Question 1

(a)

Augmented Lagrangian function (the scaled form) is as follows.

Q1

$$L(x, z, w, \rho) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda_1\|z\|_1 + \frac{\lambda_2}{2}\|z\|_2^2 + \frac{\rho}{2}\|x - z + w\|_2^2 + \frac{\rho}{2}\|w\|_2^2$$

Update step for **x** is derived and shown as below.

$$x_k = \underset{x}{\arg\min}\ \frac{1}{2}\|Ax-b\|_2^2 + \frac{\rho}{2}\|x - z_{k-1} + w_{k-1}\|_2^2$$

After taking derivative, set it equal to 0, we have

$$\underline{A^T(Ax}-b) + \underline{\rho(x} - z_{k-1} + w_{k-1}) = 0$$

$$(A^TA + \rho I)x - A^Tb - \rho z_{k-1} + \rho w_{k-1} = 0$$

$$x_k = \frac{A^Tb + \rho(z_{k-1} - w_{k-1})}{A^TA + \rho I}$$

Update step for **z** is derived and shown as below.

$$Z_k = \underset{Z}{\operatorname{argmin}}\ \lambda_1 \|Z\|_1 + \frac{\lambda_2}{2}\|Z\|_2^2 + \frac{\rho}{2}\|X_k - Z + W_{k-1}\|_2^2$$

$$\lambda_1 \operatorname{sign}(Z) + \lambda_2 Z - \rho(X_k - Z + W_{k-1}) = 0$$

$$\lambda_2 Z + \rho Z + \lambda_1 \operatorname{sign}(Z) - \rho X_k - \rho W_{k-1} = 0$$

$$(\lambda_2 + \rho) Z = \rho(X_k + W_{k-1}) - \lambda_1 \operatorname{sign}(Z)$$

$$Z_k = \frac{\rho(X_k + W_{k-1}) - \lambda_1 \operatorname{sign}(Z_{k-1})}{\lambda_2 + \rho}$$

Update step for **w** is shown as below.

$$W_k = W_{k-1} + X_k - Z_k$$

(b)

Note:

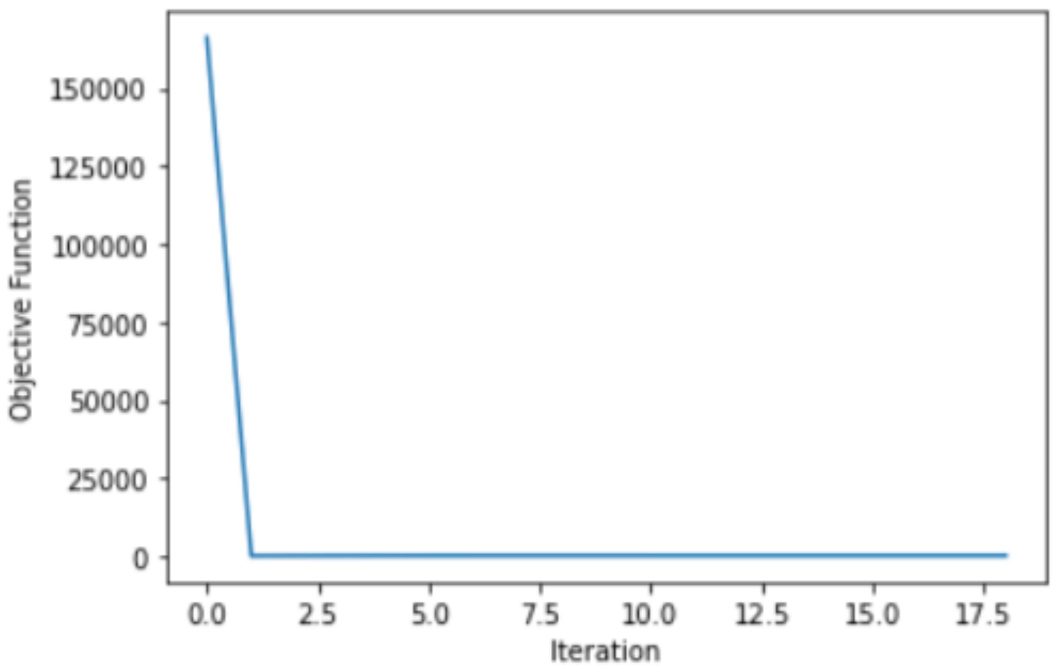
I **standardized both input variables A and response variables b.** This takes into account the effect of intercept.

After the training session, I **scaled response variables b back to its original range of [0.95,1]** in order to compare.

My coefficients x is

```
Coefficients x:

[[  0.94775251]
 [  0.55266321]
 [  2.29673911]
 [ -0.34106   ]
 [  1.84779417]
 [ -4.09344076]
 [  1.78671412]
 [-13.39916652]
 [  5.64934424]
 [  7.27344785]
 [ -1.47063122]
 [ -0.71332257]
 [ -0.32671017]]
```

Objective function versus iterations is plotted as follows.



Sum of Absolute Errors on the test set is 1.9967727070434012

```
Sum of Absolute Error on test set is 1.9967727070434012
```

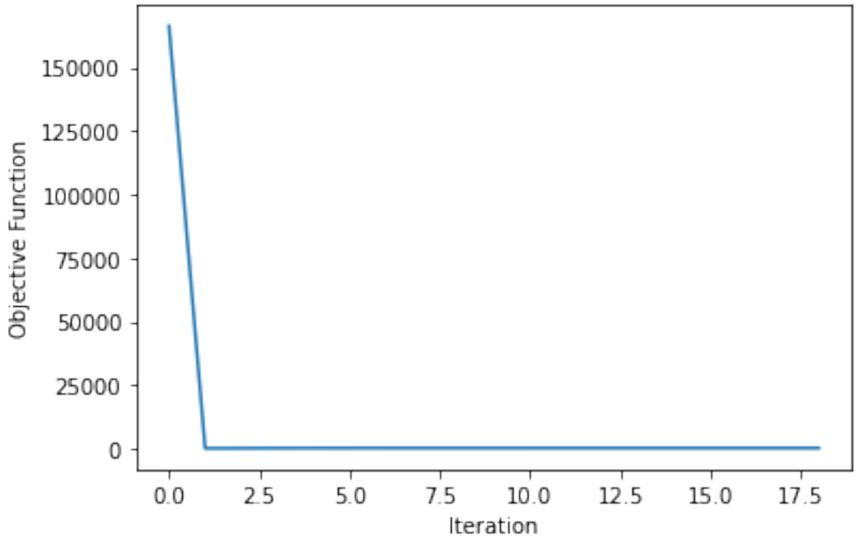
In [24]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import Normalizer
from sklearn.metrics import r2_score
from sklearn.preprocessing import StandardScaler
```

In [127]:
```
data = pd.read_csv('Question1-1.csv',header=None).to_numpy()

#preprocessing: standardizing both A and b
A_train = data[:2000,:13]
b_train = data[:2000,13][:,np.newaxis]
A_test = data[2000:,:13]
b_test = data[2000:,13][:,np.newaxis]

scaler1 = StandardScaler()
A_train = scaler1.fit_transform(A_train)
A_test = scaler1.transform(A_test)

scaler2 = StandardScaler()
b_train = scaler2.fit_transform(b_train)
b_test = scaler2.transform(b_test)
```

```
#r2 score on standardized A and b
beta = np.linalg.lstsq(A_train,b_train,rcond=None)[0]
b_pred = A_train@beta
print(f'R2 score on training dataset with OLS is {r2_score(b_train, b_
pred)}')

#ADMM
rho = 1
lam1 = 0.1
lam2 = 0.9
x = np.ones((13,1))
z = np.ones((13,1))
w = np.ones((13,1))
xz = [np.linalg.norm(x-z)]
obj = [0.5*np.linalg.norm(A_train@x-b_train)**2+lam1*np.linalg.norm(z,
ord=1)+0.5*lam2*np.linalg.norm(z)**2]
diff = np.inf
tol = 1e-3
k = 0
while diff>tol :
    k += 1
    x = np.linalg.inv(A_train.T@A_train+rho*np.eye(13))@(A_train.T@b_t
rain+rho*(z-w))
    z = (rho*(x+w)-lam1*np.sign(z))/(lam2+rho)
    w = w+x-z
    obj.append(0.5*np.linalg.norm(A_train@x-b_train)**2+lam1*np.linalg
.norm(z,ord=1)+0.5*lam2*np.linalg.norm(z)**2)
    diff = abs(obj[-2]-obj[-1])
    xz.append(np.linalg.norm(x-z))

plt.figure()
plt.plot(range(k+1),obj)
plt.xlabel('Iteration')
plt.ylabel('Objective Function')
plt.show()
print(f'Coefficients x: \n\n{x} ')

#plt.figure()
#plt.plot(range(k+1),xz)
#plt.xlabel('Iteration')
#plt.ylabel('||x - z||')
#plt.show()

#SAE on test set after scaling response variables b back to its origin
al scale
b_pred = A_test@x
b_pred = scaler2.inverse_transform(b_pred)
b_test = scaler2.inverse_transform(b_test)
print(f'\nSum of Absolute Error on test set is {sum(abs(b_pred-b_test)
```

```
)[0]}')
```

```
R2 score on training dataset with OLS is 0.8602850250401106
```




```
Coefficients x:

[[  0.94775251]
 [  0.55266321]
 [  2.29673911]
 [ -0.34106   ]
 [  1.84779417]
 [ -4.09344076]
 [  1.78671412]
 [-13.39916652]
 [  5.64934424]
 [  7.27344785]
 [ -1.47063122]
 [ -0.71332257]
 [ -0.32671017]]

Sum of Absolute Error on test set is 1.9967727070434012
```

In [ ]: