

Homework 2 of Computational Mathematics

AM15 黃琦翔 111652028

March 24, 2024

1. $x^3 = x + 1 \implies x^2 = 1 + \frac{1}{x} \implies x = \sqrt{1 + \frac{1}{x}} = g(x)$. $p_1 = g(p_0) = \sqrt{1+1} = \sqrt{2} \approx 1.414$. $p_2 = g(p_1) = \sqrt{1 + \frac{1}{\sqrt{2}}} \approx 1.3065$. $p_3 = g(p_2) \approx 1.3172$. $p_4 = g(p_3) \approx 1.326$. $p_5 = g(p_4) \approx 1.324$. Then, p_4 is the answer that we want to find.

2. Let $f(x) = x^3 + x - 4$, $f'(x) = 3x^2 + 1 < 49$ for all $x \in [1, 4]$. Thus, for $|x - y| < \frac{10^{-3}}{49} \approx 2.0409e - 5$, $|f(x) - f(y)| < 10^3$. Find n s.t. $3 \cdot 2^{-n} < 2.0409e - 5$, $n > -\log_2\left(\frac{2.0409e - 5}{3}\right) \approx 17.1653$. Thus, the bound of the number of iteration is 18. Then, by python code below, the root is about 1.3787.

```
HW2 > HW2.py > ...
1  a = [1]
2  b = [4]
3
4  def f(x):
5      return x**3 + x - 4
6
7  while 1:
8      mid = (a[-1] + b[-1])/2
9      val = f(mid)
10     print(mid, val)
11
12     if abs(val) < 0.0001:
13         break
14     elif val > 0:
15         b.append(mid)
16     else:
17         a.append(mid)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\ryan\OneDrive - 國立陽明交通大學\HW\計數> &
2.5 14.125
1.75 3.109375
1.375 -0.025390625
1.5625 1.377197265625
1.46875 0.637176513671875
1.421875 0.2965202331542969
1.3984375 0.13326025009155273
1.38671875 0.05336350202560425
1.380859375 0.013844214379787445
1.3779296875 -0.005808685906231403
1.37939453125 0.004008884658105671
1.378662109375 -0.0009021193400258198
1.3790283203125 0.0015528278327110456
1.37884521484375 0.00032521555817766057
1.378753662109375 -0.00028848656066315925
1.3787994384765625 1.8355831034710945e-05
```

3. In this question, we let $p_{n+1} = g(p_n)$ and $p = \sqrt[3]{21}$. And by $\lim_{n \rightarrow \infty} \frac{|p_{n+1} - p|}{|p_n - p|^\alpha} = \lambda$,

$$\frac{|p_{n+1} - p|}{|p_n - p|} \approx \frac{\lambda |p_n - p|^\alpha}{\lambda |p_{n-1} - p|^\alpha} \approx \left| \frac{p_n - p}{p_{n-1} - p} \right|^\alpha, \text{ then } \alpha \approx \frac{\ln |(p_{n+1} - p)/(p_n - p)|}{\ln |(p_n - p)/(p_{n-1} - p)|}.$$

Then, running python and observe the function, we get a is linearly convergence, b is quadratic convergence (Newton's method), and d is suplinear converges. Meanwhile, though $\sqrt[3]{21}$ is a fixed point of c, but it will diverges or converges to 0 for any open interval contains $\sqrt[3]{21}$.

Thus, order of speed of convergence is $b > d > a > c$.

```

20
21 def problem_3():
22     import math
23     p = 21**(1.0/3)
24     def a(x):
25         return (20*(x**3)+21)/(21*(x**2))
26
27     def b(x):
28         return x- (x**3 - 21)/(3*(x**2))
29
30     def c(x):
31         return x - (x**4 - 21*x)/(x**2 - 21)
32
33     def d(x):
34         return math.sqrt(21/x)
35
36     functions = {"a": a, "b": b, "c":c, "d": d}
37
38     for i in functions.keys():
39         f = functions[i]
40         p_now = 1
41         for k in range(10):
42             p_now = f(p_now)
43             # print(k+1, "err:", abs(p_now - p))
44             try:
45                 alpha = math.log(abs((f(f(p_now)) - p)/(f(p_now) - p)))/math.log(abs
46                 print(i, alpha)
47             except:
48                 print(i, "Cannot compute alpha by this method.")
49
50         print("---")
51
52

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\9ryan\OneDrive - 國立陽明交通大學\HW\計數> & C:/Users/9ryan/AppData/Local/Programs/Python
a 0.9904058224210771
---
b Cannot compute alpha by this method.
---
c Cannot compute alpha by this method.
---
d 1.0016102080959117
---
PS C:\Users\9ryan\OneDrive - 國立陽明交通大學\HW\計數>

```