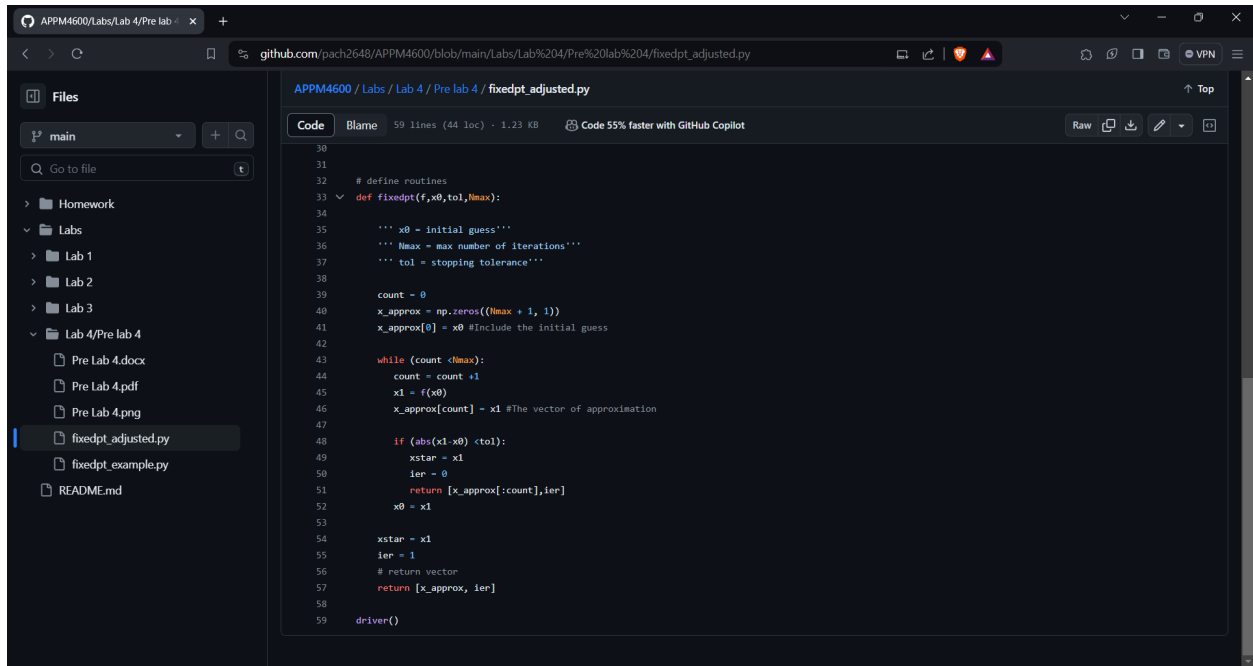


Pre-lab 4

1. Create a new fixed point iteration subroutine that returns a vector whose entries are the approximations of the fixed point at all the iterations in order.



The screenshot shows a GitHub web interface for the repository 'pach2648/APPM4600'. The file 'fixedpt_adjusted.py' is selected in the 'Lab 4/Pre lab 4' directory. The code is a Python script implementing a fixed-point iteration algorithm. It includes a function 'def fixedpt(f, x0, tol, Nmax):' which takes a function 'f', an initial guess 'x0', a tolerance 'tol', and a maximum number of iterations 'Nmax'. The code initializes 'count' to 0 and 'x_approx' as a zero vector of size 'Nmax + 1'. It then enters a 'while' loop that iterates until the tolerance is met or the maximum number of iterations is reached. Inside the loop, it calculates 'x1 = f(x0)', updates 'x_approx[count] = x1', and checks if the absolute difference between 'x1' and 'x0' is less than 'tol'. If so, it sets 'xstar = x1', 'ier = 0', and returns the vector 'x_approx' and 'ier'. Otherwise, it increments 'count' and continues the loop. After the loop, it sets 'xstar = x1', 'ier = 1', and returns the vector 'x_approx' and 'ier'. Finally, it calls 'driver()' at the bottom.

```
30
31
32 # define routines
33 def fixedpt(f, x0, tol, Nmax):
34
35     ''' x0 = initial guess'''
36     ''' Nmax = max number of iterations'''
37     ''' tol = stopping tolerance'''
38
39     count = 0
40     x_approx = np.zeros((Nmax + 1, 1))
41     x_approx[0] = x0 #Include the initial guess
42
43     while (count < Nmax):
44         count = count + 1
45         x1 = f(x0)
46         x_approx[count] = x1 #The vector of approximation
47
48         if (abs(x1 - x0) < tol):
49             xstar = x1
50             ier = 0
51             return [x_approx[:count], ier]
52         x0 = x1
53
54     xstar = x1
55     ier = 1
56     # return vector
57     return [x_approx, ier]
58
59 driver()
```

Github link:

https://github.com/pach2648/APPM4600/blob/main/Labs/Lab%204/Pre%20lab%204/fixedpt_adjusted.py