```
# import libraries
import numpy as np
import matplotlib.pyplot as plt


def driver():

# test functions
     f1 = lambda x: -np.sin(2*x) + 5*x/4 - 3/4
# fixed point is alpha1 = 1.4987....

     f2 = lambda x: 3+2*np.sin(x)
#fixed point is alpha2 = 3.09...

     Nmax = 100
     tol = 0.5e-10

# test f1 '''
     x0 = 1.5
     [xstar,ier] = fixedpt(f1,x0,tol,Nmax)
     print('the approximate fixed point is:',xstar)
     print('f1(xstar):',f1(xstar))
     print('Error message reads:',ier)


     x_values = np.linspace(-5, 10, 400)  # Adjust the range as needed

     # Calculate y values
     y_values = equation(x_values)

     # Plot the function
     plt.figure(figsize=(8, 6))
     plt.plot(x_values, y_values, label='$x - 4\sin(2x) - 3 = 0$')
     plt.axhline(0, color='black', linewidth=0.5)  # Add x-axis
     plt.axvline(0, color='black', linewidth=0.5)  # Add y-axis
     plt.title('Plot of $x - 4\sin(2x) - 3 = 0$')
     plt.xlabel('x')
     plt.ylabel('y')
     plt.grid(True)
     plt.legend()
     plt.show()


# define routines
def fixedpt(f,x0,tol,Nmax):

    ''' x0 = initial guess'''
    ''' Nmax = max number of iterations'''
    ''' tol = stopping tolerance'''

    count = 0

    while (count <Nmax):
       count = count +1
       x1 = f(x0)
       if (abs(x1-x0) <tol):
          xstar = x1
```

```
            ier = 0
            return [xstar,ier]
        x0 = x1

    xstar = x1
    ier = 1
    return [xstar, ier]

# Define the function
def equation(x):
    return x - 4 * np.sin(2 * x) - 3

# Generate x values


driver()
```