Homework 4

Hanyuan Zhu

October 2, 2018

1 Bisection Method

The smallest positive root is 0.6662394324925154

The bisection algorithm is below by Python:

```
import numpy as np
import math
#iteration counter: to count how many steps to reach the threshold value
i=0

#Define the target funciton
def f(x):
    return math.tan(x)-math.cos(x)

def randompostive(x):
    return np.multiply(np.random.random(),x)

## NOTE: Inital generator, range:[0,1), half open interval, choosing a number uniformally randomly
a = randompostive(1);
b = randompostive(1);

## Here guarantees f(b)<= 0 <= f(a)
while f(a)<0:
    b = a;
    a = randompostive(1);

while f(b)>0:
    b = randompostive(1);

#to print the initial range
print abs(a-b)

### Setting the threshold epsilon
epsilon_ = math.pow(10,-15)

### Begin the bisection iterations. # NOTE: Return r as root.
if f(a)*f(b) != 0:
    while abs(a-b) > epsilon_ :
        c = (a+b)/2
        if f(c) > 0:
            a = c;
        elif f(c) < 0:
            b = c;
        else:
            a = b;
        i = i+1;
    r = (a+b)/2;
elif  f(a) == 0:
    r = a;
else:
    r = b;

# print the root = 0.6662394324925154
print r

# print the number of iteratoins to get the root 50
print i
```

Newton's Method

The smallest positive root is 0.6662394324925153

The basin of attraction of the smallest positive root is the largest continuous interval of $f'(x)$ which contains the root.

$$f'(x) = 1 + \tan^2 x + \sin x \tag{1}$$

We know $\sin x$ is continuous on $\mathbb{R}$, and $\tan^2 x$ is continuous on $[-4/\pi, 4/\pi]$, therefore the basin of attraction is $[-4/\pi, 4/\pi]$

The Newton's algorithm is below by Python:

```
import numpy as np
import math
#iteration counter: to count how many steps to reach the threshold value
i=0

def randompostive(x):
    return np.multiply(np.random.random(),x)

# here iter(x) is x - f(x)/f'(x)
def iter(x):
    f_x = math.tan(x)-math.cos(x);
    df_x = 1+math.pow(math.tan(x),2)+math.sin(x);
    return x - f_x/df_x

## NOTE: Inital generator, range:[0,5). Since we get smallest root from bisection around 1.57.
x_1 = 0

x_2 = randompostive(2);

### Setting the threshold epsilon
epsilon_ = math.pow(10,-15)

while abs(x_1-x_2) > epsilon_:
    x_1 = x_2;
    x_2 = iter(x_1);
    i = i+1;

# x_2= 0.6662394324925153
print x_2

# i = 9
print i
```