Homework 6

This homework assignment is about root finding for nonlinear equations. The bisection method from section 3.1 and the newton method from section 3.2.

For the written exercises, you should upload a scanned PDF to Gradescope and then follow the prompts given by Gradescope to assign certain pages of your PDF document to the correct problems.

For the coding exercises, you will be prompted to upload your python file directly to Gradescope. Please make sure you submit only one file and that your variables are assigned to the correct variable names.

The course syllabus found on Canvas has information on how homework is graded and how homework should be presented and submitted. Please let me or the TAs know if you have any questions or concerns.

This assignment is due on Tuesday, February 20 at 11:59pm.

Written Assignment

1. From section 3.1 (pg. 123) Complete problem 1. Use an initial guess of $x_0 = 0$ and compute x_5. Circle your final answer.
Answer:

$$f(x) = e^x - 3x$$

Note that $x_l = 0$ and $x_r = 1$. Furthermore, $f(x_l) = 1$ and $f(x_r) = -0.3$.

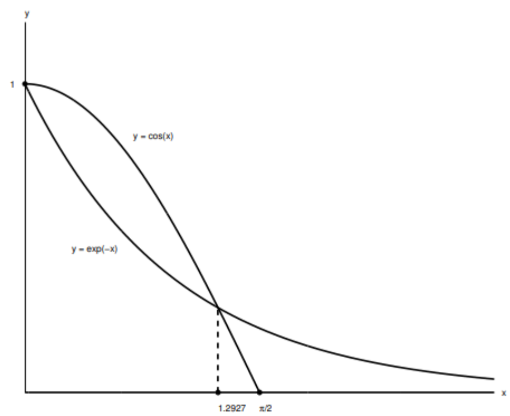
(i) $x_l = 0$, $x_r = 1$ so we have $\frac{x_r + x_l}{2} = x_c = 0.5$ and $f(x_c) = 0.2$, so $x_c \to x_l$
(ii) $x_l = 0.5$, $x_r = 1$ so we have $x_c = 0.75$ and $f(x_c) = -0.1$, so $x_c \to x_r$
(iii) $x_l = 0.5$, $x_r = 0.75$ so we have $x_c = 0.625$ and $f(x_c) = -0.007$, so $x_c \to x_r$
(vi) $x_l = 0.5$, $x_r = 0.625$ so we have $x_c = 0.5625$ and $f(x_c) = 0.07$, so $x_c \to x_l$
(v) $x_l = 0.5625$, $x_r = 0.625$ so we have $x_c = 0.59375$.

From section 3.2 complete

a.) problem 1 (*hint: solving the equation $x^2 = R$ is the same as finding the roots of the function $f(x) = x^2 - R$.* Circle your final answer.
Answer:

3.2.1. $f(x) = x^2 - R, f'(x) = 2x$. $x - \dfrac{f(x)}{f'(x)} = x - \dfrac{x^2 - R}{2x} = \dfrac{x^2 + R}{2x} = \dfrac{1}{2}\left(x + \dfrac{R}{x}\right)$. So $x_{n+1} = \dfrac{1}{2}\left(x_n + \dfrac{R}{x_n}\right)$.

b.) problem 17 Circle your final answer.
Answer:



3.2.17. $f(x) = x^5 - x^3 + 3,\ f(1) = 3;\ f'(x) = 5x^4 - 3x^2,\ f'(1) = 2$. So

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = 1 - \frac{3}{2} = -\frac{1}{2}$$

c.) problem 25 Circle your final answer.
Answer:

3.2.25. $f(x) = x^2 - R$, $f'(x) = 2x$. So $x - \dfrac{f}{f'} = x - \dfrac{x^2 - R}{2x} = (x^2 + R)/(2x)$. Therefore, $x_{n+1} = \dfrac{1}{2}[x_n + (R/x_n)]$.

$R = 2$: $x_0 = 1$, $x_1 = 1.5$, $(\varepsilon_1 = 8.5 \times 10^{-2})$, $x_2 = 1.4167$ $(\varepsilon_2 = 2.5 \times 10^{-3})$, $x_3 = 1.4142$ $(\varepsilon_3 = 2.1 \times 10^{-6})$.

x	$f(x) = x^2 - 2$
1	-1.00
2	2.00
1.5	0.25
1.25	-0.44
1.375	-0.11
1.4375	

For Bisection

$$n \geq -1 + \log\left(\frac{b-a}{c}\right) \Big/ \log 2$$
$$n \geq -1 + \log\left(\frac{1}{10^{-6}}\right) \Big/ \log 2 = -1 + 6/\log 2 = 18.9$$

Therefore, 19 steps.

For Newton's method, $|ce_n| \leq (ce_0)^{2^n}$ where $e_0 = |1 - \sqrt{2}| = 0.41 \leq \dfrac{1}{2} = \delta$ and

$$c = \frac{1}{2}\left(\max_{|x-\sqrt{2}|\leq\delta} |f''(x)|\right) \Big/ \left(\min_{|x-\sqrt{2}|\leq\delta} |f'(x)|\right)$$
$$= \frac{1}{2}(2)/[2(\sqrt{2} - \delta)] = 1/(2\sqrt{2} - 1) = 0.55 \approx 2^{-1}$$

Now $|e_n| \leq 2(2^{-1}2^{-1})^{2^n} < 10^{-6}$. Want

$$2(2^{-2})^{2^n} < 10^{-6}$$
$$\log 2 + 2^n \log 2^{-2} < -6$$
$$2^n > \frac{6 + \log 2}{2\log 2} = \frac{3}{\log 2} + \frac{1}{2} = 10.47$$

$n = 3, \text{LHS} = 8$; $n = 4, \text{LHS} = 16$. Therefore, 4 iterations.

Coding Assignment

The coding assignment for this week will ask you to write a python function for Newtons method for root finding. In the week 6 modules on Canvas, there is a file called Newton_method.py. In this code a function called Newton_method(f,df,x0,n) is defind. The arguments are f for the function you want to find the root of, df is the derivative of the function, x0 is the initial guess, and n is the number of iterations of Newtons method you want to calculate.

Another stopping criteria for Newton's method is to stop iterations when $|f(x_k)| < \epsilon$, where ϵ is some predefined threshold. Ideally, the smaller the threshold, the closer our root will be to the exact root (provided our initial guess is close enough to the exact root). The hw6.py template on Canvas is a template for Newtons method that uses this stopping criteria. There are many ways one could write this code, however the hw6_template.py template makes use of a while loop.

1. In 1685, John Wallis published a book called *Algebra*, in which he described a method devised by Newton for solving equations. In slightly modified form, this method was also published by Joseph Raphson in 1690. This form is the one now commonly called Newton's method or the Newton-Raphson method. Newton himself discussed the method in 1669 and illustrated it with the equation $x^3 - 2x - 5 = 0$. Wallis used the same example.

   Use your Newton's method code to approximate the root of $f(x) = x^3 - 2x - 5$, thus finding the root of the function Newton used to describe the method. Use an initial guess of $x_0 = 1$, and use a threshold value of $\epsilon = 10^{-8}$. Your function should return the approximate root and the number of iterations it took to get this root. The autograder will ask for the approximate root which is a float variable named x and the number of iterations which is an int variable named n.