104283 - Introduction to Numerical Analysis
Spring 2023
Python Assignment 1

Name: Haoyi Yang
Student ID: 999009798

March 29, 2023

Part *I*

1. Question

Write a function in Python: bisection(a, b, func, tol) The function gets the following parameters as input:
- Parameters a, b representing the interval.
- Some function f(x) defined by func.
- Convergence tolerance - tol.

Using the Bisection method, the function searches for a root $f(x) = 0$ in the interval [a,b] (page 49 algorithm 2.1 in the textbook).
The function will output the following:
- List of approximations x_i
- Final approximation x for which the method converges (if it converges).
- Number of iterations.

2. Code and explanation

According to the pseudocode in page 49 algorithm 2.1 in the textbook, We can easily generate the python code:

```
list = []
 def bisection(a,b,func,tol):
    if func(a) * func(b) < 0: # f(a) and f(b) should be different sign.
        i = 1
        FA = func(a)
        while True:
            p = a + (b - a)/2
            list.append(p)
            FP = func(p)
            if FP == 0 or (b - a)/2 < tol:# Here jump out the loop.
                print(list)
                print(f'The number of iteration is {i}')
                print(f'The approximation x = {p}')
                return
            i += 1
            if FA * FP > 0:
                a = p
                FA = FP
            else:
                b = p
    else:
        return ("Can not use the Bisection Method!")
```

Everything is almost based on the pseudocode so I didn't write many comments. The difference between my code and the pseudocode is that I added an if part before the whole code(We should determine whether f(a) and f(b) have the same sign, or it violate the Intermediate Value Theorem so we can't use Bisection method), and I deleted the Max iterations part N_0 because the question seems not ask us to do this. And last I print three outputs that the question ask us to do.

Part II

1. Question

Find solutions (if found) to the problems using bisection in the given intervals:
a) $2x^3 - 2x - 5 = 0; 1 \leq x \leq 2$
b) $e^x - x^2 + 3x - 2 = 0; 0 \leq x \leq 1$
c) $-3.55x^3 + 1.1x^2 + 0.765x - 0.74 = 0; -1 \leq x \leq 1$
d) $x^6 + 6x^5 + 9x^4 - 2x^3 - 6x^2 + 1 = 0; -3 \leq x \leq -2$

2. Code and explanation

First let's add a math python library so that we can generate some mathmatical symbols:

```
import math
```

Then I define four functions in python(I just use $a(x), b(x), c(x), d(x)$ to represent these four functions):

```python
def a(x):
    a = 2*x**3 - 2*x - 5
    return a

def b(x):
    b = math.exp(x) - x**2 + 3*x - 2
    return b

def c(x):
    c = -3.55*x**3 + 1.1*x**2 + 0.765*x - 0.74
    return c

def d(x):
    d = x**6 + 6*x**5 + 9*x**4 - 2*x**3 - 6*x**2 + 1
    return d
```

And next I plug these four functions into the python bisection code I wrote in Part 1 and added the interval and tolerance the question gave us:

```python
bisection(1,2,a,10**-5)
bisection(0,1,b,10**-5)
bisection(-1,1,c,10**-5)
bisection(-3,-2,d,10**-5)
```

Then I got four outputs.

For question a), output is: [1.5, 1.75, 1.625, 1.5625, 1.59375, 1.609375, 1.6015625, 1.59765625, 1.599609375, 1.6005859375, 1.60107421875, 1.600830078125, 1.6007080078125, 1.60064697265625, 1.600616455078125, 1.6006011962890625, 1.6005935668945312]
The number of iteration is 17
The approximation x = 1.6005935668945312

For question b), output is: [0.5, 0.25, 0.375, 0.3125, 0.28125, 0.265625, 0.2578125, 0.25390625, 0.255859375, 0.2568359375, 0.25732421875, 0.257568359375, 0.2574462890625, 0.25750732421875, 0.257537841796875, 0.2575225830078125, 0.25753021240234375]
The number of iteration is 17
The approximation x = 0.25753021240234375

For question c), output is: [0.0, -0.5, -0.75, -0.625, -0.5625, -0.59375, -0.609375, -0.6015625, -0.60546875, -0.607421875, -0.6083984375, -0.60791015625, -0.608154296875, -0.6080322265625, -0.60809326171875, -0.608123779296875, -0.6081390380859375, -0.6081314086914062]

The number of iteration is 18
The approximation x = -0.6081314086914062

For question d), output is: Can not use the Bisection Method!

It can be seen that only question d can't find the answer. That's because for function $d = x^6 + 6x^5 + 9x^4 - 2x^3 - 6x^2 + 1$ and the interval [-3,-2], $d(a) * d(b) \geq 0$. It means $d(a)$ and $d(b)$ have the same sign, so that it violate the Intermediate Value Theorem. So we can't apply Bisection here.

3. Table Conclusion

It seems to be easier to know what's going on if we make a table to show the detailed data of each iteration, so I did. Here are the tables:
Table for question a):

number of iterations i	**a**	**b**	**approximation x_i**
1	**1**	**2**	**1.5**
2	**1.5**	**2**	**1.75**
3	**1.5**	**1.75**	**1.625**
4	**1.5**	**1.625**	**1.5625**
5	**1.5625**	**1.625**	**1.59375**
6	**1.59375**	**1.625**	**1.609375**
7	**1.59375**	**1.609375**	**1.6015625**
8	**1.59375**	**1.6015625**	**1.59765625**
9	**1.59765625**	**1.6015625**	**1.599609375**
10	**1.599609375**	**1.6015625**	**1.6005859375**
11	**1.599609375**	**1.6005859375**	**1.60107421875**
12	**1.6005859375**	**1.60107421875**	**1.600830078125**
13	**1.6005859375**	**1.600830078125**	**1.6007080078125**
14	**1.6005859375**	**1.6007080078125**	**1.60064697265625**
15	**1.6005859375**	**1.6006469726562**	**1.60061645507812**
16	**1.6005859375**	**1.6006164550781**	**1.60060119628906**
17	**1.6005859375**	**1.6006011962890**	**1.60059356689453**

Table for questions b) and c):

number of iterations i	a	b	approximation x_i
1	0	1	0.5
2	0	0.5	0.25
3	0.25	0.5	0.375
4	0.25	0.375	0.3125
5	0.25	0.3125	0.28125
6	0.25	0.28125	0.265625
7	0.25	0.265625	0.2578125
8	0.25	0.2578125	0.25390625
9	0.25390625	0.2578125	0.255859375
10	0.255859375	0.2578125	0.2568359375
11	0.2568359375	0.2578125	0.25732421875
12	0.25732421875	0.2578125	0.257568359375
13	0.25732421875	0.257568359375	0.2574462890625
14	0.2574462890625	0.257568359375	0.25750732421875
15	0.2575073242187	0.257568359375	0.25753784179687
16	0.2574462890625	0.257537841796	0.25752258300781
17	0.2574462890625	0.257522583007	0.25753021240234

number of iterations i	a	b	approximation x_i
1	-1	1	0
2	-1	0	-0.5
3	-1	-0.5	-0.75
4	-0.75	-0.5	-0.625
5	-0.625	-0.5	-0.5625
6	-0.625	-0.5625	-0.59375
7	-0.625	-0.59375	-0.609375
8	-0.609375	-0.59375	-0.6015625
9	-0.609375	-0.6015625	-0.60546875
10	-0.609375	-0.60546875	-0.607421875
11	-0.609375	-0.607421875	-0.6083984375
12	-0.6083984375	-0.607421875	-0.60791015625
13	-0.6083984375	-0.60791015625	-0.608154296875
14	-0.608154296875	-0.60791015625	-0.6080322265625
15	-0.608154296875	-0.60803222656	-0.6080932617187
16	-0.608154296875	-0.60809326171	-0.6081237792968
17	-0.608154296875	-0.60812377929	-0.6081390380859
18	-0.608139038085	-0.60812377929	-0.6081314086914