18.1 consider the example

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \\ 1 & 1.0001 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 0.0001 \\ 4.0001 \end{bmatrix}$$

(a) What are the matrices A^+ and P for this example? Give exact answers.

Solution

$A^+ = (A^*A)^{-1}A^*$ and $P = AA^+$.

First we compute A^+.

$$A^*A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & 1.0001 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \\ 1 & 1.0001 \end{bmatrix} = \begin{bmatrix} 3 & 3.0002 \\ 3.0002 & 3.0004 \end{bmatrix}$$

$$(A^*A^{-1}) = \frac{1}{3(3.0004) - (3.0002)(3.0002)} \begin{bmatrix} 3.0004 & -3.0002 \\ -3.0002 & 3 \end{bmatrix}$$

$$= -4.0 \times 10^{-8} \begin{bmatrix} 3.0004 & -3.0002 \\ -3.0002 & 3 \end{bmatrix}$$

$$(A^*A)^{-1}A = -4.0 \times 10^{-8} \begin{bmatrix} 3.0004 & -3.0002 \\ -3.0002 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1.0001 & 1.0001 \end{bmatrix}$$

$$A^+ = -4.0 \times 10^{-8} \begin{bmatrix} 0.0002 & 0.00010002 & 0.00010002 \\ 0.0002 & -6.0005 & -6.0005 \end{bmatrix}$$

$$P = AA^+$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \\ 1 & 1.0001 \end{bmatrix} \begin{bmatrix} 0.0002 & 0.00010002 & 0.00010002 \\ 0.0002 & -6.0005 & -6.0005 \end{bmatrix}$$

$$= -4.0 \times 10^{-8} \begin{bmatrix} 0.0004 & -6.00039998 & -6.00039998 \\ -2 \times 10^{-8} & -6.0010003 & -6.0010003 \end{bmatrix}$$

(b) Find exact solutions x and $y = Ax$ to the least squares problem $Ax \approx b$.

Solution:

$$x = A^+ b$$

$$= -4.0 \times 10^{-8} \begin{bmatrix} 0.0002 & 0.00010002 & 0.00010002 \\ 0.0002 & -6.0005 & -6.0005 \end{bmatrix} \begin{bmatrix} 2 \\ 0.0001 \\ 4.0001 \end{bmatrix}$$

$$= -4.0 \times 10^{-8} \begin{bmatrix} 0.000800100004 \\ -24.0028001 \end{bmatrix}$$

and

$$b \approx y = Ax = AA^+ b$$

$$= -4.0 \times 10^{-8} \begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \\ 1 & 1.0001 \end{bmatrix} \begin{bmatrix} 0.000800100004 \\ -24.0028001 \end{bmatrix}$$

$\downarrow$

$$= -4.0 \times 10^{-8} \begin{bmatrix} -24.0019999999996 \\ -24.0044002800006 \\ -24.0044002800006 \end{bmatrix}$$

(c) See Jupyter Notebook

(d) See Jupyter Notebook

(e) See Jupyter Notebook.

18.2 One might think that the more variables one included in such a model, the more information one would obtain, but this is not always true. Explain this phenomenon from the point of view of conditioning, make specific reference to the results of Theorem 18.1.

Solution Three of the four quantities in the table for Theorem 18.1 are proportional to $\kappa(A)$. This implies that poor conditioning on A will result in large perturbations in solutions x & y. Thus, including additional variables that cause A to be poorly conditioned may return undesirable results, particularly if data is noisy. For example, a minor mis-reporting in a parents IQ or years of education may result in a major change in predicted annual income, making the model untrustworthy.

In [ ]:
```
import numpy as np
```

Problem 18.1

Below are the computations for Parts (c), (d), and (e). These computation use the following information:

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \\ 1 & 1.0001 \end{bmatrix}$$

$$b = \begin{bmatrix} 2 \\ 0.0001 \\ 4.0001 \end{bmatrix}$$

and the computed

$$x = 4.0 \times 10^{-8} \begin{bmatrix} 0.000800100004 \\ -24.0028001 \end{bmatrix}$$

$$y = 4.0 \times 10^{-8} \begin{bmatrix} 24.001999999996 \\ 24.004400280006 \\ 24.004400280006 \end{bmatrix}$$

In [ ]:
```
A = np.array([[1.0, 1.0], [1.0, 1.0001], [1.0, 1.0001]], dtype=np.float64)
b = np.array([2, 0.0001, 4.0001], dtype=np.float64)
x = np.array([0.000800100004,  -24.0028001], dtype=np.float64)
y = 4.0e-8 * np.array([24.001999999996, 24.004400280006, 24.004400280006], dtype=np
```

(c) What $\kappa(A)$, θ, and η?

In [ ]:
```
kappa_A = np.linalg.cond(A, p=2)
theta = np.arccos(np.linalg.norm(y, ord=2) / np.linalg.norm(b, ord=2))
eta = (np.linalg.norm(A, ord=2) * np.linalg.norm(x, ord=2)) / np.linalg.norm(y, ord
print('kappa(A): ' + str(kappa_A))
print('theta: ' +  str(theta))
print('eta: ' + str(eta))
```

```
kappa(A): 42429.235416083044
theta: 1.5707959549401582
eta: 35355339.07897037
```

(d) What are the 4 condition numbers of Theorem 18.1?

$$\kappa_{y \mapsto b} = \frac{1}{\cos\theta}$$

$$\kappa_{y \mapsto A} = \frac{\kappa(A)}{\cos\theta}$$

$$\kappa_{x \mapsto b} = \frac{\kappa(A)}{\eta\cos\theta}$$

$$\kappa_{x \mapsto A} = \kappa(A) + \frac{\kappa(A)^2\tan\theta}{\eta}$$

In [ ]:
```
# Computations for each of the 4 condition numbers listed above.
kappa_y_b = 1/np.cos(theta)
kappa_y_A = np.linalg.cond(A, p=2)/np.cos(theta)
kappa_x_b = np.linalg.cond(A, p=2)/(eta * np.cos(theta))
kappa_x_A = ((np.linalg.cond(A, p=2) ** 2) * np.tan(theta))/(eta)

print('Sensitivity of y to perterbations in b is kappa_y_b: ' + str(kappa_y_b))
print('Sensitivity of y to perterbations in A is kappa_y_A: ' + str(kappa_y_A))
print('Sensitivity of x to perterbations in b is kappa_x_b: ' + str(kappa_x_b))
print('Sensitivity of x to perterbations in A is kappa_y_A: ' + str(kappa_x_A))
```

```
Sensitivity of y to perterbations in b is kappa_y_b: 2689222.1524829348
Sensitivity of y to perterbations in A is kappa_y_A: 114101639793.84401
Sensitivity of x to perterbations in b is kappa_x_b: 3227.281727916238
Sensitivity of x to perterbations in A is kappa_y_A: 136931096.18777186
```

(e) Give examples of perturbations δb and δA that approximately attain these four condition numbers.

In [ ]:
```
# Perturbations of b and A
delta_b = np.array([2 + 1e-14, 0.0001, 4.0001], dtype=np.float64)
delta_A = A = np.array([[1.0 + 1e-14, 1.0], [1.0, 1.0001], [1.0, 1.0001]], dtype=np
```

In [ ]:
```
# Computations for each of the 4 condition numbers listed above with delta_b and de
kappa_delta_A = np.linalg.cond(delta_A, p=2)
theta = np.arccos(np.linalg.norm(y, ord=2) / np.linalg.norm(b, ord=2))
eta = (np.linalg.norm(delta_A, ord=2) * np.linalg.norm(x, ord=2)) / np.linalg.norm(
print('kappa(delta_A): ' + str(kappa_delta_A))
print('theta: ' +  str(theta))
print('eta: ' + str(eta))


kappa_y_b = 1/np.cos(theta)
kappa_y_delta_A = np.linalg.cond(delta_A, p=2)/np.cos(theta)
kappa_x_b = np.linalg.cond(delta_A, p=2)/(eta * np.cos(theta))
kappa_x_delta_A = ((np.linalg.cond(delta_A, p=2) ** 2) * np.tan(theta))/(eta)

print('Sensitivity of y to perterbations in b is kappa_y_b: ' + str(kappa_y_b))
print('Sensitivity of y to perterbations in A is kappa_y_delta_A: ' + str(kappa_y_d
print('Sensitivity of x to perterbations in b is kappa_x_b: ' + str(kappa_x_b))
print('Sensitivity of x to perterbations in A is kappa_ydelta__A: ' + str(kappa_x_d
```

```
kappa(delta_A): 42429.2354118404
theta: 1.5707959549401582
eta: 35355339.07897044
Sensitivity of y to perterbations in b is kappa_y_b: 2689222.1524829348
Sensitivity of y to perterbations in A is kappa_y_delta_A: 114101639782.43459
Sensitivity of x to perterbations in b is kappa_x_b: 3227.281727593525
Sensitivity of x to perterbations in A is kappa_ydelta__A: 136931096.16038716
```