EE263 Homework 6 Solutions

Fall 2023

3.660. Some true/false questions. Determine if the following statements are true or false. No justification or discussion is needed for your answers. What we mean by "true" is that the statement is true for all values of the matrices and vectors given. You can't assume anything about the dimensions of the matrices (unless it's explicitly stated), but you can assume that the dimensions are such that all expressions make sense. For example, the statement "$A+B = B+A$" is true, because no matter what the dimensions of A and B (which must, however, be the same), and no matter what values A and B have, the statement holds. As another example, the statement $A^2 = A$ is false, because there are (square) matrices for which this doesn't hold. (There are also matrices for which it does hold, *e.g.*, an identity matrix. But that doesn't make the statement true.)

a) If $x^T Ax = x^T Bx$ for all x, then $A = B$.

b) If $x^T Ay = x^T By$ for all x and y, then $A = B$.

c) If $\|Ax\| = \|Bx\|$ for all x, then $A = B$.

d) If A and B are both stable, then $A + B$ is also stable.

e) The matrix $\begin{bmatrix} 2a & 3b \\ 4c & 5d \end{bmatrix}$ is equal to $A \begin{bmatrix} a & b \\ c & d \end{bmatrix} B$ for some matrices A and B.

f) If R is upper triangular and orthogonal, then R is diagonal.

g) If A is square, then there always exists a matrix C such that $AC = CA^T$.

h) If $x, y \in \mathbf{R}^n$ then the $n \times n$ matrix xy^T is diagonalizable.

Solution.

a) False. Consider

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Then, we have that

$$x^T Ax = x_1^2 + 2x_1x_2 + x_2^2 = x^T Bx$$

for all x, but $A \neq B$.

b) True. For $x = e_i$ and $y = e_j$, we have that

$$a_{ij} = e_i^T Ae_j = e_i^T Be_j = b_{ij}.$$

Since this holds for all i and j, we have that $A = B$.

c) False. Consider
$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \; B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$
Then, we have that
$$\|Ax\| = \|x\| = \|Bx\|$$
for all x, but $A \neq B$.

d) False. Consider
$$A = \begin{bmatrix} -1 & 2 \\ 0 & -1 \end{bmatrix} \qquad \text{and} \qquad B = \begin{bmatrix} -1 & 0 \\ 2 & -1 \end{bmatrix}.$$
Then, -1 is the only eigenvalue of A, and also the only eigenvalue of B. Therefore, A and B are both stable. However, the eigenvalues of $A + B$ are -4 and 0, so $A + B$ is not stable.

e) False. Suppose $a = b = c = d = 1$. Then, we have that
$$\operatorname{rank}\left(\begin{bmatrix} 2a & 3b \\ 4c & 5d \end{bmatrix}\right) = 2 \qquad \text{and} \qquad \operatorname{rank}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = 1.$$
For any A and B, we have that
$$\operatorname{rank}\left(A \begin{bmatrix} a & b \\ c & d \end{bmatrix} B\right) \leq \operatorname{rank}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = 1.$$
Thus, it is impossible to find matrices A and B such that the matrix $\begin{bmatrix} 2a & 3b \\ 4c & 5d \end{bmatrix}$ is equal to $A \begin{bmatrix} a & b \\ c & d \end{bmatrix} B$.

f) True. Suppose $R \in \mathbb{R}^{n \times n}$ is upper triangular and orthogonal. We will argue by induction on n that R is diagonal. If $n = 1$, then any matrix is diagonal, so the claim is trivially true. Now suppose $n \geq 2$. The norm of the first column of R is equal to 1:
$$\|r_1\|^2 = r_{11}^2 = 1.$$
The inner product of the first column of R with any other column of R is
$$r_1^T r_j = r_{11} r_{1j} = 0.$$
Since $r_{11} \neq 0$, this implies that $r_{1j} = 0$ for all $j = 2, \ldots, n$. Thus, R has the form
$$R = \begin{bmatrix} r_{11} & 0 \\ 0 & R_{22} \end{bmatrix}.$$
Since R is upper triangular, it is clear that $R_{22} \in \mathbb{R}^{(n-1)\times(n-1)}$ is upper triangular. Moreover, R_{22} is orthogonal because
$$R^T R = \begin{bmatrix} \|r_{11}\|^2 & 0 \\ 0 & R_{22}^T R_{22} \end{bmatrix} = I = \begin{bmatrix} 1 & 0 \\ 0 & I \end{bmatrix}.$$
The induction hypothesis implies that R_{22} is diagonal, and hence that R is diagonal. By induction, this proves the claim for all n.

g) True. The matrix $C = 0$ always satisfies the condition.

h) False. For $x = (1, 0)$ and $y = (0, 1)$, the matrix

$$xy^T = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}^T = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

is not diagonalizable.

8.160. Designing an equalizer for backwards-compatible wireless transceivers. You want to design the equalizer for a new line of wireless handheld transceivers (more commonly called walkie-talkies). The transmitter for the new line of transceivers has already been designed (and cannot be changed) – if the input signal is $x \in \mathbb{R}^n$, then the transmitted signal is $y = A_{\text{new}}x \in \mathbb{R}^m$, where $A_{\text{new}} \in \mathbb{R}^{m \times n}$ is known. An equalizer for A_{new} is a matrix $B \in \mathbb{R}^{n \times m}$ such that $By = x$ for every $x \in \mathbb{R}^n$.

The new line of transceivers will replace an older model. Given an input signal $x \in \mathbb{R}^n$, the old line of transceivers transmit a signal $y_{\text{old}} = A_{\text{old}}x \in \mathbb{R}^m$, where $A_{\text{old}} \in \mathbb{R}^{m \times n}$ is known. In addition to providing exact equalization for the new line of transceivers, you want your equalizer to be able to at least partially equalize signals transmitted using the old line of transceivers. In other words, to the extent that it is possible, you want the new line of transceivers to be backwards compatible with the old line of transceivers.

a) Explain how to find an equalizer B that minimizes

$$J = \|BA_{\text{old}} - I\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} (BA_{\text{old}} - I)_{ij}^2$$

among all B that exactly equalize A_{new}. Such a B is an exact equalizer for A_{new}, and an approximate equalizer for A_{old}. State any assumptions that are needed for your method to work.

b) The file `backwards_compatible_transceiver_data.json` defines the following variables.

- `Anew`, the $m \times n$ matrix that describes the transmitter used in the new line of transceivers
- `Aold`, the $m \times n$ matrix that describes the transmitter used in the old line of transceivers
- `x`, a vector of length n that serves as an example input signal

Apply your method to this example data. Report the optimal value of J. The pseudoinverse $A_{\text{new}}^\dagger$ is another exact equalizer for A_{new}. Compare the optimal value of J, and the value of J achieved by $A_{\text{new}}^\dagger$.

c) The example signal x defined in the data file is a binary signal. Form the signal $y_{\text{old}} = A_{\text{old}}x$ transmitted by the old line of transceivers, and construct an estimate of x by equalizing y_{old} using B, and then rounding the result to a binary signal. More concretely, compute the estimate $\hat{x} \in \mathbb{R}^n$, where

$$\hat{x}_i = \begin{cases} 1 & (By_{\text{old}})_i > \frac{1}{2}, \\ 0 & \text{otherwise.} \end{cases}$$

Report the bit error rate of your estimate, which is defined as

$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{I}(x_i \neq \hat{x}_i),$$

where $\mathbf{I}(x_i \neq \hat{x}_i)$ is an indicator function:

$$\mathbf{I}(x_i \neq \hat{x}_i) = \begin{cases} 1 & x_i \neq \hat{x}_i, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, report the bit error rate if $A_{\text{new}}^{\dagger}$ is used as the equalizer.

Solution.

a) Write the equalizer $B \in \mathbb{R}^{n \times m}$ in terms of its rows:

$$B = \begin{bmatrix} b_1^{\mathsf{T}} \\ \vdots \\ b_n^{\mathsf{T}} \end{bmatrix}.$$

We require that B be an exact equalizer for A_{new}: that is, $BA_{\text{new}} = I$. We can express this condition in terms of the rows of B as

$$A_{\text{new}}^{\mathsf{T}} b_i = e_i, \qquad i = 1, \ldots, n,$$

where e_i denotes the ith standard basis vector in $\mathbb{R}^n$. Similarly, we can write our objective in terms of the rows of B:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{n} (BA_{\text{old}} - I)_{ij}^2 = \sum_{i=1}^{n} \|(BA_{\text{old}} - I)_{i*}\|^2 = \sum_{i=1}^{n} \|A_{\text{old}}^{\mathsf{T}} b_i - e_i\|^2.$$

Thus, we want to solve the following optimization problem.

$$\begin{array}{ll} \underset{B \in \mathbb{R}^{n \times m}}{\text{minimize}} & \sum_{i=1}^{n} \|A_{\text{old}}^{\mathsf{T}} b_i - e_i\|^2 \\ \text{subject to} & A_{\text{new}}^{\mathsf{T}} b_i = e_i \qquad i = 1, \ldots, n \end{array}$$

This problem is separable in the rows of B, allowing us to decompose it into n vector optimization problems:

$$\begin{array}{ll} \underset{b_i \in \mathbb{R}^{m}}{\text{minimize}} & \|A_{\text{old}}^{\mathsf{T}} b_i - e_i\|^2 \\ \text{subject to} & A_{\text{new}}^{\mathsf{T}} b_i = e_i \end{array}$$

for $i = 1, \ldots, n$. Each of these problems is a linearly constrained minimum-norm problem; the solution of such a problem can be obtained by solving the following system of equations:

$$\begin{bmatrix} A_{\text{old}} A_{\text{old}}^\mathsf{T} & A_{\text{new}} \\ A_{\text{new}}^\mathsf{T} & 0 \end{bmatrix} \begin{bmatrix} b_i \\ \lambda_i \end{bmatrix} = \begin{bmatrix} A_{\text{old}} e_i \\ e_i \end{bmatrix}, \qquad i = 1, \ldots, n.$$

This method works as long as each of these optimization problems is feasible – that is, as long as we can find a matrix $B \in \mathbb{R}^{n \times m}$ such that

$$BA_{\text{new}} = I.$$

In other words, we require that A_{new} be skinny and full rank (or, equivalently, left invertible). In order for the KKT system to have a unique solution, we require that $A_{\text{new}}^\mathsf{T}$ be fat and full rank, and

$$\begin{bmatrix} A_{\text{old}}^\mathsf{T} \\ A_{\text{new}}^\mathsf{T} \end{bmatrix}$$

be skinny and full rank. Equivalently, we require that A_{new} be skinny and full rank, and $\begin{bmatrix} A_{\text{old}} & A_{\text{new}} \end{bmatrix}$ be fat and full rank.

b) The optimal value of J is 3.2361; in comparison, the value of J achieved by $A_{\text{new}}^\dagger$ is 8.0901, which is significantly higher.

c) The bit error rate using the equalizer B is 0.0333, while the bit error rate using $A_{\text{new}}^\dagger$ is 0.1000. Thus, we see that B has a much lower bit error rate than $A_{\text{new}}^\dagger$.

```
using LinearAlgebra

include("readclassjson.jl")

data = readclassjson("backwards_compatible_transceiver_data.json")

A_new = data["Anew"]

A_old = data["Aold"]

x = data["x"]

block_matrix = [2*A_old*A_old’ A_new; A_new’ zeros(30, 30)]

B = zeros(30, 50)

E = I(30)

for i in 1:30
    B[i,:] = (block_matrix\[(2*A_old*E[i,:])’ E[i,:]’]’)[1:50]
end
```

```
B

B*A_new

norm(B*A_old - E)^2

A_pinv = inv(A_new'*A_new)*A_new'

norm(A_pinv*A_old - E)^2

y_old = A_old*x

x_est = B*y_old

x_est_pinv = A_pinv*y_old

for i in 1:30
    if x_est[i]>0.5
        x_est[i] = 1
    else
        x_est[i] = 0
    end
end

x_est

for i in 1:30
    if x_est_pinv[i]>0.5
        x_est_pinv[i] = 1
    else
        x_est_pinv[i] = 0
    end
end

x_est_pinv

function get_bit_rate(x_est, x, n)
    wrong = 0
    for i in 1:n
        if x_est[i] != x[i]
            wrong+=1
        end
    end
    return wrong/n
end
```

```
get_bit_rate(x_est, x, 30)

get_bit_rate(x_est_pinv, x, 30)
```

■

15.2150. Norm expressions for quadratic forms. Let $f(x) = x^\mathsf{T} Ax$ (with $A = A^\mathsf{T} \in \mathbb{R}^{n\times n}$) be a quadratic form.

a) Show that f is positive semidefinite (*i.e.*, $A \geq 0$) if and only if it can be expressed as $f(x) = \|Fx\|^2$ for some matrix $F \in \mathbb{R}^{k\times n}$. Explain how to find such an F (when $A \geq 0$). What is the size of the smallest such F (*i.e.*, how small can k be)?

b) Show that f can be expressed as a difference of squared norms, in the form $f(x) = \|Fx\|^2 - \|Gx\|^2$, for some appropriate matrices F and G. How small can the sizes of F and G be?

Solution.

a) We know that the norm expression $f(x) = \|Fx\|^2$ is a positive semidefinite quadratic form simply because $f(x) \geq 0$ for all x and $f(x) = x^\mathsf{T} Ax$ with $A = F^\mathsf{T} F \geq 0$. In this problem we will show the converse, *i.e.*, any positive semidefinite quadratic form $f(x) = x^\mathsf{T} Ax$ can be written as a norm expression $f(x) = \|Fx\|^2$. Suppose the eigenvalue decomposition of $A \geq 0$ is $Q\Lambda Q^\mathsf{T}$, with $Q^\mathsf{T} Q = I$ and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ where λ_i are the eigenvalues of A. Since $\lambda_i \geq 0$ (because $A \geq 0$) then $\Lambda^{1/2} = \mathrm{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n})$ is a real matrix. Let $F = \Lambda^{1/2} Q^\mathsf{T} \in \mathbb{R}^{n\times n}$. Then we have $\|Fx\|^2 = x^\mathsf{T} F^\mathsf{T} Fx = Q\Lambda^{1/2}\Lambda^{1/2}Q^\mathsf{T} = x^\mathsf{T} Ax = f(x)$. To get smallest F suppose that $\mathrm{rank}(A) = r$. Therefore, $A \in \mathbb{R}^{n\times n}$ has exactly r nonzero eigenvalues $\lambda_1, \ldots, \lambda_r$. Suppose $\Lambda_+ = \mathrm{diag}(\lambda_1, \ldots, \lambda_r)$. Hence, the eigenvalue decomposition of A can be written as

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} \Lambda_+ & 0_{r\times(n-r)} \\ 0_{(n-r)\times r} & 0_{(n-r)\times(n-r)} \end{bmatrix} \begin{bmatrix} Q_1^\mathsf{T} \\ Q_2^\mathsf{T} \end{bmatrix}$$

and as a result $A = Q_1 \Lambda_+ Q_1^\mathsf{T}$ where $Q_1 \in \mathbb{R}^{n\times r}$. Now we can take $F = \Lambda_+^{1/2} Q_1^\mathsf{T} \in \mathbb{R}^{r\times n}$. Therefore, k can be as small as r, *i.e.*, $k = \mathrm{rank}(r)$. Note that k cannot be any smaller than $\mathrm{rank}(A)$ because $A = F^\mathsf{T} F$ implies that $\mathrm{rank}(A) \leq k$.

b) In general, a quadratic form need not to be positive semidefinite. In this problem we show that any quadratic form can be decomposed into its "positive" and "negative" parts. In other words, we can write $f(x)$ as the difference of two norm expressions, *i.e.*, $f(x) = \|Fx\|^2 - \|Gx\|^2$. Suppose A has n_1 positive eigenvalues $\lambda_1, \ldots, \lambda_{n_1}$, n_2 negative eigenvalues $\lambda_{n_1+1}, \ldots, \lambda_{n_1+n_2}$, and therefore $n - n_1 - n_2$ zero eigenvalues. Let

$$\Lambda_+ = \mathrm{diag}(\lambda_1, \ldots, \lambda_{n_1}), \quad \Lambda_- = \mathrm{diag}(-\lambda_{n_1+1}, \ldots, -\lambda_{n_1+n_2}).$$

The eigenvalue decomposition of A can be written as

$$A = \begin{bmatrix} Q_1 & Q_2 & Q_3 \end{bmatrix} \begin{bmatrix} \Lambda_+ & 0_{n_1\times n_2} & 0_{n_1\times(n-n_1-n_2)} \\ 0_{n_2\times n_1} & -\Lambda_- & 0_{n_2\times(n-n_1-n_2)} \\ 0_{(n-n_1-n_2)\times n_1} & 0_{(n-n_1-n_2)\times n_2} & 0_{(n-n_1-n_2)\times(n-n_1-n_2)} \end{bmatrix} \begin{bmatrix} Q_1^\mathsf{T} \\ Q_2^\mathsf{T} \\ Q_3^\mathsf{T} \end{bmatrix}$$

so $A = Q_1\Lambda_+Q_1^\mathsf{T} - Q_2^\mathsf{T}\Lambda_-Q_2$. Now simply take $F = \Lambda_+^{1/2}Q_1^\mathsf{T} \in \mathbb{R}^{n_1\times n}$ and $G = \Lambda_-^{1/2}Q_2^\mathsf{T} \in \mathbb{R}^{n_2\times n}$. It is easy to verify that $A = F^\mathsf{T}F - G^\mathsf{T}G$ and therefore $x^\mathsf{T}Ax = \|Fx\|^2 - \|Gx\|^2$. In fact, this method gives the smallest sizes for F and G.

15.2790. Ellipsoids.

a) Write a function that, given a a 2×2 real, positive definite symmetric matrix $A > 0$, plots the ellipse

$$E = \left\{\, x \in \mathbb{R}^2 \mid x^T A x = 1 \,\right\}$$

Make sure that your plot is shown so that horizontal and vertical lengths are the same, that is, with aspect ratio 1. Turn in your code.

Julia hint: use the following to draw a plot with correct aspect ratio.

```
Using Plots;  plot(x, y, aspect_ratio=:equal)
```

b) Use your code to plot the ellipsoid for the matrix

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

c) Use your code to plot the ellipsoid for the matrix

$$A = \begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.4 \end{bmatrix}$$

On your plot, also show semiaxes.

d) Consider an estimation problem, where we have three sensors, define by $b_i \in \mathbb{R}^2$ for $i = 1, 2, 3$. We measure $y_i = b_i^\mathsf{T} x$. The vectors b_i are

$$b_1 = \begin{bmatrix} 0.89 \\ 0.45 \end{bmatrix} \quad b_2 = \begin{bmatrix} 0.45 \\ 0.89 \end{bmatrix} \qquad b_3 = \begin{bmatrix} -0.71 \\ 0.71 \end{bmatrix}$$

Plot the set of $x \in \mathbb{R}^2$ for which $\|y\| \le 1$. On your plot, show also the b_i (that is, plot a line from the origin to b_i).

Solution.

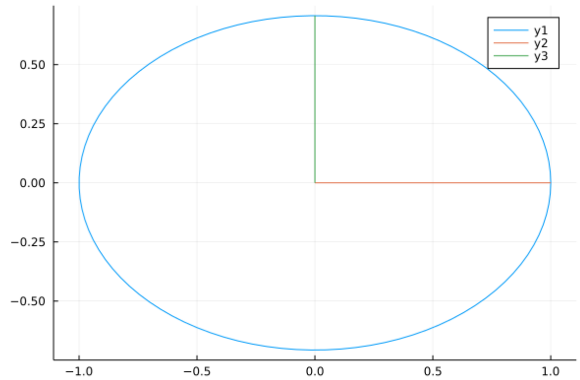
a) The following function plots the ellipse

```
using Plots
using LinearAlgebra

function ellipse(A)
    D, V = eigen(A)
    B = V' * diagm(1 ./sqrt.(D)) * V
    xy = [B*[cos(t), sin(t)] for t in  range(0,2pi,length=100)]
    x = [a[1] for a in xy]
```

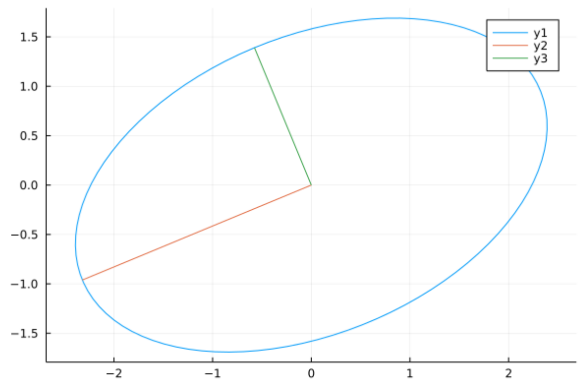
```
    y = [a[2] for a in xy]
    plot(x, y, aspect_ratio=:equal)
end

function plotaxes(p, A)
    for x in getaxes(A)
        plot!(p,[0, x[1]], [0,x[2]])
    end
end

function getaxes(A)
    D, V = eigen(A)
    n = size(A,2)
    return [V[:,i] / sqrt(D[i]) for i=1:n]
end

function partb()
    A = [1 0 ; 0 2]
    p = ellipse(A)
    plotaxes(p, A)
    savefig(p, "partb.png")
    display(p)
end

function partc()
    A = [0.2 -0.1 ; -0.1 0.4]
    p = ellipse(A)
    plotaxes(p, A)
    savefig(p, "partc.png")
    display(p)
end


function partd()
    A =  [0.89   0.45
          0.45 0.89
          -0.71  0.71]
    p=ellipse(A'*A)
    for x in eachrow(A)
        plot!(p,[0, x[1]], [0,x[2]])
    end
    savefig(p, "partd.png")
    display(p)
end
```

```
partb()
partc()
partd()
```

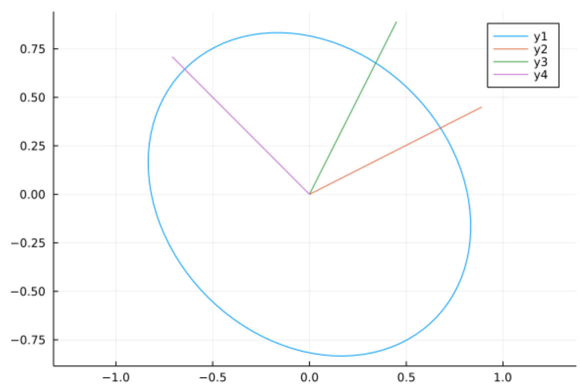
b) The plot is below.



c) The plot is below.



d) The plot is below.



16.2980. Smoothing. We have a discrete-time signal given by $x \in \mathbb{R}^n$. We get to measure $y \in \mathbb{R}^n$, given by

$$y_i = \sum_{k=-h}^{h} c_k x_{i+k} + w_i \qquad \text{for } i = 1, \ldots, n$$

where w_i is noise. Here we use the convention that $x_i = 0$ for $i < 1$ or $i > n$. That is, y is c convolved with x plus noise. In applications, very often the effect of convolution with c is to smooth or blur x, and we would like to undo this.

The file `regl_data.json` contains c, w and x.

a) In Julia, construct the $n \times n$ matrix such that $y = Ax + w$. Plot the singular values σ_k against k.

b) Plot the first 6 right singular vectors of A (i.e. plot V_{ij} against i for $j = 1, \ldots, 6$.) Explain what you see.

c) Find and plot the least-squares estimate of x given y_{meas}, computng y_{meas} using c, x and w given in `regl_data.json`. Explain what happens.

d) Many of the singular values of A are very small; this means that the measurement in the directions of the corresponding right singular vectors is being swamped by the noise.

   If we believe these components are small, we can remove them from our estimate of x altogether by *truncating* the SVD of A and using the truncated SVD to compute the estimate. This is called the *truncated SVD regularization* of least-squares.

   Suppose we decided only to keep the first r components. Then truncate by letting $\tilde{V}$ and $\tilde{U}$ be the first r columns of V and U, and letting $\tilde{\Sigma}$ be the top-left $r \times r$ submatrix of Σ. Then we can construct an estimator that ignores the noise components by

$$A_{\text{est}} = \tilde{V} \tilde{\Sigma}^{-1} \tilde{U}^T$$

   and set

$$x_{\text{est}} = A_{\text{est}} y_{\text{meas}}$$

   For values of r in $5, 10, 15, 30, 50$, compute and plot the corresponding estimates of x. Explain what you see.

e) For each r between 1 and 35, compute the norm of the error

$$\|x - x_{\text{est}}\|$$

   Plot this against r. Explain what you see.

f) Pick the 'best' r and plot the corresponding estimate.

g) Another approach is to use Tychonov regularization. Find and plot the vector $x_{\text{reg}} \in \mathbb{R}^n$ that minimizes the function

$$\|Ax - y\|^2 + \mu \|x\|^2,$$

   where $\mu > 0$ is the regularization parameter. Pick a value of μ that gives a good estimate, in your opinion.

h) The regularized solution is a linear function of y, so it can be expressed as $x_{\text{reg}} = By$ where $B \in \mathbb{R}^{n\times n}$. Express the SVD of B in terms of the SVD of A. To be more specific, let

$$B = \sum_{i=1}^{n} \tilde{\sigma}_i \tilde{u}_i \tilde{v}_i^T$$

denote the SVD of B. Express $\tilde{\sigma}_i$, $\tilde{u}_i$, $\tilde{v}_i$, for $i = 1, \dots, n$, in terms of σ_i, u_i, v_i, $i = 1, \dots, n$ (and, possibly, μ). Recall the convention that $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_n$.

i) Find the norm of B. Give your answer in terms of the SVD of A (and μ).

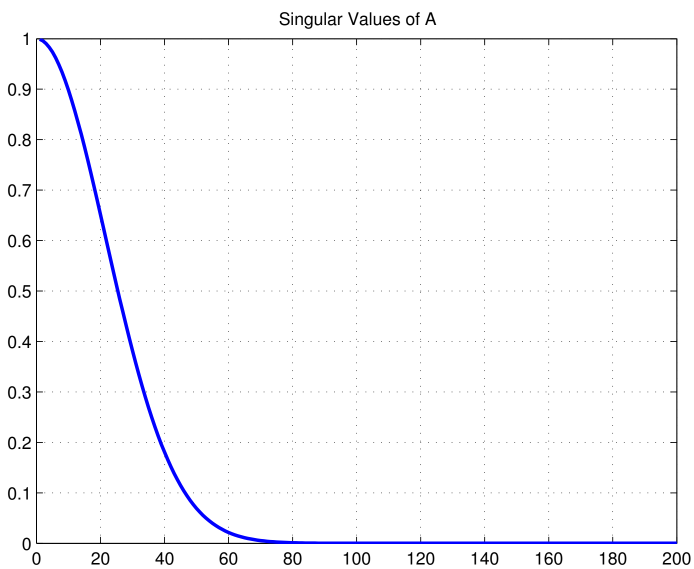
j) Find the worst-case relative inversion error, defined as

$$\max_{y\neq 0} \frac{\|ABy - y\|}{\|y\|}.$$

Give your answer in terms of the SVD of A (and μ).

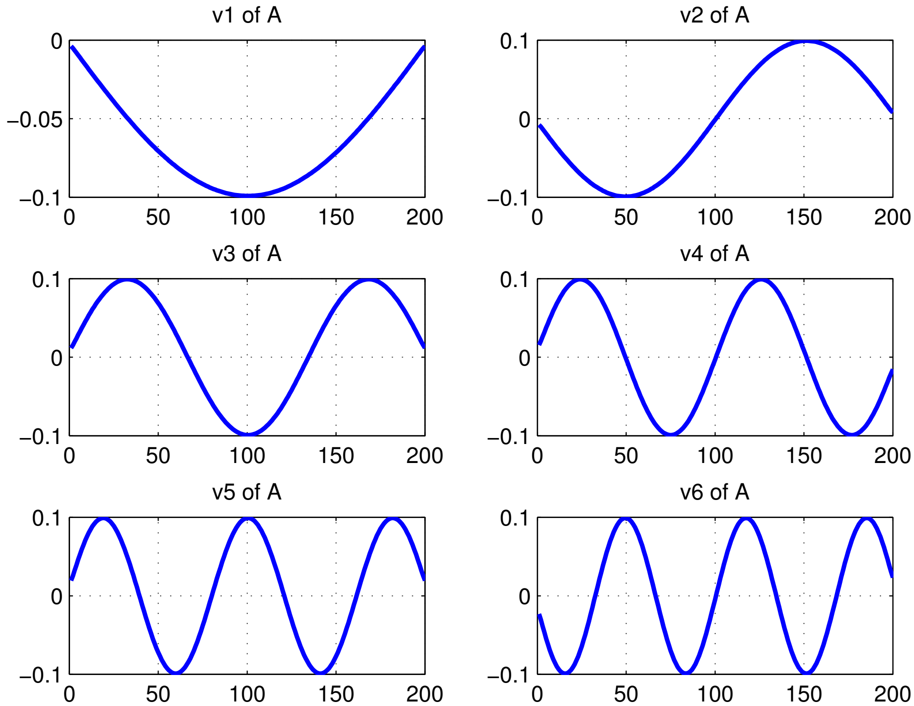
Solution.

a) A can be constructed from c easily. The singular values of A is as follows.



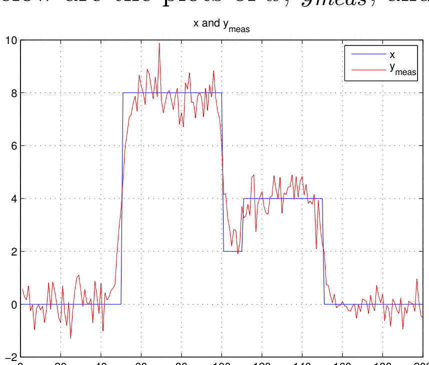
Note that they decay very fast, and some are extremely small. In particular, this shows that this estimation problem will be extremely sensitive to noise and numerical errors in the measurement, since the condition number is large.

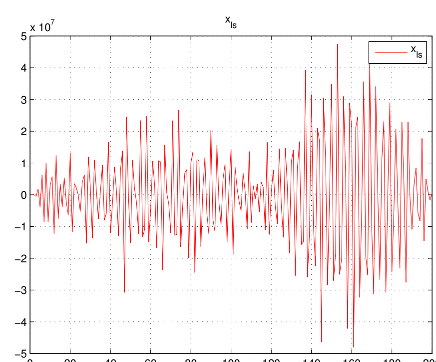
b) The first six singular vectors are below.



The matrix A is a smoother, or low-pass filter. So we should expect that it has a gain that depends on the 'frequency' of the input. The plot show that this intuition is correct; the input is broken down into different frequency components by the matrix of right singular vectors V, each is scaled by the corresponding singular value, and then the output is constructed. The left singular vectors look similar. Since its a low-pass filter, the larger singular values correspond to low frequencies.

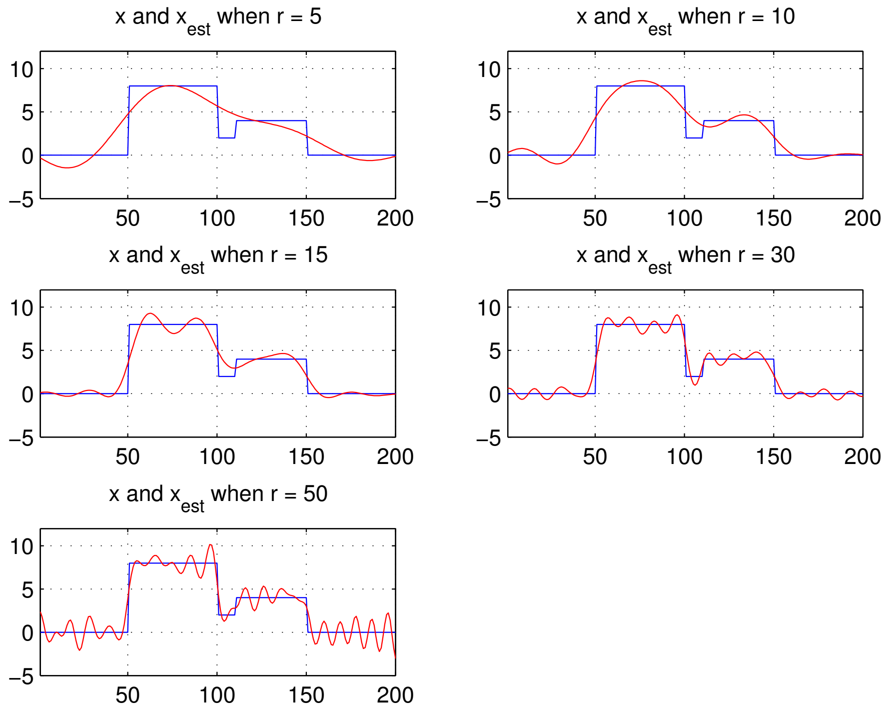
c) Below are the plots of x, y_{meas}, and x_{ls}.





Our estimate is extremely badly corrupted by the noise, because of the high condition number of A; the estimation ellipsoid is long and thin, and we have extremely poor estimates of the components of x corresponding to small singular values of A. These are the high frequency components. They are multiplied by the small singular values, and so are swamped by the noise.

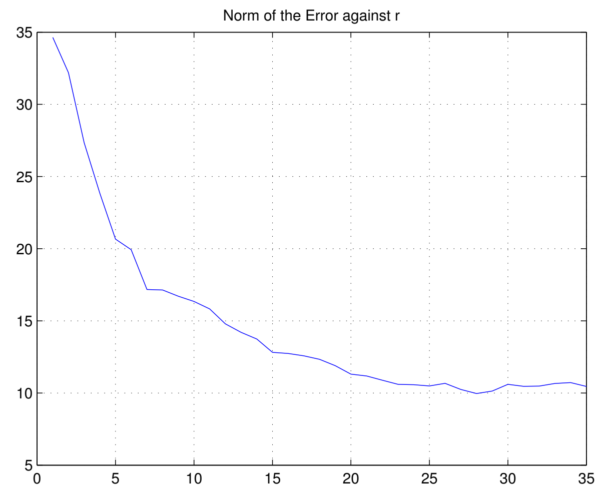
d) The plots below show the estimates obtained with different values of r.



The regularization ignores components of the frequency corresponding to right singular vectors v_k when $k > r$. These are the higher-frequency and more noise components. Removing them is equivalent to making the assumption that that component is actually zero, rather than using the measured data.

The estimate is poor for r very small, because much of the signal is being ignored by the regularization. It is also poor at large r, when very noisy measurements of the high-frequency components are used.

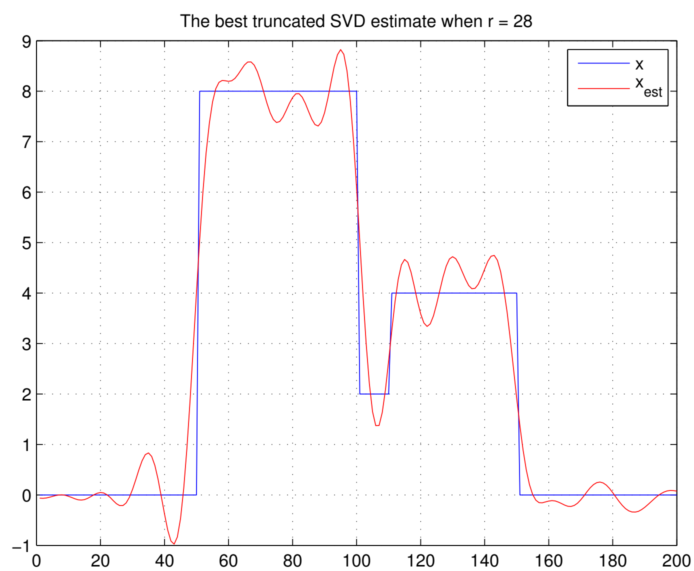
e) The error is plotted below.



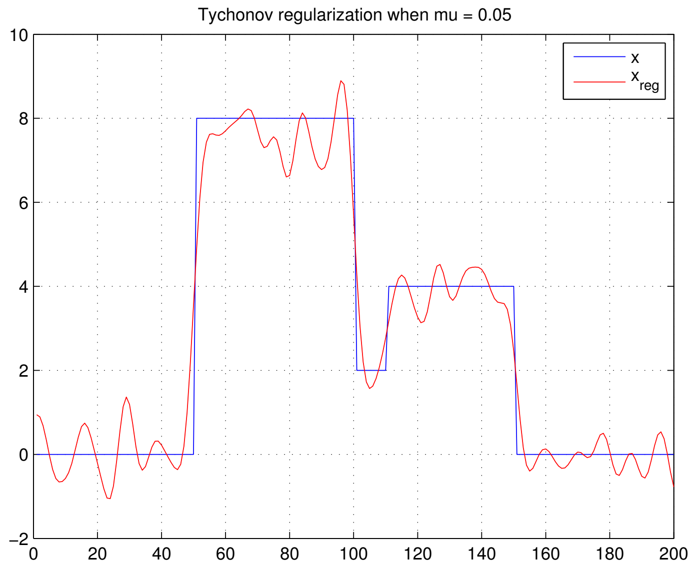
The graph shows the error phenomena described in part (d). Using too few singular values means we lose too much information, using too many means we use information which is very badly corrupted by noise.

f) From the above graph, we find the number of singular values which results in the minimum error is 28.

The corresponding estimate is below.



g) One of good values of μ can be chosen as $\mu = 0.05$ by computing errors for various values of μ. Below is the estimate corresponding to $\mu = 0.05$.



h) The regularized least-squares solution is given by $x_{\mathrm{rls}(\mu)} = \left(A^TA + \mu I\right)^{-1} A^Ty$, and thus

$$\begin{aligned} B &= \left(A^TA+\mu I\right)^{-1}A^T \\ &= \left(\left(U\Sigma V^T\right)^T U\Sigma V^T + \mu I\right)^{-1}\left(U\Sigma V^T\right)^T \\ &= \left(V\Sigma U^TU\Sigma V^T + \mu I\right)^{-1} V\Sigma U^T \\ &= \left(V\left(\Sigma^2+\mu I\right)V^T\right)^{-1} V\Sigma U^T \\ &= \left(V\left(\Sigma^2+\mu I\right)^{-1}V^T\right) V\Sigma U^T \\ &= V\left(\Sigma^2+\mu I\right)^{-1}\Sigma U^T \\ &= V\mathrm{diag}\left(\frac{\sigma_i}{\sigma_i^2+\mu}\right)U^T. \end{aligned}$$

This is almost the SVD of B, except for one detail: the numbers

$$\frac{\sigma_i}{\sigma_i^2+\mu}$$

aren't necessarily ordered from largest to smallest. Thus we have

$$\tilde{\sigma}_i = \frac{\sigma_{[i]}}{\sigma_{[i]}^2+\mu} \qquad \tilde{u}_i = v_{[i]}, \qquad \tilde{v}_i = u_{[i]},$$

where the notation $x_{[i]}$ means the ith largest element of x. (We accepted all sorts of descriptions of this!) One common misconception was that the numbers

$$\frac{\sigma_i}{\sigma_i^2+\mu}$$

were simply in reverse order, so all that had to be done was to reverse the ordering. That isn't true; just sketch the function $\sigma/(\sigma^2+\mu)$ as a function of μ to see that it is not always decreasing. (It increases first, then decreases.)

i) The norm of B is its largest singular value, *i.e.*,

$$\|B\| = \max_i \frac{\sigma_i}{\sigma_i^2 + \mu}.$$

j) The worst-case relative inversion error is the matrix norm of $AB - I$:

$$\begin{aligned}
AB - I &= U\Sigma V^T V \left(\Sigma + \mu\Sigma^{-1}\right)^{-1} U^T - I \\
&= U\Sigma \left(\Sigma + \mu\Sigma^{-1}\right)^{-1} U^T - I \\
&= U\left(\left(I + \mu\Sigma^{-2}\right)^{-1} - I\right) U^T \\
&= U\mathrm{diag}\left(\frac{1}{1 + \mu/\sigma_i^2} - 1\right) U^T \\
&= -U\mathrm{diag}\left(\frac{\mu}{\sigma_i^2 + \mu}\right) U^T
\end{aligned}$$

This is the SVD of $AB - I$ (to within reordering). Its largest singular value, *i.e.*, the norm of $AB - I$, is given by

$$\|AB - I\| = \frac{\mu}{\sigma_n^2 + \mu}.$$

Note that we had to absorb the negative sign in the lefthand orthogonal matrix; one common error was to keep the negative sign in the norm. Obviously that couldn't be right because norms are always nonnegative!