1. Quadratic extrapolation of a time series. We are given a series $z$ up to time $t$. Using a quadratic model, we want to extrapolate, or predict, $z(t+1)$ based on the three previous elements of the series, $z(t)$, $z(t-1)$, and $z(t-2)$. We'll denote the predicted value of $z(t+1)$ by $\hat{z}(t+1)$. More precisely, you will find $\hat{z}(t+1)$ as follows.

a) Find the quadratic function $f(\tau) = a_2\tau^2 + a_1\tau + a_0$ which satisfies $f(t) = z(t)$, $f(t-1) = z(t-1)$, and $f(t-2) = z(t-2)$. Then the extrapolated value is given by $\hat{z}(t+1) = f(t+1)$.

Show that $\hat{z}(t+1) = c \begin{bmatrix} z(t) \\ z(t-1) \\ z(t-2) \end{bmatrix}$, where $c \in \mathbb{R}^{1 \times 3}$, and does not depend on $t$. In other words, the quadratic extrapolator is a linear function. Find $c$ explicitly.

b) Use the following matlab code to generate a time series $z$:

```matlab
t = 1:1000;
z = 5*sin(t/10 + 2) + 0.1*sin(t) + 0.1*sin(2*t - 5);
```

Use the quadratic extrapolation method from part (a) to find $\hat{z}(t)$ for $t = 4, \ldots, 1000$. Find the relative root-mean-square (RMS) error, which is given by

$$\left( \frac{1}{997} \sum_{j=4}^{1000} (\hat{z}(j) - z(j))^2 \right)^{1/2} \left( \frac{1}{997} \sum_{j=4}^{1000} z(j)^2 \right)^{1/2}.$$ 

Solution.

a) Setting $f(t) = z(t)$, $f(t-1) = z(t-1)$ and $f(t-2) = z(t-2)$ gives the following system of linear equations:

$$a_2 t^2 + a_1 t + a_0 = z(t)$$
$$a_2 (t-1)^2 + a_1 (t-1) + a_0 = z(t-1)$$
$$a_2 (t-2)^2 + a_1 (t-2) + a_0 = z(t-2)$$
with solution

\[
\begin{align*}
a_0 &= (0.5t^2 - 1.5t + 1)z(t) + (2t - t^2)z(t - 1) + (0.5t^2 - 0.5t)z(t - 2) \\
a_1 &= (1.5 - t)z(t) + (2t - 2)z(t - 1) + (0.5 - t)z(t - 2) \\
a_2 &= 0.5z(t) + \left(\frac{5}{2} - t\right)z(t - 1) + 0.5z(t - 2).
\end{align*}
\]

Substituting in \( \dot{z}(t + 1) = a_2(t + 1)^2 + a_1(t + 1) + a_0 \) and simplifying, we get

\[
\dot{z}(t + 1) = 3z(t) - 3z(t - 1) + z(t - 2).
\]

Hence,

\[
c = \begin{bmatrix} 3 & -3 & 1 \end{bmatrix}.
\]

Observe that \( c \) does not depend on \( t \), but the coefficients \( a_0, a_1 \) and \( a_2 \) do. In other words, the quadratic extrapolator \( f \) varies between samples, but its value at \( t + 1 \) is always given by the same combination of \( z(t), z(t - 1) \) and \( z(t - 2) \).

An alternative solution in which \( c \) is found directly without solving for \( a_0, a_1, a_2 \) is possible by equating the coefficients for the various powers of \( t \).

\[
\begin{align*}
\dot{z}(t + 1) &= c_1 z(t) + c_2 z(t - 1) + c_3 z(t - 2) \\
a_2(t + 1)^2 + a_1(t + 1) + a_0 &= c_1 \left[ a_2(t)^2 + a_1(t) + a_0 \right] \\
&\quad + c_2 \left[ a_2(t - 1)^2 + a_1(t - 1) + a_0 \right] \\
&\quad + c_3 \left[ a_2(t - 2)^2 + a_1(t - 2) + a_0 \right] \\
t^2(a_2) + t(2a_2 + a_1) + (a_2 + a_1 + a_0) &= c_1 \left[ t^2(a_2) + t(a_1) + (a_0) \right] \\
&\quad + c_2 \left[ t^2(a_2) + t(-2a_2 + a_1) + (a_2 - a_1 + a_0) \right] \\
&\quad + c_3 \left[ t^2(a_2) + t(-4a_2 + a_1) + (4a_2 - 2a_1 + a_0) \right] 
\end{align*}
\]

In order for this equation to hold for all \( t \), the coefficients for each power of \( t \) \((t^2, t^1, t^0)\) must be equal. First consider the coefficients for \( t^2 \):

\[
a_2 = c_1a_2 + c_2a_2 + c_3a_2 \\
\therefore c_1 + c_2 + c_3 = 1
\]

For \( t \), and using the equation for \( t^2 \) during simplifications, we get

\[
\begin{align*}
2a_2 + a_1 &= c_1a_1 + c_2a_1 - 2c_2a_2 + c_3a_1 - 4c_3a_2 \\
2a_2 + a_1 &= a_1(c_1 + c_2 + c_3) - 2a_2(c_2 + 2c_3) \\
\therefore c_2 + 2c_3 &= -1
\end{align*}
\]
Finally, consider the constant terms:

\[
\begin{align*}
a_2 + a_1 + a_0 &= c_1a_0 + c_2a_2 - c_2a_1 + c_2a_0 + 4c_3a_2 - 2x_3a_1 + c_3a_0 \\
a_2 + a_1 + a_0 &= a_0(c_1 + c_2 + c_3) - a_1(c_2 + 2c_3) + a_2(c_2 + 4c_3) \\
\therefore c_2 + 4c_3 &= 1
\end{align*}
\]

We now have three equations in three unknowns, and can easily solve for \(c\) to find

\[
c = \begin{bmatrix} 3 & -3 & 1 \end{bmatrix}.
\]

b) The following matlab code computes the predicted values and finds that the relative RMS error is 0.097:

```matlab
t = 1:1000;
z = 5*sin(t/10 + 2) + 0.1*sin(t) + 0.1*sin(2*t - 5);
c = [3 -3 1];
for j=4:1000
    zhat(j) = c*z(j-1:-1:j-3)';
end
residual = zhat(4:1000)-z(4:1000);
rel_rms = sqrt(mean(residual.^2)/mean(z(4:end).^2))
figure;
plot(t, z, '-', t, zhat, '--');
xlim([0, 100])
xlabel('t'); ylabel('value');
legend('z(t)', 'hatz(t)');
```

In order to get an idea of how good the approximation is, we plot the first 100 samples:

![Graph of z(t) and hatz(t)](image)
2. **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:

\[
L_{\text{cone}} = \sum_{i=1}^{20} l_i p_i, \quad M_{\text{cone}} = \sum_{i=1}^{20} m_i p_i, \quad S_{\text{cone}} = \sum_{i=1}^{20} s_i p_i,
\]

where \( p_i \) is the incident power in the \( i \)th 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 \( p_{\text{test}} \). 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.m`, 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 \( i = 1, \ldots, 20 \) 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 \((p - \tilde{p}) \in \text{null}(A)\).

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
\[
Ap_{\text{test}} = A \begin{bmatrix} u & v & w \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix},
\]

but 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 \(p_{\text{test}}\), a match is always possible if \(B\) is invertible. In this case, if \(a_i < 0\), the \(i\)th light should be mixed with \(p_{\text{test}}\) 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.

The following Julia code illustrates the steps.

```julia
# Extraction of the data

include("readJSON263.jl");
mydata = readJSON263("color_perception.json");

L_coefficients = mydata["L_coefficients"]
M_coefficients = mydata["M_coefficients"]
S_coefficients = mydata["S_coefficients"]
R_phosphor = mydata["R_phosphor"]
G_phosphor = mydata["G_phosphor"]
B_phosphor = mydata["B_phosphor"]
test_light = mydata["test_light"]
tungsten = mydata["tungsten"]
sunlight = mydata["sunlight"]

A = [L_coefficients; M_coefficients; S_coefficients];
B = A*[R_phosphor' G_phosphor' B_phosphor'];
weights = B\A*test_light
```

Equivalently, the following matlab code illustrates the steps.

```matlab
close all; clear all;
color_perception;
A = [L_coefficients; M_coefficients; S_coefficients]; B = A*[R_phosphor' G_phosphor' B_phosphor'];
weights = inv(B)*A*test_light;
```

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

\[
A \begin{bmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{20} \end{bmatrix} = I
\]

Note that \( I \) here is not the identity matrix, but the vector of incoming light spectrum!

This is equivalent to saying \((R - \tilde{R})I \in \text{null}(A)\). The objects will look different under \( \tilde{I} \) if, additionally, \( ARI \neq A\tilde{R}\tilde{I} \) which means that \((R - \tilde{R})\tilde{I} \not\in \text{null}(A)\). 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.

```matlab
n = N(:,1);
```
n = n*10;

for i in 1:20
n[i] = n[i]/tungsten[i];
end

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 - n;

t1 = zeros(20);
t2 = zeros(20);

for i in 1:20
t1[i] = r1[i]*tungsten[i];
t2[i] = r2[i]*tungsten[i];
end

color1_tungsten = A*t1'
color2_tungsten = A*t2'

for i in 1:20
s1[i] = r1[i]*sunlight[i];
s2[i] = r2[i]*sunlight[i];
end

color1_sunlight = A*s1'
color2_sunlight = A*s2'

Or, in Matlab:

close all; clear all;
color_perception;
A = [L_coefficients; M_coefficients; S_coefficients]; N = null(A);
n = N(:,1);
n= n*10;
for i = 1:20
n(i) = n(i)/tungsten(i);
end
r1 = [0; .2; .3; .7; .7; .8; .8; .2; .9; .8; .2; .8; .9; .2; .8; .3; .8; .7; .2; .4];
r2 = r1-n;
for i = 1:20
t1(i) = r1(i)*tungsten(i);
t2(i) = r2(i)*tungsten(i);
end color1_tungsten = A*t1'; color2_tungsten = A*t2';
for i = 1:20
\[ s_1(i) = r_1(i) \times \text{sunlight}(i); \]
\[ s_2(i) = r_2(i) \times \text{sunlight}(i); \]
\[ \text{end} \]
\[ \text{color}_1\_\text{sun} = A \times s_1'; \]
\[ \text{color}_2\_\text{sun} = A \times s_2'; \]

253.5187

3. **Gambler’s ruin.** Consider a gambling situation involving two players \textit{A} and \textit{B}. An example is roulette where, say, player \textit{A} is a \textit{guest} and player \textit{B} is the \textit{house}. During any one play of the game there is a probability \( p, 0 < p < 1 \), that player \textit{A} wins a chip (or coin) from player \textit{B}, and a probability \( q = 1 - p \) that Player \textit{B} wins a chip from player \textit{A}. The players begin with initial holdings of \( a \) and \( b \) chips, respectively. A player wins overall if she obtains all the chips.

a) Find the probability that player \textit{A} wins.

\textit{Hint.} This might sound like a problem for a probability and statistics course, but we want you to approach this problem from a linear dynamical systems point of view. Consider the general situation where \textit{A} has \( k \) chips. Denote the probability under these circumstances that player \textit{A} eventually wins by \( u(k) \). Assume \( u(k) \) is the state of the system you are analyzing. Can you write a difference equation that describes the dynamics of \( u(k) \)? To solve your difference equation you can assume the solution has the general form \( u(k) = \lambda^k \) (we will see why later in the class). You will also need to come up with two initial conditions to uniquely solve your difference equation. Think of \( u(k) \) when player \textit{A} has no chips, or has \( a + b \) chips.

b) As a specific example, suppose you play a roulette wheel that has 37 divisions: 18 are red, 18 are black and one is green. If you bet on either red or black, you win a sum equal to your bet if the outcome is a division of that color (You cannot bet on green). Otherwise you loose your bet. If the bank has 1000 chips and you have 100 chips, what is the chance that you can \textit{break the bank}, betting only one chip on red or black each spin of the wheel?

**Solution.**

a) Assuming player \textit{A} has \( k \) chips, at the conclusion of the next play she will have either \( k + 1 \) or \( k - 1 \) chips, depending on whether she wins or loses that play. The probabilities of eventually winning must therefore satisfy the difference equation:

\[ u(k) = u(k|\text{win kth play})P(\text{win kth play}) + u(k|\text{loose kth play})P(\text{loose kth play}) \]
\[ = pu(k + 1) + qu(k - 1) \]

In addition, we have the two initial (boundary) conditions:

\[ u(0) = 0 \quad u(a + b) = 1 \]

This difference equation for \( u(k) \) is linear and has constant coefficients. Assuming a general solution of the form \( u(k) = \lambda^k \) and plugging it into the difference equation, after some simplification we get:

\[ -p\lambda^2 + \lambda - q = 0 \]
The corresponding roots are \( \lambda = 1, \lambda = q/p \). Accordingly, the general solution assuming \( q \neq p \) is:

\[
 u(k) = c_1 + c_2(q/p)^k
\]

The two initial conditions give the equations:

\[
\begin{align*}
0 &= c_1 + c_2 \\
1 &= c_1 + c_2(q/p)^{a+b}
\end{align*}
\]

After solving for \( c_1 \) and \( c_2 \) and substituting the results into the general solution, we get:

\[
 u(k) = \frac{1 - (q/p)^k}{1 - (q/p)^{a+b}}
\]

Finally, at the original position where player \( A \) has \( a \) chips, the corresponding probability of winning is:

\[
 u(a) = \frac{1 - (q/p)^a}{1 - (q/p)^{a+b}}
\]

b) In this case:

\[
 p = \frac{18}{37}, \quad q = \frac{19}{37}, \quad a = 100, \quad b = 1000
\]

Thus:

\[
 u(100) = \frac{1 - (19/18)^{100}}{1 - (19/18)^{1100}} = 3.29 \times 10^{-24}
\]

4. Some properties of the product of two matrices. For each of the following statements, either show that it is true, or give a (specific) counterexample.

- If \( AB \) is full rank then \( A \) and \( B \) are full rank. False. Consider the following counterexample:

  \[
  A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \quad AB = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}
  \]

  Clearly \( AB \) is full rank while \( B \) is not.

- If \( A \) and \( B \) are full rank then \( AB \) is full rank.

- If \( A \) and \( B \) have zero nullspace, then so does \( AB \).

- If \( A \) and \( B \) are onto, then so is \( AB \).

You can assume that \( A \in \mathbb{R}^{m \times n} \) and \( B \in \mathbb{R}^{n \times p} \). Some of the false statements above become true under certain assumptions on the dimensions of \( A \) and \( B \). As a trivial example, all of the statements above are true when \( A \) and \( B \) are scalars, i.e., \( n = m = p = 1 \). For each of the statements above, find conditions on \( n, m, \) and \( p \) that make them true. Try to find the most general conditions you can. You can give your conditions as inequalities involving \( n, m, \) and \( p \), or you can use more informal language such as “\( A \) and \( B \) are both skinny.”

Solution. First note that an \( m \times n \) matrix is full rank if and only if the maximum number of independent columns or rows is equal to \( \min\{m, n\} \).

- If \( AB \) is full rank then \( A \) and \( B \) are full rank. False. Consider the following counterexample:

  \[
  A = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, \quad AB = \begin{bmatrix} 1 & 1 \end{bmatrix}
  \]

  Clearly \( AB \) is full rank while \( B \) is not.
• If $A$ and $B$ are full rank then $AB$ is full rank. \textit{False}. Consider:

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

Clearly, $A$ and $B$ are full rank while $AB$ is not.

• If $A$ and $B$ have zero null space, then so does $AB$. \textit{True}. The proof is easy. We will prove that $ABx = 0$ implies that $x = 0$ and hence $\text{null}(AB) = \{0\}$. If $ABx = 0$, since $A$ has zero null space then $Bx = 0$. Now since $B$ has zero null space this implies that $x = 0$ and we are done.

• If $A$ and $B$ are onto, then so is $AB$. \textit{True}. We need to show that $y = ABx$ can be solved in $x$ given any $y$. Suppose that $y \in \mathbb{R}^m$ is arbitrary. Since $A$ is onto, then $y = A\tilde{x}$ holds for some $\tilde{x} \in \mathbb{R}^n$. Now consider the equation $\tilde{x} = Bx$. Since $B$ is onto, then $\tilde{x} = Bx$ holds for some $x \in \mathbb{R}^p$. This proves that $y = ABx$ is solvable in $x$ with $y = A\tilde{x}$ and $\tilde{x} = Bx$ and we are done.

Now we will find conditions under which the first two statements are correct. We will give these conditions based on the relative sizes of $m$, $n$ and $p$, \textit{i.e.}, when $A$ is fat or skinny, $B$ is fat or skinny, or $AB$ is fat or skinny. We consider a square matrix to be both fat and skinny. There are 8 possible cases to check, but by using transposes we can reduce that down to 4 cases. For example lets consider the case when $AB$ is full rank and fat, $A$ is fat and $B$ is fat we are considering wheither $A$ and $B$ are full rank. Since $AB$ is full rank, $(AB)^T$ will also be full rank. We know that $(AB)^T = B^T A^T$ so the same results apply for $AB$ skinny, $B$ and $A$ skinny. First we consider the statement: “If $AB$ is full rank then $A$ and $B$ are full rank.”

• \textit{A fat, $B$ fat, $AB$ fat} (or \textit{A skinny, $B$ skinny, $AB$ skinny}). The statement is not true for this case. Consider the counter example:

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

• \textit{A fat, $B$ skinny, $AB$ fat} (or \textit{A fat, $B$ skinny, $AB$ skinny}). The statement is not true in this case. Consider:

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

However, if we add the constraint that $AB$ is square then the statement becomes correct. To show this we use the facts that for a full rank fat matrix $A$ all rows are independent, so $x^T A = 0$ implies $x = 0$, and for a full rank skinny matrix $B$ all columns are independent, so $Bx = 0$ implies that $x = 0$. We first prove by contradiction that $AB$ full rank implies that $A$ is full rank. If $A$ (fat) is not full rank, then there exists an $x \neq 0$ such that $x^T A = 0$, and therefore, $x^T AB = 0$. This implies that the rows of $AB$ (a square matrix) are dependent which is impossible since $AB$ is full rank and we are done. Now we prove
that $B$ should be full rank as well. If $B$ (skinny) is not full rank, then $Bx = 0$ for $x \neq 0$ which implies that $ABx = 0$, or the columns of $AB$ (a full rank square matrix) are dependent which is a contradiction. Hence $B$ is full rank too and we are done.

• $A$ skinny, $B$ fat, $AB$ fat (or $A$ skinny, $B$ fat, $AB$ skinny.) The statement is true in this case. First note that if $AB$ is full rank then $A$ should be square. We have

$$\text{rank}(AB) \leq \min\{\text{rank}(A), \text{rank}(B)\}$$

and since $A$ is skinny and $B$ is fat, $\text{rank}(A) \leq n$ and $\text{rank}(B) \leq n$ and therefore

$$\text{rank}(AB) \leq n.$$

Now since $AB$ is full rank and fat, then $\text{rank}(AB) = m$ so $m \leq n$. However, $A$ is skinny so $m \geq n$ and therefore we can only have $m = n$ or that $A$ is square. Now it is easy to prove that $AB$ full rank implies that $A$ and $B$ are full rank. We first prove that $A$ is full rank by contradiction. Suppose that $A$ (square) is not full rank so there exists a non-trivial linear combination of its rows that is equal to zero, i.e., $x \neq 0$ and $x^TA = 0$. Therefore, $x^TAB = 0$ which implies that a linear combination of the rows of $AB$ (a fat matrix) is zero which is impossible because $AB$ is full rank. This shows that $A$ should be full rank. Now we show that $B$ should be full rank as well. Since $A$ is full rank and square, then $A^{-1}$ exists so $B = A^{-1}(AB)$. Suppose that $B$ (fat) is not full rank so there exists an $x \neq 0$ such that $x^TB = 0$ and therefore $x^TA^{-1}(AB) = 0$. But $x^TA^{-1}$ is nonzero because $x$ is nonzero and $A^{-1}$ is invertible, which implies that a linear combination of the rows of $AB$ (a full rank fat matrix) is zero. This is impossible of course and we have shown by contradiction that $B$ should be full rank and we are done.

• $A$ fat, $B$ fat, $AB$ skinny (or $A$ skinny, $B$ skinny, $AB$ fat.) If $A$ is fat, $B$ is fat and $AB$ is skinny, then $A$, $B$ and $AB$ can only be square matrices. $A$ being fat implies that $m \leq n$ and $B$ being fat implies that $n \leq p$ and we get $p \geq m$. However, $p \leq m$ because $AB$ is skinny, so we can only have $m = p$, and therefore $m = n$ as well. In other words, $A$, $B$ and $AB$ are square. As a result, this case ($A$ square, $B$ square, $AB$ square) falls into the previous category ($A$ skinny, $B$ fat, $AB$ fat) and hence the statement is true.

To summarize, the most general conditions for the statement to be true are:

• $A$ fat, $B$ skinny, $AB$ square,
• $A$ square, $B$ fat, $AB$ fat,
• $A$ skinny, $B$ square, $AB$ skinny.

Comment: Another way to do this part:
The following inequalities are always true, regardless of the sizes of $A$, $B$ and $AB$:

$$\text{rank}(A) \leq \min\{m, n\}, \quad \text{rank}(B) \leq \min\{n, p\}$$

$$\text{rank}(AB) \leq \min\{\text{rank}(A), \text{rank}(B)\}$$
Since $AB$ is full rank, we also have $\text{rank}(AB) = \min\{m, p\}$. From this and the last inequality above we get the following:

$$\min\{m, p\} \leq \text{rank}(A) \leq \min\{m, n\}, \quad \min\{m, p\} \leq \text{rank}(B) \leq \min\{n, p\}$$

Now, with the three numbers $m$, $n$ and $p$, there are six different cases. However, as mentioned before, we only need to check three cases, since the other three can be obtained by taking transposes. Using the above inequalities in each case, we get:

- $m \leq n \leq p$: $\text{rank}(A) = m$, $m \leq \text{rank}(B) \leq n$
  
  Thus in this case $A$ will be full rank, but we can’t say anything about $B$. The only way to be able to infer that $B$ is also full rank is to have $m = n$. So the claim will be true if $m = n \leq p$.

- $m \leq p \leq n$: $\text{rank}(A) = m$, $m \leq \text{rank}(B) \leq p$
  
  Similar to the previous case, to be able to infer both $A$ and $B$ are full rank, we should have $m = p$. So the condition in this case will be $m = p \leq n$.

- $n \leq m \leq p$: $m \leq \text{rank}(A) \leq n$, but $n \leq m$, so we must have $m = n \leq p$, yielding $\text{rank}(A) = \text{rank}(B) = m$.

Therefore, the most general conditions where the claim is true are:

$m = n \leq p$, $n = p \leq m$, $m = p \leq n$

Which are the same conditions as the ones obtained before.

Now we consider the second statement: “If $A$ and $B$ are full rank then $AB$ is full rank.” Again we consider different cases:

- $A$ fat, $B$ fat, $AB$ fat (or $A$ skinny, $B$ skinny, $AB$ skinny.) The statement is true in this case. Since $AB$ is fat, we need to prove that $x^T AB = 0$ implies that $x = 0$. But this is easy: $x^T AB = 0$ implies that $x^T A = 0$ (because $B$ is fat and full rank) and $x^T A = 0$ implies that $x = 0$ (because $A$ is fat and full rank) and we are done.

- $A$ fat, $B$ skinny, $AB$ fat (or $A$ fat, $B$ skinny, $AB$ skinny.) The statement is not true in this case. Consider the counter example:

$$\begin{bmatrix} 1 & 0 \\ \hline \text{full rank} & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \hline \text{full rank} \end{bmatrix} = \begin{bmatrix} 0 \\ \hline \text{not full rank} \end{bmatrix}.$$ 

- $A$ skinny, $B$ fat, $AB$ fat (or $A$ skinny, $B$ fat, $AB$ skinny.) The statement is not true in this case. Consider:

$$\begin{bmatrix} 1 \\ \hline \text{full rank} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ \hline \text{full rank} & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \hline \text{full rank} & 0 \end{bmatrix}.$$ 

- $A$ fat, $B$ fat, $AB$ skinny (or $A$ skinny, $B$ skinny, $AB$ fat.) As shown previously, if $A$ is fat, $B$ is fat and $AB$ is skinny, then $A$, $B$ and $AB$ can only be square matrices. Therefore, this case falls into the category of $A$ fat, $B$ fat, $AB$ fat for which the statement is true.

To summarize, the statement is true only if

- $A$ fat, $B$ fat, $AB$ fat,
- $A$ skinny, $B$ skinny, $AB$ skinny.
5. Linearizing range measurements. Consider a single (scalar) measurement \( y \) of the distance or range of \( x \in \mathbb{R}^n \) to a fixed point or beacon at \( a \), i.e., \( y = \|x - a\| \).

a) Show that the linearized model near \( x_0 \) can be expressed as \( \delta y = k^T \delta x \), where \( k \) is the unit vector (i.e., with length one) pointing from \( a \) to \( x_0 \). Derive this analytically, and also draw a picture (for \( n = 2 \)) to demonstrate it.

b) Consider the error \( e \) of the linearized approximation, i.e.,
\[
e = \|x_0 + \delta x - a\| - \|x_0 - a\| - k^T \delta x.
\]
The relative error of the approximation is given by \( \eta = e / \|x_0 - a\| \). We know, of course, that the absolute value of the relative error is very small provided \( \delta x \) is small. In many specific applications, it is possible and useful to make a stronger statement, for example, to derive a bound on how large the error can be. You will do that here. In fact you will prove that
\[
0 \leq \eta \leq \frac{\alpha^2}{2}
\]
where \( \alpha = \|\delta x\| / \|x_0 - a\| \) is the relative size of \( \delta x \). For example, for a relative displacement of \( \alpha = 1\% \), we have \( \eta \leq 0.00005 \), i.e., the linearized model is accurate to about 0.005\%. To prove this bound you can proceed as follows:

- Show that \( \eta = -1 + \sqrt{1 + \alpha^2 + 2\beta - \beta} \) where \( \beta = k^T \delta x / \|x_0 - a\| \).
- Verify that \( |\beta| \leq \alpha \).
- Consider the function \( g(\beta) = -1 + \sqrt{1 + \alpha^2 + 2\beta - \beta} \) with \( |\beta| \leq \alpha \). By maximizing and minimizing \( g \) over the interval \(-\alpha \leq \beta \leq \alpha \) show that
\[
0 \leq \eta \leq \frac{\alpha^2}{2}.
\]

Solution.

a) For the linearized model we have
\[
\delta y = \begin{pmatrix} \partial y \\ \partial x \end{pmatrix} \delta x
\]
so all we have to do is to compute the matrix \( \partial y / \partial x \). Since \( y = \|x - a\| \) we have \( y^2 = (x - a)^T (x - a) \) and differentiating both sides with respect to \( x \) gives
\[
2 \frac{\partial y}{\partial x} y = 2(x - a)^T
\]
and therefore
\[
\frac{\partial y}{\partial x} = \frac{(x - a)^T}{y} = \frac{(x - a)^T}{\|x - a\|},
\]
so \( \delta y = k^T \delta x \) with \( k = (x - a) / \|x - a\| \). Clearly, \( k \) points from \( a \) to \( x \) and is of length one since
\[
k^T k = \frac{(x - a)^T (x - a)}{\|x - a\|^2} = 1.
\]
b) First we show that \( \eta = -1 + \sqrt{1 + \alpha^2 + 2\beta} - \beta \) where \( \beta = \frac{k^T \delta x}{\|x_0 - a\|} \). Note that

\[
e = \|x_0 + \delta x - a\| - \|x_0 - a\| - k^T \delta x
= \|x_0 - a\| \left(\frac{x_0 - a}{\|x_0 - a\|} + \frac{\delta x}{\|x_0 - a\|}\right) - 1 - \frac{k^T \delta x}{\|x_0 - a\|},
\]

and after dividing both sides by \(\|x_0 - a\|\) and using \( k = (x_0 - a)/\|x_0 - a\| \) and \( \eta = e/\|x_0 - a\| \) we get

\[
\eta = \|k + \frac{\delta x}{\|x_0 - a\|}\| - 1 - \beta.
\]

But

\[
\|k + \frac{\delta x}{\|x_0 - a\|}\| = \sqrt{(k + \frac{\delta x}{\|x_0 - a\|})^T (k + \frac{\delta x}{\|x_0 - a\|})}
= \sqrt{\|k\|^2 + \frac{k^T \delta x}{\|x_0 - a\|} + \frac{\|\delta x\|^2}{\|x_0 - a\|^2}}.
\]

Since \(\|k\| = 1\) and by substituting the values for \(\alpha\) and \(\beta\) we have

\[
k + \frac{\delta x}{\|x_0 - a\|} = \sqrt{1 + 2\beta + \alpha^2}.
\]

Therefore (1) can be written as

\[
\eta = \sqrt{1 + 2\beta + \alpha^2} - 1 - \beta.
\]

It is easy to see that \(|\beta| \leq \alpha\). Simply we can use the Cauchy-Schwarz inequality for the vectors \( k \) and \( \delta x/\|x_0 - a\| \), i.e.,

\[
\left| \frac{k^T \delta x}{\|x_0 - a\|} \right| \leq \|k\| \frac{\|\delta x\|}{\|x_0 - a\|}
\]

and since \(\|k\| = 1\) we immediately get \(|\beta| \leq \alpha\).
At this point, all we need to do to derive a bound on how large the error can be is to maximize and minimize the function \( g(\beta) = \sqrt{1 + 2\beta + \alpha^2} - 1 - \beta \) over the interval \(|\beta| \leq \alpha\) or \(-\alpha \leq \beta \leq \alpha\). The maximum or minimum of a smooth function \( g(\beta) \) over a given interval \((-\alpha \leq \beta \leq \alpha\) can only occur at the endpoints of the interval \((\beta = \pm \alpha)\) or at the extrema (points \( \beta \) with \( g'(\beta) = 0 \)). For \( g(\beta) \) we have:

- **Value at endpoint \( \beta = \alpha \).**

  \[
  g(\alpha) = \sqrt{1 + 2\alpha + \alpha^2} - 1 - \alpha \\
  = \sqrt{(1 + \alpha)^2} - 1 - \alpha \\
  = 1 + \alpha - 1 - \alpha \\
  = 0.
  \]

- **Value at endpoint \( \beta = -\alpha \).**

  \[
  g(-\alpha) = \sqrt{1 - 2\alpha + \alpha^2} - 1 + \alpha \\
  = \sqrt{(1 - \alpha)^2} - 1 + \alpha \\
  = |1 - \alpha| - (1 - \alpha) \\
  = \begin{cases} 
  0; & 0 \leq \alpha \leq 1 \\
  2(\alpha - 1); & \alpha > 1.
  \end{cases}
  \]

Therefore \( g(-\alpha) \geq 0 \) for all \( \alpha \) because \( 2(\alpha - 1) > 0 \) for \( \alpha > 1 \).

- **Extremum value.**

  \[
  g'(\beta) = \frac{1}{\sqrt{1 + 2\beta + \alpha^2}} - 1.
  \]

Setting \( g'(\beta) = 0 \) we get \( \sqrt{1 + 2\beta + \alpha^2} = 1 \) or \( 1 + 2\beta + \alpha^2 = 1 \) and therefore \( \beta_{\text{ex.}} = -\alpha^2/2 \). The function value at the extremum \( \beta_{\text{ex.}} = -\alpha^2/2 \) is

\[
\begin{align*}
g(\beta_{\text{ex.}}) &= \sqrt{1 - \alpha^2 + \alpha^2} - 1 + \frac{\alpha^2}{2} \\
&= \frac{\alpha^2}{2}.
\end{align*}
\]

Clearly, \( g(\beta) \geq 0 \) for all \( \beta \) satisfying \(|\beta| \leq \alpha \) because the value of \( g(\beta) \) at the endpoints \( \beta = \pm \alpha \) and at the extremum \( \beta = \alpha^2/2 \) are all non-negative. Thus we have achieved the lower bound on the relative error \( \eta \), i.e., we have shown that \( \eta \geq 0 \). For the upper bound we need to be a bit more careful. The upper bound we get is either \( g(\alpha) \), \( g(-\alpha) \) or \( g(\beta_{\text{ex.}}) \). First note that \( g(\alpha) = 0 \) is always less than or equal to \( g(\beta_{\text{ex.}}) = \alpha^2/2 \geq 0 \) so the choice of \( g(\alpha) \) is immediately ruled out as the maximum of \( g \). Now consider \( g(-\alpha) \) and \( g(\beta_{\text{ex.}}) \). For \( 0 \leq \alpha \leq 1 \) we obviously have \( g(\beta_{\text{ex.}}) \geq g(-\alpha) = 0 \). For \( \alpha > 1 \) we also have \( g(\beta_{\text{ex.}}) \geq g(-\alpha) = 2(\alpha - 1) \) because \( \alpha^2/2 \geq 2(\alpha - 1) \) is equivalent to \( \alpha^2 - 4\alpha + 4 \geq 0 \) which is true since \( \alpha^2 - 4\alpha + 4 = (\alpha - 2)^2 \) is a complete square. Thus, we achieve an upper bound on \( g(\beta) \) for all \( \beta \) satisfying \(|\beta| \leq \alpha \) as \( g(\beta) \leq \alpha^2/2 \). Therefore we have shown that
\( \eta \leq \alpha^2/2 \) and we are done.

(Note: when \( \beta_{\text{ex.}} \) falls outside the interval \( \beta \leq |\alpha| \), it is possible to achieve a tighter upper bound for \( g \). In this case, the maximum of \( g \) over \( \beta \leq |\alpha| \) is obtained at the endpoint \( \beta = -\alpha/2 \). The extremum \( \beta_{\text{ex.}} = -\alpha^2/2 \) falls outside \( \beta \leq |\alpha| \) when \( \alpha^2/2 > \alpha \) or \( \alpha > 2 \). Therefore, a tighter upper bound on \( \eta \) for \( \alpha > 2 \) becomes \( \eta \leq g(-\alpha) = 2(\alpha - 1) \).

6. Temperatures in a multi-core processor. We are concerned with the temperature of a processor at two critical locations. These temperatures, denoted \( T = (T_1, T_2) \) (in degrees C), are affine functions of the power dissipated by three processor cores, denoted \( P = (P_1, P_2, P_3) \) (in W). We make 4 measurements. In the first, all cores are idling, and dissipate 10 W. In the next three measurements, one of the processors is set to full power, 100 W, and the other two are idling. In each experiment we measure and note the temperatures at the two critical locations.

<table>
<thead>
<tr>
<th>( P_1 )</th>
<th>( P_2 )</th>
<th>( P_3 )</th>
<th>( T_1 )</th>
<th>( T_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10W</td>
<td>10W</td>
<td>10W</td>
<td>27(^\circ)</td>
<td>29(^\circ)</td>
</tr>
<tr>
<td>100W</td>
<td>10W</td>
<td>10W</td>
<td>45(^\circ)</td>
<td>37(^\circ)</td>
</tr>
<tr>
<td>10W</td>
<td>100W</td>
<td>10W</td>
<td>41(^\circ)</td>
<td>49(^\circ)</td>
</tr>
<tr>
<td>10W</td>
<td>10W</td>
<td>100W</td>
<td>35(^\circ)</td>
<td>55(^\circ)</td>
</tr>
</tbody>
</table>

Suppose we operate all cores at the same power, \( p \). How large can we make \( p \), without \( T_1 \) or \( T_2 \) exceeding 70\(^\circ\)?

You must fully explain your reasoning and method, in addition to providing the numerical solution.

**Solution.** The temperature vector \( T \) is an affine function of the power vector \( P \), i.e., we have \( T = AP + b \) for some matrix \( A \in \mathbb{R}^{2\times 3} \) and some vector \( b \in \mathbb{R}^2 \). Once we find \( A \) and \( b \), we can predict the temperature \( T \) for any value of \( P \).

The first approach is to (somewhat laboriously) write equations describing the measurements in terms of the elements of \( A \). Let \( a_{ij} \) denote the \((i,j)\) entry of \( A \). We can write out the relations \( T = AP + b \) for the 4 experiments listed above as the set of 8 equations

\[
\begin{align*}
10a_{11} + 10a_{12} + 10a_{13} + b_1 &= 27, \\
10a_{21} + 10a_{22} + 10a_{23} + b_2 &= 29, \\
100a_{11} + 10a_{12} + 10a_{13} + b_1 &= 45, \\
100a_{21} + 10a_{22} + 10a_{23} + b_2 &= 37, \\
10a_{11} + 100a_{12} + 10a_{13} + b_1 &= 41, \\
10a_{21} + 100a_{22} + 10a_{23} + b_2 &= 49, \\
10a_{11} + 10a_{12} + 100a_{13} + b_1 &= 35, \\
10a_{21} + 10a_{22} + 100a_{23} + b_2 &= 55.
\end{align*}
\]
Next, we define a vector of unknowns, \( x = (a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, b_1, b_2) \in \mathbb{R}^8 \). We rewrite the 8 equations above as \( Cx = d \), where

\[
C = \begin{bmatrix}
10 & 10 & 10 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 10 & 10 & 10 & 0 & 1 \\
100 & 10 & 10 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 100 & 10 & 10 & 0 & 1 \\
10 & 100 & 10 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 10 & 100 & 10 & 0 & 1 \\
10 & 100 & 100 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 10 & 10 & 100 & 0 & 1
\end{bmatrix}, \quad d = \begin{bmatrix}
27 \\
29 \\
45 \\
37 \\
41 \\
49 \\
35 \\
55
\end{bmatrix}.
\]

We solve for \( x \) as \( x = C^{-1}d \). (It turns out that \( C \) is invertible.) Putting the entries of \( x \) into the appropriate places in \( A \) and \( b \), we have

\[
A = \begin{bmatrix}
0.200 & 0.156 & 0.089 \\
0.089 & 0.222 & 0.289
\end{bmatrix}, \quad b = \begin{bmatrix}
22.6 \\
23.0
\end{bmatrix}.
\]

At this point we can predict \( T \) for any \( P \) (assuming we trust the affine model).

Substituting \( P = (p, p, p) \) into \( T = AP + b \), we get

\[
T_1 = 0.444p + 22.6, \quad T_2 = 0.600p + 23.0.
\]

Both of these temperatures are increasing in \( p \) (it would be quite surprising if this were not the case). The value of \( p \) for which \( T_1 = 70 \) is \( p = (70 - 22.6)/0.444 = 106.8 \). The value of \( p \) for which \( T_2 = 70 \) is \( p = (70 - 23)/0.6 = 78.3 \). Thus, the maximum value of \( p \) for which both temperatures do not exceed 70° is \( p = 78.3 \).
**Alternative solution.** Another way of solving this problem is to directly exploit the fact that $T$ is an affine function of $P$. This means that if we form any linear combination of the power vectors used in the experiment, with the coefficients summing to one, the temperature vector will also be the same linear combination of the temperatures.

By averaging the last three experiments we find if the powers are $P = (40, 40, 40)$, then the temperature vector is $T = (40.33, 47.00)$. (Note that this is really a prediction, based on the observed experimental data and the affineness assumption; it’s not a new experiment!)

Now we form a new power vector of the form

$$P = (1 - \theta)(10, 10, 10) + \theta(40, 40, 40) = (10 + 30\theta, 10 + 30\theta, 10 + 30\theta),$$

where $\theta \in \mathbb{R}$. The coefficients $1 - \theta$ and $\theta$ sum to one, so since $T$ is affine, we find that the corresponding temperature vector is

$$T = (1 - \theta)(27, 29) + \theta(40.33, 47.00) = (27 + 13.33\theta, 29 + 18\theta),$$

just as above. The first coefficient hits 70 at $\theta = 3.22$; the second coefficient hits 70 at $\theta = 2.23$. Thus, $\theta$ can be as large as $\theta = 2.27$. This corresponds to the powers $P = (78.3, 78.3, 78.3)$.

7. **Relative deviation between vectors.** Suppose $a$ and $b$ are nonzero vectors of the same size. The relative deviation of $b$ from $a$ is defined as the distance between $a$ and $b$, divided by the norm of $a$,

$$\eta_{ab} = \frac{\|a - b\|}{\|a\|}.$$ 

This is often expressed as a percentage. The relative deviation is not a symmetric function of $a$ and $b$; in general, $\eta_{ab} \neq \eta_{ba}$.

Suppose $\eta_{ab} = 0.1$ (i.e., 10%). How big and how small can be $\eta_{ba}$ be? How big and how small can $\angle(a,b)$ be? Explain your reasoning. For bounding $\angle(a,b)$, you can just draw some pictures; you don’t have to give a formal argument.

**Solution.** We’ll work out a more general case. We have

$$\|a - b\| = \eta_{ab}\|a\|.$$ 

We need to get upper and lower bounds on $\|b\|$. We can use the triangle inequality to get an upper bound:

$$\|b\| = \|a + (-a + b)\|$$ 
$$\leq \|a\| + \|a - b\|$$ 
$$= (1 + \eta_{ab})\|a\|.$$ 

This inequality is tight, if $a$ and $a - b$ are anti-aligned, which is the same as $a$ and $b$ being aligned. Now we can say that

$$\eta_{ba} = \frac{\|a - b\|}{\|b\|} \geq \frac{\eta_{ab}\|a\|}{(1 + \eta_{ab})\|a\|} = \frac{\eta_{ab}}{(1 + \eta_{ab})}.$$ 

18
This is a general bound, and it is tight when \( a \) and \( b \) are aligned. For \( \eta_{ab} = 0.1 \), we find that \( \eta_{ba} \geq 0.1111 \).

Now let’s get a lower bound on \( \|b\| \), again using the triangle inequality:

\[
\|a\| = \|b + (a - b)\|
\leq \|b\| + \|a - b\|
= \|b\| + \eta_{ab}\|a\|.
\]

This inequality is tight if \( a \) and \( a - b \) are aligned, which is the same as \( a \) and \( b \) being anti-aligned. Subtracting, we get

\[
(1 - \eta_{ab})\|a\| \leq \|b\|.
\]

Assuming that \( \eta_{ab} < 1 \) (which is the case for \( \eta_{ab} = 0.1 \)), we then have

\[
\eta_{ba} = \frac{\|a - b\|}{\|b\|} \leq \frac{\eta_{ab}\|a\|}{(1 - \eta_{ab})\|a\|} = \frac{\eta_{ab}}{1 - \eta_{ab}}.
\]

This is a general bound, tight when \( a \) and \( b \) are anti-aligned. For \( \eta_{ab} = 0.1 \), we find that \( \eta_{ba} \geq 0.1/0.9 = 0.1111 \).

In summary, when \( \eta_{ab} = 0.1 \), \( \eta_{ba} \) can range between 0.0909 and 0.1111. The lower limit occurs when \( a \) and \( b \) are aligned; the upper limit occurs when \( a \) and \( b \) are anti-aligned.

Now let’s look at the angle. We first give a geometric argument. Let’s look at the plane spanned by \( a \) and \( b \). Then vector \( b \) must be in a ball of radius \( \eta_{ab}\|a\| \), centered in \( a \), as shown below. Assuming that \( \eta_{ab} < 1 \) (which is the case here), the ball does not include the origin.

Now, we look on how small and how large the angle between \( a \) and \( b \) can be, as \( b \) varies over the ball. When \( a \) and \( b \) are aligned, \( \angle(a, b) = 0 \). Now let’s see how large the angle can be. The largest angle is obtained when \( b \) and \( a - b \) are orthogonal; in this case \( (0, a, b) \) are the vertices of a right triangle. In this case we have \( \angle(a, b) = \arcsin \eta_{ab} \). For \( \eta_{ab} \), we find that \( \angle(a, b) = 0.1002 \). Therefore \( \angle(a, b) \) can take values in the interval \([0, 0.1002]\).
8. 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 \( k \)th entry). If all sensors are operating correctly, we have \( y = \tilde{y} \). If the \( k \)th sensor fails, we have \( \tilde{y}_i = y_i \) for all \( i \neq k \).

The file `one_bad_sensor.m`, available on the course web site, defines \( A \) and \( \tilde{y} \) (as \( A \) and \( \tilde{y} \)). 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 \( i \)th entry removed. Likewise, let \( A^{(i)} \) be the measurement matrix with the \( i \)th row of \( A \) removed. This corresponds to the system without the \( i \)th sensor.

If the \( i \)th 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

```matlab
test = 1:
for k=1:m
    withoutk=[1:k-1 k+1:m];
    Atent = A(withoutk,:);
    ytent = ytilde(withoutk);
    if rank([ Atent ytent ]) == rank(Atent) == k
        break
    end
end
```

The 11th sensor is faulty.

9. Projection matrices. A matrix \( P \in \mathbb{R}^{n \times n} \) is called a projection matrix if \( P = P^T \) and \( P^2 = P \).

a) Show that if \( P \) is a projection matrix then so is \( I - P \).

b) Suppose that the columns of \( U \in \mathbb{R}^{n \times k} \) are orthonormal. Show that \( UU^T \) is a projection matrix. (Later we will show that the converse is true: every projection matrix can be expressed as \( UU^T \) for some \( U \) with orthonormal columns.)

c) Suppose \( A \in \mathbb{R}^{n \times k} \) is full rank, with \( k \leq n \). Show that \( A(A^T A)^{-1}A^T \) is a projection matrix.
d) If \( S \subseteq \mathbb{R}^n \) and \( x \in \mathbb{R}^n \), the point \( y \) in \( S \) closest to \( x \) is called the \textit{projection of} \( x \) \textit{on} \( S \). Show that if \( P \) is a projection matrix, then \( y = Px \) is the projection of \( x \) on \( \text{range}(P) \). (Which is why such matrices are called projection matrices . . . )

\textbf{Solution.}

a) To show that \( I - P \) is a projection matrix we need to check two properties:

i. \( I - P = (I - P)^T \)

ii. \( (I - P)^2 = I - P \).

The first one is easy: \( (I - P)^T = I - P^T = I - P \) because \( P = P^T \) (\( P \) is a projection matrix.) The show the second property we have

\[
(I - P)^2 = I - 2P + P^2 \\
= I - 2P + P \quad \text{(since} \; P = P^2\text{)} \\
= I - P
\]

and we are done.

b) Since the columns of \( U \) are orthonormal we have \( U^T U = I \). Using this fact it is easy to prove that \( U U^T \) is a projection matrix, \textit{i.e.,} \( (U U^T)^T = U U^T \) and \( (U U^T)^2 = U U^T \). Clearly, \( (U U^T)^T = (U^T)^T U^T = U U^T \)

\[
(U U^T)^2 = (U U^T)(U U^T) \\
= U(U^T U)U^T \\
= U U^T \quad \text{(since} \; U^T U = I\text{).}
\]

c) First note that \( (A(A^T A)^{-1} A^T)^T = A(A^T A)^{-1} A^T \) because

\[
(A(A^T A)^{-1} A^T)^T = (A^T)^T \left( (A^T A)^{-1} \right)^T A^T \\
= A \left( (A^T A)^{-1} \right)^{-1} A^T \\
= A(A^T A)^{-1} A^T.
\]

Also \( (A(A^T A)^{-1} A^T)^2 = A(A^T A)^{-1} A^T \) because

\[
\left( (A(A^T A)^{-1} A^T) \right)^2 = \left( A(A^T A)^{-1} A^T \right) \left( A(A^T A)^{-1} A^T \right) \\
= A \left( (A^T A)^{-1} A^T \right) (A^T A)^{-1} A^T \\
= A(A^T A)^{-1} A^T \quad \text{(since} \; (A^T A)^{-1} A^T A = I\text{).}
\]

d) To show that \( Px \) is the projection of \( x \) on \( \text{range}(P) \) we verify that the “error” \( x - Px \) is orthogonal to \textit{any} vector in \( \text{range}(P) \). Since \( \text{range}(P) \) is nothing but the span of the
columns of $P$ we only need to show that $x - Px$ is orthogonal to the columns of $P$, or in other words, $P^T(x - Px) = 0$. But

$$P^T(x - Px) = P(x - Px) \quad \text{ (since } P = P^T)$$

$$= Px - P^2x$$

$$= 0 \quad \text{ (since } P^2 = P)$$

and we are done.