EECS 551/453: HW SOLUTIONS

PROBLEM 1 (∗)

Here $A \in \mathbb{R}^{n\times n}$ have the SVD $A = U\Sigma V^T$.

$$B = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$$

We have that:

$$\begin{aligned}
\det(B - zI) &= \begin{bmatrix} -zI & A \\ A^T & -zI \end{bmatrix} \\
&= \det(-zI) \cdot \det(-zI - A^T(-zI)^{-1}A) \quad \text{By Property 16 of Laub} \\
&= (-z)^n \cdot \det((z^2I - A^TA)/(-z)) \quad \text{By Property 10 of Laub, } \det(-zI) = (-z)^n \\
&= \det(z^2I - A^TA)
\end{aligned}$$

Recall that the eigenvalues of A^TA, which are simply $\sigma_1^2, \ldots, \sigma_n^2$ are the solutions of the equation $\det(A^TA - zI) = 0$ or equivalently $\det(zI - A^TA) = 0$. Thus the solutions of the equation $\det(z^2I - A^TA) = 0$, which are exactly the eigenvalues of B, are exactly $\sigma_1, \ldots, \sigma_n, -\sigma_1, \ldots, -\sigma_n$.

PROBLEM 2

Here $A = xy^T$ is written in outerproduct form. We can write $A = \sigma uv^T$ where $\sigma = \sqrt{\langle x, x\rangle \cdot \langle y, y\rangle}$, $u = x/\sqrt{\langle x, x\rangle}$ and $v = y/\sqrt{\langle y, y\rangle}$. Comparing this with the SVD form $A = U\Sigma V^T$ we have that the first column of U and V are equal to the u and v derived. The remaining columns of U and V can be any set of orthogonal vectors that is in the ortho-complement of u and v. Further, Σ has only one non zero element, given by σ. Since there is only one non-zero singular value, the rank of A is 1, which implies A has only one linearly independent column.

PROBLEM 3 (∗)

Let $A \in \mathbb{R}_n^{n\times n}$ be a square matrix with (full) rank n. The SVD of A always exists and is given by $A = U\Sigma V^T$. Since the rank of A is n, Σ is an $n \times n$ diagonal matrix with strictly positive entries. Since V is an orthogonal matrix, $V^TV = I$ and hence we can express A as :

$$\begin{aligned}
A &= U\Sigma V^T \\
&= UI\Sigma V^T \quad \text{Note that since } A \text{ is square } U\text{ , } V \text{ and } I \text{ have the same dimensions} \\
&= \underbrace{UV^T}_{=Q}\underbrace{V\Sigma V^T}_{=P}.
\end{aligned}$$

We note that $Q = UV^T$ is an orthogonal matrix since $Q^TQ = VU^TUV^T = VIV^T = I$ and $QQ^T = UV^TVU^T = UIU^T = I$. The matrix $P = V\Sigma V^T$ is symmetric since $P^T = V\Sigma V^T = P$. It is positive definite since its eigenvalues, which are exactly equal to the n strictly positive singular values of A, are strictly positive (Note this is a sufficient and necessary test for positive-definiteness of a matrix).

PROBLEM 4

Let z be one of the n roots of unity, i.e. $z \in \{e^{-\frac{j2\pi k}{n}} := \omega^k \,|\, k = 0, \ldots, n-1\}$, where $\omega := e^{-\frac{j2\pi}{n}}$. Let v_k denote the vector v when $z = \omega^k$.

Notice that:

$$
\begin{aligned}
v_k^H v_l &= \sum_{i=0}^{n-1} \omega^{-ik}\omega^{il} \\
&= \sum_{i=0}^{n-1} \omega^{i(l-k)} \\
&= \begin{cases} 0 & l \neq k \\ \sum_{i=0}^{n-1} 1 & l = k \end{cases} \\
&= \begin{cases} 0 & l \neq k \\ n & l = k \end{cases}. \qquad (1)
\end{aligned}
$$

This may be computed as the sum of the geometric series $\{1, \omega^{l-k}, \omega^{2(l-k)}, \ldots, \omega^{(n-1)(l-k)}\}$, or by using the fact that the n roots of unit solve $z^n - 1 = 0$, and therefore sum to $\frac{\text{coefficient of (n−1) order term}}{\text{coefficient of n order term}} = 0$.
Now,

$$
Cv_k = \begin{bmatrix} c_0 & c_{n-1} & \ldots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-1} & c_{n-2} & \ldots & c_1 & c_0 \end{bmatrix} \begin{bmatrix} 1 \\ \omega^k \\ \omega^{2k} \\ \vdots \\ \omega^{(n-1)k} \end{bmatrix}
$$

$$
= \begin{bmatrix} c_0 + c_{n-1}\omega^k + \ldots + c_2\omega^{(n-2)k} + c_1\omega^{(n-1)k} \\ c_1 + c_0\omega^k + c_{n-1}\omega^{2k} + \ldots + c_2\omega^{(n-1)k} \\ \vdots \\ c_{n-1} + c_{n-2}\omega^k + \ldots + c_1\omega^{(n-2)k} + c_0\omega^{(n-1)k} \end{bmatrix}
$$

$$
= \begin{bmatrix} c_0 + c_{n-1}\omega^k + \ldots + c_2\omega^{(n-2)k} + c_1\omega^{(n-1)k} \\ \omega^k(c_1\omega^{(n-1)k} + c_0 + c_{n-1}\omega^k + \ldots + c_2\omega^{(n-2)k}) \\ \vdots \\ \omega^{(n-1)k}(c_{n-1}\omega^k + c_{n-2}\omega^{2k} + \ldots + c_1\omega^k + c_0) \end{bmatrix}
$$

$$
= \left(c_0 + c_{n-1}\omega^k + \ldots + c_2\omega^{(n-2)k} + c_1\omega^{(n-1)k}\right) \begin{bmatrix} 1 \\ \omega^k \\ \omega^{2k} \\ \vdots \\ \omega^{(n-1)k} \end{bmatrix}
$$

$$
= \lambda_k v_k
$$

Therefore, v_k is an eigenvector of C corresponding to eigenvalue λ_k.
Relation to DFT: Notice that the k^{th} eigenvalue λ_k is the expression for the DFT of the time reversed version of the signal $c = \begin{bmatrix} c_0 & c_1 & \ldots & c_{n-1} \end{bmatrix}^T$.

We have also shown that $\{v_k, k = 0, \ldots, n-1\}$ are orthogonal to each other in (1). In order to ensure unit norm for each of the eigenvectors, we divide each vector v_k by $\sqrt{v_k^H v_k} = \sqrt{n}$, so that the orthogonal matrix of eigenvectors becomes $V = \frac{1}{\sqrt{n}} \begin{bmatrix} v_0 & v_1 & \ldots & v_{n-1} \end{bmatrix}$. If we define a diagonal matrix $\Lambda \in \mathbb{C}^{n \times n}$ such that $\Lambda_{ii} = \lambda_{i-1}$, $i = 1, \ldots, n$, we can write the eigendecomposition of C as $C = V\Lambda V^{-1}$. Notice that the eigenevectors do not depend upon $\{c_i\}$, which shows that all circulant matrices are diagonalized by DFT matrices.

NOTE: If we define a new matrix $\widetilde{C}$ as:

$$\widetilde{C} = C^T = \begin{bmatrix} c_0 & c_1 & \ldots & c_{n-1} \\ c_{n-1} & c_0 & \ldots & c_{n-2} \\ & \vdots & \vdots & \\ c_1 & c_2 & \ldots & c_0 \end{bmatrix}$$

then the eigenvalues of $\widetilde{C}$ will be exactly the DFT coefficients of the vector $\begin{bmatrix} c_0 & c_1 & \ldots & c_{n-1} \end{bmatrix}^T$. `MATLAB`'s computation of eigenvalues might not correspond to the ordering defined by the DFT. Recollect that the DFT of the time reversed version of a signal is a time reversal of the DFT of the original signal. Hence, while the *set* of eigenvalues returned by the `eig` command will be the same, the eigenvalues of C and $\widetilde{C}$ will correspond to different eigenvectors.

The following `MATLAB` code illustrates a few of these concepts:

```
% Create c vector : [c_0 ... c_{n-1}]
n = 3;
c = randn(n,1);

% Generate circulant matrix C
C = zeros(n);
for idx = 1:n
    C(:,idx) = circshift(c,(idx-1));
end

% Compute the eigendecomposition of C
%
% Sort the eigenvalues in descending order, and reorder the eigenvectors
% correspondingly, to make comparison with DFT coefficients easier.
[QC,LambdaC] = eig(C);
[lambdaC_ord,qCidx] = sort(diag(LambdaC),'descend');
QC = QC(:,qCidx);


% Since eigenvalues are the DFT of the time reversed signal version of c,
% compute its time reversal ctil, and then its DFT.
ctil = [c(1); flipud(c(2:end))];
dftctil = fft(ctil);
[ctilft_ord, ctilftidx] = sort(dftctil, 'descend');

disp('----------------------------------------------------------');
```

```
disp('Eigenvalues of C (in descending order): ');
disp(lambdaC_ord);
disp('DFT of ctil = [c_0, c_{n-1} ... c_1] (in descending order): ');
disp(ctilft_ord);


% Generate the matrix V of eigenvectors, and reorder them so that they
% correspond to the order of the sorted DFT values.
e = ones(n,1);
gridkn = (e*(0:n-1)).*((0:n-1)'*e');
V = 1/sqrt(n)*exp((-1j*2*pi/n).*gridkn);

Vctil = V(:,ctilftidx);

% Compare with the eigenvectors computed using eig()
disp('');
disp('abs(inner product) of eigenvectors using eig() and solution');
disp(num2str(abs(QC'*Vctil)));


% Repeat the process for Ctil = transpose(C) as in 'NOTE'
Ctil = transpose(C);
[QCtil, LambdaCtil] = eig(Ctil);
[lambdaCtil_ord, qCtilidx] = sort(diag(LambdaCtil),'descend');
QCtil = QCtil(:,qCtilidx);

dftc = fft(c);
[cft_ord, cftidx] = sort(dftc, 'descend');

disp('-----------------------------------------------------------');
disp('Eigenvalues of Ctil = transpose(C) (in descending order): ');
disp(lambdaCtil_ord);
disp('DFT of c = [c_0, c_1, ... c_{n-1}] (in descending order): ');
disp(cft_ord);

Vc = V(:,cftidx);
disp('');
disp('abs(inner product) of eigenvectors using eig() and solution');
disp(num2str(abs(QCtil'*Vc)));


% Compare the eigenvector matrices for C and Ctil:
disp('');
disp('-----------------------------------------------------------');
disp('abs(inner product) of eigenvectors of C and transpose(C):');
disp(abs(QCtil'*QC));
```

a. We have that

$$A = xy^T = \frac{x}{||x||_2}||x||_2||y||_2\frac{y^T}{||y||_2}$$

$$= \begin{bmatrix} \frac{x}{||x||_2} & U^{\perp} \end{bmatrix} \begin{bmatrix} ||x||_2||y||_2 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ & & \ddots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \frac{y}{||y||_2}^T & V^{\perp} \end{bmatrix}. \tag{2}$$

We know that xy^T is a rank–1 matrix, and thus has only one singular value. Furthermore, $\frac{x}{||x||_2}$ and $\frac{y}{||y||_2}$ are unit norm vectors, and $||x||_2||y||_2$ is a positive real number. Therefore, $\sigma_1 = ||x||_2||y||_2$ is the only singular value of A, and $u_1 = \frac{x}{||x||_2}$, $v_1 = \frac{y}{||y||_2}$ are the corresponding left and right singular vectors. The full Singular Value Decomposition of A is given by (2), where $U^{\perp}$ is any set of orthonormal vectors orthogonal to u_1, and $V^{\perp}$ is any set of orthonormal vectors, orthonogonal to v_1. Using the definition of pseudoinverse,

$$A^{\dagger} = \frac{y}{||y||_2}\frac{1}{||x||_2||y||_2}\frac{x^T}{||x||_2} = \frac{yx^T}{||x||_2^2||y||_2^2}.$$

b. This is a special case of Problem 6a in which $y = x$. Thererfore,

$$A^{\dagger} = \frac{x}{||x||_2}\frac{1}{||x||_2||x||_2}\frac{x^T}{||x||_2} = \frac{xx^T}{||x||_2^4}.$$

PROBLEM 6 ($*$)

From HW 2, we have that for fixed i, the matrix

$$A_i = \begin{bmatrix} \alpha_i^T & 0^T & -\widetilde{x}_i\alpha_i^T \\ 0^T & \alpha_i^T & -\widetilde{y}_i\alpha_i^T \end{bmatrix}$$

satisfies the equation

$$A_i h = 0.$$

Therefore, for $i = 1, \dots, n$,

$$\underbrace{\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix}}_{:=A} h = 0.$$

gives the required matrix A. $h \in \mathcal{N}(A)$. For h to be unique up to a scaling, we require that $\dim\mathcal{N}(A) = 1$, for if $\dim\mathcal{N}(A) = q > 1$, then we can find q basis vectors $\{b_1, \dots b_q\}$ for $\mathcal{N}(A)$, and every linear combination $h = \alpha_1 b_1 + \dots + \alpha_q b_q$ satisfies the equation $Ah = 0$. By rank-plus-nullity theorem,

$$\dim\mathcal{N}(A) + \text{rank}(A) = \#\text{ of columns of } A$$
$$1 + \text{rank}(A) = 9$$
$$\text{rank}(A) = 8.$$

Since $\text{rank}(A) \leq \min(2n, 9)$, we set $2n = 8$ to get $n = 4$.

(b) One possible `MATLAB` implementation is

```
function H = hw3p6b(XY1,XY2)
%
```

```
%
% Syntax:       H = hw3p6b(XY1,XY2);
%
% Inputs:       XY1 and XY2 are n x 2 matrices of [x, y] coordinates for n
%               corresponding points in coordinate systems 1 and 2,
%               respectively
%
% Outputs:      H is the unique 3 x 3 projective transformation matrix that
%               maps XY1 to XY2 with H(3,3) = 1. That is, in the noiseless
%               case, the following relationship should hold:
%
%               tmp = H * [XY1, ones(n,1)]';
%               XY2 = tmp(1:2,:)' ./ (tmp(3,:)' * [1 1]);
%

% Compute transformation
n = size(XY1,1);
A = zeros(0,9);
for i = 1:n
    XY1i = [XY1(i,:), 1];
    Ai = [XY1i       , zeros(1,3), -XY2(i,1) * XY1i;
          zeros(1,3), XY1i,        -XY2(i,2) * XY1i];
    A = [A; Ai]; %#ok
end
[~,~,V] = svd(A);
H = reshape(V(:,end) / V(end),3,3)';
```

PROBLEM 7 ($*$)

This was an exploratory problem.

PROBLEM 8

Approach 1 Let $0 = A^T x$, then $x \in \mathcal{N}(A^T)$. Let y be any arbitrary vector. We have that

$$y^T A^T x = 0$$
$$\implies (Ay)^T x = 0.$$

Therefore $Ay \perp x$, for any arbitrary vector y and $x \in \mathcal{N}(A^T)$. But $\mathcal{R}(A) = \{Ay\}$, which allows us to conclude that $\mathcal{R}(A) \perp \mathcal{N}(A^T)$, or that $\mathcal{R}(A)^\perp = \mathcal{N}(A^T)$.

Approach 2 Let $A \in \mathbb{R}_r^{m\times n}$. Then, $A = U\Sigma V^T$ and $A^T = V\Sigma^T U^T$. Using our discussion on the anatomy of the SVD,

$$A = \left[\underbrace{u_1 \quad \dots \quad u_r}_{\text{basis for } \mathcal{R}(A)} \quad \underbrace{u_{r+1} \quad \dots \quad u_m}_{\text{basis for } \mathcal{R}(A)^\perp} \right] \Sigma \left[\underbrace{v_1 \quad \dots \quad v_r}_{\text{basis for } \mathcal{N}(A)^\perp} \quad \underbrace{v_{r+1} \quad \dots \quad v_n}_{\text{basis for } \mathcal{N}(A)} \right]^T$$

$$A^T = \left[\underbrace{v_1 \quad \dots \quad v_r}_{\text{basis for } \mathcal{R}(A^T)} \quad \underbrace{v_{r+1} \quad \dots \quad u_n}_{\text{basis for } \mathcal{R}(A^T)^\perp} \right] \Sigma^T \left[\underbrace{u_1 \quad \dots \quad u_r}_{\text{basis for } \mathcal{N}(A^T)^\perp} \quad \underbrace{u_{r+1} \quad \dots \quad u_m}_{\text{basis for } \mathcal{N}(A^T)} \right]^T$$

We see that $R(A)^\perp = \text{span}\{u_{r+1}, \dots, u_m\} = \mathcal{N}(A^T)$.

PROBLEM 9 ($*$)

a) The problem of finding the line $y = \alpha x + b$ that best fits the points $(1, 2)$, $(2, 1)$ and $(3, 3)$ is equivalent to the problem:

$$\text{Find } \beta \text{ that minimizes } ||z - A\beta||_2$$

with $z = \begin{bmatrix} 2 & 1 & 3 \end{bmatrix}^T$, $A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix}$ and $\beta = \begin{bmatrix} \alpha & b \end{bmatrix}^T$.

Thus the optimal $\widehat{\beta} = A^\dagger z = V\Sigma^+ U^T z$, where $A = U\Sigma V^T$ is the singular value decomposition of A. From MATLAB we have that:

$$U = \begin{bmatrix} -0.3231 & 0.8538 & 0.4082 \\ -0.5475 & 0.1832 & -0.8165 \\ -0.7719 & -0.4873 & 0.4082 \end{bmatrix}$$

,

$$V = \begin{bmatrix} -0.9153 & -0.4027 \\ -0.4027 & 0.9153 \end{bmatrix}$$

and

$$\Sigma = \begin{bmatrix} 4.0791 & 0 \\ 0 & 0.6005 \\ 0 & 0 \end{bmatrix}$$

so that

$$\widehat{\beta} = V\Sigma^\dagger U^T = V \begin{bmatrix} 0.2451 & 0 & 0 \\ 0 & 1.6653 & 0 \end{bmatrix} U^T z = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}.$$

b) Here we are asked to find the line $y = \alpha x + b$ that best fits the points $(2, 1)$, $(1, 2)$ and $(3, 3)$. We can repeat the above computation or we can recognize that all that has changed is that points 1 and 2 in a) have been swapped with points 2 and 1 in b). Thus the solution will be identical since the least-squares criterion does not differentiate between ordering of points. A more formal way of seeing is to first note that the new A matrix for b) is exactly same as the A matrix in a) except for the first two rows swapping places. Denote the "A" matrix in part a) by A_a and the "A" matrix in part b) by A_b. Then

$$A_b = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{=P} A_a.$$

Notice that P is a orthogonal matrix since $PP^T = I$. Such a P is referred to as a permutation matrix. We also have that the new $z_b = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^T = P z_a$. Thus the new problem is

$$\text{Find } \beta_b \text{ that minimizes } ||z_b - A_b\beta||_2 = ||Pz_a - PA_a\beta||_2 = ||z_a - A_a\beta||_2,$$

since P is an orthogonal matrix. Thus we get the same solution.

PROBLEM 10 (*)

For this problem, let us define for integer values d:

$$y = \begin{bmatrix} f(t_1) \\ \vdots \\ f(t_{16}) \end{bmatrix} \in \mathbb{R}^{16}$$

$$A = \begin{bmatrix} 1 & t_1 & t_1^2 & \dots & t_1^d \\ \dots & \dots & \dots & \dots & \\ 1 & t_{16} & t_{16}^2 & \dots & t_{16}^d \end{bmatrix} \in \mathbb{R}^{16\times(d+1)}$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_{d+1} \end{bmatrix} \in \mathbb{R}^{d+1}.$$

Then the solution:

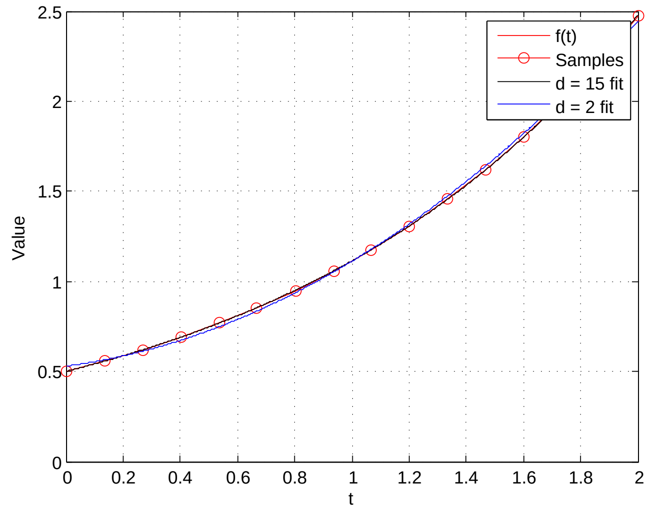
$$\widehat{x} = A^{\dagger} y$$

yields the the desired optimal least-squares estimate of the coefficients of the degree-d polynomial $p_d(t) = \sum_{i=1}^{d+1} x_i t^{i-1}$ that minimizes the error $||y - Ax||_2$. A of the form above is referred to as a Vandermonde matrix so that my code in MATLAB for the first part is given below:

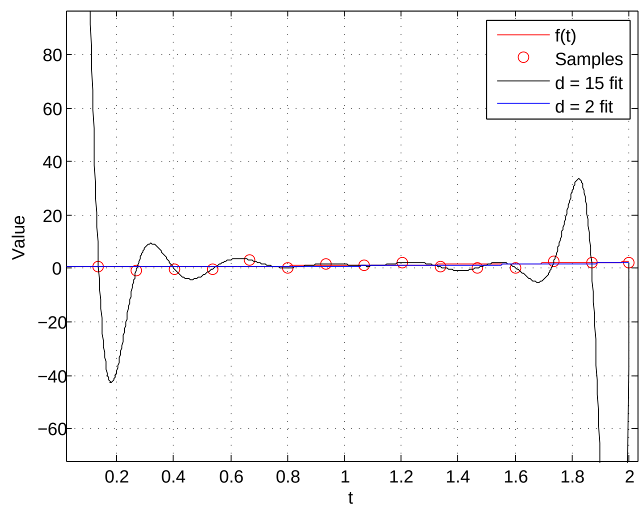
```
T = [2.1e-3 0.136 0.268 0.402 0.536 0.668 0.802 0.936 1.068 1.202 1.336 1.468 1.602 1.736 ...
    1.868 2.000];
f =  0.5* exp(0.8*T');
d = 15;
A15 = [];
for idx = 1 : d+1,
A15 = [A15 T'.^(idx-1)];
end

A2 = A15(:,1:3); % for degree 2
x16 = pinv(A15)*f;
x2 = pinv(A2)*f;

t = [0:1000]'/500;
plot(t,0.5*exp(0.8*t),'red');
hold on
plot(T,f,'o');
a15 = []'
for idx = 1 : d+1,
a16 = [a15 t.^(idx-1)];
end

a2 = a15(:,1:3);
plot(t,a15*x15,'black');
plot(t,a2*x2,'blue');
legend('f(t)','Samples','d = 15 fit','d = 2 fit')
grid on
xlabel('t'); ylabel('Value');
```

This yielded Figure 1a. Notice that the error for $d = 15$ is much smaller. In fact it will be exactly zero since $rank(A) = 16 = dim(y)$ so that we have an exact solution! Comparing the least-squares coefficients obtained for $d = 2$ and $d = 15$ reveals that the first three terms are about the same. When noise is added we get Figure 1b. Notice that the polynomial for $d = 15$ passes *exactly* through the noisy samples. This follows from the previous argument since $rank(A) = 16$ so that error will be zero, relative to the samples, but large relative to the function. This is another example of why choosing large model orders ($= d$) can make things worse. Let us pursue this line of inquiry further and examine the condition number for the $d = 15$ case. In MATLAB I type:


```
>> cond(A15)

ans =

3.0448e+013

>> min(svd(A15))

ans =

1.3343e-009

>> max(svd(A15))

ans =

4.0626e+004
```



(A) Problem 3-part a) - when there is no noise.



(B) Problem 3-part b) - when there is noise.

FIGURE 1. Problem 3.

to obtain the condition number which is the ratio of the largest and smallest singular value of a matrix. The condition number is important because the error manifests as a δb so that the error in the least squares solution (relative to the noise-less setting) is given by exactly by $\delta\widehat{x} = A^{\dagger}(b+\delta b) - A^{\dagger}b = A^{\dagger}\delta b$. The large condition means that small δb will get amplified producing large δx and hence instabilities as manifested in the overfitting. Theorem 6.12 of Elden provides an estimate of the degree to which δx can change as a function of the condition number κ.

PROBLEM 11 $(*)$

Here we have

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

so that by definition,

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

and hence the optimal least-squares estimate is:

$$\widehat{x} = A^{\dagger} b = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

a) Consider

$$A_1 = \begin{bmatrix} 1 & \delta \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & \delta \end{bmatrix}$$

so that by inspection we can deduce that the SVD of $A_1 = \sigma_1 u_1 v_1^T$ where $\sigma_1 = \sqrt{1+\delta^2}$, $u_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ and $v_1 = \begin{bmatrix} 1/\sqrt{1+\delta^2} & \delta/\sqrt{1+\delta^2} \end{bmatrix}^T$. We do not need to compute the other singular vectors since the optimal (minimum-norm) least-squares solution is simply:

$$\widehat{x}_1 = A_1^{\dagger} b = \frac{1}{\sigma_1} v_1 u_1^T b = \frac{1}{1+\delta^2} \begin{bmatrix} 1 \\ \delta \end{bmatrix}.$$

Notice that

$$\widehat{x}_1 \longrightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \text{as } \delta \longrightarrow 0$$

so that $||\widehat{x}_1 - x||_2 \longrightarrow 0$.

b) Consider now the setting where

$$A_2 = \begin{bmatrix} 1 & 0 \\ 0 & \delta \end{bmatrix}$$

which has $rank(A) = 2$ for $\delta > 0$. Here we have that the optimal least-squares solution is:

$$\widehat{x}_2 = A_2^{\dagger} b = \begin{bmatrix} 1 \\ \frac{1}{\delta} \end{bmatrix}$$

so that $||\widehat{x} - \widehat{x}_2||_2 \longrightarrow \infty$ as $\delta \longrightarrow 0$.

The point of this exercise is that the manner in which the error is introduced can either impact the problem negligibly or dramatically!