Homework 3

Panupong (Peach) Chaphuphuang

February 16, 2024

Github URL for homework 3:
`https://github.com/pach2648/APPM4600/tree/main/Homework/HW3`

Problem 1

Solution

Consider the equation $2x - 1 = \sin x$.

a) Find a closed interval $[a, b]$ on which the equation has a root r, and use the Intermediate Value Theorem to prove that r exists.

From the Intermediate Value Theorem:

$$f(a) < \kappa < f(b)$$
$$a < r < b$$
$$f(r) = \kappa$$

where r is the root of an equation

From $2x - 1 = \sin x$, it is difficult to solve it directly, so I let $f(x) = 2x - 1$ and $g(x) = \sin x$. To solve it, I find the interception of both equations to find the range of a and b.

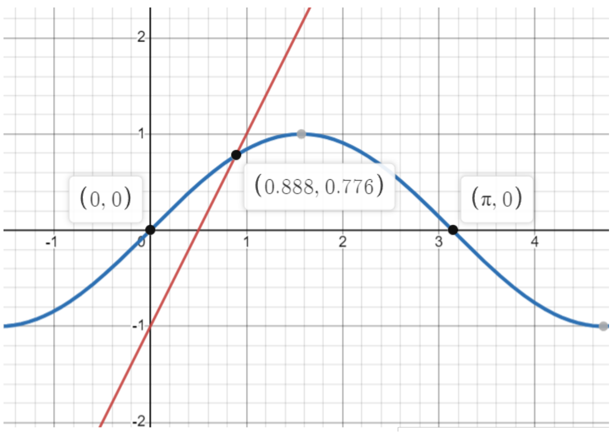

Figure 1: The plot of both $f(x)$ and $g(x)$

From figure 1, we can see that the root is in between $a = 0$ and $b = \pi$ which should be the interval on the equation.

We can recheck by using the Intermediate Value Theorem.

$$
\begin{aligned}
F(x) = 2x - 1 - \sin x &= 0 \\
F(r) &= 0 = \kappa \\
F(a) = F(0) &= -1 < 0 \\
F(b) = F(\pi) &= 2\pi - 1 > 0 \\
\underbrace{F(0)}_{-1} < \underbrace{\kappa}_{0} &< \underbrace{F(\pi)}_{2\pi-1}
\end{aligned}
$$

It matches the theorem, so a closed interval $[a, b]$ on which the equation has a root r can be $[0, \pi]$

b) Prove that r from (a) is the only root of the equation (on all of $\mathbb{R}$).

$$
\begin{aligned}
F(x) &= 2x - 1 - \sin x = 0 \\
F'(x) &= \underbrace{2 - \cos x > 0}_{\text{Always greater than zero since cos(x) must be in [-1,1] only}}
\end{aligned}
$$

From the first derivative of the $F(x)$, we can know that this function is always an increasing function which guarantees that it cannot turn and go back to get another root.

c) Use the bisection code from class (or your own) to approximate r to eight correct decimal places. Include the calling script, the resulting final approximation, and the total number of iterations used.

For the code, click **Github Link for 1c**

```
Bisection method with nmax=100 and tol=1.0e-08

|--n--|--an--|--bn--|----xn----|-|bn-an|--|---|f(xn)|---|
|--0--|0.0000|3.1416|1.57079633|3.14159265|1.14159265|
|--1--|0.0000|1.5708|0.78539816|1.57079633|0.13631045|
|--2--|0.7854|1.5708|1.17809725|0.78539816|0.43231496|
|--3--|0.7854|1.1781|0.98174770|0.39269908|0.13202580|
|--4--|0.7854|0.9817|0.88357293|0.19634954|0.00586459|
|--5--|0.8836|0.9817|0.93266032|0.09817477|0.06211311|
|--6--|0.8836|0.9327|0.90811663|0.04908739|0.02788683|
|--7--|0.8836|0.9081|0.89584478|0.02454369|0.01095233|
|--8--|0.8836|0.8958|0.88970886|0.01227185|0.00252925|
|--9--|0.8836|0.8897|0.88664090|0.00613592|0.00167132|
|--10--|0.8866|0.8897|0.88817488|0.00306796|0.00042805|
|--11--|0.8866|0.8882|0.88740789|0.00153398|0.00062186|
|--12--|0.8874|0.8882|0.88779138|0.00076699|0.00009696|
|--13--|0.8878|0.8882|0.88798313|0.00038350|0.00016553|
|--14--|0.8878|0.8880|0.88788725|0.00019175|0.00003428|
|--15--|0.8878|0.8879|0.88783932|0.00009587|0.00003134|
|--16--|0.8878|0.8879|0.88786329|0.00004794|0.00000147|
|--17--|0.8878|0.8879|0.88785130|0.00002397|0.00001493|
|--18--|0.8879|0.8879|0.88785729|0.00001198|0.00000673|
|--19--|0.8879|0.8879|0.88786029|0.00000599|0.00000263|
|--20--|0.8879|0.8879|0.88786179|0.00000300|0.00000058|
|--21--|0.8879|0.8879|0.88786254|0.00000150|0.00000045|
|--22--|0.8879|0.8879|0.88786216|0.00000075|0.00000007|
|--23--|0.8879|0.8879|0.88786235|0.00000037|0.00000019|
|--24--|0.8879|0.8879|0.88786226|0.00000019|0.00000006|
|--25--|0.8879|0.8879|0.88786221|0.00000009|0.00000000|
|--26--|0.8879|0.8879|0.88786223|0.00000005|0.00000003|
|--27--|0.8879|0.8879|0.88786222|0.00000002|0.00000001|
|--28--|0.8879|0.8879|0.88786222|0.00000001|0.00000001|
```

Figure 2: Python results (29 iterations)

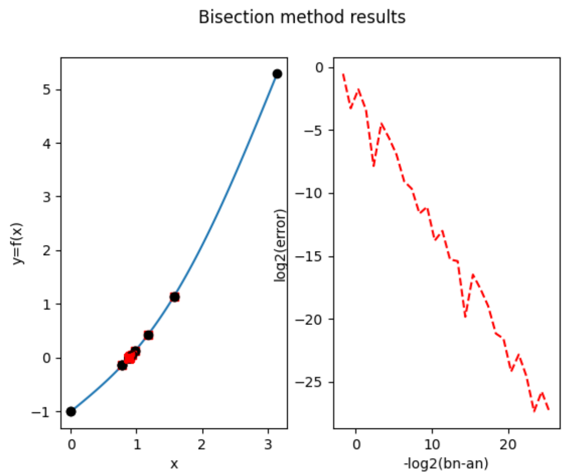

Figure 3: Error plots

Problem 2

The function $f(x) = (x-5)^9$ has a root (with multiplicity 9) at $x = 5$ and is monotonically increasing (decreasing) for $x > 5$ ($x < 5$) and should thus be a suitable candidate for your function above. Use **a=4.82** and **b=5.2** and **tol = 1e-4** and use **bisection** with:

Solution

For the code, click **Github Link for 2a and 2b**

a) $f(x) = (x-5)^9$

From the code, the approximate root is 5.000073242187501 **and** $f(x) = 6.065292655789404 \times 10^{-38}$

b) The expanded expanded version of $(x-5)^9$, that is, $f(x) = x^9 - 45x^8 + \cdots - 1953125$.

From the code, the approximate root is 5.12875 **and** $f(x) = 0.0$

c) Explain what is happening.

Part a is more accurate, since in part b we have multiple two closed number subtraction when expanding $f(x) = (x-5)^9$ which make part b lose precision while calculating the root.

Problem 3

Solution

a) Use a theorem from class (Theorem 2.1 from text) to find an upper bound on the number of iterations in the bisection needed to approximate the solution of $x^3+x-4=0$ lying in the interval $[1,4]$ with an accuracy of 10^{-3}

$$
\begin{aligned}
|p_N - p| \leq 2^{-N}(b-a) = 2^{-N}(4-1) &< 10^{-3} \\
2^{-N} &< \frac{10^{-3}}{3} \\
-N\log_{10} 2 &< \log_{10}\frac{10^{-3}}{3} \\
N &> -\frac{\log_{10}\frac{10^{-3}}{3}}{\log_{10} 2} \approx 11.55
\end{aligned}
$$

We need $N > 11.55$ and N needs to be an integer, so $N = 12$

b) Find an approximation of the root using the bisection code from class to this degree of accuracy. How does the number of iterations compare with the upper bound you found in part (a)?

For the code, click **Github Link for 3b**

```
Bisection method with nmax=100 and tol=1.0e-03

|--n--|--an--|--bn--|----xn----|-|bn-an|--|---|f(xn)|---|
|--0--|1.0000|4.0000|2.50000000|3.00000000|14.12500000|
|--1--|1.0000|2.5000|1.75000000|1.50000000|3.10937500|
|--2--|1.0000|1.7500|1.37500000|0.75000000|0.02539062|
|--3--|1.3750|1.7500|1.56250000|0.37500000|1.37719727|
|--4--|1.3750|1.5625|1.46875000|0.18750000|0.63717651|
|--5--|1.3750|1.4688|1.42187500|0.09375000|0.29652023|
|--6--|1.3750|1.4219|1.39843750|0.04687500|0.13326025|
|--7--|1.3750|1.3984|1.38671875|0.02343750|0.05336350|
|--8--|1.3750|1.3867|1.38085938|0.01171875|0.01384421|
|--9--|1.3750|1.3809|1.37792969|0.00585938|0.00580869|
|--10--|1.3779|1.3809|1.37939453|0.00292969|0.00400888|
|--11--|1.3779|1.3794|1.37866211|0.00146484|0.00090212|
```

Figure 4: Python results (12 iterations)

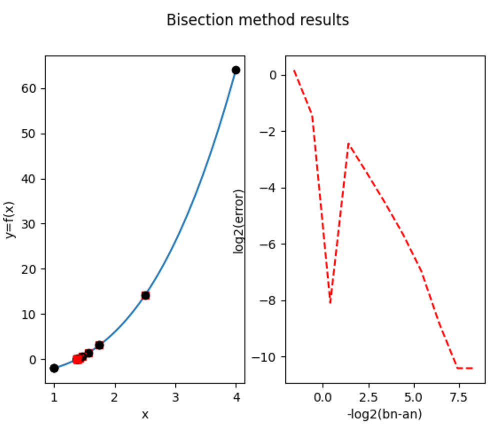

Figure 5: Error plots

Problem 4

Definition 1 *Suppose $\{p_n\}_{n=0}^{\infty}$ n=0 is a sequence that converges to p with $p_n \neq p$ for all n. If there exists postive constants λ and α such that*

$$\lim_{n\to\infty} \frac{|p_{n-1} - p|}{|p_n - p|^{\alpha}} = \lambda$$

Then $\{p_n\}_{n=0}^{\infty}$ converges to p with an order α and asympotic error constant λ. If $\lambda = 1$ and $\lambda < 1$ then the sequence converges linearly. If $\alpha = 2$, the sequence is quadratically convergent. Which of the following iterations will converge to the indicated fixed point x_* (provided x_0 is sufficiently close to x_*)? If it does converge, give the order of convergence; for linear convergence, give the rate of linear convergence.

Solution

a) (10 points) $x_{n+1} = -16 + 6x_n + 12x_n, x_* = 2$

$$\begin{aligned} f(x_n) &= -16 + 6x_n + 12x_n^{-1} \\ f'(x_n) &= 6 - \frac{12}{x_n^2} \\ f'(x_n = x_* = 2) &= 6 - \frac{12}{2^2} = 3 > 1 \end{aligned}$$

Since the absolute of the first derivative (slope) at the fixed point is greater than 1, this iteration does not converge.

b) (10 points) $x_{n+1} = \frac{2}{3}x_n + \frac{1}{x_n^2}, x_* = 3^{1/3}$

$$\begin{aligned} f(x_n) &= \frac{2}{3}x_n + \frac{1}{x_n^2} \\ f'(x_n) &= \frac{2}{3} + (-2)\frac{1}{x_n^3} \\ f'(x_n = x_* = 3^{1/3}) &= \frac{2}{3} - \frac{2}{3} = 0 < 1 \end{aligned}$$

Since the absolute of the first derivative (slope) at the fixed point is less than 1, this iteration does converge. However, the slope is zero at the fixed point which is a special case (double roots case)

Consider the Taylor expansion of f about the fixed point r to find the order of convergence:

$$\begin{aligned} \underbrace{f(x_{n-1})}_{x_n} &= \underbrace{f(r)}_{r} + \cancel{\underbrace{f'(r)}_{0}(x_{n-1} - r)} + \frac{1}{2}f''(c)(x_{n-1} - r)^2 \\ \frac{x_n - r}{(x_{n-1} - r)^2} &= \frac{f''(c)}{2} \\ \lim_{n\to\infty}\left|\frac{x_n - r}{(x_{n-1} - r)^2}\right| &= \lim_{n\to\infty}\left|\frac{f''(c)}{2}\right| = \underbrace{\lambda}_{constant} \end{aligned}$$

From the definition of the order of convergence

$$\lim_{n\to\infty}\left|\frac{p_{n+1}-p}{(p_n-p)^\alpha}\right| = \underbrace{\lambda}_{constant}$$

where α is the order of convergence, and p is a fixed point.

When comparing with the taylor expansion above, we realize that if $f'(r) = 0$, the α must be 2. It means that this iteration converges quadratically

c) (10 points) $x_{n+1} = \frac{12}{1+x_n}, x_* = 3$

$$\begin{aligned} f(x_n) &= \frac{12}{1+x_n} \\ f'(x_n) &= \frac{-12}{(1+x)^2} \\ f'(x_n = x_* = 3) &= \frac{-12}{4^2} = \frac{-12}{16} \\ \left|f'(x_* = 3)\right| &= \frac{12}{16} < 1 \end{aligned}$$

Since the absolute of the first derivative (slope) at the fixed point is less than 1, this iteration does converge.

Consider the Taylor expansion of f about the fixed point r to find the order of convergence:

$$\begin{aligned} \underbrace{f(x_{n-1})}_{x_n} &= \underbrace{f(r)}_{r} + f'(r)(x_{n-1}-r) \\ \frac{x_n - r}{(x_{n-1}-r)^1} &= f'(c) \\ \lim_{n\to\infty}\left|\frac{x_n-r}{(x_{n-1}-r)}\right| &= \lim_{n\to\infty}\left|f'(c)\right| = \underbrace{\lambda}_{constant} \end{aligned}$$

The order of convergence of this iteration (α) should be 1 (You can compare the taylor expansion to the definition of the order of convergence above like part (b)) since we use the fixed point iteration and we know that it should make the iteration converge linearly

Problem 5

All the roots of the scalar equation

$$x - 4sin(2x) - 3 = 0,$$

are to be determined with at least 10 accurate digits[1].

Solution

a) Plot $f(x) = x - 4sin(2x) - 3$ (using your **Python** toolbox). All the zero crossings should be in the plot. How many are there?

For the code, click **Github Link for 5a**

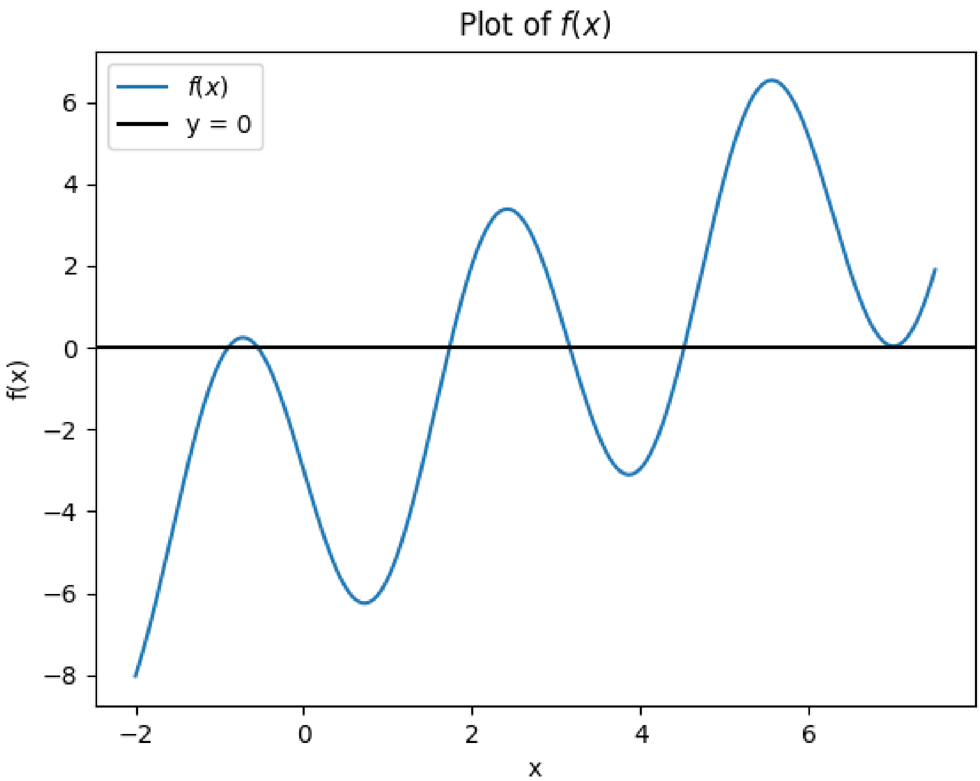

Figure 6: Plot which has 5 zero crossings

b) Write a program or use the code from class to compute the roots using the fixed point iteration

$$x_{n+1} = -sin(2x_n) + 5x_n/4 - 3/4.$$

Use a stopping criterium that gives an answer with ten correct digits. (*Hint: you may have to change the error used in determing the stopping criterion.*) Find, empirically which of the roots that can be found with the above iteration. Give a theoretical explanation.

For the code, click **Github Link for 5b**

In this iteration, after testing with two initial guesses ($x_0 = 1$ & $x_0 = 3$), the results from both guesses are completely different. For $x_0 = 1$, the result is -0.5444424 and use 19 numbers of iteration to get 10 digits of accuracy. On the other hand, for $x_0 = 3$, the result is 3.16182649 and use 78 numbers of iteration to get 10 digits of accuracy.

If there are multiple roots like this case, the fixed point iteration's behaviour can vary based on the initial guess. It means that the initial guess plays a significant role in finding the root of the fixed point iteration method.