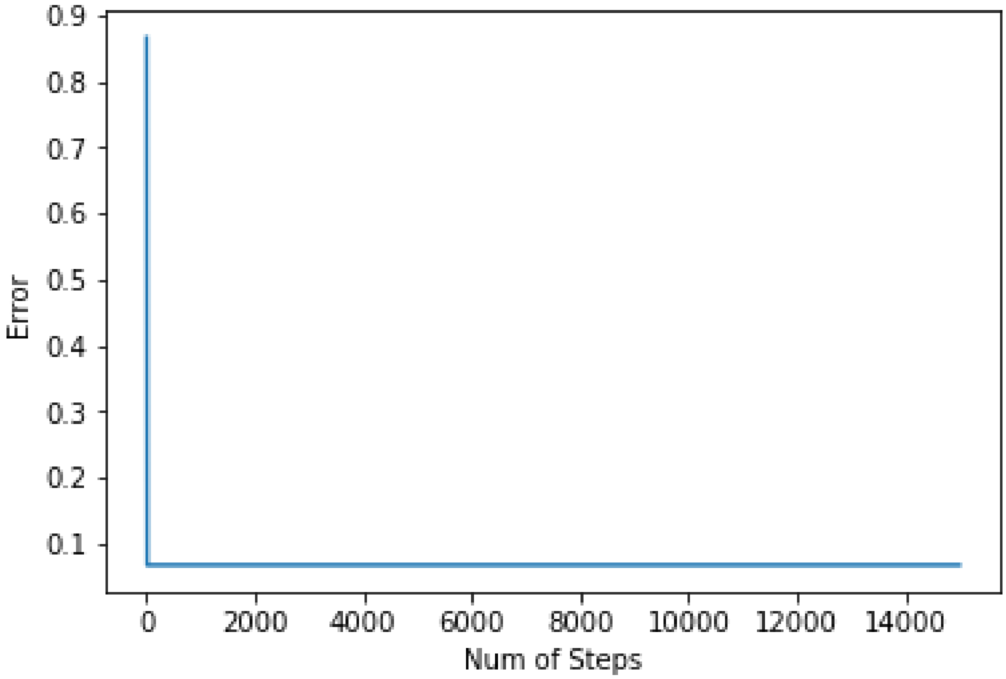
# HW2_problem2

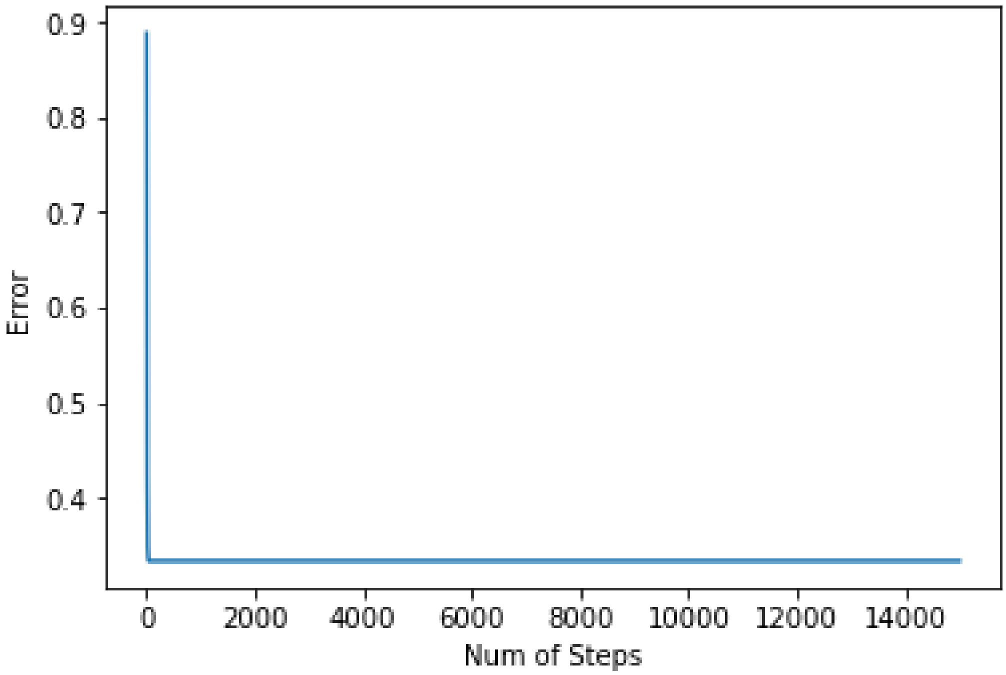
September 17, 2018

```
In [35]: import numpy as np
         import matplotlib.pyplot as plt

In [36]: b = [4.5, 6]
         x_0 = [0, 0]
         x_opt = b

         def error_cal(x, x_opt):
             error = np.linalg.norm(x - x_opt) / np.linalg.norm(x_opt)
             return error

In [37]: # Problem 2.2
         errors = []
         x = np.zeros((2, len(x_opt)))
         error = 100
         alpha = 1
         i = 1
         while (error > 0.01) & (i < 15000):
             x[i,:] = x[i-1,:] - alpha * (x[i-1,:] - b) / np.linalg.norm(x[i-1,:]-b)
             error = error_cal(x[i,:], x_opt)
             x_temp = x[i,:]
             x = np.vstack((x, x_temp))
             errors.append(error)
             i += 1

         if i==15000:
             print("Does not converge!")
         else:
             print(i-1)
         plt.plot(range(1,i), errors)
         plt.xlabel("Num of Steps")
         plt.ylabel("Error")
         plt.show()

Does not converge!
```



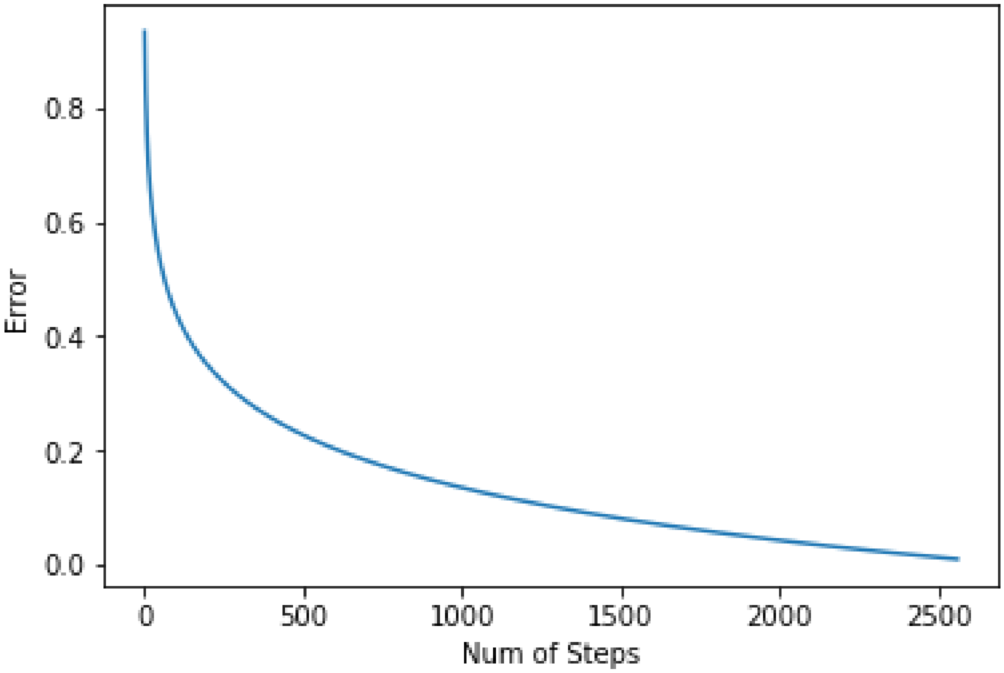
```
In [38]: # Problem2.3
         errors = []
         x = np.zeros((2, len(x_opt)))
         error = 100
         alpha = 5/6
         i = 1
         while (error > 0.01) & (i < 15000):
             x[i,:] = x[i-1,:] - (alpha ** i) * (x[i-1,:] - b) / np.linalg.norm(x[i-1,:]-b)
             error = error_cal(x[i,:], x_opt)
             x_temp = x[i,:]
             x = np.vstack((x, x_temp))
             errors.append(error)
             i += 1

         if i==15000:
             print("Does not converge!")
         else:
             print(i-1)
         plt.plot(range(1,i), errors)
         plt.xlabel("Num of Steps")
         plt.ylabel("Error")
         plt.show()
```

```
Does not converge!
```



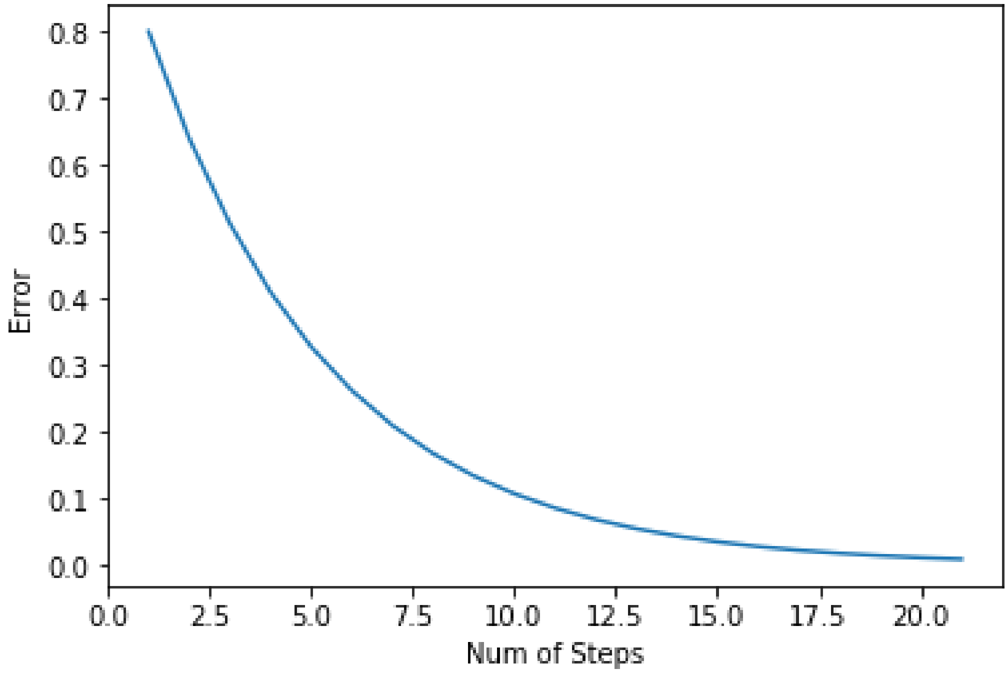
```
In [40]: # Problem2.4
         errors = []
         x = np.zeros((2, len(x_opt)))
         error = 100
         i = 1
         while (error > 0.01) & (i < 15000):
             x[i,:] = x[i-1,:] - (1/(1 + i)) * (x[i-1,:] - b) / np.linalg.norm(x[i-1,:]-b)
             error = error_cal(x[i,:], x_opt)
             x_temp = x[i,:]
             x = np.vstack((x, x_temp))
             errors.append(error)
             i += 1

         if i==15000:
             print("Does not converge!")
         else:
             print(i-1)
         plt.plot(range(1,i), errors)
         plt.xlabel("Num of Steps")
         plt.ylabel("Error")
         plt.show()
```

```
2559
```



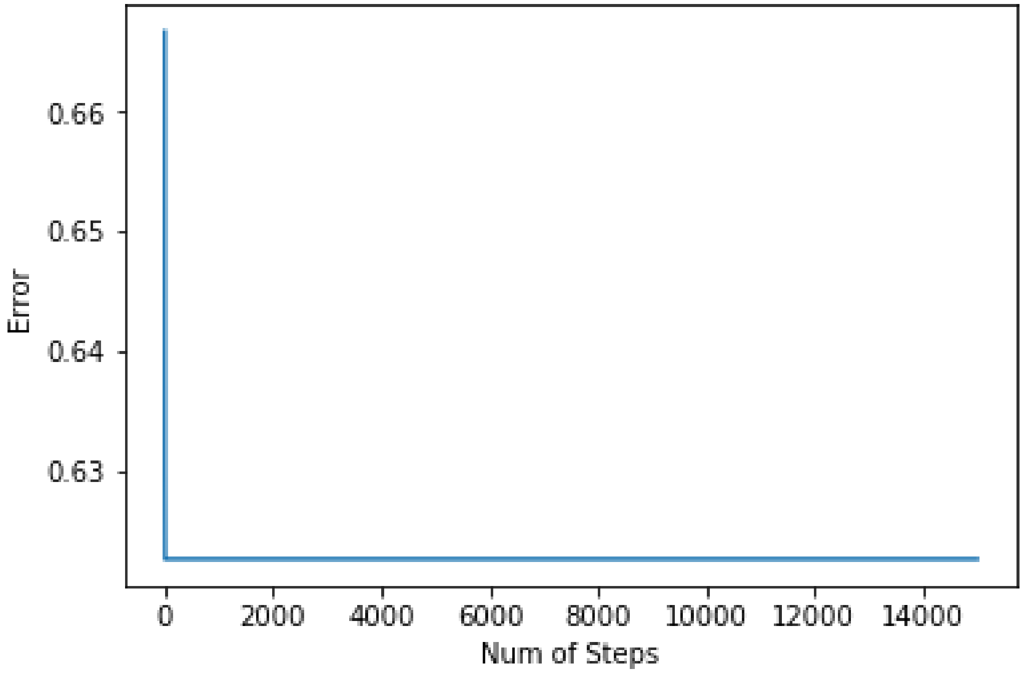
```
In [42]: # Problem2.5
         errors = []
         x = np.zeros((2, len(x_opt)))
         error = 100
         alpha = 0.1
         i = 1
         while (error > 0.01) & (i < 15000):
             x[i,:] = x[i-1,:] - alpha * 2*(x[i-1,:] - b)
             error = error_cal(x[i,:], x_opt)
             x_temp = x[i,:]
             x = np.vstack((x, x_temp))
             errors.append(error)
             i += 1
         if i==15000:
             print("Does not converge!")
         else:
             print(i-1)
         plt.plot(range(1,i), errors)
         plt.xlabel("Num of Steps")
         plt.ylabel("Error")
         plt.show()
```

```
21
```



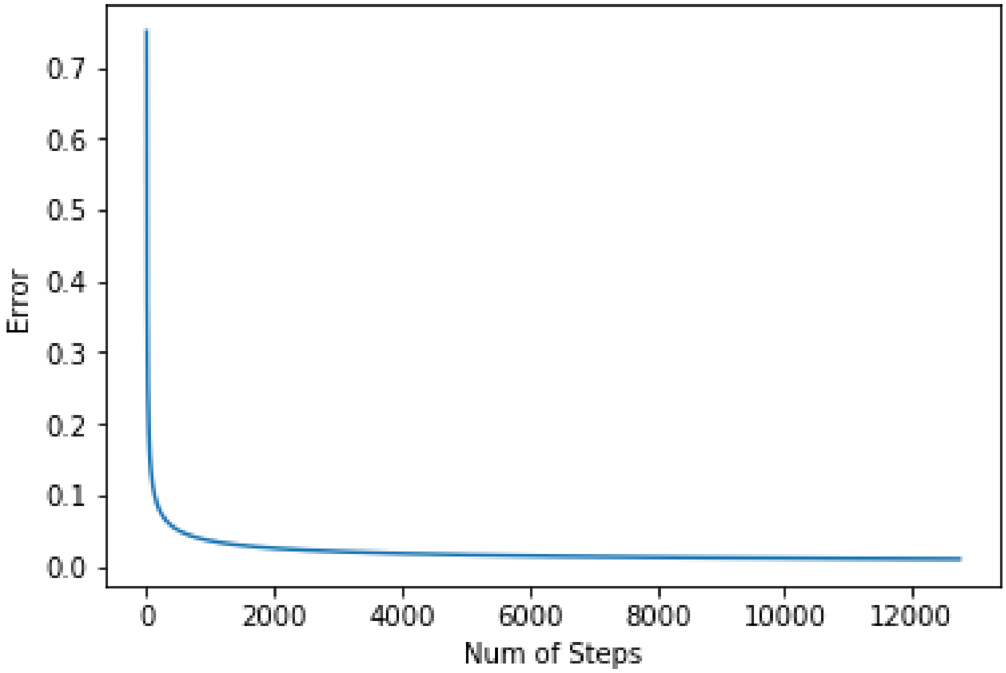
```
In [45]: # Problem2.6
         errors = []
         x = np.zeros((2, len(x_opt)))
         error = 100
         alpha = 1/6
         i = 1
         while (error > 0.01) & (i < 15000):
             x[i,:] = x[i-1,:] - (alpha ** i) * 2*(x[i-1,:] - b)
             error = error_cal(x[i,:], x_opt)
             x_temp = x[i,:]
             x = np.vstack((x, x_temp))
             errors.append(error)
             i += 1
         if i==15000:
             print("Does not converge!")
         else:
             print(i-1)
         plt.plot(range(1,i), errors)
         plt.xlabel("Num of Steps")
         plt.ylabel("Error")
         plt.show()
```

```
Does not converge!
```



```
In [46]: # Problem2.7
         errors = []
         x = np.zeros((2, len(x_opt)))
         error = 100
         i = 1
         while (error > 0.01) & (i < 15000):
             x[i,:] = x[i-1,:] - (1/(4*(1+i))) * 2*(x[i-1,:] - b)
             error = error_cal(x[i,:], x_opt)
             x_temp = x[i,:]
             x = np.vstack((x, x_temp))
             errors.append(error)
             i += 1
         if i==15000:
             print("Does not converge!")
         else:
             print(i-1)
         plt.plot(range(1,i), errors)
         plt.xlabel("Num of Steps")
         plt.ylabel("Error")
         plt.show()
```

```
12732
```


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Error
0
2000
4000
6000
8000
10000
12000
Num of Steps