19CS10071

Q7) Consider E = L.S error $= \| \hat{y} - y \|_2^2 = \| Ax - b \|_2^2$

$$\therefore \frac{\partial E}{\partial x_i} = \left(2A^T(Ax-b)\right)_i$$

$$2A^T(A\hat{x} - b) = 0 \Rightarrow A^T(A\hat{x} - b) = 0$$

$$x^{k+1} = x^k - \frac{1}{\|A\|^2} A^T(Ax^k - b)$$

$$= x^k - \frac{1}{\|A\|^2} A^T(A(x^k) - b) + \frac{1}{\|A\|} A^T(A(\hat{x}) - b)$$

$$x^{k+1} = x^k + \frac{1}{\|A\|^2} A^T(A(\hat{x} - x^k))$$

$$\Rightarrow \hat{x} - x^{k+1} = \hat{x} - x^k - \frac{1}{\|A\|^2} A^T(A(\hat{x} - x^k))$$

Let $r^k = \hat{x} - x^k$ denote difference between $\hat{x}$ & x^k at k^{th} iteration

$$\therefore r^{k+1} = r^k - \frac{1}{\|A\|^2} A^T A r^k$$

$$= \left(I - \frac{A^T A}{\|A\|^2}\right) r^k \qquad \therefore r^k = \left(I - \frac{AA}{\|A\|}\right)^k r^{(0)}$$

As $k \to \infty$ $\left(I - \frac{A^T A}{\|A\|}\right)^k \to 0$

So it converges

b) for determining x^k,

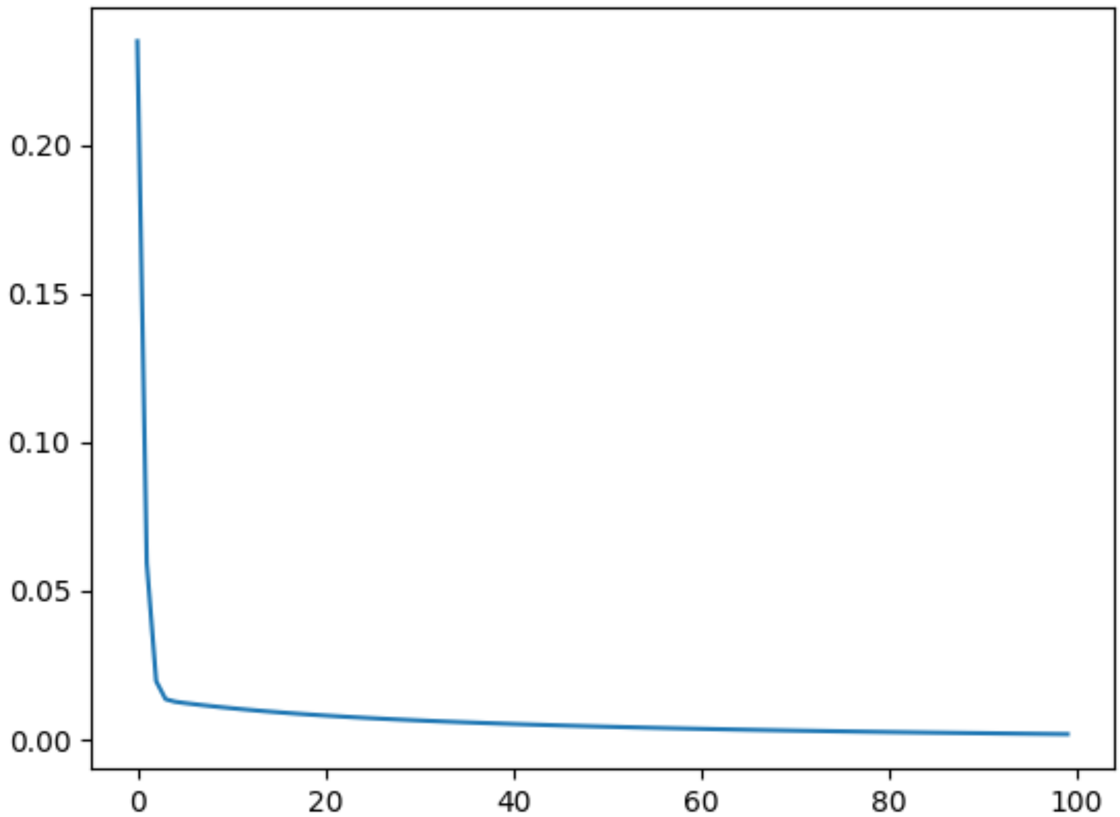
$$\underset{m\times n}{A}\,\underset{n\times 1}{x} \to O(mn)$$

$$\underset{n\times m}{A^T}\,(Ax)_{m\times 1} \to O(n.m.1) = O(nm)$$

So for k^{th} iteration we get $O(nmk)$ as time complexity

c)

↑ Diff bet Successive Values



itr →

So as we can see the algorithm is efficient and converges rapidly

Python Code

```
import numpy as np
import matplotlib.pyplot as plt
A = np.random.rand(30,10)
b = np.random.rand(30,1)
while np.linalg.matrix_rank(A)<10:
    A = np.random.rand(30,10)

x =  np.zeros((10,1))
X =[]
Y = []
for i in range(100):
    oldx = x
    x = x-np.dot(A.T,(np.dot(A,x)-b))/(np.linalg.norm(A)**2)
    diff = np.linalg.norm(oldx-x)
    print('Iteration ',i,'Norm of Difference between old and new x: ',diff)
    X.append(i)
    Y.append(diff)
plt.plot(X,Y)
plt.show()
```

d) Yes it is more efficient than normal way of computation using A^+x or using QR factorisation to compute A^+x