Nohan Joemon
CH19B072

CH2061 - Assignment 2 Nonlinear equations

17th September 2020

QUESTION 1 Bisection method (10 marks)

Water is flowing in a trapezoidal channel at a rate of Q = 40 m^3/s. The critical depth y for such a channel must satisfy the equation

$0 = 1 - (Q^2B / gAc^3)$

where g = 9.81 m/s^2, Ac = the cross-sectional area (m^2), and B = the width of the channel at the surface (m). For this case, the width and the cross-sectional area can be related to y by

$B = 3 + y$ and $Ac = 3y + y^2/2$

Solve for the critical depth using bisection method. Use initial guesses of $x_l = 1.0$ and $x_u = 3.0$ and iterate until the approximate percent relative error falls below 1% or the number of iterations exceed 10.

ANSWER 1

$f(x) = 1 - (Q^2B / gAc^3)$

substituting g = 9.81 m/s^2, Q = 40 m^3/s, $B = 3 + y$ and $Ac = 3y + y^2/2$, we have:

$f(x) = 1 - 163.09(3+y)/(3y+y^2/2)^3$

Initial guess: $x_l = 1.0$ and $x_u = 3.0$

$f(x_l)f(x_u) = f(1.0)f(3.0)=(-14.21)X(0.98) <0$, hence the initial guesses are safe to be used

Iter 1:

[$\mathbf{x_l = 1.0}$ **and** $\mathbf{x_u = 3.0}$

$f(x_l) = -14.21 \quad f(x_u) = 0.98$

$\mathbf{x_r = (x_l+x_u)/2=2.0}$

$f(x_r=2.0) = -0.5926$

$f(x_r)f(x_u)<0 \Rightarrow x_l = 2.0,\ x_u = 3.0$ for next iteration

Approx relative error : Not defined (first iteration)]

Iter 2:

[$\mathbf{x_l = 2.0}$ **and** $\mathbf{x_u = 3.0}$

$f(x_l) = -0.5926 \quad f(x_u) = 0.98$

$\mathbf{x_r = (x_l+x_u)/2=2.5}$

$f(x_r=2.5) = 0.2521$

$f(x_r)f(x_l)<0 \Rightarrow x_l = 2.0,\ x_u = 2.5$ for next iteration

Approx relative error :

$(|\ x_r(iter2) - x_r(iter1)\ |) / x_r(iter2) = (2.5 - 2.0)/2.5 = 0.2$

]

Iter 3:

[$\mathbf{x_l = 2.0}$ **and** $\mathbf{x_u = 2.5}$

$f(x_l) = -0.5926 \quad f(x_u) = 0.2521$

$\mathbf{x_r = (x_l+x_u)/2=2.25}$

$f(x_r=2.25) = -\ 0.0709$

$f(x_r)f(x_u)<0 \Rightarrow x_l = 2.25,\ x_u = 2.5$ for next iteration

Approx relative error :

$(|\ x_r(iter3) - x_r(iter2)\ |) / x_r(iter3) = |(2.25 - 2.5)|/2.25 = 0.11$

]

Iter 4:

[**x_l = 2.25 and x_u = 2.5**

$f(x_l) = -0.0709$ $f(x_u) = 0.2521$

$x_r = (x_l+x_u)/2=2.375$

$f(x_r=2.375) = 0.1088$

$f(x_r)f(x_l)<0 \Rightarrow x_l = 2.25$, $x_u = 2.375$ for next iteration

Approx relative error :

$(|\ x_r(\text{iter4}) - x_r(\text{iter3})\ |) / x_r(\text{iter4}) = |(2.375 - 2.25)|/2.375 = 0.05$

]

Iter 5:

[**x_l = 2.25 and x_u = 2.375**

$f(x_l) = -0.0709$ $f(x_u) = 0.1088$

$x_r = (x_l+x_u)/2=2.3125$

$f(x_r=2.3125) = 0.0241$

$f(x_r)f(x_l)<0 \Rightarrow x_l = 2.25$, $x_u = 2.3125$ for next iteration

Approx relative error :

$(|\ x_r(\text{iter5}) - x_r(\text{iter4})\ |) / x_r(\text{iter5}) = |(2.3125 - 2.375)|/2.3125 = 0.02$

]

Iter 6:

[**x_l = 2.25 and x_u = 2.3125**

$f(x_l) = -0.0709$ $f(x_u) = 0.0241$

$x_r = (x_l+x_u)/2=2.28125$

$f(x_r=2.28125) = -\ 0.0219$

$f(x_r)f(x_u)<0 \Rightarrow x_l = 2.28125$, $x_u = 2.3125$ for next iteration

Approx relative error :

($|x_r(iter6) - x_r(iter5)|$) $/x_r(iter6)$ = |(2.28125 - 2.3125)|/2.28125 = 0.013

]

Iter 7:

[**x_l = 2.28125 and x_u = 2.3125**

$f(x_l) = -0.0219$ $f(x_u) = 0.0241$

$x_r = (x_l+x_u)/2=2.296875$

$f(x_r=2.296875) = 1.4235 \times 10^{-3}$

Approx relative error :

($|x_r(iter7) - x_r(iter6)|$) $/x_r(iter7)$ = |(2.296875 - 2.28125)|/2.296875 = 0.006 <0.01 ; hence stop iteration

]

Therefore the value of critical depth is y = 2.296875m

QUESTION 2 Regula fasli/False position method (10 marks)

The velocity v of a falling parachutist is given by $v = gmc(1 - \exp(-c/m)t)$

where g=9.81 m/s^2. For a parachutist with a drag coefficient c = 15 kg/s, compute the mass m so that the velocity is v = 35 m/s at t = 9 s. Use the false position/regula fasli method to determine m until the approximate error falls below 1%. You can choose the initial guesses to cover a wide range of mass for an adult human.

ANSWER 2

$f(x) = v - gm(1 - \exp(-c/m)t)/c$

substituting g = 9.81 m/s^2, c = 15 kg/s, v = 35 m/s, t = 9 s, we have:

$f(x) = 35 - 0.654m(1 - e^{(-135/m)})$

Initial guess: $x_l = 30.0$ and $x_u = 100.0$

$f(x_l) = 15.5979$ $f(x_u) = -13.4456$

$f(x_l)f(x_u) < 0$, hence the initial guesses are safe to be used

Iter 1:

[$x_l = 30.0$ and $x_u = 100.0$

$f(x_l) = 15.5979$ $f(x_u) = -13.4456$

$x_r = x_u - (f(x_u)(x_l - x_u))/(f(x_l) - f(x_u)) = 67.5936$

$f(x_r) = -3.2069$

$f(x_r)f(x_l) < 0 \Rightarrow x_l = 30.0$, $x_u = 67.5936$ for next iteration

Approx relative error :

Not defined as it is the first iteration

]

Iter 2:

[x_l = 30.0 and x_u = 67.5936

$f(x_l)$ = 15.5979 $f(x_u)$ = - 3.2069

$x_r = x_u - (f(x_u)(x_l - x_u))/(f(x_l) - f(x_u))$ = 61.1834

$f(x_r)$ = -0.6085

$f(x_r)f(x_l)<0 \Rightarrow x_l$ = 30.0, x_u = 61.1834 for next iteration

Approx relative error :

(| x_r(iter2) - x_r(iter1) |) /x_r(iter2) = |(61.1834- 67.5936)|/61.1834 = 0.1047

]

Iter 3:

[x_l = 30.0 and x_u = 61.1834

$f(x_l)$ = 15.5979 $f(x_u)$ = - 0.6085

$x_r = x_u - (f(x_u)(x_l - x_u))/(f(x_l) - f(x_u))$ = 60.0117

$f(x_r)$ = -0.1092

$f(x_r)f(x_l)<0 \Rightarrow x_l$ = 30.0, x_u = 60.0117 for next iteration

Approx relative error :

(| x_r(iter2) - x_r(iter1) |) /x_r(iter2) = |(60.0117- 61.1834)|/60.0117 = 0.0195

]

Iter 4:

[x_l = 30.0 and x_u = 60.0117

$f(x_l)$ = 15.5979 $f(x_u)$ = - 0.1092

$x_r = x_u - (f(x_u)(x_l - x_u))/(f(x_l) - f(x_u))$ = 59.8031

$f(x_r)$ = - 0.0193

Approx relative error :

$(|x_r(\text{iter2}) - x_r(\text{iter1})|)/x_r(\text{iter2}) = |(59.8031 - 60.0117)|/59.8031 = 0.0034 < 0.01;$

Hence stop iteration

]

Therefore the value of mass is m = 59.8031kg

QUESTION 3 Newton-Raphson method (20 marks)

a) We have a defective calculator that could only add, subtract, and multiply but not divide. Using this calculator and the Newton-Raphson method, find the value of 12.834 up to 5 decimal places. Use the initial guess as 0.4. Take assumptions wherever necessary and state them clearly. With this, write down a general equation to compute the reciprocal of any number x.

b) The same calculator has a defective square-root button too. With the help of this calculator and the Newton-Raphson method, compute the square root of 12.425 up to 5 decimal places. Use the initial guess as 3.5. With this, develop a general model to calculate the rth root of any number N using the Newton-Raphson method.

Trivia: Google and find out why such calculations are/were necessary in the present/past (not for grading).

Tip: Don't just solve it. Try to graphically interpret how the solutions are being approached with increasing iterations (you don't have to submit any graphs).

ANSWER 3

a) Reciprocal of k = solution of the equation f(x) = 0;

where f(x) = k - (1/x)

$f'(x) = 1/x^2$

Therefore, in Newton - Raphson method gives the iteration

$x_{i+1} = x_i - f(x) / f'(x) = x_i - (k -(1/x_i)) / (1/x_i^2) = x_i (2 - kx_i)$

$x_{i+1} = x_i (2 - kx_i)$ => This equation can be iterated until convergence to get the reciprocal of any number k without using division (This uses addition and

multiplication only)

In this case, k = 2.834: Newton - Raphson method: $x_{i+1} = x_i\ (2 - 2.834x_i)$

Initial guess= $x_0 = 0.4$

Iter 1:

[$x_0 = 0.4$

$x_1 = x_0\ (2 - 2.834x_0) = 0.34656$

approx. relative error: not defined as it is the first iteration]

Iter 2:

[$x_1 = 0.34656$

$x_2 = x_1\ (2 - 2.834x_1) = 0.35274$

approx. relative error: $(| x_2 - x_1 |)/x_2 = 0.0175$]

Iter 3:

[$x_2 = 0.35274$

$x_3 = x_2\ (2 - 2.834x_2) = 0.35285$

approx. relative error: $(| x_3 - x_2 |)/x_3 = 3.117 \times 10^{-4} < 0.001$

Taking the threshold as 0.1 % approx error, it is safe to stop the iteration here]

Therefore, the value of reciprocal of 2.834, calculated from Newton-Raphson method is 0.35285

b) r^{th}-root of N = solution of the equation f(x) = 0;

where $f(x) = x^r - N$

$f'(x) = rx^{r-1}$

Therefore, in Newton - Raphson method gives the iteration

$x_{i+1} = x_i - f(x) / f'(x) = (x_i^r\ (r - 1) + N) / rx^{r-1}$

$x_{i+1} = (x_i^r (r - 1) + N) / rx^{r-1}$ => This equation can be iterated until convergence to get the r^{th} root of any number N without using root - operator

In this case, N = 12.425, r = 2:

Newton - Raphson method: $x_{i+1} = (x_i^2 + 12.425) / 2x_i$

Initial guess= $x_0 = 3.5$

Iter 1:

[$x_0 = 3.5$

$x_1 = (x_0^2 + 12.425) / 2x_0 = 3.525$

approx. relative error: not defined as it is the first iteration]

Iter 2:

[$x_1 = 3.525$

$x_2 = (x_1^2 + 12.425) / 2x_1 = 3.52491$

approx. relative error: $(| x_2 - x_1 |)/x_2 = 2.5532 \times 10^{-5} < 0.0001$

Taking threshold as 0.01% approx error, it is safe to stop the iteration here, but since it is just the second iteration, it is better to do one more iteration to ensure the correct answer]

Iter 3:

[$x_2 = 3.52491$

$x_3 = (x_2^2 + 12.425) / 2x_2 = 3.52491$

approx. relative error: $(| x_3 - x_2 |)/x_3 = 0$ (considering only five decimal places)

Hence, 3.52491 can be taken as the value]

Therefore, the value of square root of 12.425, calculated from Newton-Raphson method is 3.52491

GOOGLE: In the early days of computing, the technique for finding 1/a described above was of great practical importance. Computers had addition, subtraction, and multiplication “hard-wired.” But division was not hard-wired, and had to be done by software. Note that x/y = x(1/y), so as multiplication is hard-wired, we can do division if we can find reciprocals. And Newton’s Method was used to do that.

QUESTION 4   Secant method (15 marks)

Using the Secant method, find the positive root of the following circle (i.e., the point where the circle intersects the x-axis)

$(x+1)^2 + (y-2)^2 = 16$

Comment on the following

• How can the negative root of the same circle be found?

• You have the freedom to choose the initial conditions. What initial conditions will you choose and what are the limits for choosing the same?

ANSWER 4

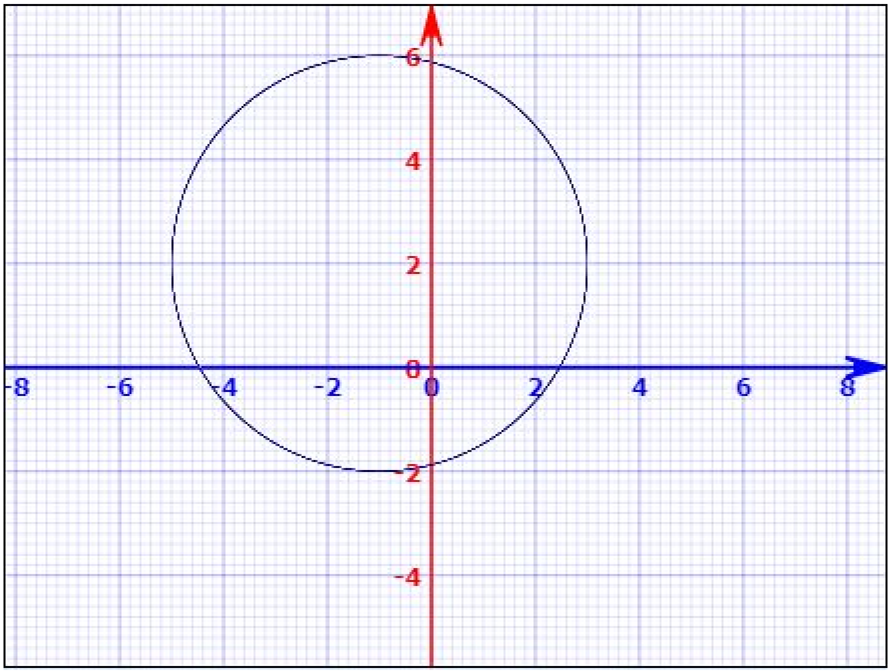
$(x+1)^2 + (y-2)^2 = 16 \Rightarrow$ **$y = 2 \pm \sqrt{(16 - (x+1)^2)}$**

We need to find x for which y = 0

Domain conditions: restriction is that $16 - (x+1)^2 >= 0$ because root is defined only for +ve real numbers

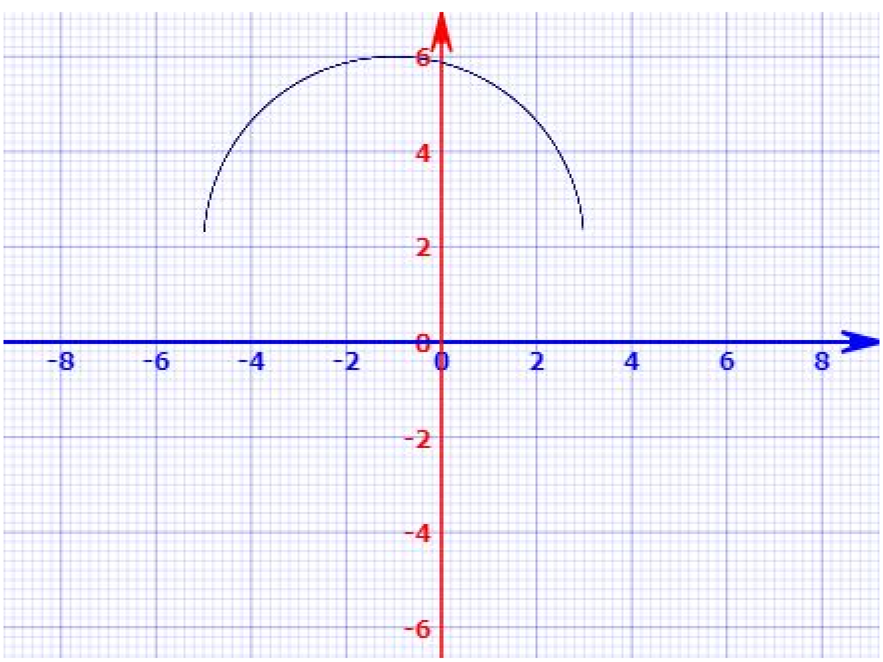
ie $-5 <= x <= 3$

The above restriction should be considered for all x, including initial guesses and x on each iteration



Case 1: Consider y = $2 + \sqrt{(16 - (x+1)^2)}$:

This corresponds to the upper half of the circle:



In this case, for any two initial guesses x_{-1} , x_0 in [-5,3], the resulting secant meets the x-axis at a point x_1 outside the domain [-5,3], hence we cannot find $f(x_1)$.

For example, I tried using initial guesses: we take $x_{-1} = 0.5$ and $x_0 = 3$;

Secant method : $x_{i+1} = x_i - (f(x_i)(x_{i-1} - x_i)) / (f(x_{i-1}) - f(x_i))$

Iter 1: [$x_{-1} = 0.5$ and $x_0 = 3$

$f(x_{-1}) = 5.7081$ and $f(x_0) = 2.0$

$x_1 = x_0 - (f(x_0)(x_{-1} - x_0)) / (f(x_{-1}) - f(x_0)) = 4.3484$ (outside the domain [-5,3])

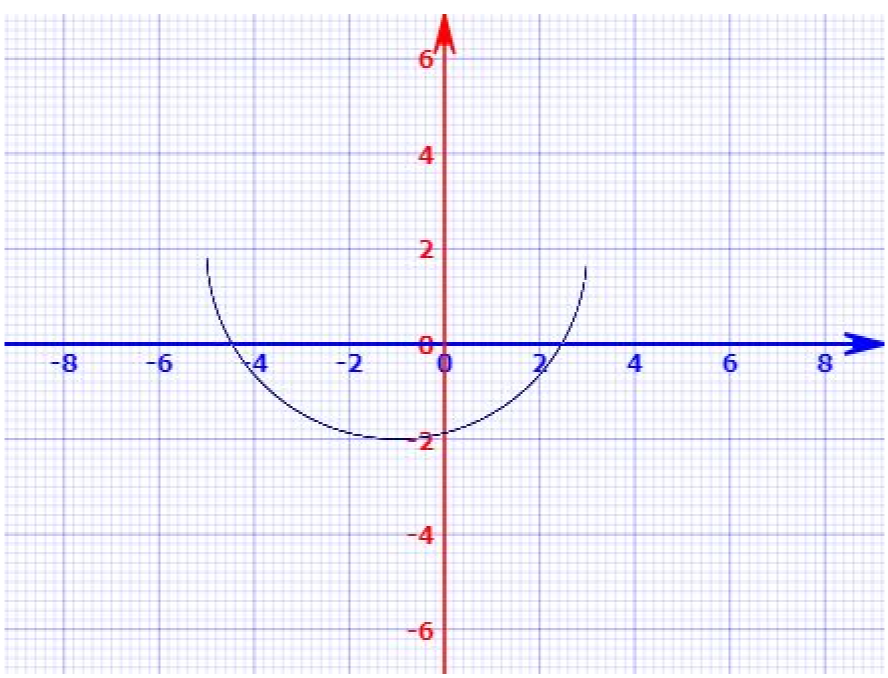
Therefore, $f(x_1)$ = not defined].

In this case, no set of initial guesses can yield the solution

Hence in this case, the solution cannot be obtained using Secant method

Case 2: Consider $y = f(x) = 2 - \sqrt{16 - (x+1)^2}$:

This corresponds to the lower half of the circle:



In this case, if we choose proper initial guesses, the secant can intersect at points on x axis in the domain of the function and thus, it is possible to find the solution

To find the positive root :

We need to use initial guesses such that it is closer to positive root than negative root. We consider initial guesses as $x_{-1} = 0.5$ and $x_0 = 3.0$

$f(x_{-1}) = -1.7081$ and $f(x_0) = 2.0$

Secant method : $x_{i+1} = x_i - (f(x_i)(x_{i-1} - x_i)) / (f(x_{i-1}) - f(x_i))$

Iter 1: [

$x_1 = x_0 - (f(x_0)(x_{-1} - x_0)) / (f(x_{-1}) - f(x_0)) = 1.6516$

$f(x_1) = -0.9948$

Approx relative error: not defined as its the first iteration]

Iter 2: [

$x_2 = x_1 - (f(x_1)(x_0 - x_1)) / (f(x_0) - f(x_1)) = 2.0995$

$f(x_2) = -0.5284$

Approx relative error:$(|x_2 - x_1| / x_2 = 0.42$]

Iter 3: [

$x_3 = x_2 - (f(x_2)(x_1 - x_2)) / (f(x_1) - f(x_2)) = 2.6070$

$f(x_3) = 0.2710$

Approx relative error:$(|x_3 - x_2| / x_3 = 0.194$]

Iter 4: [

$x_4 = x_3 - (f(x_3)(x_2 - x_3)) / (f(x_2) - f(x_3)) = 2.4349$

$f(x_4) = -0.0496$

Approx relative error:$(|x_4 - x_3| / x_4 = 0.070$]

Iter 5: [

$x_5 = x_4 - (f(x_4)(x_3 - x_4)) / (f(x_3) - f(x_4)) = 2.4615$

$f(x_5) = -0.0043$

Approx relative error: $(|x_5 - x_4| / x_5 = 0.0108$]

Iter 6: [

$x_6 = x_5 - (f(x_5)(x_4 - x_5)) / (f(x_4) - f(x_5)) = 2.4641$

$f(x_5) = 7.5 \times 10^{-5}$

Approx relative error: $(|x_6 - x_5| / x_6 = 0.0010 < 0.005$

Hence, % approx error is less than 0.5% and it is safe to stop the iteration]

Therefore, the positive value of the root is 2.4641

a) **Finding the negative root:**

It can be found in the same way as positive root in case 2 (lower half of circle). But, here also, care must be taken so that in each iteration x value is in the domain of the function. For that, we need to use proper initial guesses.

<u>Given below is JUST ONE set of plausible initial guesses :</u>

We know that the circle is symmetrical about x= -1. Also, we know that $x_{-1} = 0.5$, $x_0 = 3.0$ as initial guesses work successfully in finding positive root. Therefore, the symmetrical points on the other side of x = -1, which are - 2.5 and - 5.0 can be used as x_{-1} and x_0 respectively in finding negative root. In this case, the obtained secants are also symmetrical (wrt x = -1) and hence, domain issues will not occur before convergence.

Hence we can use $x_{-1} = -2.5$ and $x_0 = -5.0$ as initial guesses in finding negative roots

b) LIMITATIONS ON INITIAL GUESSES : I have given explanation for initial guesses for each case.

QUESTION 5 Fixed-point iteration method (15 marks)

We will use a modified fixed-point iteration procedure to find the roots of equation:

$G(x) = 2 - x + ln(x) = 0$

For fixed-point iteration of the form $x_{i+1} = g(x_i)$, let us define $g(x) = x + \beta G(x)$. Find the sufficient condition that fixed-point iteration with the above choice of $g(x)$ converges. Choose the initial guess for x that satisfies the sufficient condition and use fixed-point iteration until the approximate percent relative error is less than 1%.

ANSWER 5

$g(x) = x + \beta G(x) = x + \beta(2 - x + ln(x))$

Therefore, if we use $x_{i+1} = g(x_i)$, we get

$x_{i+1} = x_i + \beta(2 - x_i + ln(x_i))$

(NOTE: HOW DOES THIS MODIFICATION GIVE THE SAME ANSWER?

> When x converges, $x_{i+1} \fallingdotseq x_i$ and hence the equation becomes $2 - x_i + ln(x_i) = 0$ (if $\beta \neq 0$) , which was our original equation. Hence, using modified function g(x) for fixed-point iteration gives the correct value of x_i **)**

CONDITION FOR CONVERGENCE:

Condition for convergence: If g(x) is continuous and differentiable over an interval, then condition for convergence is $| g'(x) | < 1$ for that interval

$g(x) = x + \beta(2 - x + ln(x))$ is continuous and differentiable for all x>0 (ln(x) is defined only for x>0)

$g'(x) = 1 - \beta + \beta/x$

Therefore, condition for convergence in this case is $\mathbf{| 1 - \beta + \beta/x | < 1}$

<u>**One possibility is $\beta = 1$ and x>1 :**</u>

In this case, g(x) = 2 + ln (x)

Therefore, Fixed point iteration method : $x_{i+1} = 2 + \ln(x_i)$

We choose our initial guess as $x_0 = 2$ (This satisfies the condition $x>1$. Also, in further iterations value of x increases, hence $x > 1$ for all iterations)

Iter 1 : [$x_1 = 2 + \ln(x_0) = 2 + \ln(2) = 2.6931$

Approx relative error: not defined (first iteration)]

Iter 2 : [$x_2 = 2 + \ln(x_1) = 2 + \ln(2.6931) = 2.9907$

Approx relative error : $| x_2 - x_1 | / x_2 = 0.099$]

Iter 3 : [$x_3 = 2 + \ln(x_2) = 2 + \ln(2.9907) = 3.0955$

Approx relative error : $| x_3 - x_2 | / x_3 = 0.034$]

Iter 4 : [$x_4 = 2 + \ln(x_3) = 2 + \ln(3.0955) = 3.1299$

Approx relative error : $| x_4 - x_3 | / x_4 = 0.1099$]

Iter 5 : [$x_5 = 2 + \ln(x_4) = 2 + \ln(3.1299) = 3.1410$

Approx relative error : $| x_5 - x_4 | / x_5 = 3.533 \times 10^{-3} < 1\%$]

Therefore, the value of x = 3.1410

QUESTION 6 Multiple roots (10 marks)

The function $x^3 - 2x^2 - 4x + 8$ has a double root at $x = 2$. Use (a) the standard Newton-Raphson, (b) the modified Newton-Raphson (method 1 from class notes) and (c) the modified Newton- Raphson (method 2 from class notes) to solve for the root at $x = 2$. Compare and discuss the rate of convergence using an initial guess of $x_0 = 1.2$.

ANSWER 6

$f(x) = x^3 - 2x^2 - 4x + 8$

$f'(x) = 3x^2 - 4x - 4$

$f''(x) = 6x - 4$

True solution: $x = 2$

a) Standard Newton - Raphson method:

$x_{i+1} = x_i - f(x_i) / f'(x_i)$

Initial guess : $x_0 = 1.2$

Iter 1 [$x_1 = x_0 - f(x_0) / f'(x_0) = 1.6571$

true relative error = $| 2 - x_1 |/2 = 0.1714$]

Iter 2 [$x_2 = x_1 - f(x_1) / f'(x_1) = 1.8369$

true relative error = $| 2 - x_2 |/2 = 0.0816$]

Iter 3 [$x_3 = x_2 - f(x_2) / f'(x_2) = 1.9202$

true relative error = $| 2 - x_3 |/2 = 0.0399$]

Iter 4 [$x_4 = x_3 - f(x_3) / f'(x_3) = 1.9605$

true relative error = $| 2 - x_4 |/2 = 0.0198$]

Iter 5 [$x_5 = x_4 - f(x_4) / f'(x_4) = 1.9803$

true relative error = | $2 - x_5$ |/2 = 9.85×10^{-3}]

Iter 6 [$x_6 = x_5 - f(x_5) / f'(x_5) = 1.9901$

true relative error = | $2 - x_6$ |/2 = 4.95×10^{-3}]

x = 1.9901 (after 6 iterations)

b) modified Newton - Raphson method 1:

$x_{i+1} = x_i - m\,(f(x_i) / f'(x_i))$; m=2 for double root cases

Initial guess : $x_0 = 1.2$

Iter 1 [$x_1 = x_0 - 2f(x_0) / f'(x_0) = 2.1142$

true relative error = | $2 - x_1$ |/2 = 0.0571]

Iter 2 [$x_2 = x_1 - 2f(x_1) / f'(x_1) = 2.0015$

true relative error = | $2 - x_2$ |/2 = 7.5×10^{-4}]

Iter 3 [$x_3 = x_2 - 2f(x_2) / f'(x_2) = 2.0000$

true relative error = | $2 - x_0$ |/2 = 0.0000]

x = 2.0000 after 3 iterations

c) modified Newton - Raphson method 2:

$x_{i+1} = x_i - [\ f(x_i)f'(x_i)\ ] / [\ f'(x_i)^2 - f(x_i)\, f''(x_i)\ ]$

Initial guess : $x_0 = 1.2$

Iter 1 [$x_1 = x_0 - [\ f(x_0)f'(x_0)\ ] / [\ f'(x_0)^2 - f(x_0)\, f''(x_0)\ ] = 1.8788$

true relative error = | $2 - x_1$ |/2 = 0.0606]

Iter 2 [$x_2 = x_1 - [\ f(x_1)f'(x_1)\ ] / [\ f'(x_1)^2 - f(x_1)\, f''(x_1)\ ] = 1.9980$

true relative error = | $2 - x_2$ | / 2 = 1×10^{-3}]

Iter 3 [$x_3 = x_2 - [\ f(x_2)f'(x_2)\ ] / [\ f'(x_2)^2 - f(x_2)\, f''(x_2)\ ] = 2.0000$

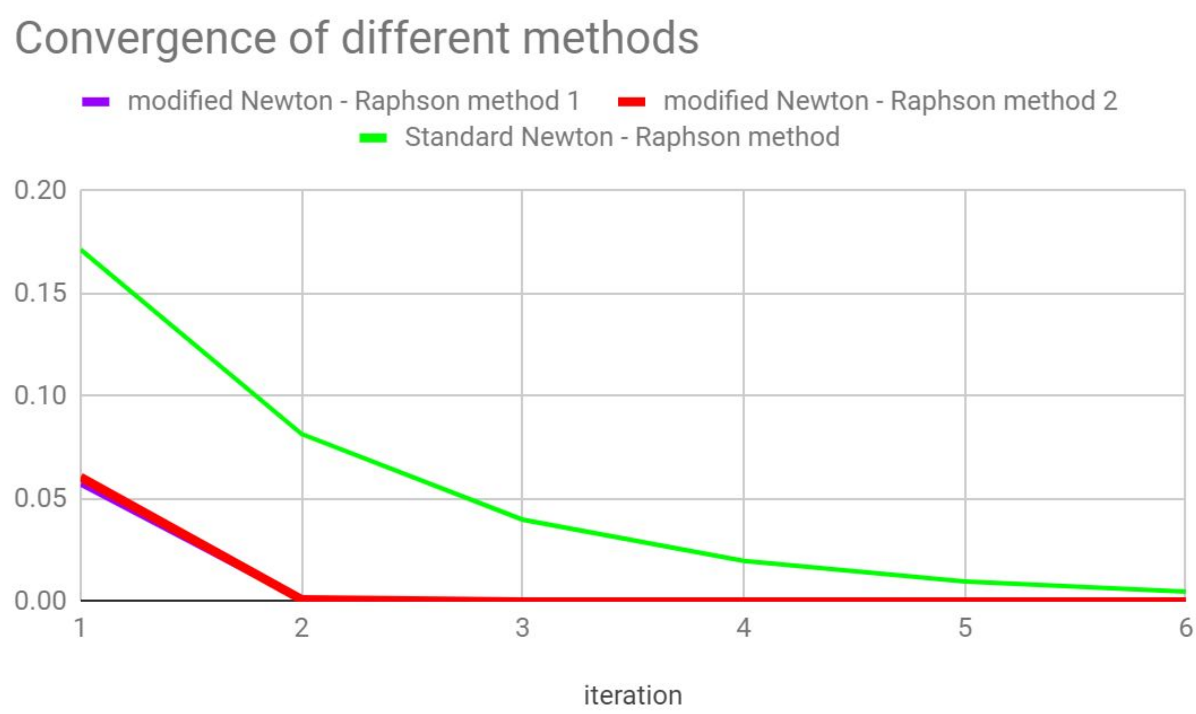
true relative error = | 2 - x_0 |/2 = 0.0000]

x = 2.0000 (after 3 iterations)

Comparison of all three methods - true relative errors:

TRUE RELATIVE ERRORS

iteration	Standard Newton - Raphson method	modified Newton - Raphson method 1	modified Newton - Raphson method 2
1	0.1714	0.0571	0.0606
2	0.0816	7.5×10^{-4}	1×10^{-3}
3	0.0399	0.0000	0.0000
4	0.0198	0.0000	0.0000
5	9.85×10^{-3}	0.0000	0.0000
6	4.95×10^{-3}	0.0000	0.0000



As expected, standard Newton-Raphson method(linear convergence) is slower when compared to the other two(quadratic convergence).

QUESTION 7 System of non-linear equations (10 marks)

Determine the roots of the following simultaneous nonlinear equations using (a) fixed-point iteration and (b) the Newton-Raphson method:

$y = -x^2 + x + 0.75$

$y + 5xy = x^2$

Employ initial guesses of $x = y = 1.2$ and discuss the results

ANSWER 7

a) Modified Fixed point iteration method:

$y = -x^2 + x + 0.75$

$y + 5xy = x^2$

Take $u(x,y) = x_i^2 / (1+ 5x_i)$ and $v(x,y) = \sqrt{(x_i + 0.75 - y_i)}$

(This does not satisfy the condition $|\partial u/\partial x| + |\partial u/\partial y| <1$, $|\partial v/\partial x| + |\partial v/\partial y| <1$, but that is a sufficient condition, not a necessary condition, so still there is hope of convergence)

Take $y_{i+1} = x_i^2 / (1+ 5x_i)$

Take $x_{i+1} = \sqrt{(x_i + 0.75 - y_i)}$

Approx relative error(x) = $(x_{i+1} - x_i)/x_{i+1}$

Approx relative error(y) = $(y_{i+1} - y_i)/y_{i+1}$

Initial guess: $x_0 = 1.2$, $y_0 = 1.2$

Iter 1 [

$x_1 = 0.8660$

$y_1 = 0.1407$

Approximate relative error = not defined]

Iter 2 [

$x_2 = 1.2146$

$y_2 = 0.2085$

Approximate relative error(x) = 0.287

Approximate relative error(y) = 0.325]

Iter 3 [

$x_3 = 1.3251$

$y_3 = 0.2303$

Approximate relative error(x) = 0.083

Approximate relative error(y) = 0.094]

Iter 4 [

$x_4 = 1.3582$

$y_4 = 0.2367$

Approximate relative error(x) = 0.0244

Approximate relative error(y) = 0.027]

Iter 5 [

$x_5 = 1.3680$

$y_5 = 0.2387$

Approximate relative error(x) = $7.16 \times 10^{-3} < 1\%$

Approximate relative error(y) = $8.37 \times 10^{-3} < 1\%$, hence iteration is stopped]

x = 1.3680

$y = 0.2387$

b) Modified Newton-Raphson method:

$u(x,y) = -x^2 + x + 0.75 - y$

$v(x,y) = y + 5xy - x^2$

$\partial u/\partial x = -2x + 1$, $\partial u/\partial y = -1$

$\partial v/\partial x = -2x + 5y$, $\partial v/\partial y = 1 + 5x$

Modified Newton - Raphson method:

$x_{i+1} = x_i - [\, u_i (\partial v_i/\partial y) - v_i (\partial u_i/\partial y)\,] / [\, (\partial u_i/\partial x)(\partial v_i/\partial y) - (\partial u_i/\partial y)(\partial v_i/\partial x)\,]$

$y_{i+1} = y_i - [\, v_i (\partial u_i/\partial x) - u_i (\partial v_i/\partial x)\,] / [\, (\partial u_i/\partial x)(\partial v_i/\partial y) - (\partial u_i/\partial y)(\partial v_i/\partial x)\,]$

Approx relative error(x) = $(x_{i+1} - x_i)/x_{i+1}$

Approx relative error(y) = $(y_{i+1} - y_i)/y_{i+1}$

Initial guesses: $x_0 = 1.2$, $y_0 = 1.2$

Iter 1 [

$x_1 = 1.5435$

$y_1 = 0.0290$

Approx relative error = not defined]

Iter 2 [

$x_2 = 1.3941$

$y_2 = 0.2228$

Approx relative error (x) = 0.1072

Approx relative error (y) = 0.8698]

Iter 2 [

$x_2 = 1.3724$

$y_2 = 0.2393$

Approx relative error (x) = 0.0158

Approx relative error (y) = 0.0689]

Iter 3 [

$x_3 = 1.3720$

$y_3 = 0.2395$

Approx relative error (x) = $2.91 \times 10^{-4} < 0.1\%$

Approx relative error (y) = $8.35 \times 10^{-4} < 0.1\%$]

Hence stop iteration.

x = 1.3720

y = 0.2395

Therefore, comparing (a) and (b), we can say that Modified Newton-Raphson method converges faster than Modified fixed point iteration method in this case. Also, Modified fixed point iteration method is very restrictive and it is difficult to decide which combination to consider while solving.

QUESTION 8 Mueller’s method (10 marks)

A two-dimensional circular cylinder is placed in a high-speed uniform flow. Vortices shed from the cylinder at a constant frequency, and pressure sensors on the rear surface of the cylinder detect this frequency by calculating how often the pressure oscillates. Given three data points, use Müller’s method to find the time where the pressure was zero. Hint: You can fit a polynomial through the given data set

Time	0.6	0.62	0.64
Pressure	20	50	60

ANSWER 8

[Here, we take x = time and f(x) = Pressure.]

Assume the parabola is f(x) (We don't know the exact expression of parabola yet).

To solve f(x), we have 3 points(x) and their values f(x).

We take these 3 points as our initial guesses x_0 , x_1, x_2 in Mullers method

Therefore, we have,

$x_0 = 0.6$, $x_1 = 0.62$, $x_2 = 0.64$

$f(x_0) = 20$, $f(x_1) = 50$, $f(x_2) = 60$

Fitting a quadratic equation/parabola to the three points:

$g(x) = a(x - x_2)^2 + b(x - x_2) + c$

Iter 1:

($x_0 = 0.6$, $x_1 = 0.62$, $x_2 = 0.64$

$f(x_0) = 20$, $f(x_1) = 50$, $f(x_2) = 60$)

$h_0 = x_1 - x_0 \quad = 0.02$

$h_1 = x_2 - x_1 \quad = 0.02$

$\delta_0 = (f(x_1) - f(x_0)) / (x_1 - x_0) = 1500$

$\delta_1 = (f(x_2) - f(x_1)) / (x_2 - x_1) = 500$

$a = (\delta_1 - \delta_0) / (h_1 + h_0) = - 25000$

$b = ah_1 + \delta_1 = 0$

$c = f(x_2) = 60$

$\Rightarrow f(x) = a(x- x_2)^2 + b(x-x_2) + c$ --> (1)

$x_3 - x_2 = -2c / (b \pm \sqrt{(b^2 - 4ac)}) = \pm\ 0.04899$

$x_3 = x_2 \pm 0.04899$

$x_3 = 0.68899$ or $x_3 = 0.59101$ (Since both are equally close to x_2, we need to take both of them). However, in the question, we are asked to find the time when Pressure **was** zero. Therefore, we consider $x < x_1, x_2, x_3$, which is 0.59101

Substituting in (1):

$f(x_3) = - 5.025 \times 10^{-4}$

Iter 2: (Just for confirmation):

$x_0 = 0.62$, $x_1 = 0.64$, $x_2 = 0.59101$

$f(x_0) = 50$, $f(x_1) = 60$, $f(x_2) = - 5.025 \times 10^{-4}$

$h_0 = x_1 - x_0 \quad = 0.02$

$h_1 = x_2 - x_1 \quad = - 0.04899$

$\delta_0 = (f(x_1) - f(x_0)) / (x_1 - x_0) = 500$

$\delta_1 = (f(x_2) - f(x_1)) / (x_2 - x_1) = 1224.75$

$a = (\delta_1 - \delta_0) / (h_1 + h_0) = - 25000$

$b = ah_1 + \delta_1 = 2449.5$

$c = f(x_2) = - 5.025 \times 10^{-4}$

=> $f(x) = a(x- x_2)^2 + b(x-x_2) + c$ --> (1)

$x_3 - x_2 = -2c / (b \pm \sqrt{(b^2 - 4ac)})$ is less than 10^{-4}, ie very low , Hence previous value was correct (New value wont make much difference, hence I am using the value from iteration 1 as the final answer)

x = 0.59101

ie time at which Pressure “was” 0 = 0.59101