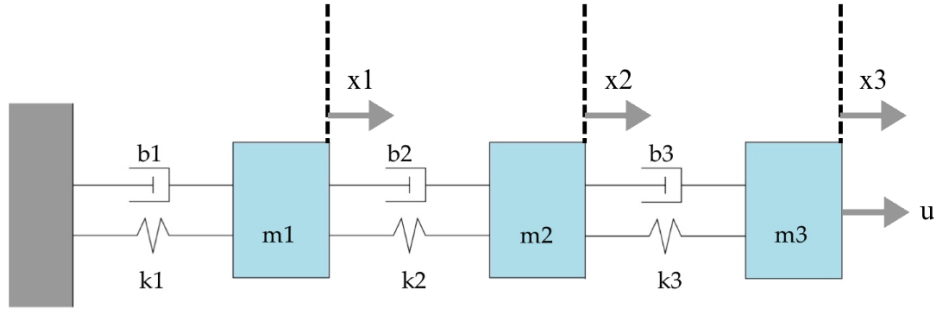
d) **False:** Consider the symmetric matrix  $A = \text{diag}(3, 0, -4)$ . The smallest eigenvalue is  $\lambda_3 = -4$  with eigenvector  $e_3$ . However,  $(x^T Ax)^2$  is minimized by the second eigenvector  $e_2$  associated with  $\lambda_2 = 0$ .

e) **True:** Suppose  $y \in \mathcal{R}(ABC)$ . Then,  $y = ABCx$  for some  $x$ . Therefore,  $y = Av$  for  $v = BCx$ , and we can conclude  $y \in \mathcal{R}(A)$ .

f) **True:** Suppose  $A \in \mathbb{R}^{m \times n}$  and  $x \in \mathbb{R}^n$ . Then,  $\|Ax\|_2 \leq \|A\|_2 \|x\|_2$  by definition of the  $l_2$ -induced operator norm. Then, since  $\|A\|_2^2 = (\sigma_{\max}(A))^2 = \lambda_{\max}(A^T A) \leq \sum_{i=1}^n \lambda_i(A^T A) = \text{tr}(A^T A) = \|A\|_F^2$ , we can conclude that  $\|Ax\|_2 \leq \|A\|_2 \|x\|_2 \leq \|A\|_F \|x\|_2$ .

g) **False:** Consider  $A = \begin{bmatrix} 0.9 & 10 \\ 0 & 0.9 \end{bmatrix}$  with eigenvalues  $|\lambda| = 0.9 < 1$ . Then,  $\|A\|_2 = \sigma_{\max}(A) \approx 10.0804$ . We compute  $A^2 = \begin{bmatrix} 0.81 & 18.0 \\ 0 & 0.81 \end{bmatrix}$  and  $\|A^2\|_2 \approx 18.0364$ . Therefore,  $\|A^2\|_2 > \|A\|_2$ .

4. **Suspension system.** Let us consider the following mechanical suspension system



where  $u$  is a force that applied to the third mass. The equations of motion for this system are

$$\begin{aligned}\ddot{q}_1 &= -k_1 q_1 - b_1 \dot{q}_1 - k_2 q_1 + k_2 q_2 - b_2 \dot{q}_1 + b_2 \dot{q}_2 \\ \ddot{q}_2 &= -k_2 q_2 + k_2 q_1 - b_2 \dot{q}_2 + b_2 \dot{q}_1 - k_3 q_2 + k_3 q_3 - b_3 \dot{q}_2 + b_3 \dot{q}_3 \\ \ddot{q}_3 &= -k_3 q_3 + k_3 q_2 - b_3 \dot{q}_3 + b_3 \dot{q}_2 + u\end{aligned}$$

a) We want to model the system as

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}$$

Find  $A$ ,  $B$  and  $C$  in terms of  $k_1, k_2, k_3, b_1, b_2, b_3$ . Use states  $x = (q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3)$  and let  $y = q_3$ .

b) We will use sample period  $h = 1$ , with all  $k_i = 1$  and all  $b_i = 1$ . Let  $x_d(k) = x(kh)$ , and  $y_d(k) = y(kh)$ . Suppose the force input  $u$  is piecewise constant, so that  $u(t) = u_d(k)$  for  $kh \leq t < kh + h$ . Find a  $A_d$ ,  $B_d$  and  $C_d$  so that these quantities satisfy the discrete-time linear dynamical system

$$\begin{aligned}x_d(k+1) &= A_d x_d(k) + B_d u_d(k) \\ y_d(k) &= C_d x_d(k)\end{aligned}$$

c) Suppose  $x(0) = 0$ . Use your discretization to find the response of the system when

$$u_d(k) = \begin{cases} 1 & \text{if } k < 3 \\ 0 & \text{otherwise} \end{cases}$$

Plot  $y$  against  $t$  for  $0 \leq t \leq 100$ .

d) Define the magnitude of the response  $J$  by

$$J = \sum_{k=0}^{99} y_d(k)^2$$

The force  $u_d$  is known to satisfy

$$\sum_{k=0}^{49} u_d(k)^2 \leq 1$$

and  $u_d(k) = 0$  for  $k \geq 50$ . Find the the force  $u_d$  which satisfies these constraints and maximizes  $J$ . Report the maximum  $J$ , and plot the force input  $u_d$ , and the resulting output  $y_d$ .

**Solution.**

a)

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -(k_1 + k_2) & k_2 & 0 & -(b_1 + b_2) & b_2 & 0 \\ k_2 & -(k_2 + k_3) & k_3 & b_2 & -(b_2 + b_3) & b_3 \\ 0 & k_3 & -k_3 & 0 & b_3 & -b_3 \end{bmatrix}$$

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

$$C = [0 \ 0 \ 1 \ 0 \ 0 \ 0]$$

b)

$$A_d = e^{hA} = I + Ah + \frac{1}{2}A^2h^2 + \dots$$

$$B_d = \int_0^h e^{\tau A} B d\tau = A^{-1}(e^{hA} - I)B$$

$$C_d = C$$

c)

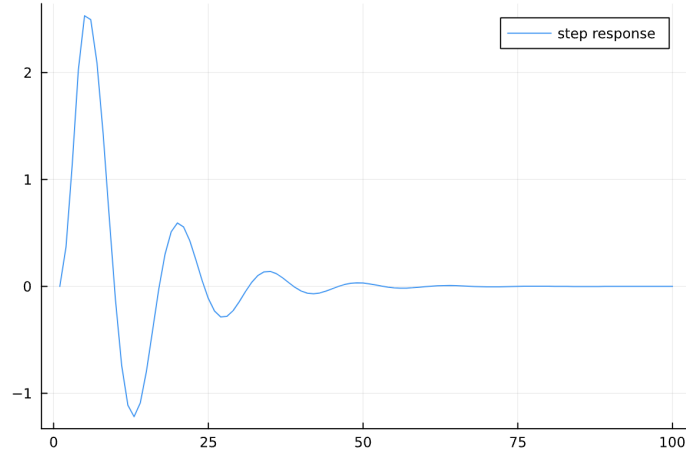


Figure 1: Plot of  $y$

d) Let us express  $y(0), \dots, y(99)$  as a function of  $u(0), \dots, u(49)$ :

$$\begin{bmatrix} y(0) \\ \vdots \\ y(99) \end{bmatrix} = T \begin{bmatrix} u(0) \\ \vdots \\ u(49) \end{bmatrix}, \quad T = \begin{bmatrix} 0 & \dots & & \\ CB & 0 & \dots & \\ CAB & CB & \dots & \\ \vdots & & & \\ CA^{98}B & CA^{97}B & \dots & CA^{49}B \end{bmatrix} \begin{bmatrix} u(0) \\ \vdots \\ u(49) \end{bmatrix}$$

To maximize  $J$  we must maximize  $\|Tu\|^2$  subject to  $\|u\| \leq 1$ . That is the norm of  $T$  (squared), ie, its maximum singular value squared. The worst case  $u$  is given by the right singular vector associated with the largest singular value of  $T$ . We have  $\max_u \sum_{t=0}^{99} y(t)^2 = \|T\|^2 \approx 35.15$ .

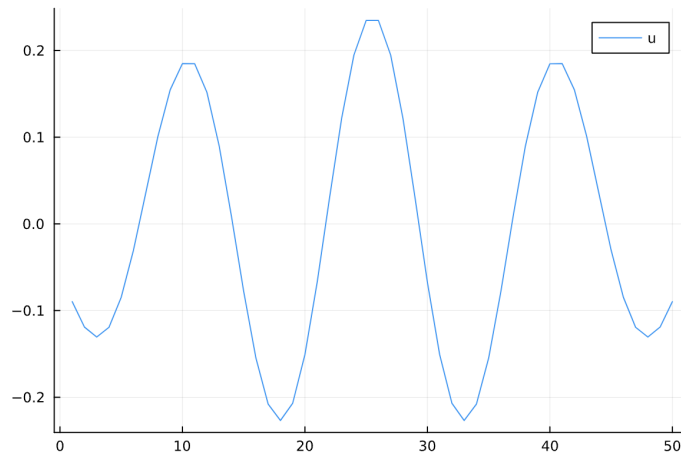


Figure 2: Force input

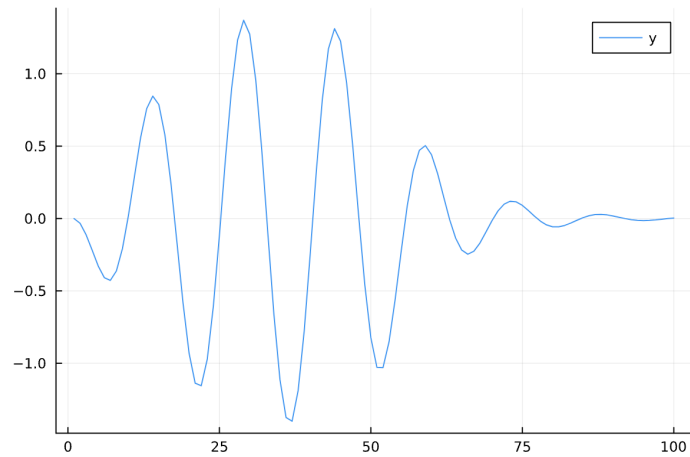


Figure 3: Output

```

using LinearAlgebra
using Plots
using ToeplitzMatrices
#choose k and b
k1 = 1
k2 = 1
k3 = 1
b1 = 1
b2 = 1
b3 = 1
#define A, B and C
A = [ 0 0 0 1 0 0; 0 0 0 0 1 0; 0 0 0 0 0 1;
      -(k1+k2) k2 0 -(b1+b2) b2 0;
      k2 -(k2+k3) k3 b2 -(b2+b3) b3;
      0 k3 -k3 0 b3 -b3]

B = [0 0 0 0 0 1]'

C = [ 0 0 1 0 0 0]

#Discretize A, B and C
Ad = exp(A)
Bd = inv(Ad)*(Ad - I(6))*B
Cd = C

#Check for stability
abs.(eigvals(Ad))

```

```

#Find step response
u_step = vec([1 1 1 zeros(97)'])

x_step = zeros(100, 6)
y_step = zeros(100)

for i in 1:99
    x_step[i+1, :] = Ad*x_step[i, :]+Bd*u_step[i]
    y_step[i] = (Cd*x_step[i, :])[1]
end

plot(y_step)

#worst case response

h = zeros(99)
A6 = I(6)

for i in 1:99
    h[i] =(Cd*A6*Bd)[1]
    A6=A6*Ad
end

T = Toeplitz(vec([0 h[:]']),vec([0 zeros(49)'])))

U, S, V = svd(Matrix(T))

plot(V[:, 1])
plot(T*V[:, 1])

```

**5. Illumination.** We have a square region  $R$  divided into  $N \times N$  pixels. It is illuminated by  $n$  lamps. We apply power  $q_j$  to lamp  $j$ . In this question we will choose the vector of powers  $q \in \mathbb{R}^n$  to illuminate the region  $R$  as uniformly as possible. The position of the  $i$ th lamp is given by the  $i$ th column of the matrix  $L$ , which is in the file `illumination.json`. We will use  $N = 25$ . The region  $R$  is

$$R = \{(x, y) \mid 0 \leq x \leq N, 0 \leq y \leq N\}$$

Suppose we apply power  $a \in \mathbb{R}$  to a lamp at position  $r \in \mathbb{R}^3$ . Let  $(x, y) \in \mathbb{R}^2$  be the center of one of the pixels in  $R$ . The distance from the lamp to the pixel is

$$d = \left( (x - r_1)^2 + (y - r_2)^2 + r_3^2 \right)^{\frac{1}{2}}$$



Then the light intensity falling on that pixel is  $ka/d^2$ . The total light intensity on any pixel is the sum of the intensities from the different lamps. We will have  $k = 10$ .

- a) Let  $l_i$  be the light intensity on pixel  $i$ . We would like to apply power to the lamps to uniformly light the region  $R$ , so that each pixel receives light intensity approximately equal to 1. Define the error

$$J = \sum_{i=1}^{N^2} (l_i - 1)^2$$

Find the optimal choice of powers  $q \in \mathbb{R}^n$ , where  $q_j$  is the power applied to lamp  $j$ . Report the total power used

$$P_{\text{tot}} = \sum_{j=1}^n q_j$$

Plot the intensity as a *heatmap*. You may do this using

```
using Plots
heatmap(Y, c = :thermal, aspectratio = :equal, clim=(0,2))
```

where  $Y$  is an  $N \times N$  matrix of intensities. Report the power applied to each lamp, and the error  $J$ .

- b) We unfortunately have a power budget; we require  $P_{\text{tot}} \leq 15$ . Here is one simple approach for achieving this. We use the optimal solution from the previous part, but scale it so that  $P_{\text{tot}} = 15$ . Do this, and report the error  $J$ .
- c) It is possible to find the choice of powers that minimizes  $J$  subject to the budget constraint. Find this choice of powers. Plot the resulting heatmap of intensity, report the total power used, and the power applied to each lamp.

### Solution.

- a) We want to minimize

$$\|l - \mathbf{1}\|^2$$

We'll order the pixels from  $i = 1, \dots, N^2$ , then

$$p_i = \begin{bmatrix} x_i \\ y_i \\ 0 \end{bmatrix} \in \mathbb{R}^3$$

The position of the  $j^{\text{th}}$  lamp is given by  $r^{(j)} = L_j$  where  $L_j$  is the  $j^{\text{th}}$  column of  $L$ . Then, we can write

$$l = k\tilde{A}q$$

where

$$\tilde{A}_{ij} = \frac{1}{\|p_i - r^{(j)}\|^2}$$

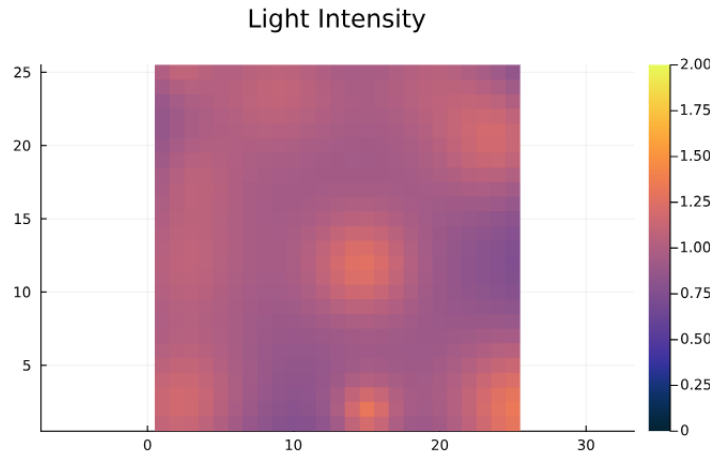
So, we can find the least-squares solution to

$$\min \|k\tilde{A}q - \mathbf{1}\|^2$$

$$\implies q_{opt} = \frac{1}{k} \left( \tilde{A}^T \tilde{A} \right)^{-1} \tilde{A}^T \mathbf{1}$$

$$q_{opt} = [2.03, 0.46, 1.69, 2.98, 3.92, 0.49, 0.26, 1.00, 1.66, 1.75]^T$$

$P_{tot} = 16.245$  and  $J = 2.415$ .



b) We will scale the magnitude of each power input by

$$u_{scaled} = \frac{15}{P_{tot}} u_{opt}$$

$J = 3.077$ .

c) We can formulate this as a constrained min-norm problem of the form

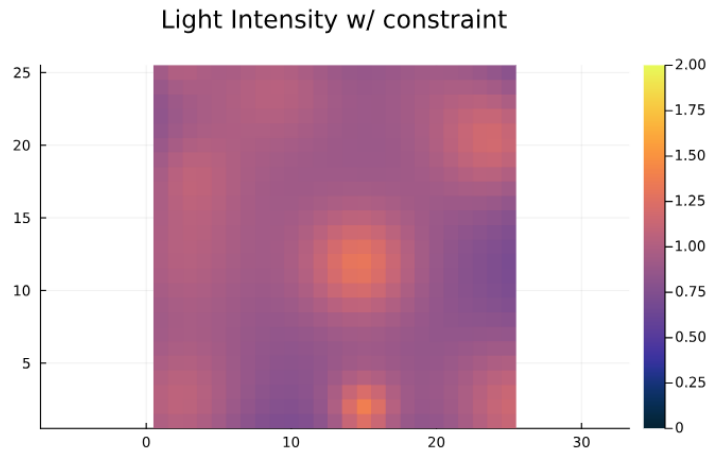
$$\min \|k\tilde{A}q - \mathbf{1}\|^2 \text{ s.t. } P_{tot} = 15$$

which is equivalent to

$$\min \|k\tilde{A}q - \mathbf{1}\|^2 \text{ s.t. } \mathbf{1}^T q = 15$$

$$\implies \begin{bmatrix} q \\ \lambda \end{bmatrix} = \begin{bmatrix} k^2 \tilde{A}^T \tilde{A} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} k \tilde{A}^T \mathbf{1} \\ 15 \end{bmatrix}$$

$$q_{opt} = [1.96, 0.58, 1.85, 2.67, 3.27, 0.42, 0.31, 1.16, 1.19, 1.59]^T$$



$P_{tot} = 15.000$  and  $J = 2.759$ .

See Julia code below:

```
import Pkg
Pkg.add("JSON")
include("readclassjson.jl")
data = readclassjson("illumination.json")
using LinearAlgebra
Pkg.add("RecipesBase")
import RecipesBase: plot
Pkg.add("Plots")

# Problem 5
# part a

L = data["L"]
N = 25
N2 = N^2
n = size(L)[2]
k = 10

# Creating matrix of centers of pixels
Px = zeros(N,N)
for i = 1:N
    for j = 1:N
        Px[i,j] = j-0.5
    end
end
Py = zeros(N,N)
```

```

for i = 1:N
    for j = 1:N
        Py[i,j] = i-0.5
    end
end
px = reshape(Px,N2,1)
py = reshape(Py,N2,1)
P_tilde = [px py zeros(N2,1)]

# Creating matrix A_tilde
A_tilde = zeros(N2,n)
for i = 1:N2
    for j = 1:n
        curr_pixel_pos = P_tilde[i,:]
        curr_lamp_pos = L[:,j]
        A_tilde[i,j] = 1/(norm(curr_pixel_pos-curr_lamp_pos))^2
    end
end

q_opt = (1/k).*inv(A_tilde'*A_tilde)*A_tilde'*ones(N2,1)
P_tot = dot(q_opt,ones(n,1))
print(String("P_total = $P_tot"))
display(q_opt)
J = norm(k.*A_tilde*q_opt-ones(N2,1))
print(String("J = $J"))

A = reshape(k.*A_tilde*q_opt,N,N)
using Plots
heatmap(A, c = :thermal, aspectratio = :equal, clim=(0,2),
        title="Light Intensity")

# Problem 5
# part b

q_opt_scaled = (15/P_tot).*q_opt
J = norm(k.*A_tilde*q_opt_scaled-ones(N2,1))
print(String("J = $J"))

# Problem 5
# part c

T = [k^2 .*A_tilde'*A_tilde ones(n,1);ones(1,n) 0]
z = [k.*A_tilde'*ones(N2,1);15]

```

```

q_opt = (inv(T)*z)[1:n]

P_tot = dot(q_opt,ones(n,1))
print(String("P_total = $P_tot"))
display(q_opt)
J = norm(k.*A_tilde*q_opt-ones(N2,1))
print(String("J = $J"))

A = reshape(k.*A_tilde*q_opt,N,N)
using Plots
heatmap(A, c = :thermal, aspectratio = :equal, clim=(0,2),title=
    "Light Intensity w/ constraint")

```

**6. Finding missing data.** We are given data from a ratings system, in the form of a matrix  $Z \in \mathbb{R}^{m \times n}$ . Here  $Z_{ij}$  is the rating that user  $i$  gives to movie  $j$ . A reasonable assumption about this setting is that such recommendation matrices have low rank. The idea is that a users rating depends on a few factors (that we don't know), such as whether they like romantic movies, comedies, etc. If there were  $r$  such hidden factors, then every row of the matrix would be a linear combination of  $r$  unknown vectors.

However, we do not have all entries of the matrix, because not every user has seen every movie. We would like to fill in the missing entries, making use of the low rank property of such matrices. We have

$$Z = \begin{bmatrix} X & P \\ R & Q \end{bmatrix}$$

where  $X \in \mathbb{R}^{m_1 \times n_1}$  is the unknown block. In more realistic settings, the unknown entries would be distributed throughout the matrix. Here, for simplicity, we consider the case when all the top-left block is unknown. That means that the first  $m_1$  users have not rated the first  $n_1$  movies.

a) Show that

$$\text{range} \begin{bmatrix} P \\ Q \end{bmatrix} \subseteq \text{range} \begin{bmatrix} X & P \\ R & Q \end{bmatrix}$$

b) Now suppose that  $Q$  is fat and full rank. Show that there exists  $X$  such that

$$\text{range} \begin{bmatrix} P \\ Q \end{bmatrix} = \text{range} \begin{bmatrix} X & P \\ R & Q \end{bmatrix}$$

c) Using the results of the previous two parts, explain how to construct  $X$  which minimizes

$$\text{rank} \begin{bmatrix} X & P \\ R & Q \end{bmatrix}$$

- d) To save money, the movie streaming service decides to only show movies which have sufficiently high ratings. For each user  $i$  there is a threshold rating  $s_i$ . The  $j$ th movie is shown if  $Z_{ij} \geq s_i$  for all  $i$ .

Using the data in `movieratings.json`, find which movies are not shown.

Important note: you must solve this question exactly. Iterative solutions are not acceptable and will not receive points.

**Solution.** Here is the solution.

- a) Suppose  $y \in \mathcal{R}\left(\begin{bmatrix} P \\ Q \end{bmatrix}\right)$ . Then,  $y = \begin{bmatrix} P \\ Q \end{bmatrix} x$  for some  $x \in \mathbb{R}^{n-n_1}$ .

Letting  $v = \begin{bmatrix} 0 \\ x \end{bmatrix} \in \mathbb{R}^n$  (i.e., the first  $n_1$  entries are all 0), we see then that

$$Zv = \begin{bmatrix} X & P \\ R & Q \end{bmatrix} \begin{bmatrix} 0 \\ x \end{bmatrix} = \begin{bmatrix} X \\ R \end{bmatrix} 0 + \begin{bmatrix} P \\ Q \end{bmatrix} x = y. \quad (1)$$

Therefore,  $y \in \mathcal{R}\left(\begin{bmatrix} X & P \\ R & Q \end{bmatrix}\right)$ .

We can therefore conclude that  $\mathcal{R}\left(\begin{bmatrix} P \\ Q \end{bmatrix}\right) \subseteq \mathcal{R}\left(\begin{bmatrix} X & P \\ R & Q \end{bmatrix}\right)$ .

- b) Since  $Q \in \mathbb{R}^{(m-m_1) \times (n-n_1)}$  is fat and full rank, the columns of  $R \in \mathbb{R}^{(m-m_1) \times n_1}$  are in the range of  $Q$ ; that is,  $R = QA$  for some  $A \in \mathbb{R}^{(n-n_1) \times n_1}$ .

More specifically, we can let  $A = Q^T(QQ^T)^{-1}R$  (since  $Q$  is fat and full rank,  $QQ^T$  is invertible).

Then, let  $X = PA$ . Therefore,

$$\begin{bmatrix} X \\ R \end{bmatrix} = \begin{bmatrix} PA \\ QA \end{bmatrix} = \begin{bmatrix} P \\ Q \end{bmatrix} A, \quad (2)$$

and we can conclude that every column of  $\begin{bmatrix} X \\ R \end{bmatrix}$  is in the range of  $\begin{bmatrix} P \\ Q \end{bmatrix}$ .

Therefore,

$$\text{rank} \begin{bmatrix} P \\ Q \end{bmatrix} = \text{rank} \begin{bmatrix} X & P \\ R & Q \end{bmatrix}. \quad (3)$$

- c) As shown in the previous problem, letting  $X = PQ^T(QQ^T)^{-1}R$  will cause

$$\text{rank} \begin{bmatrix} P \\ Q \end{bmatrix} = \text{rank} \begin{bmatrix} X & P \\ R & Q \end{bmatrix}. \quad (4)$$

Since we can't change the value of  $P$  and  $Q$ , the rank of  $Z$  is therefore minimized with this choice of  $X$ .

- d) To find the movies that aren't shown, first construct  $Z$  using  $X$  from Part C. Then, iterate through  $j = 1, \dots, n$  and if there is some  $i = 1, \dots, m$  such that  $Z_{ij} < s_i$ , then the movie is not shown.

Using this algorithm we determine that movies  $j = 15, 21, 25, 48, 61, 62, 63, 65, 82$  are not shown.

See the Julia code below:

```
include("../code/readclassjson.jl")

data = readclassjson("../data/movieratings.json")
m = data["m"]
n = data["n"]
m1 = data["m1"]
n1 = data["n1"]
Q = data["Q"]
P = data["P"]
R = data["R"]
s = data["s"]

# Construct X
A = Q \ R # A = Q' * inv(Q * Q') * R
X = P * A

# Construct Z
Z = [X P; R Q]

# Find all movies not shown
movies_not_shown = Int{[]>()
for j in 1:n
    if any(Z[:, j] .< s)
        append!(movies_not_shown, j)
    end
end
end
print("Movies Not Shown: ", movies_not_shown, "\n")
```