ECE 3331, Fall 2023, Program 04, Due Sunday 09/17 at 11:59 pm on Blackboard

1) Write a program to approximate the 5th root of a number using the Newton-Raphson algorithm discussed below this assignment.
2) Let the user input the double value c_0 (use %lf in scanf( )) for which your program is to find the 5th root $(c_0)^{\frac{1}{5}}$ by using the equation $f(x) = x^5 - c_0 = 0$ in the Newton-Raphson algorithm.
If for example, $c_0 = 243.0$ in the equation, then the 5th root of 81.0 is 3.0.
So the equation $f(x) = x^5 - c_0 = 0$ at the solution x=3.0 is satisfied (equals 0.0).

Using the Newton-Raphson algorithm, the value x computed by your program will be the 5th root $x = (c_0)^{\frac{1}{5}}$.
3) Let the user input c_0 . and then let the user input the initial guess x_0 of the 5th root of c_0 . Use double variables and use scanf( ) with the conversion specifier %lf.
4) Next, let the user input a maximum error **ε** such that your program stops the iterative solution whenever **|f(x)|< ε .**
For example, use ε = 0.0000000001.
5) Next, let the user input a maximum number of iterations for your program to run such that your program stops the iterative solution after the maximum number of iterations have been completed if the maximum error has not been reached.
For example, use max iterations = 500.
6) To update the estimate of the root, your program will compute the Newton-Raphson algorithm $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$. As you can imagine, your program will need a double variable for the previous estimate of the root x_i and another double variable for the next computed estimate of the root x_{i+1}.

For example, your program will already have the user's initial guess x_0 , so, on the first iteration, use $x_i = x_0$ and your program will compute $x_{i+1} = x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$.

At the end of your program when the iteration has been completed, your program will print out the current estimate of the root x_{i+1}, the current error **|f(x)|**, and the number of iterations that were completed. For example, if you used $c_0 = 16.0$ and $x_0 = 3.7$, you might converge to a value x=2.00000132 after perhaps 30 iterations.

Clearly identify each value in the printout.

Read about the Newton-Raphson algorithm below.

The Newton-Raphson Algorithm
An Iterative Method for Solving f(x)=0

Introduction

A solution for an equation that has been set equal to zero $f(x) = 0$ is called a root of that equation.
Such an equation may be a linear function of x or a non-linear function of x.
Often the equation will have more than one root; such as a 2nd order polynomial in x has two roots.
There are a number of methods for finding roots of an equation.
The computer is an extremely effective tool for iterative methods for finding roots.
Iterative methods begin with the user choosing an initial guess x_0 for the value of a root of the equation.
Then using the function $f(x) = 0$ and the initial guess x_0 an iterative algorithm computes a slightly better estimate x_1 of the root.
Then using the function $f(x) = 0$ and the 1st computed improved estimate x_1 , an iterative algorithm computes a slightly better estimate x_2 of the root.
This continues n times until $|f(x_n)| < \varepsilon$ where $\varepsilon > 0$ and ε is provided by the user.

A widely-used iterative algorithm for finding a root of an equation is the Newton-Raphson algorithm.

Derivation

The Newton-Raphson method is based on the principle that if the initial guess of the root of $f(x) = 0$ is at x_i, then if one draws the tangent to the curve at $f(x_i)$, the point x_{i+1} where the tangent crosses the x-axis is an improved estimate of the root (Figure 1).

Using the definition of the slope of a function, at $x = x_i$

$$f'(x_i) = \tan\theta$$
$$= \frac{f(x_i) - 0}{x_i - x_{i+1}},$$

which gives

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \qquad (1)$$

Equation (1) is called the Newton-Raphson formula for solving nonlinear equations of the form $f(x) = 0$. So starting with an initial guess , x_i, one can find the next guess, x_{i+1}, by using Equation (1). One can repeat this process until one finds the root within a desirable tolerance.

Algorithm

The steps of the Newton-Raphson method to find the root of an equation $f(x) = 0$ are

1. Given function f(x) find its derivative $f'(x)$.
2. Choose an initial guess for the root, x_i, to estimate the new value of the root, x_{i+1}, as
$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$
3. Find the absolute relative approximate error $|\in_a|$ as
$$|\in_a| = \left|\frac{x_{i+1} - x_i}{x_{i+1}}\right| \times 100$$
4. Compare the absolute relative approximate error with the pre-specified relative error tolerance, $\in_s$. If $|\in_a| > \in_s$, then go to Step 2, else stop the algorithm. Also, check if the number of iterations has exceeded the maximum number of iterations allowed. If so, one needs to terminate the algorithm and notify the user.

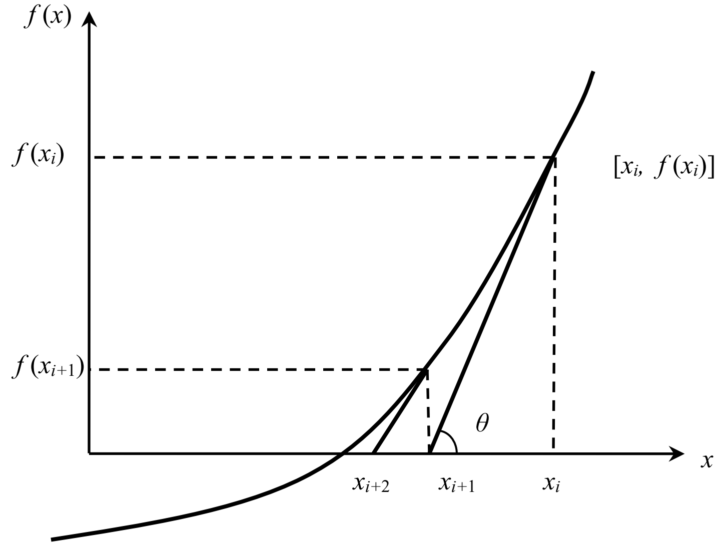

Figure 1 Geometrical illustration of the Newton-Raphson method.

Example 1

Find the solution of the equation

$$x^3 - 0.165x^2 + 3.993\times 10^{-4} = 0\,.$$

Use the Newton-Raphson method of finding roots of equations to find

a) the depth x to which the ball is submerged under water. Conduct three iterations to estimate the root of the above equation.
b) the absolute relative approximate error at the end of each iteration, and
c) the number of significant digits at least correct at the end of each iteration.

Solution

$$f(x) = x^3 - 0.165x^2 + 3.993\times 10^{-4}$$
$$f'(x) = 3x^2 - 0.33x$$

Let us assume the initial guess of the root of $f(x) = 0$ is $x_0 = 0.05\,\text{m}$. This is a reasonable guess (discuss why $x = 0$ and $x = 0.11\,\text{m}$ are not good choices) as the extreme values of the depth x would be 0 and the diameter (0.11 m) of the ball.

Iteration 1

The estimate of the root is

$$\begin{aligned}
x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} \\
&= 0.05 - \frac{(0.05)^3 - 0.165(0.05)^2 + 3.993\times 10^{-4}}{3(0.05)^2 - 0.33(0.05)} \\
&= 0.05 - \frac{1.118\times 10^{-4}}{-9\times 10^{-3}} \\
&= 0.05 - (-0.01242) \\
&= 0.06242
\end{aligned}$$

The absolute relative approximate error $|\in_a|$ at the end of Iteration 1 is

$$|\in_a| = \left|\frac{x_1 - x_0}{x_1}\right| \times 100$$
$$= \left|\frac{0.06242 - 0.05}{0.06242}\right| \times 100$$
$$= 19.90\%$$

The number of significant digits at least correct is 0, as you need an absolute relative approximate error of 5% or less for at least one significant digit to be correct in your result.

Iteration 2

The estimate of the root is

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$
$$= 0.06242 - \frac{(0.06242)^3 - 0.165(0.06242)^2 + 3.993 \times 10^{-4}}{3(0.06242)^2 - 0.33(0.06242)}$$
$$= 0.06242 - \frac{-3.97781 \times 10^{-7}}{-8.90973 \times 10^{-3}}$$
$$= 0.06242 - \left(4.4646 \times 10^{-5}\right)$$
$$= 0.06238$$

The absolute relative approximate error $|\in_a|$ at the end of Iteration 2 is

$$|\in_a| = \left|\frac{x_2 - x_1}{x_2}\right| \times 100$$
$$= \left|\frac{0.06238 - 0.06242}{0.06238}\right| \times 100$$
$$= 0.0716\%$$

The maximum value of m for which $|\in_a| \le 0.5 \times 10^{2-m}$ is 2.844. Hence, the number of significant digits at least correct in the answer is 2.

Iteration 3

The estimate of the root is

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$$
$$= 0.06238 - \frac{(0.06238)^3 - 0.165(0.06238)^2 + 3.993 \times 10^{-4}}{3(0.06238)^2 - 0.33(0.06238)}$$
$$= 0.06238 - \frac{4.44 \times 10^{-11}}{-8.91171 \times 10^{-3}}$$
$$= 0.06238 - \left(-4.9822 \times 10^{-9}\right)$$
$$= 0.06238$$

The absolute relative approximate error $|\in_a|$ at the end of Iteration 3 is

$$|\in_a| = \left|\frac{0.06238 - 0.06238}{0.06238}\right| \times 100$$
$$= 0$$

The number of significant digits at least correct is 4, as only 4 significant digits are carried through in all the calculations.

Drawbacks of the Newton-Raphson Method

1. Divergence at inflection points

If the selection of the initial guess or an iterated value of the root turns out to be close to the inflection point (see the definition in the appendix of this chapter) of the function $f(x)$ in the equation $f(x)=0$, Newton-Raphson method may start diverging away from the root. It may then start converging back to the root. For example, to find the root of the equation

$$f(x)=(x-1)^3+0.512=0$$

the Newton-Raphson method reduces to

$$x_{i+1}=x_i-\frac{(x_i^3-1)^3+0.512}{3(x_i-1)^2}$$

Starting with an initial guess of $x_0=5.0$, Table 1 shows the iterated values of the root of the equation. As you can observe, the root starts to diverge at Iteration 6 because the previous estimate of 0.92589 is close to the inflection point of $x=1$ (the value of $f'(x)$ is zero at the inflection point). Eventually, after 12 more iterations the root converges to the exact value of $x=0.2$.

Table 1 Divergence near inflection point.

Iteration Number	x_i
0	5.0000
1	3.6560
2	2.7465
3	2.1084
4	1.6000
5	0.92589
6	–30.119
7	–19.746
8	–12.831
9	–8.2217
10	–5.1498
11	–3.1044
12	–1.7464
13	–0.85356
14	–0.28538
15	0.039784
16	0.17475
17	0.19924
18	0.2

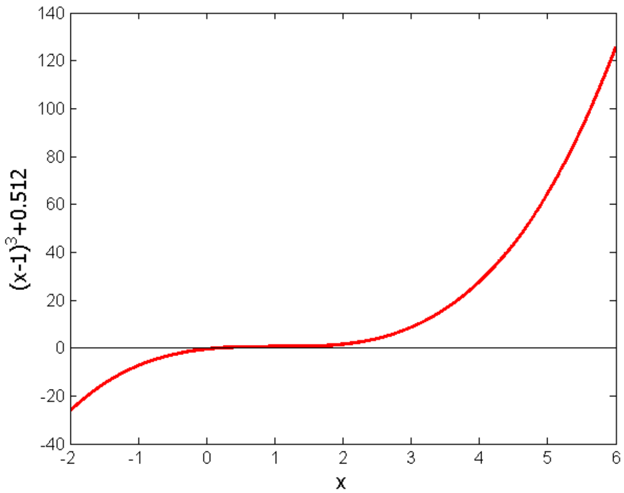

Figure 3 Divergence at inflection point for $f(x)=(x-1)^3=0$.

2. Division by zero

For the equation

$$f(x)=x^3-0.03x^2+2.4\times10^{-6}=0$$

the Newton-Raphson method reduces to

$$x_{i+1}=x_i-\frac{x_i^3-0.03x_i^2+2.4\times10^{-6}}{3x_i^2-0.06x_i}$$

For $x_0=0$ or $x_0=0.02$, division by zero occurs (Figure 4). For an initial guess close to 0.02 such as $x_0=0.01999$, one may avoid division by zero, but then the denominator in the formula is a small number. For this case, as given in Table 2, even after 9 iterations, the Newton-Raphson method does not converge.

Table 2 Division by near zero in Newton-Raphson method.

Iteration Number	x_i	$f(x_i)$	$\lvert\in_a\rvert\%$
0	0.019990	-1.60000×10^{-6}	
1	–2.6480	18.778	100.75
2	–1.7620	–5.5638	50.282
3	–1.1714	–1.6485	50.422
4	–0.77765	–0.48842	50.632
5	–0.51518	–0.14470	50.946
6	–0.34025	–0.042862	51.413
7	–0.22369	–0.012692	52.107
8	–0.14608	–0.0037553	53.127
9	–0.094490	–0.0011091	54.602

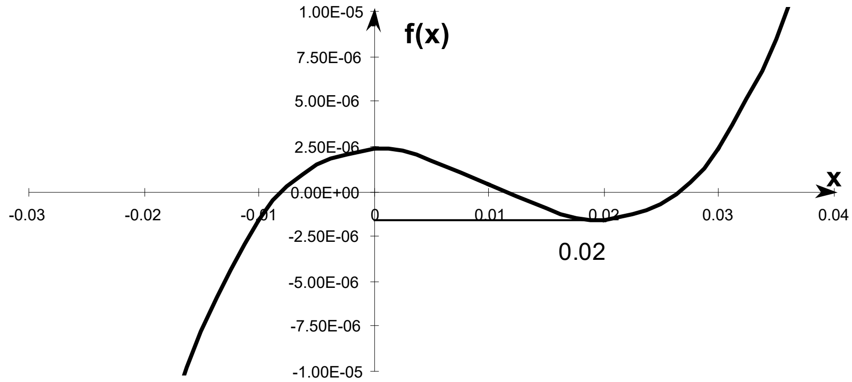

Figure 4 Pitfall of division by zero or a near zero number.

<u>3. Oscillations near local maximum and minimum</u>

Results obtained from the Newton-Raphson method may oscillate about the local maximum or minimum without converging on a root but converging on the local maximum or minimum. Eventually, it may lead to division by a number close to zero and may diverge.

For example, for

$$f(x) = x^2 + 2 = 0$$

the equation has no real roots (Figure 5 and Table 3).

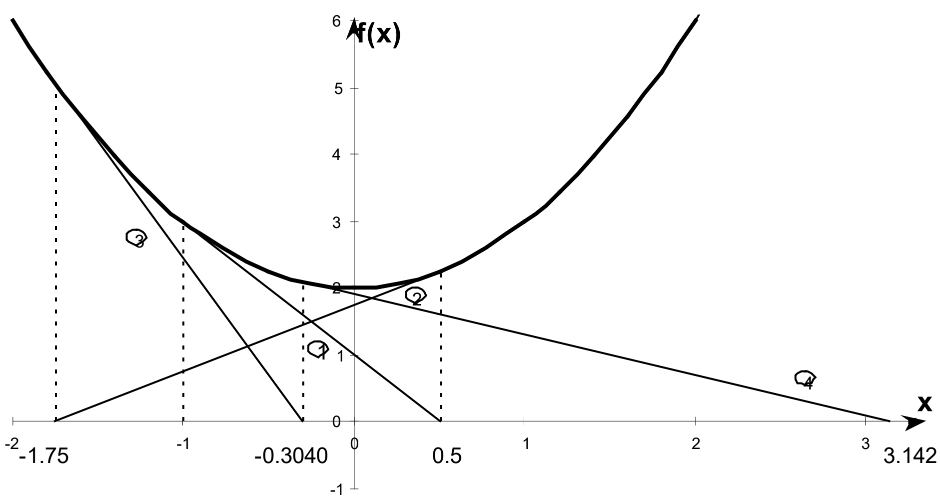

Figure 5 Oscillations around local minima for $f(x) = x^2 + 2$.

Table 3 Oscillations near local maxima and minima in Newton-Raphson method.

Iteration Number	x_i	$f(x_i)$	$\lvert\in_a\rvert\%$

0	–1.0000	3.00	
1	0.5	2.25	300.00
2	–1.75	5.063	128.571
3	–0.30357	2.092	476.47
4	3.1423	11.874	109.66
5	1.2529	3.570	150.80
6	–0.17166	2.029	829.88
7	5.7395	34.942	102.99
8	2.6955	9.266	112.93
9	0.97678	2.954	175.96

4. <u>Root jumping</u>

In some case where the function $f(x)$ is oscillating and has a number of roots, one may choose an initial guess close to a root. However, the guesses may jump and converge to some other root. For example for solving the equation $\sin x = 0$ if you choose $x_0 = 2.4\pi = (7.539822)$ as an initial guess, it converges to the root of $x = 0$ as shown in Table 4 and Figure 6. However, one may have chosen this as an initial guess to converge to $x = 2\pi = 6.2831853$.

Table 4 Root jumping in Newton-Raphson method.

Iteration Number	x_i	$f(x_i)$	$\lvert\in_a\rvert\%$
0	7.539822	0.951	
1	4.462	–0.969	68.973
2	0.5499	0.5226	711.44
3	–0.06307	–0.06303	971.91
4	8.376×10^{-4}	8.375×10^{-5}	7.54×10^{4}
5	-1.95861×10^{-13}	-1.95861×10^{-13}	4.28×10^{10}

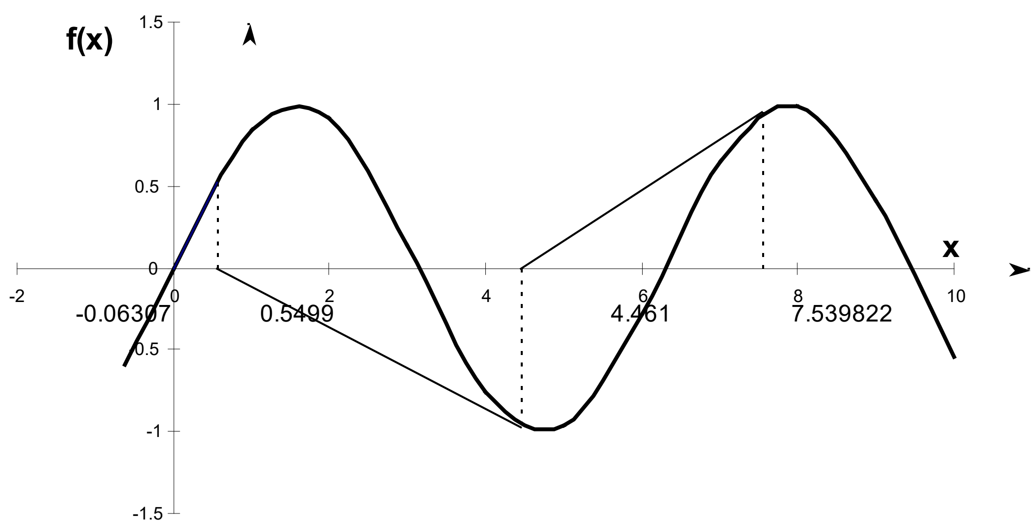

Figure 6 Root jumping from intended location of root for $f(x) = \sin x = 0$.

Appendix A. What is an inflection point?

For a function $f(x)$, the point where the concavity changes from up-to-down or down-to-up is called its inflection point. For example, for the function $f(x)=(x-1)^3$, the concavity changes at $x=1$ (see Figure 3), and hence (1,0) is an inflection point.

An inflection points MAY exist at a point where $f''(x)=0$ and where $f''(x)$ does not exist. The reason we say that it MAY exist is because if $f''(x)=0$, it only makes it a possible inflection point. For example, for $f(x)=x^4-16$, $f''(0)=0$, but the concavity does not change at $x=0$. Hence the point (0, –16) is not an inflection point of $f(x)=x^4-16$.

For $f(x)=(x-1)^3$, $f''(x)$ changes sign at $x=1$ ($f''(x)<0$ for $x<1$, and $f''(x)>0$ for $x>1$), and thus brings up the *Inflection Point Theorem* for a function $f(x)$ that states the following.

"If $f'(c)$ exists and $f''(c)$ changes sign at $x=c$, then the point $(c, f(c))$ is an inflection point of the graph of f."