Math 151A Homework #1

Nathan Solomon

October 11, 2024

Problem 0.1.

(a) The intermediate value theorem says that if a real function (real domain and codomain) is less than a somewhere and greater than a somewhere else, and continuous everywhere between those points, it must have gone through a at some point, meaning there is a point where the function is exactly equal to a.

(b) Let $a = 0.2$ and $b = 0.3$, and let $f(x) = x\cos x - 2x^2 + 3x - 1$. Then $f(a) = 0.2\cos(0.2) - 0.08 + 0.6 - 1 = 0.2 * \cos(0.2) - .48$ which is negative because $\cos(0.2)$ can't be more than 1, and $f(b) = 0.3\cos(0.3) - 0.18 + 0.9 - 1 = 0.3\cos(0.3) - 0.28$, which is positive because I plugged it into a calculator and got 0.0066. $f : \mathbb{R} \to \mathbb{R}$ is continuous everywhere, so by IVT, there must be some $x \in (a, b)$ such that $f(x) = 0$.

Problem 0.2.

$$\begin{aligned} f(x) &= \sqrt{x+1} \\ f'(x) &= \frac{1}{2}(x+1)^{-1/2} \\ f''(x) &= -\frac{1}{4}(x+1)^{-3/2} \\ f'''(x) &= \frac{3}{8}(x+1)^{-5/2} \end{aligned}$$

Then $f(0) = 1, f'(0) = \frac{1}{2}, f''(0) = -\frac{1}{4}, f'''(0) = \frac{3}{8}$. Plugging that into the formula for Taylor series, we get $P_3(x) = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \frac{1}{16}x^3$.

x	$P_3(x-1)$	$\sqrt{x}$	Absolute error $\lvert P_3(x) - \sqrt{x}\rvert$
0.50	0.71094	0.70711	0.00383
0.75	0.86621	0.86603	0.00019
1.25	1.11816	1.11803	0.00013
1.50	1.22656	1.22474	0.00182

Here is the code used to generate that:

```
>>> from math import sqrt
>>> def P_3(x):
...     return 1 + x/2 - x**2/8 + x**3/16
...
>>> for x in [.5, .75, 1.25, 1.5]:
...     print(f"{x:.2f} & {P_3(x-1):.5f} & {sqrt(x):.5f} & " +
            f"{abs(P_3(x-1) - sqrt(x)):.5f} \\\\\\\\n\\hline")
```

Problem 0.3.

(a) (i) $\frac{12}{15} + \frac{5}{15} = \frac{17}{15} = 1.13333\ldots$

(ii) $\left(\frac{11}{33} + \frac{9}{33}\right) - \frac{3}{20} = \frac{20}{33} - \frac{3}{20} = \frac{400-99}{660} = \frac{301}{660} = .456060606\ldots.$

(b) (i) $0.800 + 0.333 = 1.13$

(ii) $(0.333 + 0.272) - 0.150 = 0.605 - 0.150 = 0.455$

(c) (i) $0.800 + 0.333 = 1.13$

(ii) $(0.333 + 0.273) - 0.150 = 0.606 - 0.150 = 0.456$

(d) (i) Chopping and rounding both give relative error of $\frac{|1.13-17/15|}{17/15} = 0.0029411764$

(ii) With chopping, relative error is $\frac{|0.455-301/660|}{301/660} = 0.00232558139$. With rounding, relative error is $\frac{|0.456-301/660|}{301/660} = 0.00013289036$.

Problem 0.4.

```
>>> def f(x): return x**2 - 3
...
>>> a = 0
>>> b = 2
>>> max_possible_error = b - a
>>> iteration = 0
>>> while max_possible_error > 1e-4:
...     iteration += 1
...     midpoint = (a + b) / 2
...     max_possible_error = midpoint - a
...     print(f"{iteration=}\t{midpoint=:.7f}\t{max_possible_error=:.7f}")
...     if f(midpoint) < 0: a = midpoint
...     else: b = midpoint
...
iteration=1     midpoint=1.0000000      max_possible_error=1.0000000
iteration=2     midpoint=1.5000000      max_possible_error=0.5000000
iteration=3     midpoint=1.7500000      max_possible_error=0.2500000
iteration=4     midpoint=1.6250000      max_possible_error=0.1250000
iteration=5     midpoint=1.6875000      max_possible_error=0.0625000
iteration=6     midpoint=1.7187500      max_possible_error=0.0312500
iteration=7     midpoint=1.7343750      max_possible_error=0.0156250
iteration=8     midpoint=1.7265625      max_possible_error=0.0078125
iteration=9     midpoint=1.7304688      max_possible_error=0.0039062
iteration=10    midpoint=1.7324219      max_possible_error=0.0019531
iteration=11    midpoint=1.7314453      max_possible_error=0.0009766
iteration=12    midpoint=1.7319336      max_possible_error=0.0004883
iteration=13    midpoint=1.7321777      max_possible_error=0.0002441
iteration=14    midpoint=1.7320557      max_possible_error=0.0001221
iteration=15    midpoint=1.7319946      max_possible_error=0.0000610
```

So after 15 iterations, we get that $\sqrt{3} \approx 1.7319946$.

Problem 0.5.

(a) Let $\alpha = 1$, $p = 0$, and $\lambda = 0.1$. Then

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^\alpha} = \lim_{n\to\infty} \frac{10^{-(n+1)} - 0}{(10^{-n} - 0)^1} = \lim_{n\to\infty} \frac{10^{-n-1}}{10^{-n}} = \lambda.$$

So p_n converges to $p = 0$ with order $\alpha = 1$.

(b) Let $\alpha = 2$, $p = 0$, and $\lambda = 1$. Then

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^\alpha} = \lim_{n\to\infty} \frac{10^{-2^{n+1}} - 0}{(10^{-2^n} - 0)^2} = \lim_{n\to\infty} \frac{10^{2(-2^n)}}{10^{2(-2^n)}} = \lambda.$$

So p_n converges to $p = 0$ with order $\alpha = 2$.

Problem 0.6.

Suppose $f : [a, b] \to \mathbb{R}$ is a continuous function such that $f(a) - c$ and $f(b) - c$ have opposite signs, and you are using the bisection method to find a solution to $f(x) = c$ (a root of $f(x) - c$) on $[a, b]$. After n iterations, the bisection method will give you an approximation which is off by at most $2^{-n}(b - a)$, so define $\varepsilon_n := 2^{-n}(b - a)$. Then ε_n converges to $\varepsilon := 0$ with order $\alpha := 1$, because

$$\lim_{n\to\infty} \frac{|\varepsilon_{n+1} - \varepsilon|}{|\varepsilon_n - \varepsilon|^\alpha} = \lim_{n\to\infty} \frac{2^{-(n+1)}(b - a)}{2^{-n}(b - a)} = \frac{1}{2},$$

which is positive.

Call the root we're going to find p, so $f(p) = c$, and call the midpoint after n steps of the bisection method p_n. Since $|p_n - p| \le \varepsilon_n$ for all $n \in \mathbb{N}$, and we know ε_n converges linearly to $\varepsilon = 0$, p_n converges to p with at least order 1.

Problem 0.7.

The slope of the line is $(f(a_k) - f(b_k))/(a_k - b_k)$ and the line goes through $(a_k, f(a_k))$, so the formula for the line is

$$y = f(a_k) + \frac{f(a_k) - f(b_k)}{a_k - b_k} \cdot (x - a_k).$$

The root x_k is the x-value such that $y = 0$:

$$\begin{aligned}
0 &= f(a_k) + \frac{f(a_k) - f(b_k)}{a_k - b_k} \cdot (x_k - a_k) \\
f(a_k)(b_k - a_k) &= (f(a_k) - f(b_k))(x_k - a_k) \\
f(a_k)b_k - f(b_k)a_k &= (f(a_k) - f(b_k))x_k \\
x_k &= \frac{f(a_k)b_k - f(b_k)a_k}{f(a_k) - f(b_k)}
\end{aligned}$$

Problem 0.8.

(a) Here is the code I used:

```
clear all
close all
a=1.75; %%%%%%%%%
a=1; %%%%%%%%%
b=2.95; %%%%%%%%%
b=1.2; %%%%%%%%%
%b=2; %%%%%%%%%
tol = 1e-6;
Nmax = 100;
%f=@(x) (x-1)*(x-2)*(x-3); %%%%%%%%%
f=@(x) x^6 - x - 1; %%%%%%%%%

i = 1;
success = 0;
while (i<=Nmax) && (success==0)
    midpoint = (a+b)/2; %%%%%%%%%
    %midpoint=(f(a)*b - f(b)*a) / (f(a) - f(b)); %%%%%%%%%
    if (abs(f(midpoint)) < tol)
        success = 1;
    else
        i = i + 1;
        midpoint = (a+b)/2; %%%%%%%%%
        %midpoint=(f(a)*b - f(b)*a) / (f(a) - f(b)); %%%%%%%%%
        % disp([a midpoint b])
        if (sign(f(midpoint)) == sign(f(a)))
            a = midpoint;
        else
            b = midpoint;
        end
    end
end
if (success == 1)
    disp('success!');
else
    disp('did not converge');
end
i
format long
display(midpoint)
format long
numericalsolution = fzero(f, midpoint)
```

(b) The bisection method gives an answer of 2.000000762939453 after 19 iterations, and the Reguli-Falsi method gives an answer of 1.999999999762428 after 5 iterations. The actual solution is 2.

(c) The actual solution is 1.134724138401519.

For the interval $[1.0, 1.2]$, the bisection method gives 1.134724044799805 after 19 iterations, and the Reguli-Falsi method gives 1.134724105897137 after 8 iterations.

For the interval $[1.0, 2.0]$, the bisection method gives 1.134724140167236 after 21 iterations and the Reguli-Falsi method gives 1.134724053091091 after 92 iterations.

Since the Reguli-Falsi method took way more steps to converge in that last case, and since one iteration of the Reguli-Falsi method requires more FLOPs than one iteration of the bisection method, we can conclude that the Reguli-Falsi method is not always better than the bisection method.