Practical Numerical Training UKNum

root finding: “Nullstellenbestimmung”

Klahr & Bitsch

Max Planck Institute for Astronomy, Heidelberg

Program:

1) Introduction
2) Bisektion
3) Newton-Raphson
4) Sekanten
5) Regula falsi

1 Introduction

Task

- One of the most important tools in numerics:
- Solving arbitrary equations. Rewrite as:

$$f(x) = 0$$

- There for called Root Finding: finding the 0. (German: “Nullstelle”)

- 1 independent variable, 1 equation: 1 dimensional problem.

- N independent variables, N equations: multidimensional problem

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}$$

- Multi dimensional? Too complicated for lecture. Therefore 1D.

Dispersion relation: Stability analysis

$$u = e^{-i(\omega t - k_R R - k_z z)}$$

$$\begin{aligned}
&\omega^5 + i\frac{\omega^4}{\tau} \\
&-\omega^3 \left(c^2 \left(a_R b_R + a_z b_z + \gamma \left(k_R^2 + k_z^2\right)\right) + \kappa_R^2\right) \\
&-i\frac{\omega^2}{\tau} \\
&\quad \left(\kappa_R^2 + c^2 (k_R - i a_R)(k_R + i b_R) + (k_z - i a_z)(k_z + i b_z)))\right) \\
&+c^2\omega \\
&\quad \left[c^2 \left(b_z k_R - b_R k_z\right)\left(-k_z \left(b_R - a_R\gamma\right) + k_R \left(b_z - a_z\gamma\right)\right)\right. \\
&\quad \left. +(a_z b_z + k_z^2\gamma)\left(\kappa_R^2 - \frac{b_z a_R + k_z k_R \gamma}{a_z b_z + k_z^2\gamma}\kappa_z^2\right)\right] \\
&+\frac{i}{\tau}c^2 \left(k_z + i b_z\right)\left(\left(k_z - i a_z\right)\kappa_R^2 - \left(k_R - i a_R\right)\kappa_z^2\right) = 0
\end{aligned} \tag{55}$$

Fundamental Idea:

- iterative solution:
  - Starting from arbitrary guess for a solution, one modifies x step by step following an algorithm until |f(x)| is smaller than a desired value (criterion for convergence) … (hopefully…).

- Several different algorithms can be used:
  - Bisektion
  - Regula Falsi
  - Secants
  - Newton-Raphson
  - Or combination methods (e.g. Brent's Method)

- Each method comes with different pros and cons.

Warning

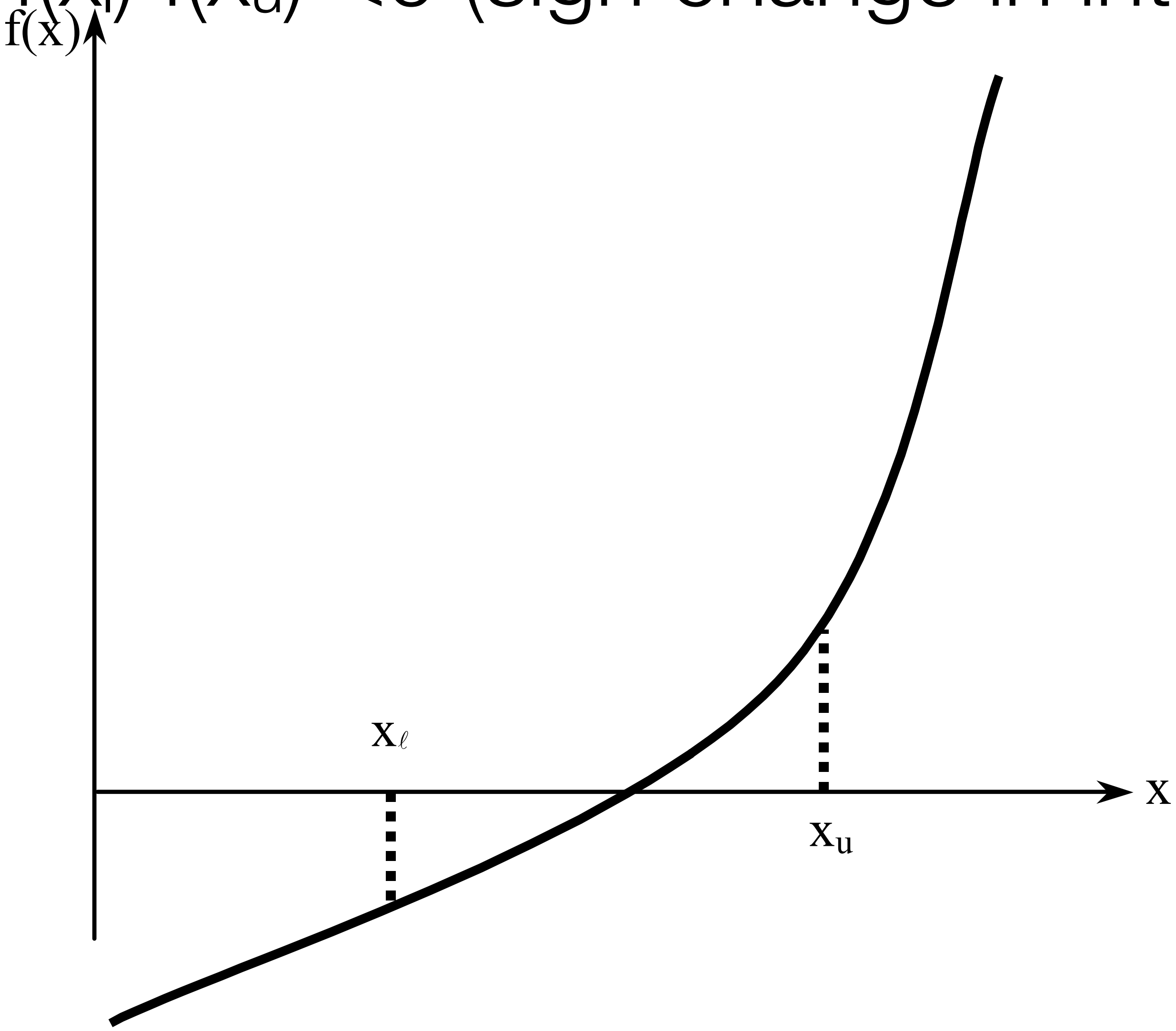
- One can either optimise for
  - “Safety” (to reliably find the correct root) or
  - “Speed” (necessary number of Iterations).

- Under certain condition some algorithms can fail!
  - Example will come shortly…

- Whenever possible:
  - Good estimate for initial x (resp. intervall)
  - And analyse the behaviour of f(x): f.i. plot the funtion

theorem *(Intermediate value theorem)*

- "Nullstellensatz" of Bolzano: Bolzano's theorem
- If a continuous function f(x) has values of opposite sign inside the interval x_l und x_u, then it has (at least) one root in that interval
- Condition: $f(x_l)\ f(x_u) < 0$ (sign change in intervall).

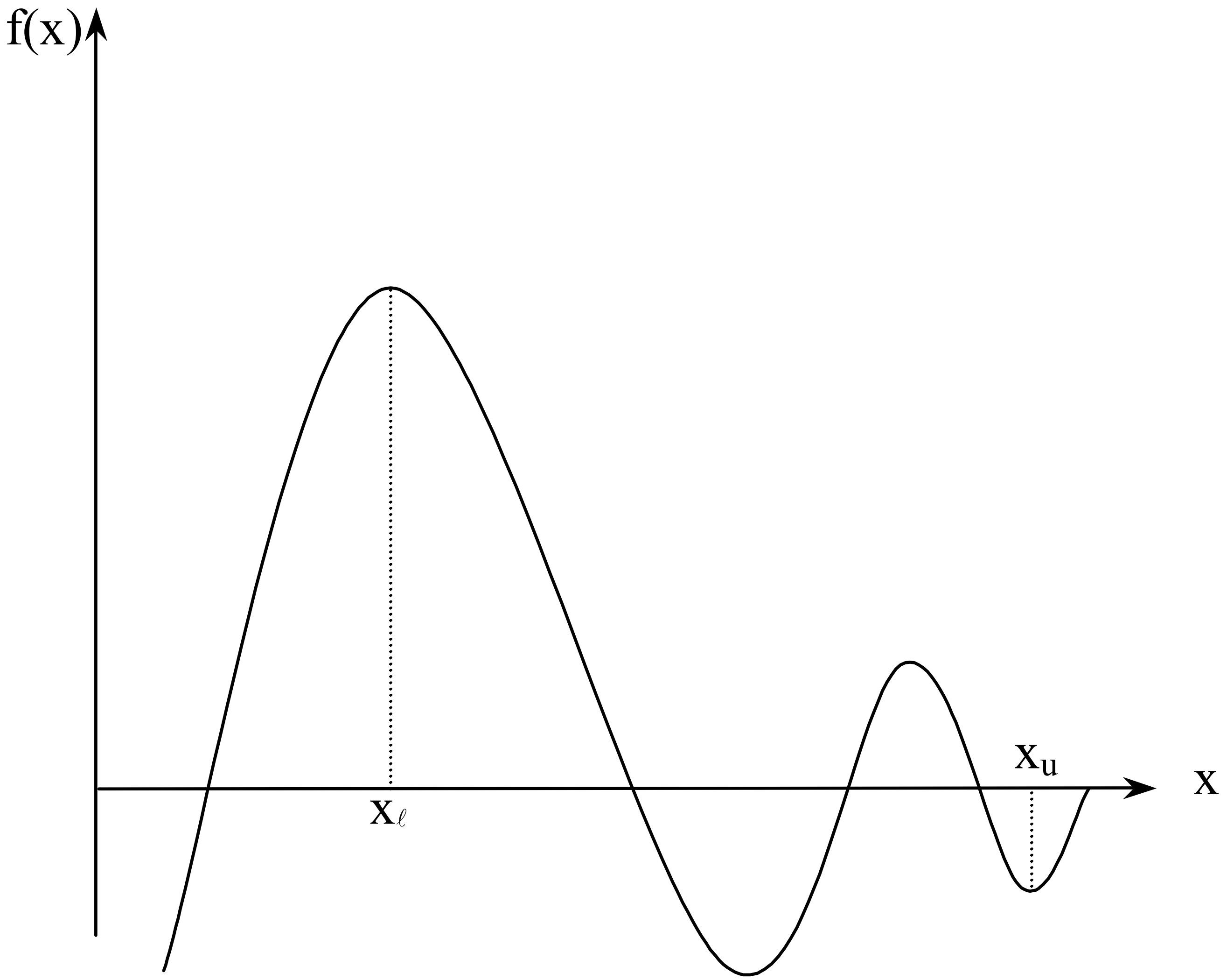


Bracketing the root

- If $f(x_l)\ f(x_u) <0$ then x_l and x_u “bracket” the root.
- First step in root finding is therefor: to find a x_l and x_u .
- Already this step is not necessarily trivial.

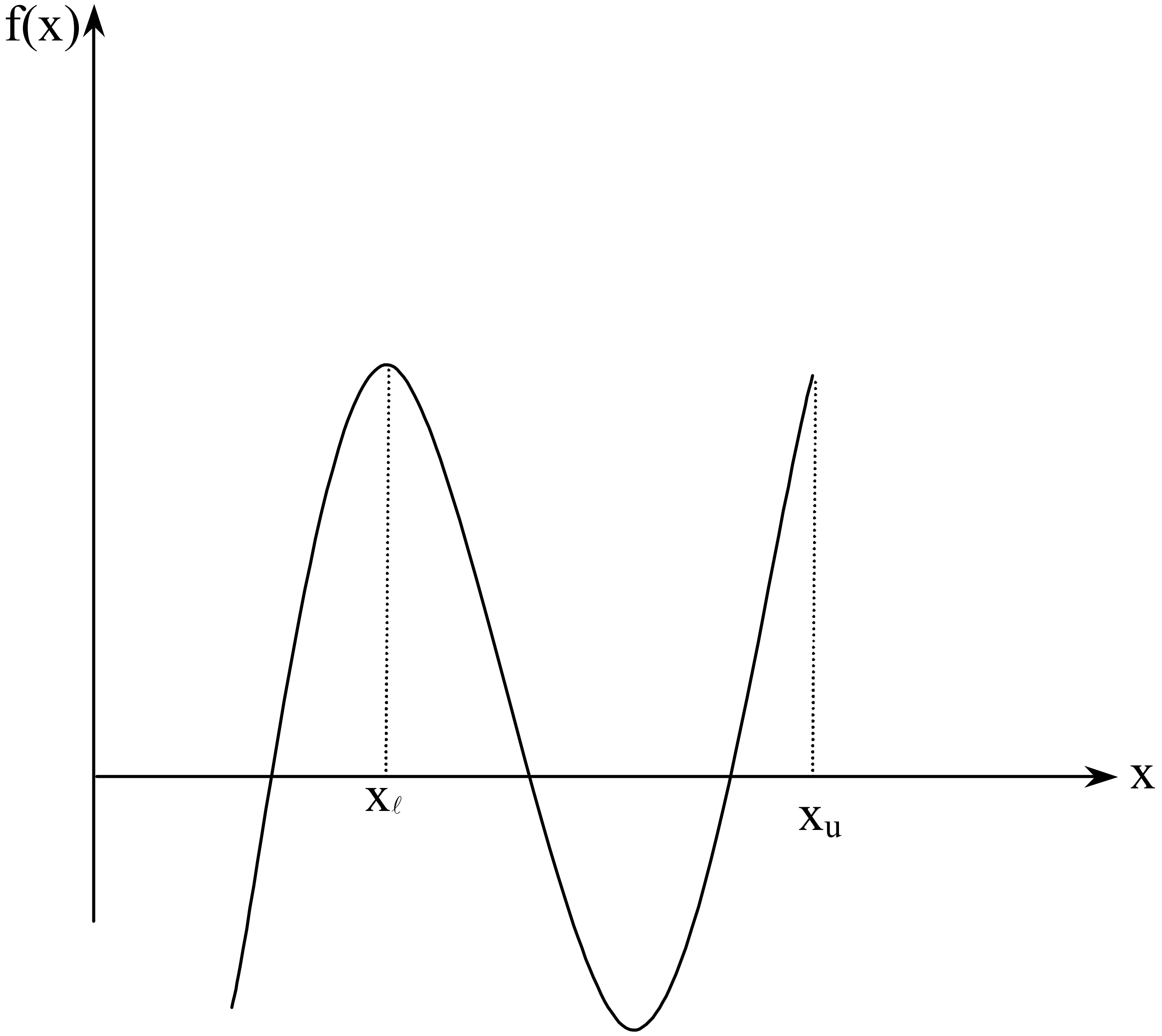
Bracketing the root: Problems I

- "…at least one root…": multiple roots possible.
- Not easy to predict which root will be found.
- Depends f.i. on initial guess.



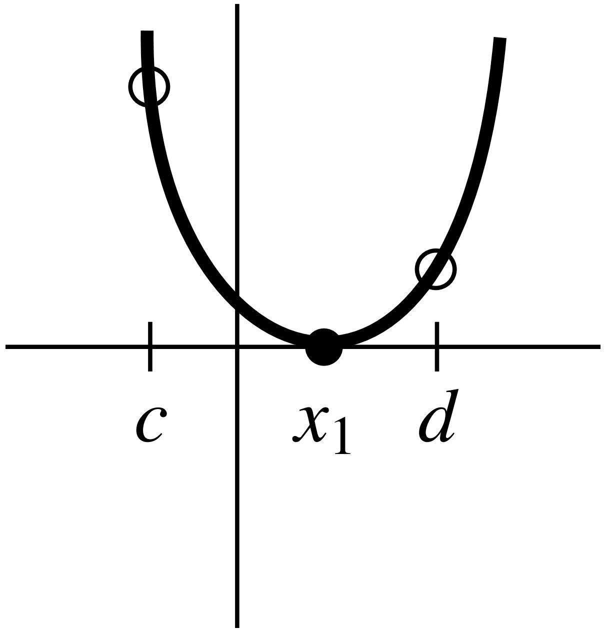
Bracketing the root: Problems II

- Double root: No change of sign.
- Condition: $f(x_l)\ f(x_u) < 0$ not fulfilled.



Bracketing the root: Problems III

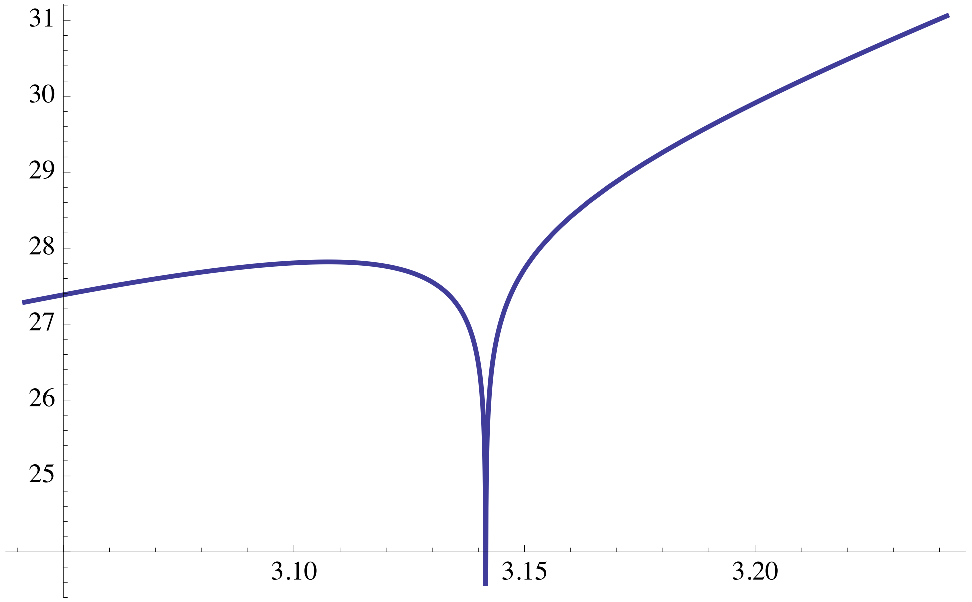
- Roots only for 1 point: Condition: f(c) f(d) <0 not fulfilled.



- Or in a very small interval

$$f(x) = 3x^2 + \frac{1}{\pi^4}\ln\left[(\pi - x)^2\right] + 1$$

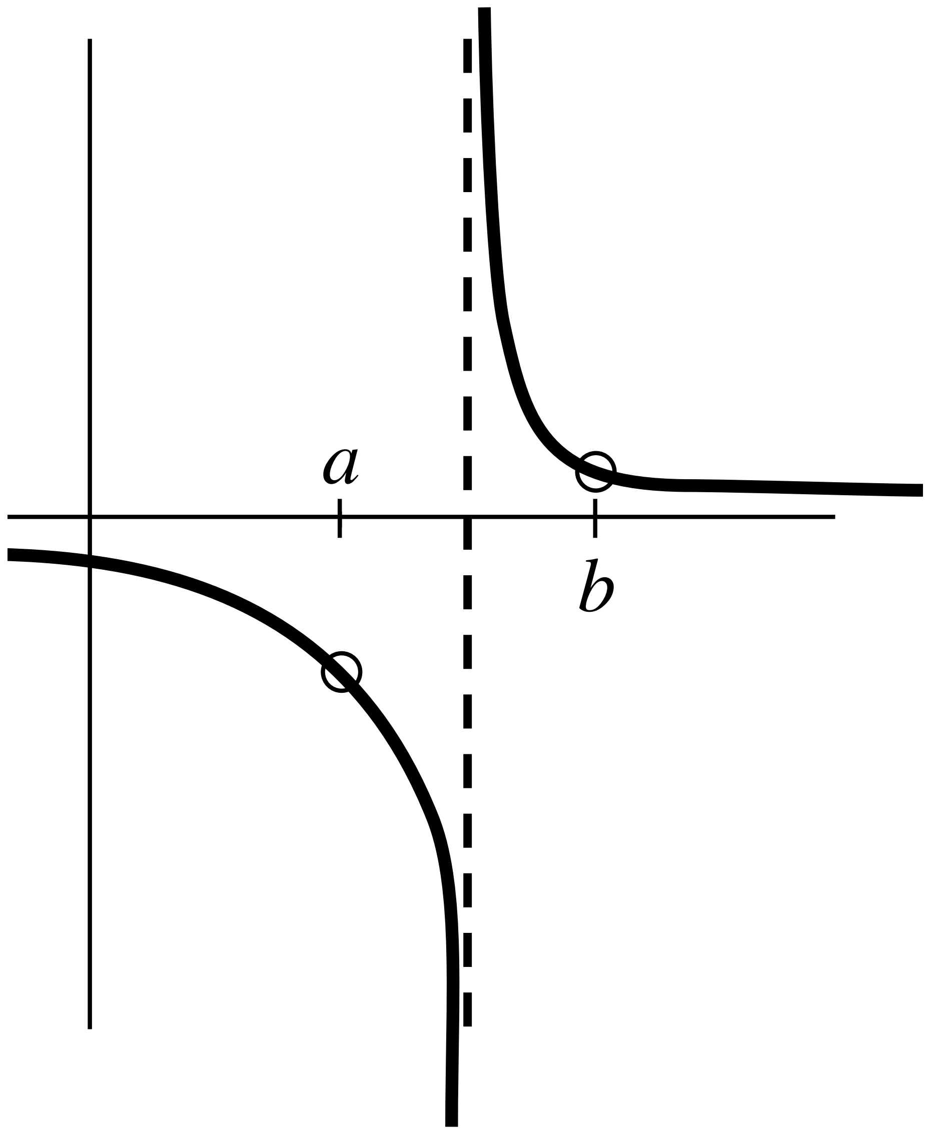
f(x)<0 only for π±10^{-667} (!)



- Impossible to determine good bracketing values.

Bracketing the root: Problems IV

- *Function not continuous*: Change of sign, but no root:

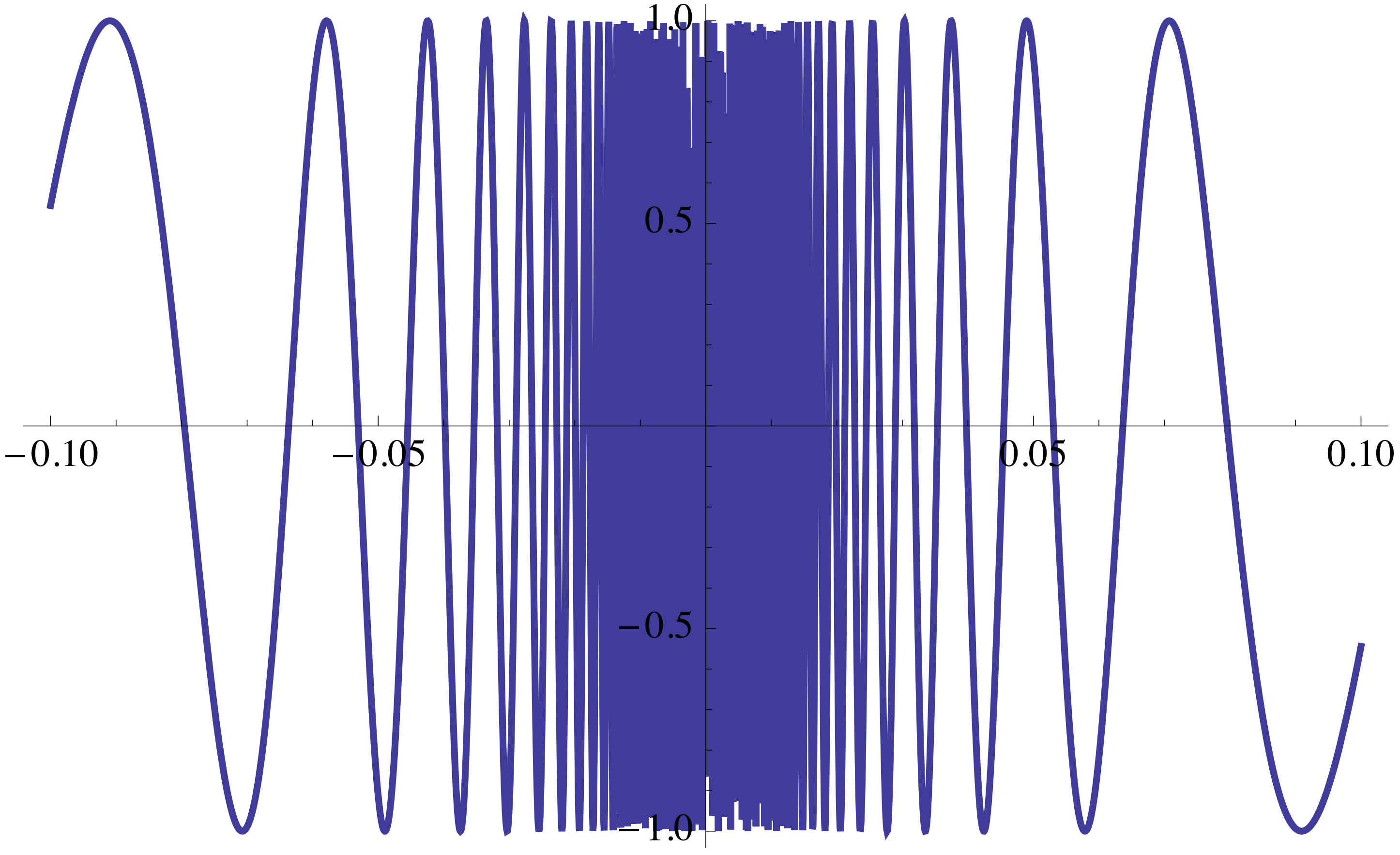


$$f(x) = \frac{1}{x-c}$$

Bisektion would converge to x = c as “root”.

Bracketing the root: Problems V

- Some more pathologic cases:

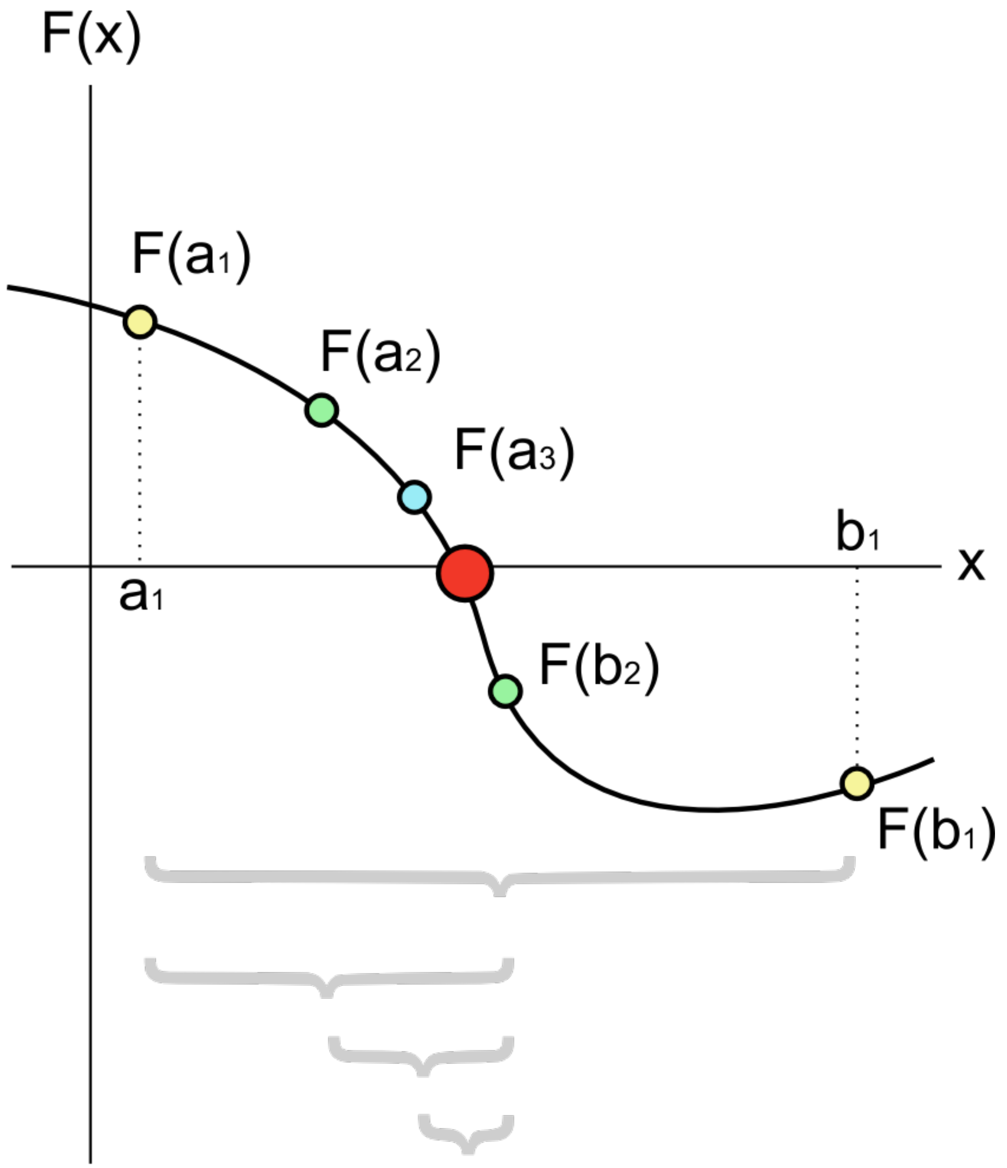


$$f(x) = \sin(1/x)$$

2 Bisektion

Algorithm

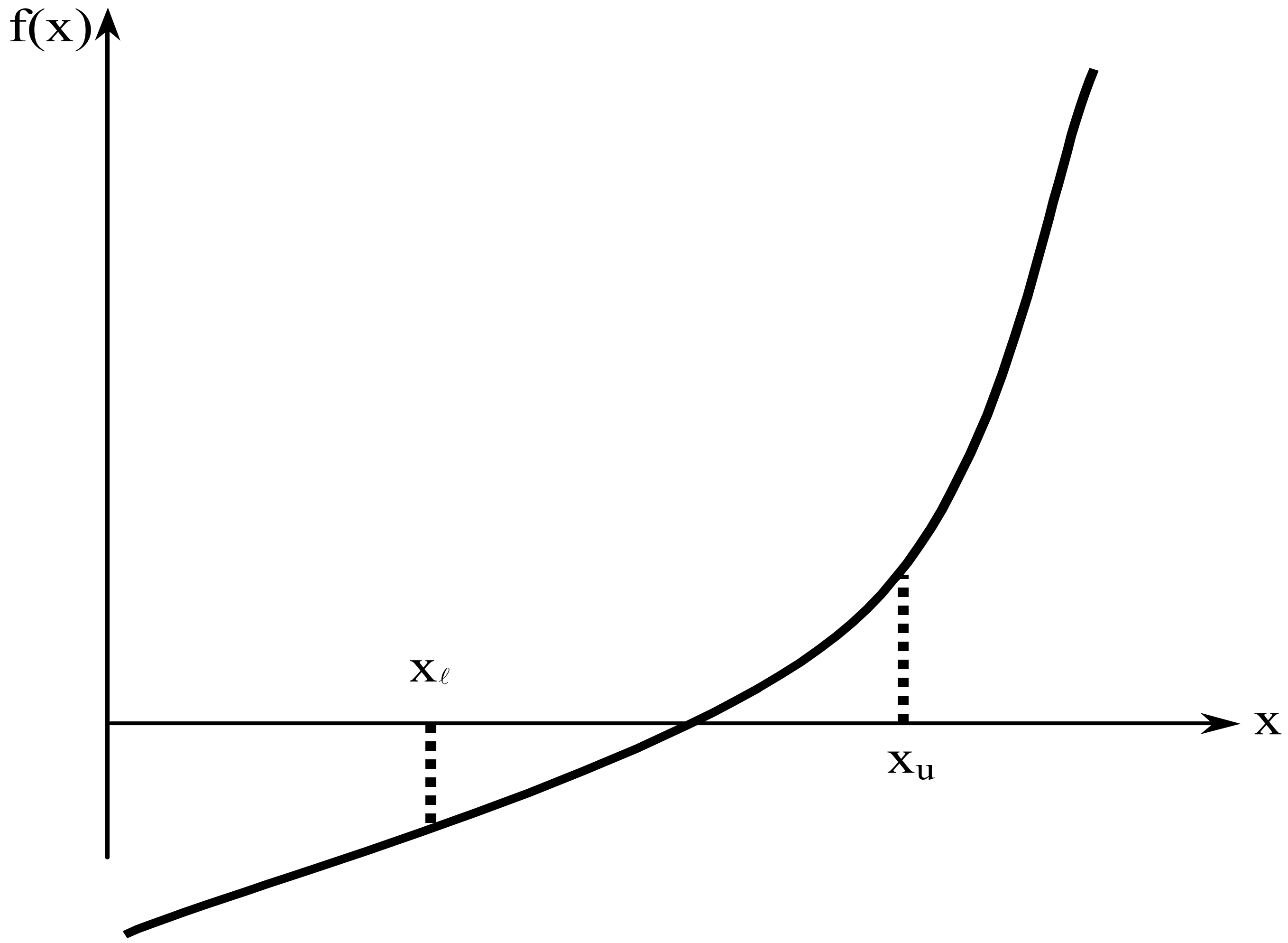
- Bisektion=”cutting in two”
- Algorithm (schematic)



- Most primitiv method (slow, but very stable).

Step 1: Bracketing

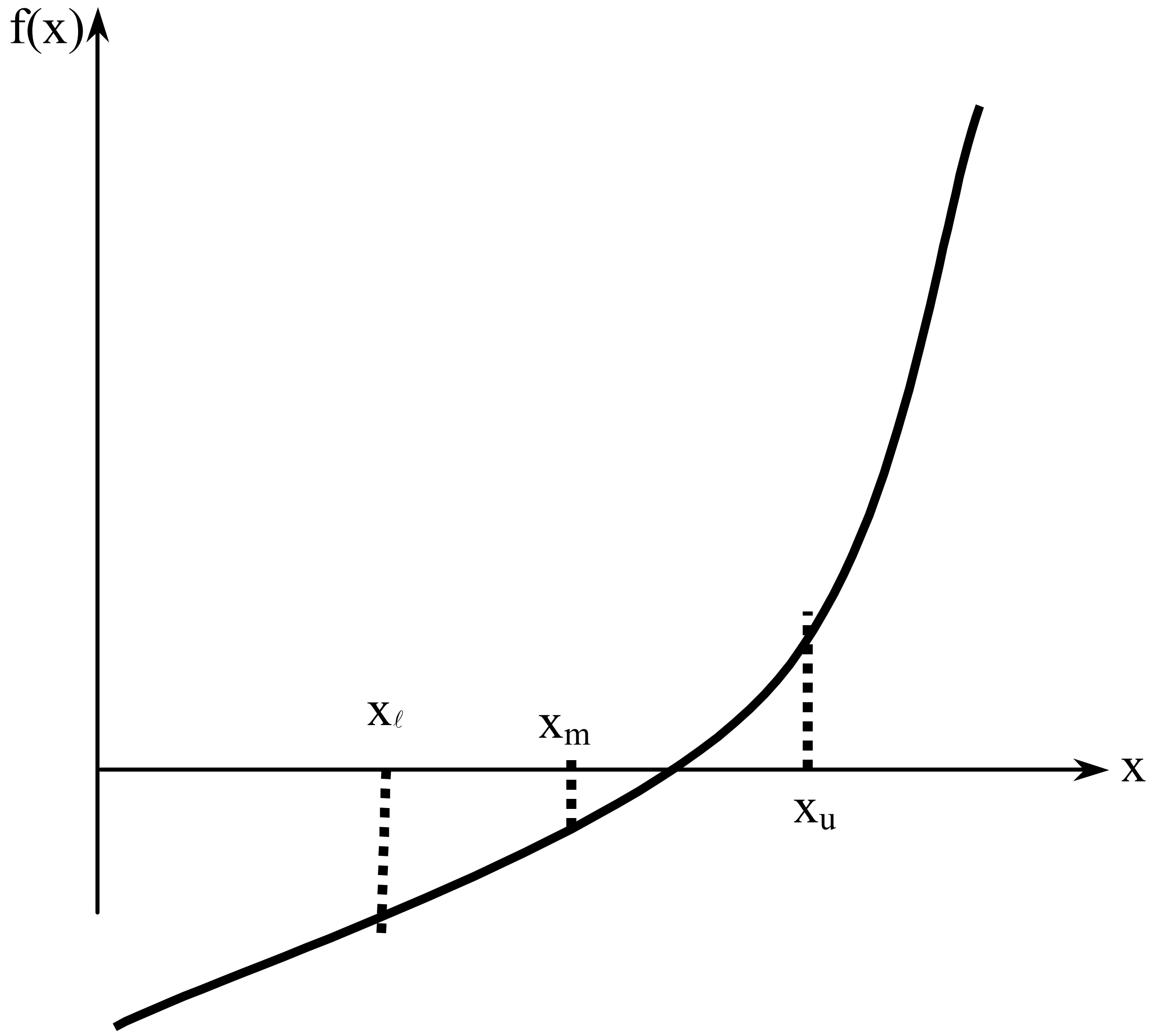
- Choose x_l and x_u that $f(x_l)\ f(x_u) < 0$. (Bracketing)
- If $x_l < x_u$.



Step 2: Middle point

- Guess: f(x) = 0 be at x_m half way between x_l and x_u.

$$x_m = \frac{x_\ell + x_u}{2}$$



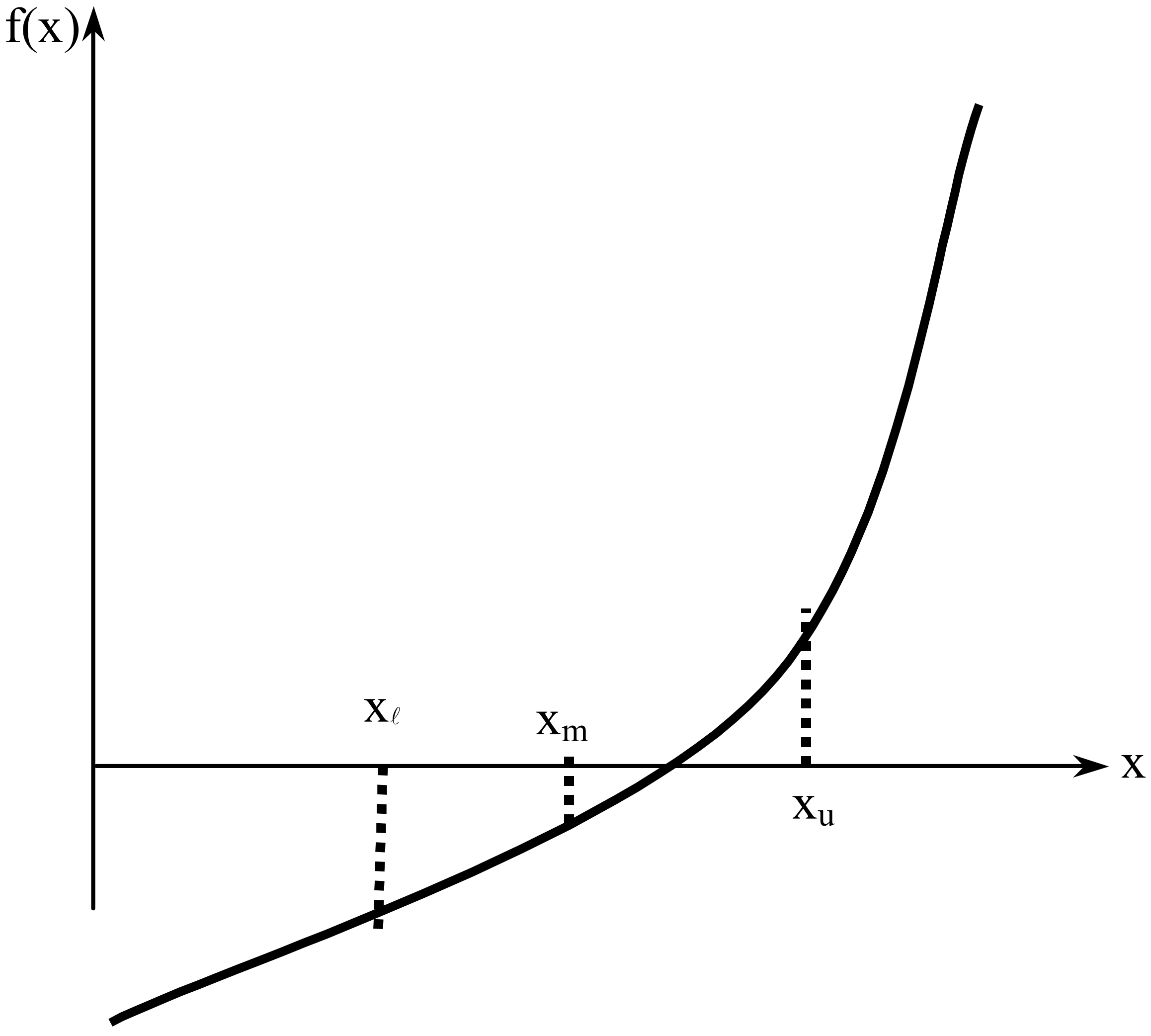
Step 3: divide interval

- Test:

a) If $f(x_l)f(x_m) < 0$, then root between x_l und x_m;
New $x_l = x_l$; $x_u = x_m$.

b) If $f(x_l)f(x_m) > 0$, the root between x_m and x_u;
Set $x_l = x_m$; $x_u = x_u$.

c) If $f(x_l)f(x_m) = 0$; then x_m is the root. Stop the algorithm.



Case b

Step 4

Find the new estimate of the root

$$x_m = \frac{x_\ell + x_u}{2}$$

Find the absolute relative approximate error

$$\left|\in_a\right| = \left|\frac{x_m^{new} - x_m^{old}}{x_m^{new}}\right| \times 100$$

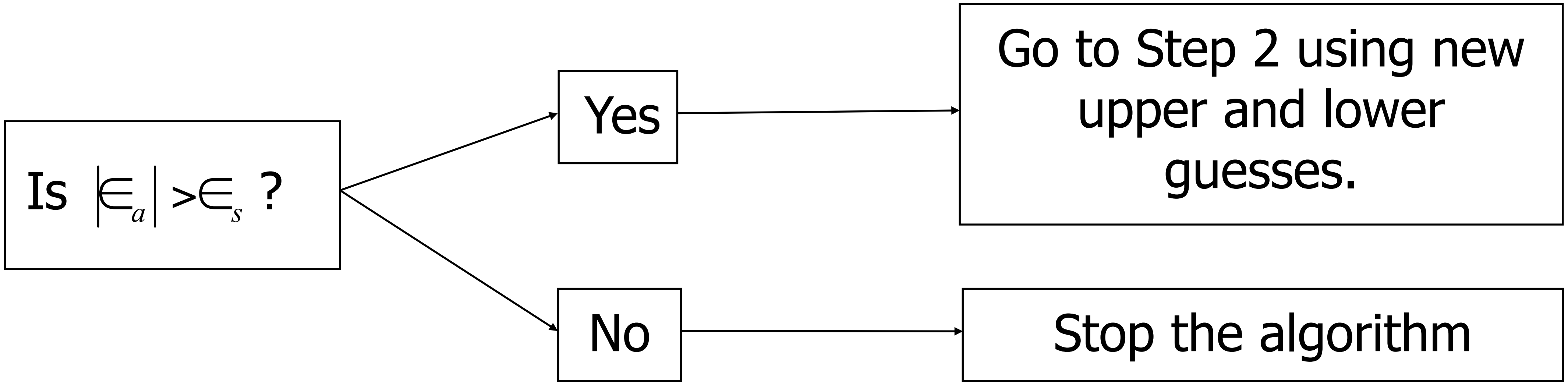
where

$$x_m^{old} = \text{previous estimate of root}$$

$$x_m^{new} = \text{current estimate of root}$$

Step 5

Compare the absolute relative approximate error $|\in_a|$ with the pre-specified error tolerance $\in_s$.



Is $|\in_a| > \in_s$?

Yes

Go to Step 2 using new upper and lower guesses.

No

Stop the algorithm

Note one should also check whether the number of iterations is more than the maximum number of iterations allowed. If so, one needs to terminate the algorithm and notify the user about it.

Example 1

You are working for 'DOWN THE TOILET COMPANY' that makes floats for ABC commodes. The floating ball has a specific gravity of 0.6 and has a radius of 5.5 cm. You are asked to find the depth to which the ball is submerged when floating in water.

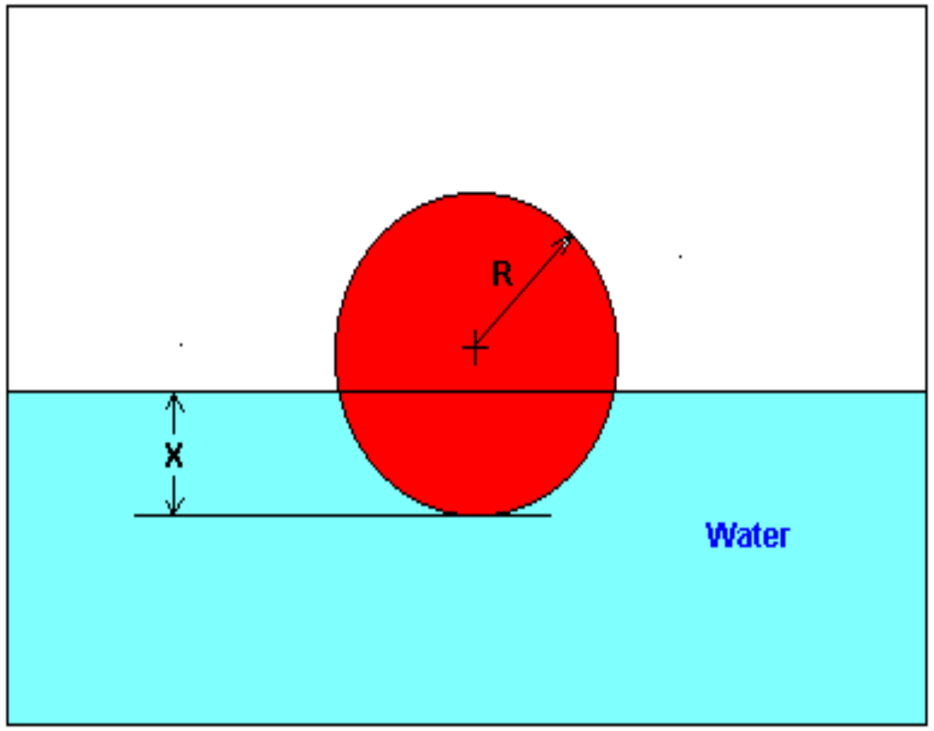


Figure 6 Diagram of the floating ball

Example 1 Cont.

The equation that gives the depth x to which the ball is submerged under water is given by

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

a) Use the bisection method of finding roots of equations to find the depth x to which the ball is submerged under water. Conduct three iterations to estimate the root of the above equation.

b) Find the absolute relative approximate error at the end of each iteration, and the number of significant digits at least correct at the end of each iteration.

Example 1 Cont.

From the physics of the problem, the ball would be submerged between x = 0 and x = 2R,

where R = radius of the ball,

that is

$$0 \le x \le 2R$$

$$0 \le x \le 2(0.055)$$

$$0 \le x \le 0.11$$

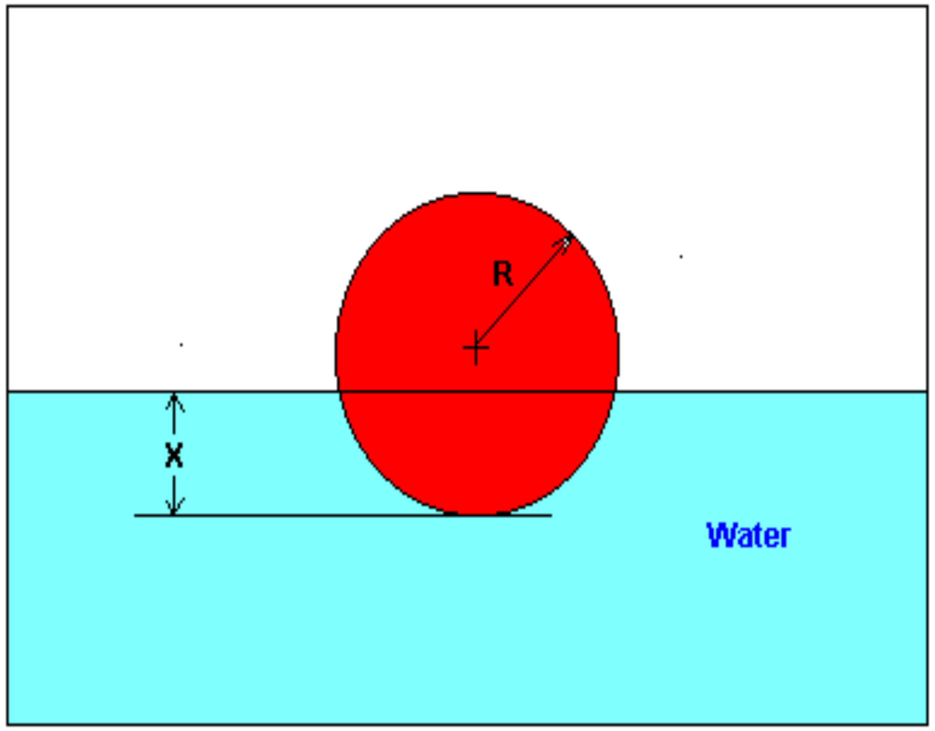


Figure 6 Diagram of the floating ball

Example 1 Cont.

Solution

To aid in the understanding of how this method works to find the root of an equation, the graph of f(x) is shown to the right,

where

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

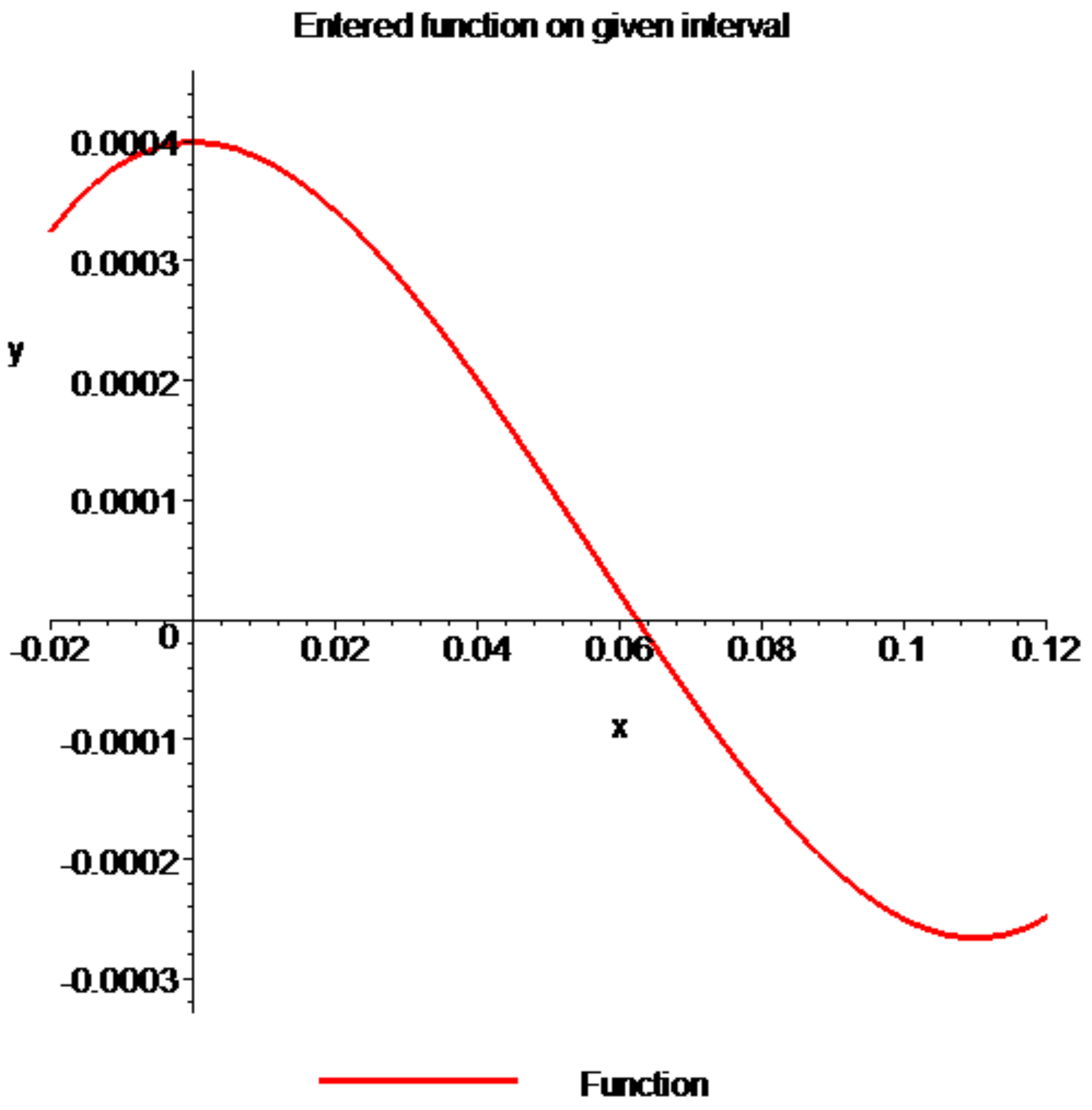


Figure 7 Graph of the function f(x)

Example 1 Cont.

Let us assume

$$x_\ell = 0.00$$

$$x_u = 0.11$$

Check if the function changes sign between $x_{\boxed{?}}$ and x_u .

$$f(x_l) = f(0) = (0)^3 - 0.165(0)^2 + 3.993\times10^{-4} = 3.993\times10^{-4}$$

$$f(x_u) = f(0.11) = (0.11)^3 - 0.165(0.11)^2 + 3.993\times10^{-4} = -2.662\times10^{-4}$$

Hence

$$f(x_l)f(x_u) = f(0)f(0.11) = (3.993\times10^{-4})(-2.662\times10^{-4}) < 0$$

So there is at least on root between $x_{\boxed{?}}$ and $x_{u,}$ that is between 0 and 0.11

Example 1 Cont.

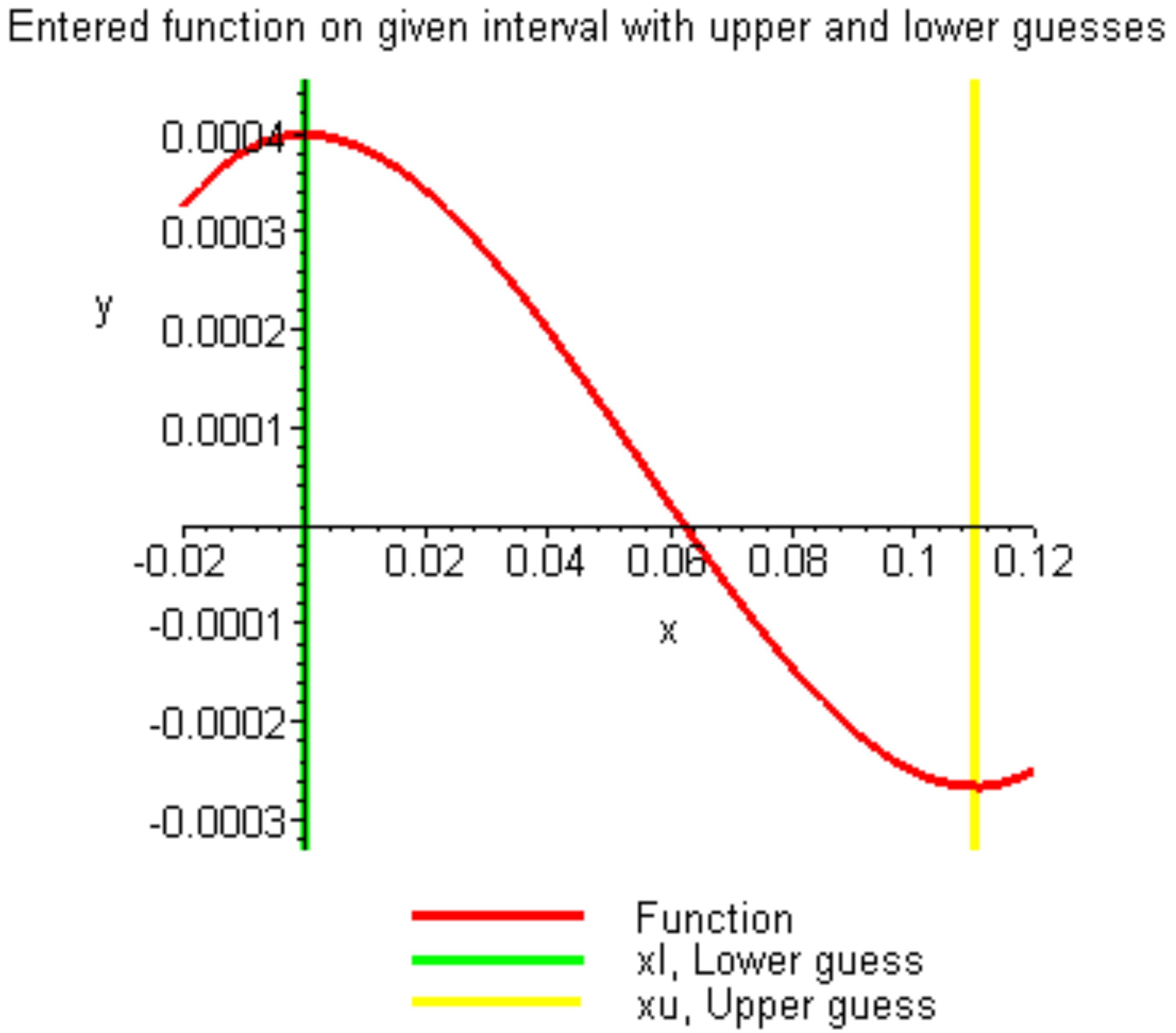


Figure 8 Graph demonstrating sign change between initial limits

Example 1 Cont.

Iteration 1

The estimate of the root is $x_m = \frac{x_\ell + x_u}{2} = \frac{0+0.11}{2} = 0.055$

$$f(x_m) = f(0.055) = (0.055)^3 - 0.165(0.055)^2 + 3.993\times10^{-4} = 6.655\times10^{-5}$$

$$f(x_l)f(x_m) = f(0)f(0.055) = (3.993\times10^{-4})(6.655\times10^{-5}) > 0$$

Hence the root is bracketed between x_m and x_u, that is, between 0.055 and 0.11. So, the lower and upper limits of the new bracket are

$$x_l = 0.055,\ x_u = 0.11$$

At this point, the absolute relative approximate error $|\in_a|$ cannot be calculated as we do not have a previous approximation.

Example 1 Cont.

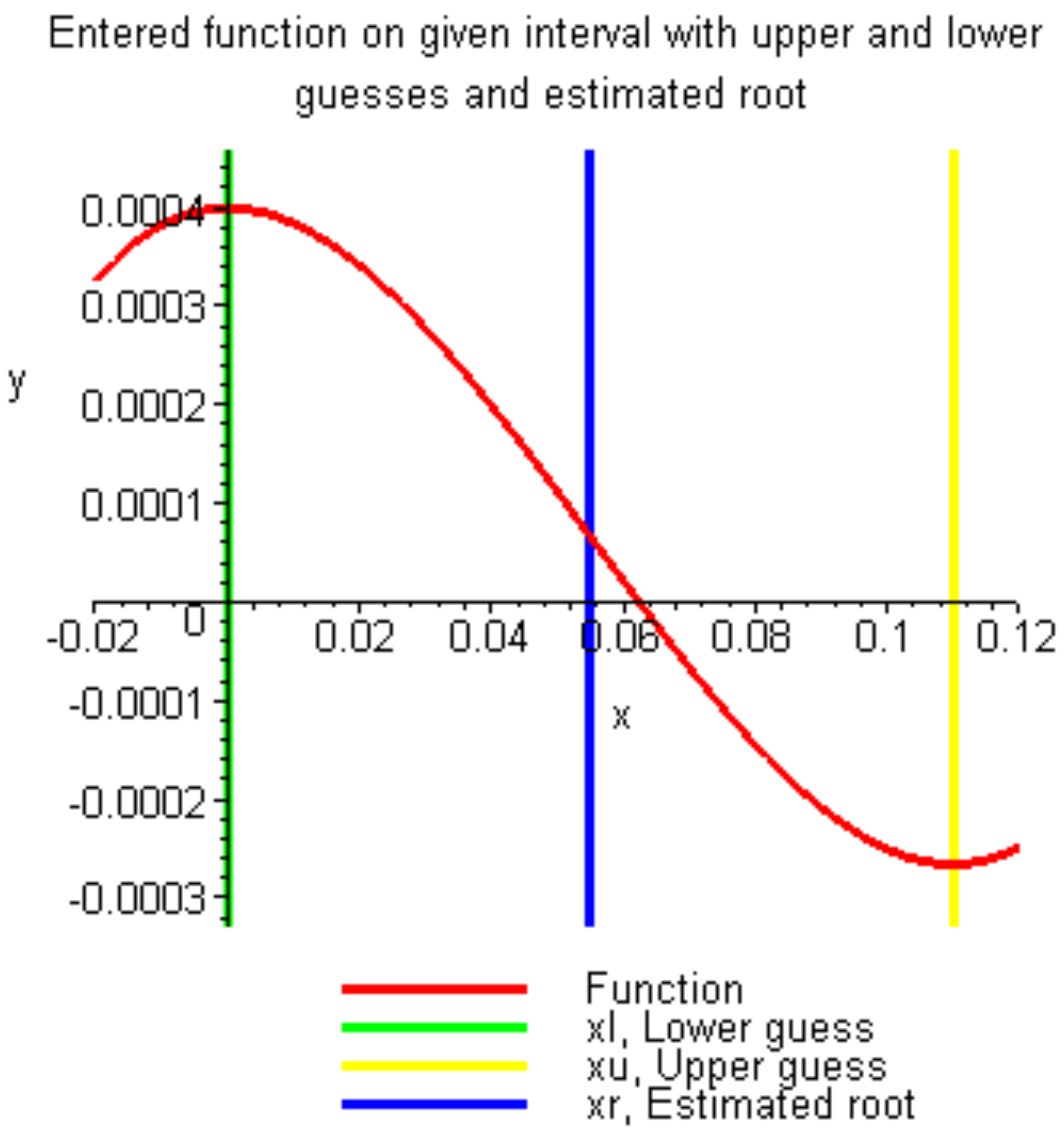


Figure 9 Estimate of the root for Iteration 1

Example 1 Cont.

Iteration 2

The estimate of the root is $x_m = \dfrac{x_\ell + x_u}{2} = \dfrac{0.055 + 0.11}{2} = 0.0825$

$$f(x_m) = f(0.0825) = (0.0825)^3 - 0.165(0.0825)^2 + 3.993\times10^{-4} = -1.622\times10^{-4}$$

$$f(x_l)f(x_m) = f(0)f(0.055) = (6.655\times10^{-5})(-1.622\times10^{-4}) < 0$$

Hence the root is bracketed between $x_{\boxed{?}\boxed{?}}$and x_m, that is, between 0.055 and 0.0825. So, the lower and upper limits of the new bracket are

$$x_l = 0.055,\ x_u = 0.0825$$

Example 1 Cont.

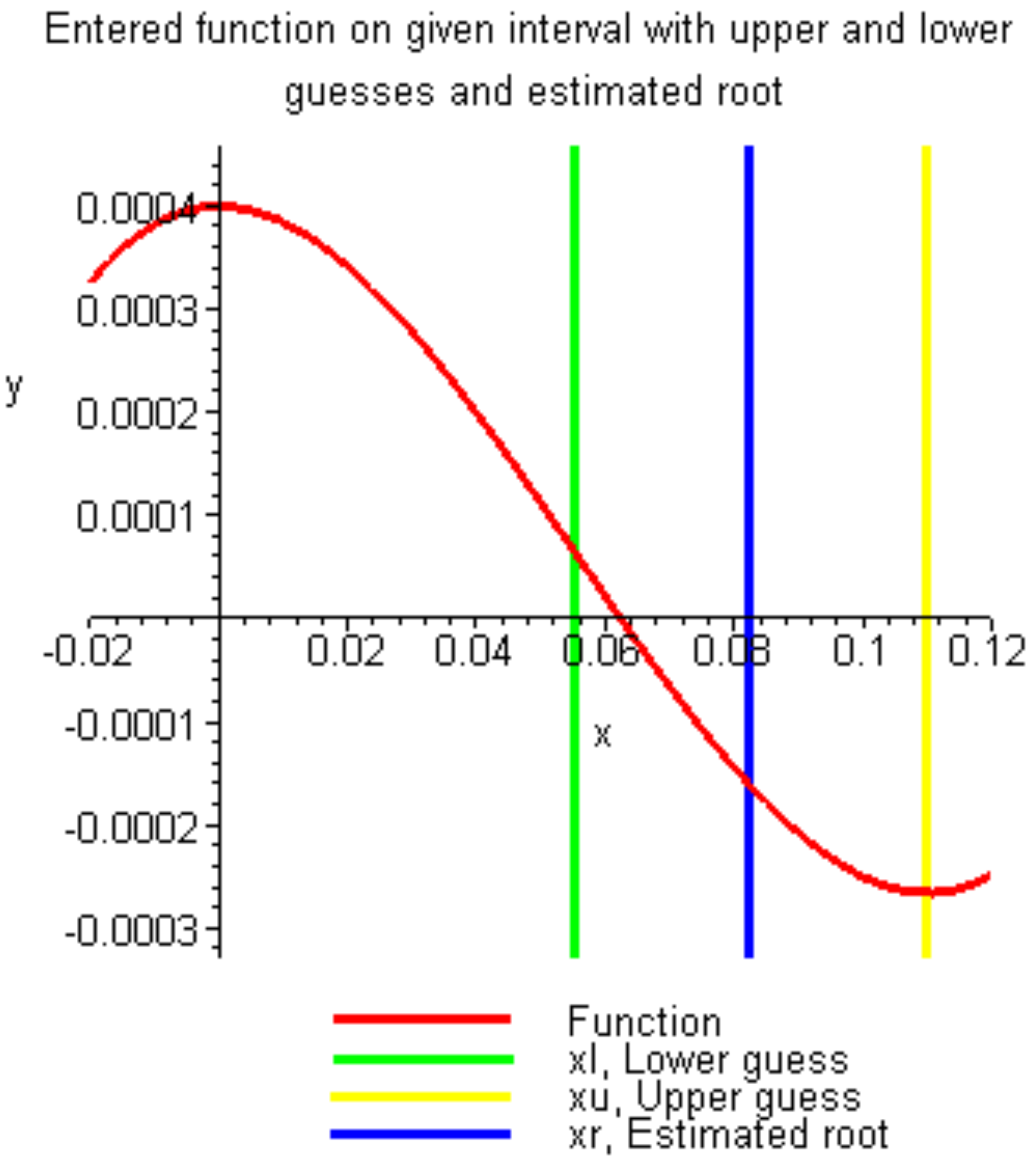


Figure 10 Estimate of the root for Iteration 2

Example 1 Cont.

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

$$\left|\in_a\right| = \left|\frac{x_m^{new} - x_m^{old}}{x_m^{new}}\right| \times 100$$

$$= \left|\frac{0.0825 - 0.055}{0.0825}\right| \times 100$$

$$= 33.333\%$$

None of the significant digits are at least correct in the estimate root of x_m = 0.0825 because the absolute relative approximate error is greater than 5%.

Example 1 Cont.

Iteration 3

The estimate of the root is $x_m = \frac{x_\ell + x_u}{2} = \frac{0.055 + 0.0825}{2} = 0.06875$

$$f(x_m) = f(0.06875) = (0.06875)^3 - 0.165(0.06875)^2 + 3.993\times10^{-4} = -5.563\times10^{-5}$$

$$f(x_l)f(x_m) = f(0.055)f(0.06875) = (6.655\times10^{-5})(-5.563\times10^{-5}) < 0$$

Hence the root is bracketed between x⍰⍰and x_m, that is, between 0.055 and 0.06875. So, the lower and upper limits of the new bracket are

$$x_l = 0.055,\ x_u = 0.06875$$

Example 1 Cont.

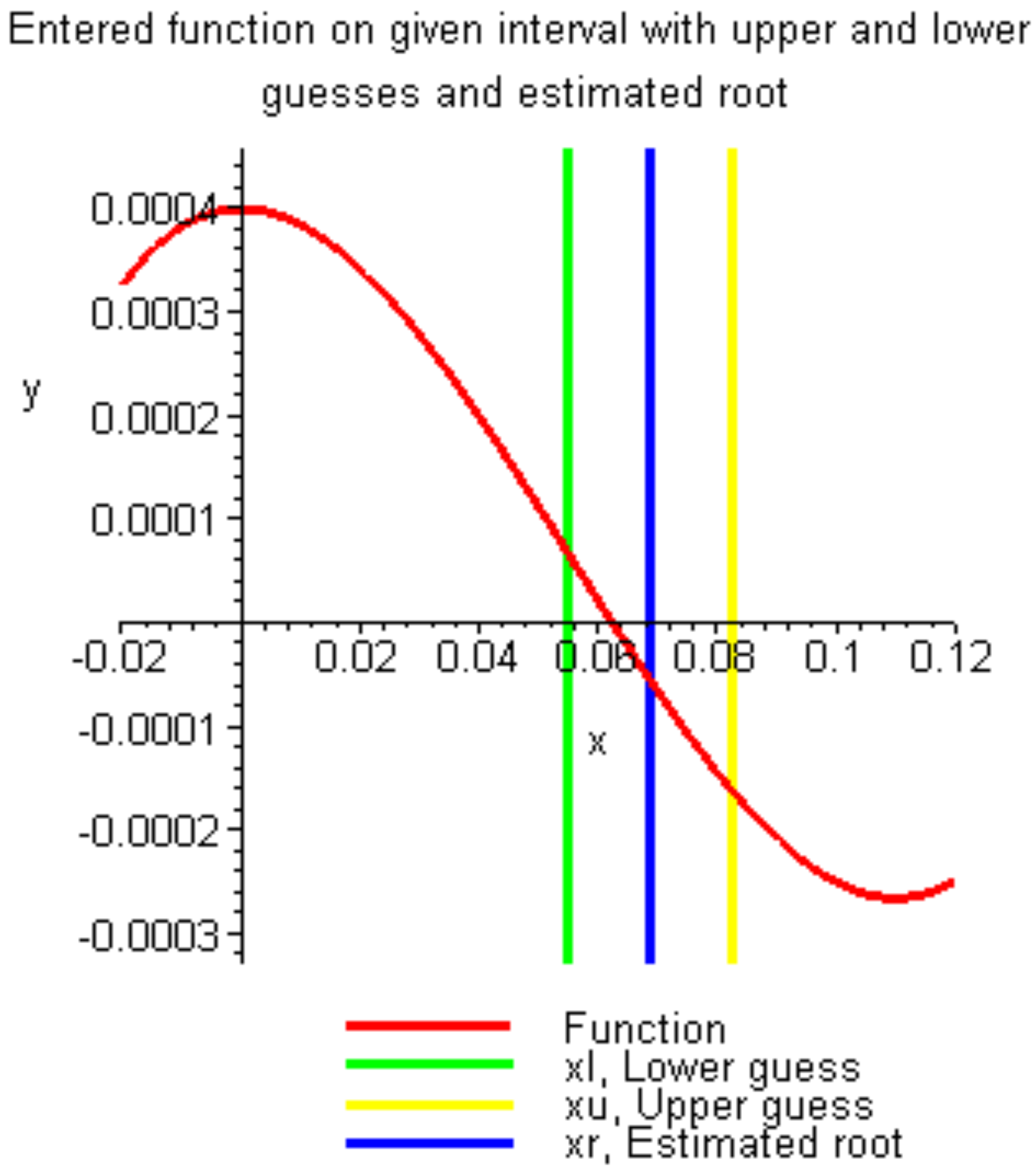


Figure 11 Estimate of the root for Iteration 3

Example 1 Cont.

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

$$|\in_a| = \left|\frac{x_m^{new} - x_m^{old}}{x_m^{new}}\right| \times 100$$

$$= \left|\frac{0.06875 - 0.0825}{0.06875}\right| \times 100$$

$$= 20\%$$

Still none of the significant digits are at least correct in the estimated root of the equation as the absolute relative approximate error is greater than 5%.
Seven more iterations were conducted and these iterations are shown in Table 1.

Table 1 Cont.

Table 1 Root of f(x)=0 as function of number of iterations for bisection method.

Iteration	x_ℓ	x_u	x_m	$\left\|\in_a\right\|\%$	$f(x_m)$
1	0.00000	0.11	0.055	----------	6.655×10^{-5}
2	0.055	0.11	0.0825	33.33	-1.622×10^{-4}
3	0.055	0.0825	0.06875	20.00	-5.563×10^{-5}
4	0.055	0.06875	0.06188	11.11	4.484×10^{-6}
5	0.06188	0.06875	0.06531	5.263	-2.593×10^{-5}
6	0.06188	0.06531	0.06359	2.702	-1.0804×10^{-5}
7	0.06188	0.06359	0.06273	1.370	-3.176×10^{-6}
8	0.06188	0.06273	0.0623	0.6897	6.497×10^{-7}
9	0.0623	0.06273	0.06252	0.3436	-1.265×10^{-6}
10	0.0623	0.06252	0.06241	0.1721	-3.0768×10^{-7}

Advantages

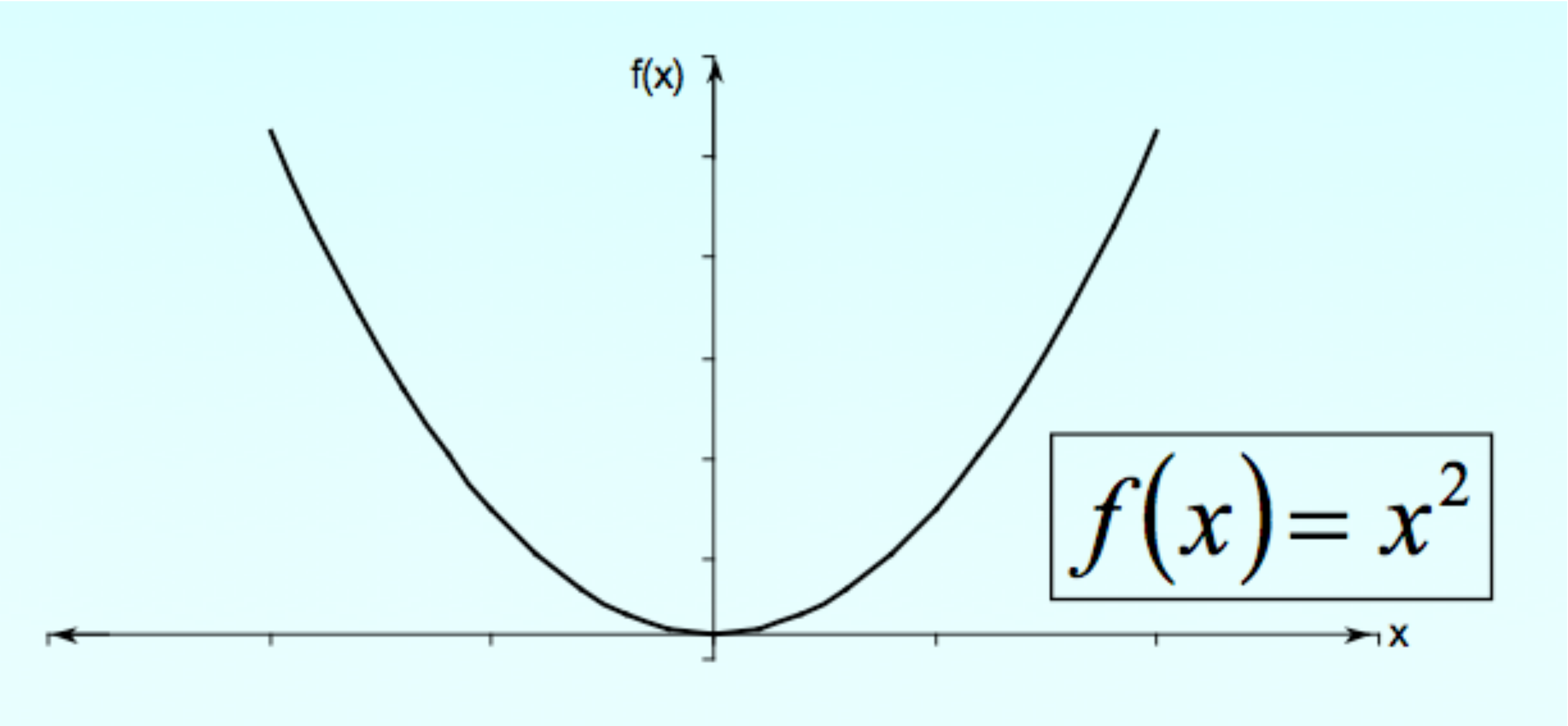
- Always convergent
- The root bracket gets halved with each iteration - guaranteed.

Drawbacks

- Slow convergence
- If one of the initial guesses is close to the root, the convergence is slower

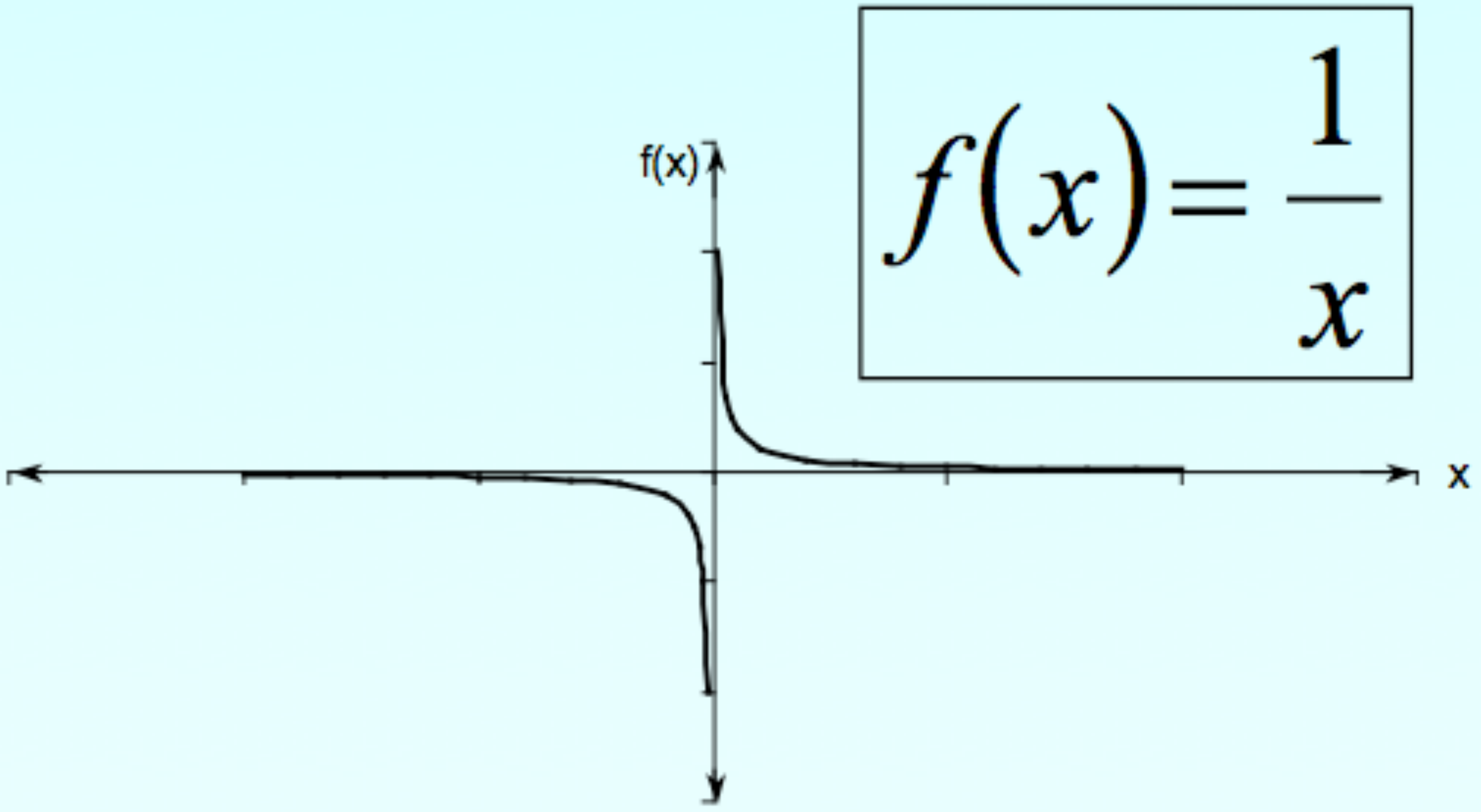
Drawbacks (continued)

- If a function f(x) is such that it just touches the x-axis it will be unable to find the lower and upper guesses.



Drawbacks (continued)

- Function changes sign but root does not exist

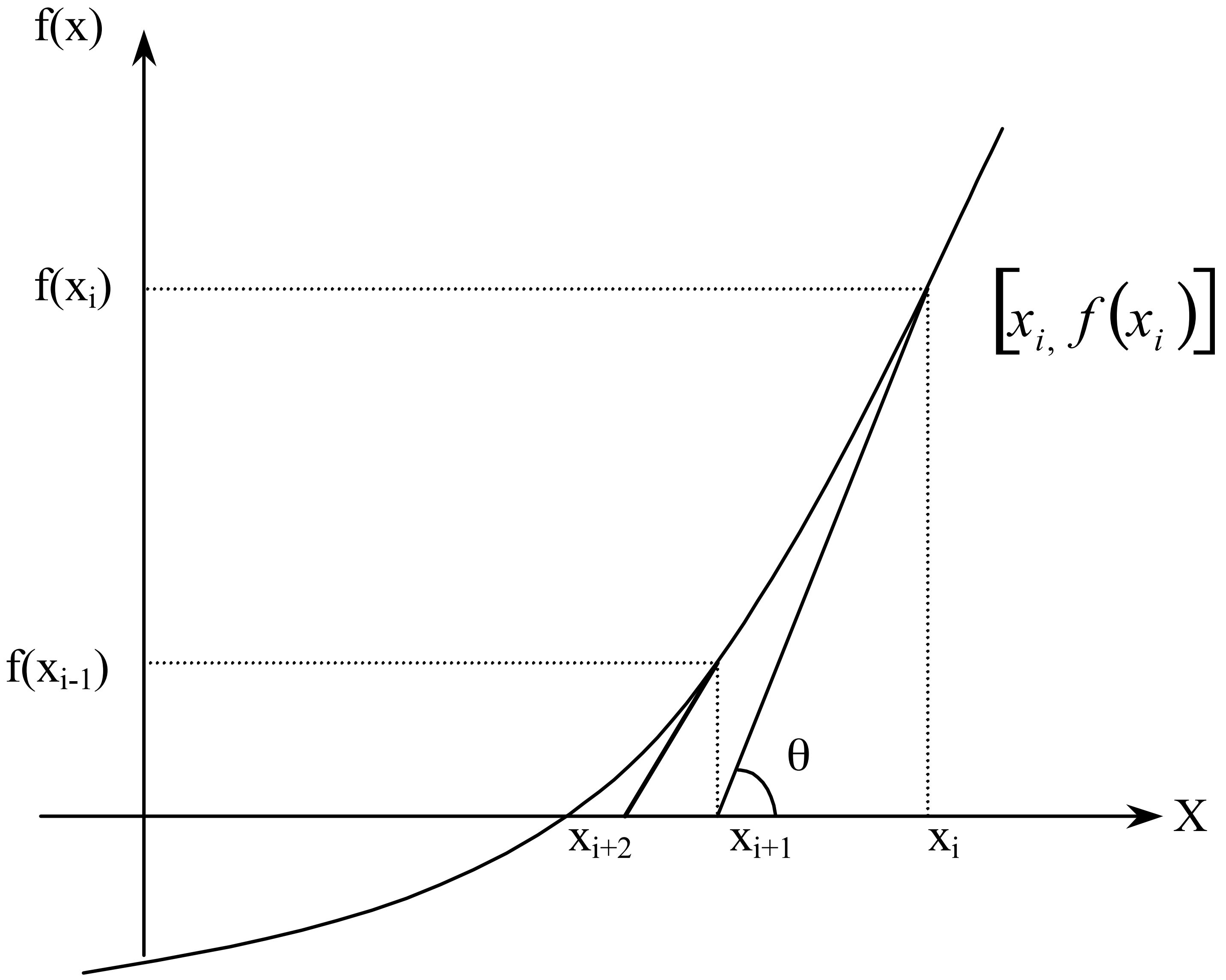


$$f(x)=\frac{1}{x}$$

Newton-Raphson Method

http://numericalmethods.eng.usf.edu

Newton-Raphson Method



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

Figure 1 Geometrical illustration of the Newton-Raphson method.

Algorithm for Newton-Raphson Method

Step 1

Evaluate $f'(x)$ symbolically.

Step 2

Use an initial guess of the root, x_i , to estimate the new value of the root, x_{i+1} , as

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

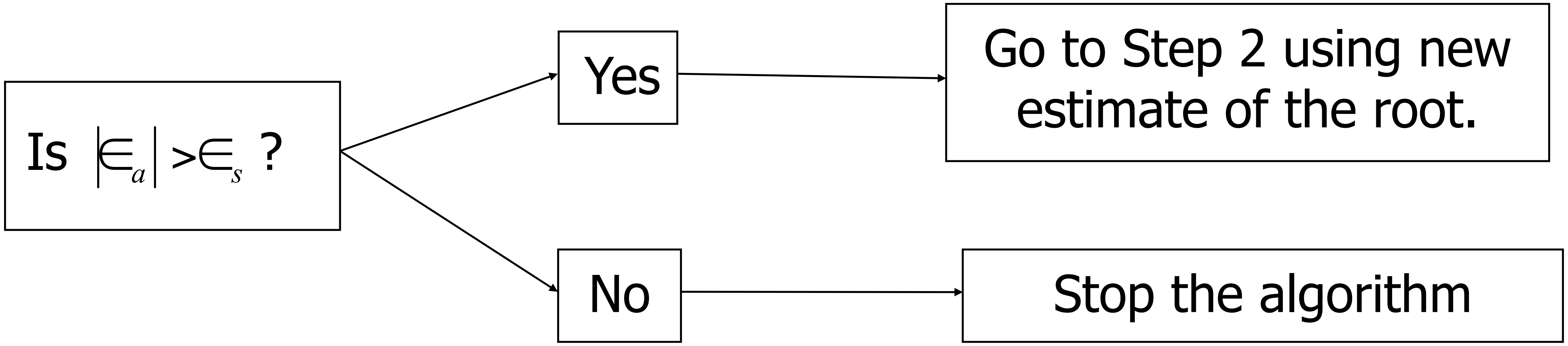
Step 3

Find the absolute relative approximate error $\left|\in_a\right|$ as

$$\left|\in_a\right| = \left|\frac{x_{i+1} - x_i}{x_{i+1}}\right| \times 100$$

Step 4

Compare the absolute relative approximate error with the pre-specified relative error tolerance $\in_s$.



Is $\left|\in_a\right| > \in_s$?
Yes
Go to Step 2 using new estimate of the root.
No
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.

Example 1

You are working for ‘DOWN THE TOILET COMPANY’ that makes floats for ABC commodes. The floating ball has a specific gravity of 0.6 and has a radius of 5.5 cm. You are asked to find the depth to which the ball is submerged when floating in water.

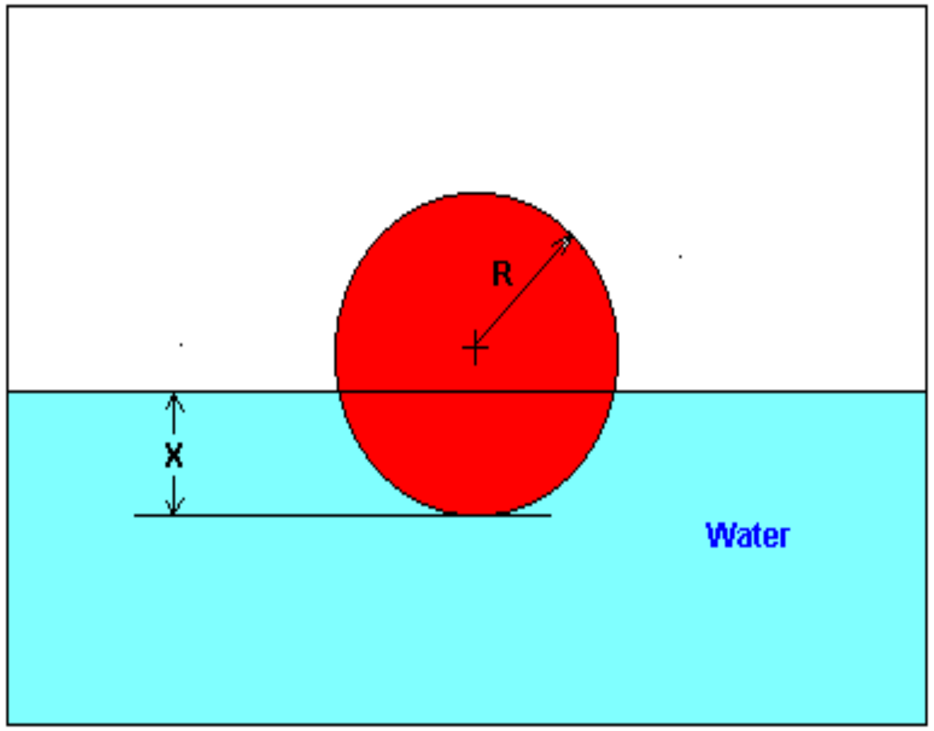


Figure 3 Floating ball problem.

Example 1 Cont.

The equation that gives the depth x in meters to which the ball is submerged under water is given by

$$f\left(x\right) = x^3 \text{-} 0.165x^2 + 3.993 \times 10^{-4}$$

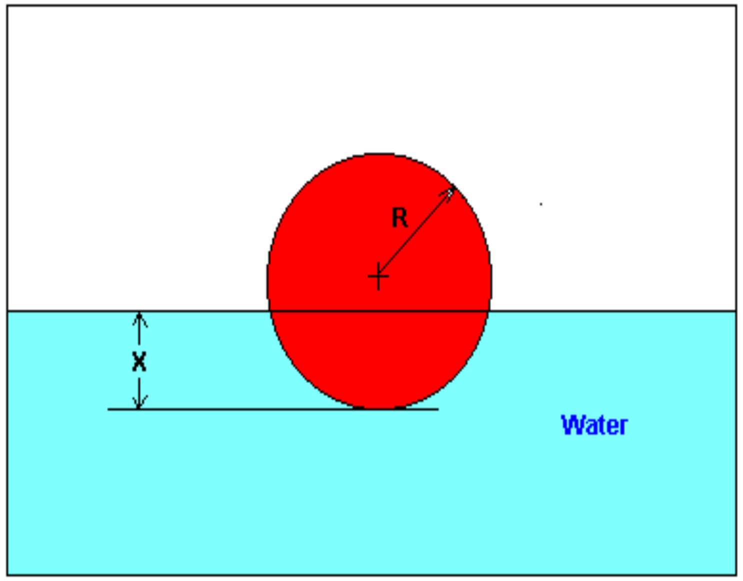


Figure 3 Floating ball problem.

Use the Newton's 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.

Example 1 Cont.

Solution

To aid in the understanding of how this method works to find the root of an equation, the graph of f(x) is shown to the right,

where

$$f(x) = x^3 - 0.165x^2 + 3.993\times10^{-4}$$

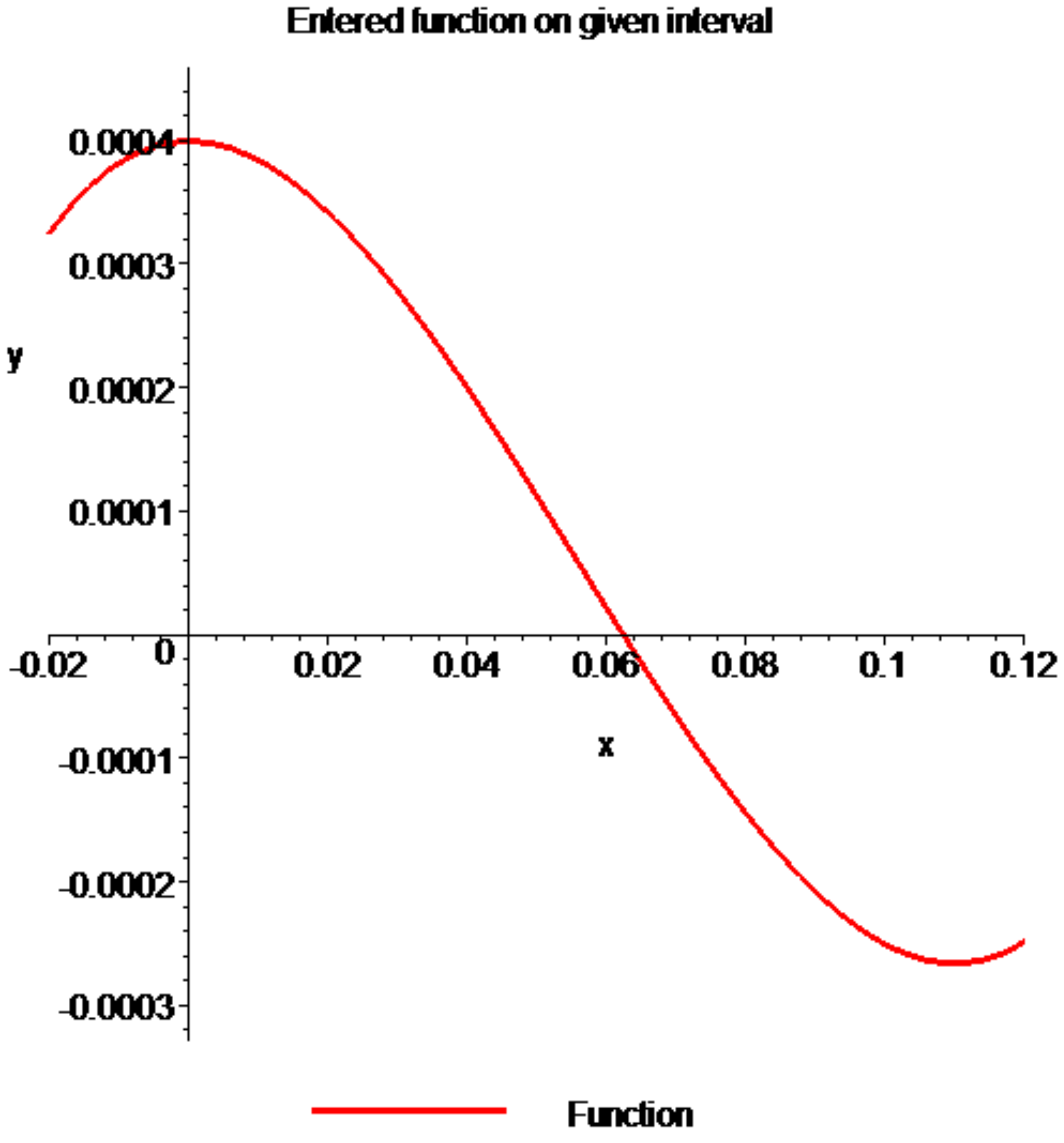


Figure 4 Graph of the function f(x)

Example 1 Cont.

Solve for $f'(x)$

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

$$f'(x) = 3x^2 \text{-} 0.33x$$

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.

Example 1 Cont.

Iteration 1

The estimate of the root is

$$
\begin{aligned}
x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} \\
&= 0.05 - \frac{(0.05)^3 - 0.165(0.05)^2 + 3.993\times10^{-4}}{3(0.05)^2 - 0.33(0.05)} \\
&= 0.05 - \frac{1.118\times10^{-4}}{-9\times10^{-3}} \\
&= 0.05 - (-0.01242) \\
&= 0.06242
\end{aligned}
$$

Example 1 Cont.

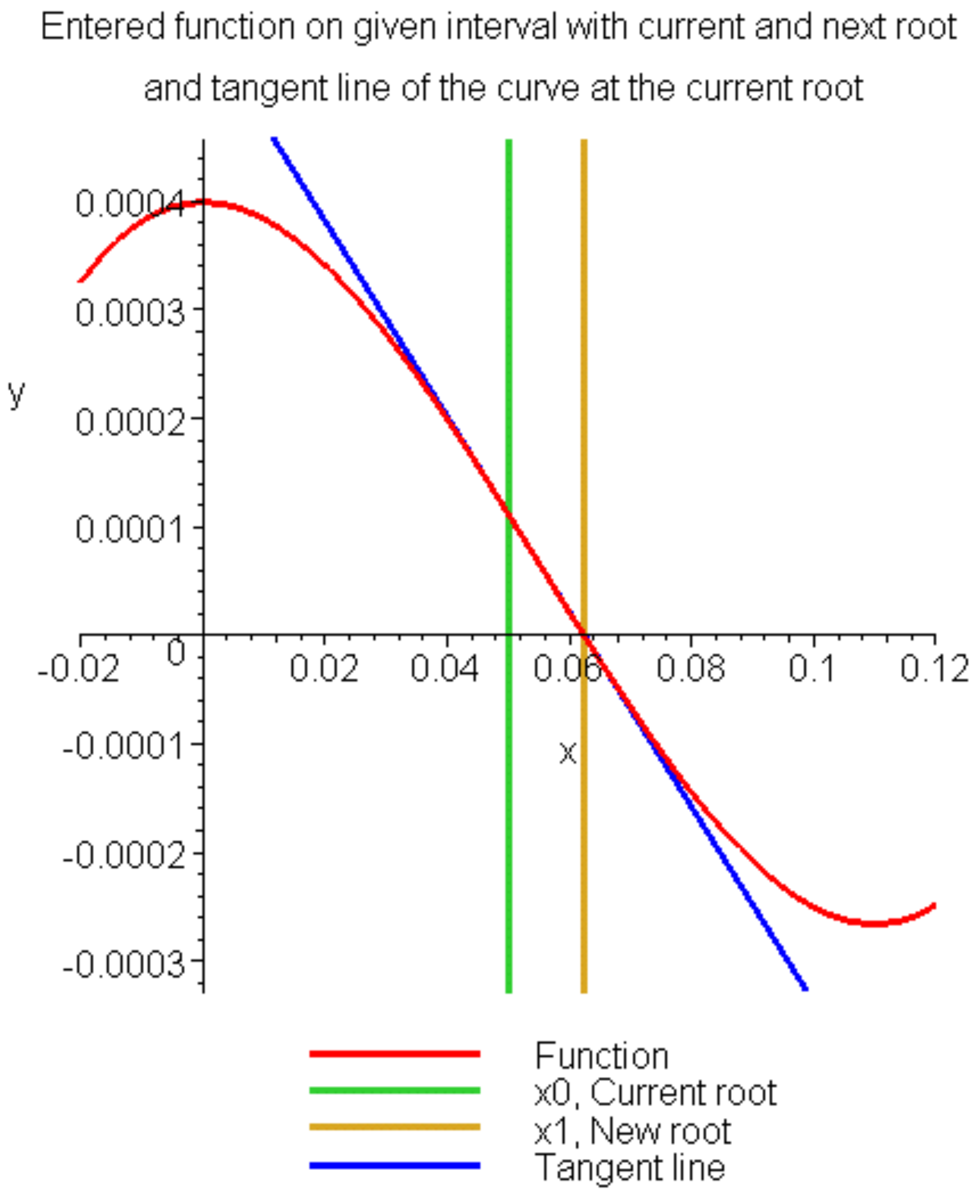


Figure 5 Estimate of the root for the first iteration.

Example 1 Cont.

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

$$\left|\in_a\right| = \left|\frac{x_1 - x_0}{x_1}\right| \times 100$$

$$= \left|\frac{0.06242 - 0.05}{0.06242}\right| \times 100$$

$$= 19.90\%$$

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 digits to be correct in your result.

Example 1 Cont.

Iteration 2

The estimate of the root is

$$
\begin{aligned}
x_2 &= x_1 - \frac{f(x_1)}{f'(x_1)} \\
&= 0.06242 - \frac{(0.06242)^3 - 0.165(0.06242)^2 + 3.993\times10^{-4}}{3(0.06242)^2 - 0.33(0.06242)} \\
&= 0.06242 - \frac{-3.97781\times10^{-7}}{-8.90973\times10^{-3}} \\
&= 0.06242 - \left(4.4646\times10^{-5}\right) \\
&= 0.06238
\end{aligned}
$$

Example 1 Cont.

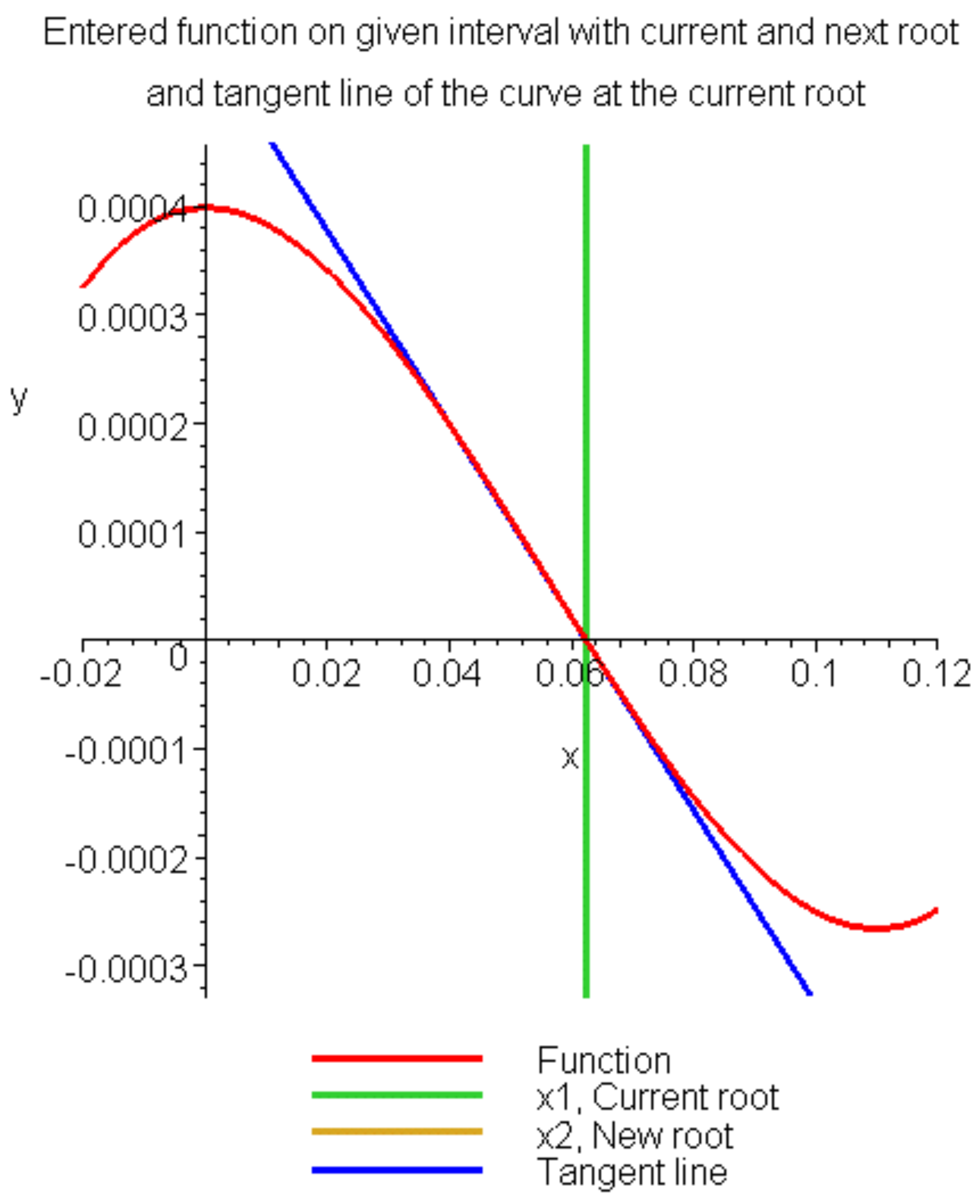


Figure 6 Estimate of the root for the Iteration 2.

Example 1 Cont.

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

$$\left|\in_a\right| = \left|\frac{x_2 - x_1}{x_2}\right| \times 100$$

$$= \left|\frac{0.06238 - 0.06242}{0.06238}\right| \times 100$$

$$= 0.0716\%$$

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

Example 1 Cont.

Iteration 3

The estimate of the root is

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

Example 1 Cont.

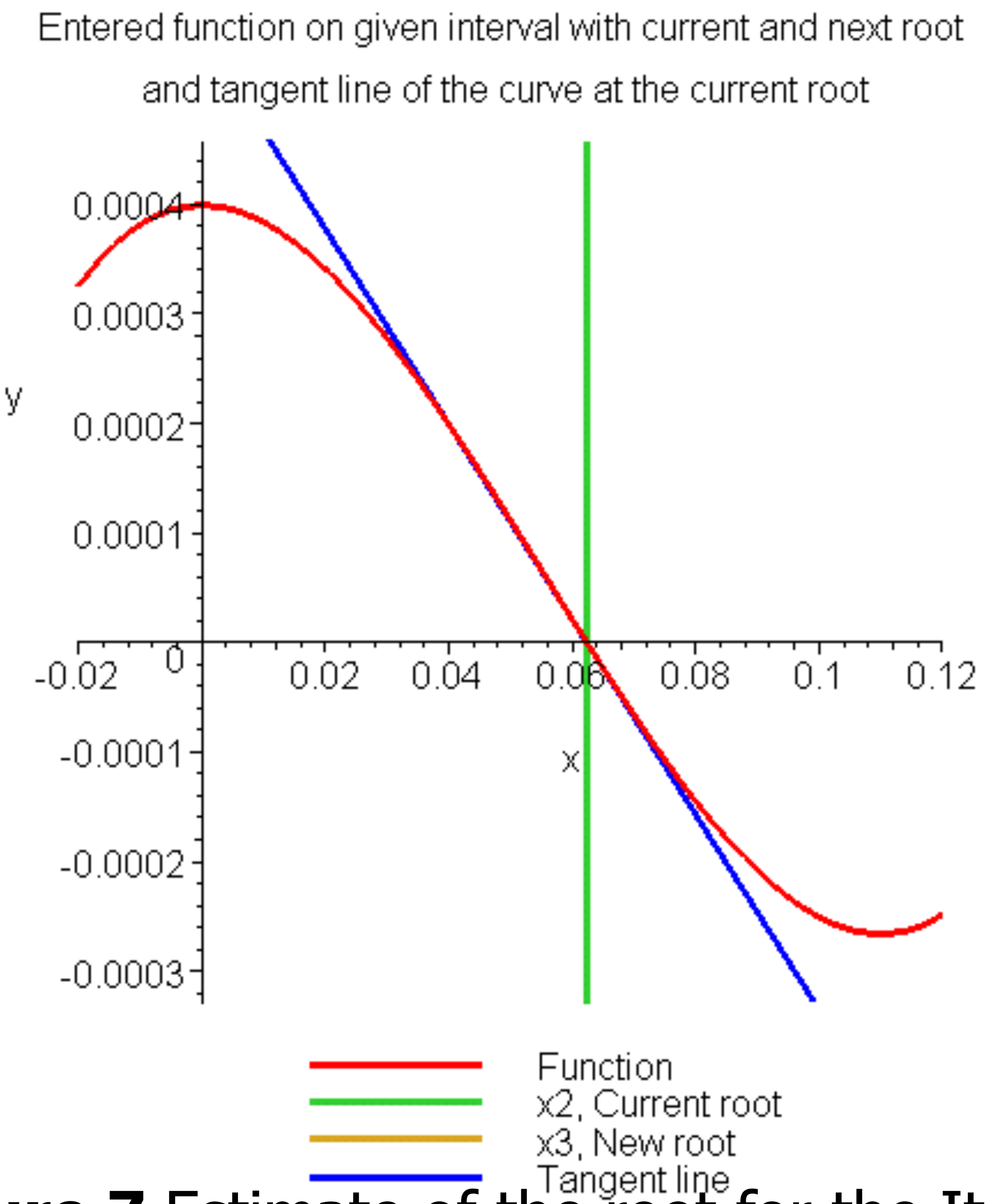


Figure 7 Estimate of the root for the Iteration 3.

Example 1 Cont.

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

$$\begin{aligned}\left|\in_a\right| &= \left|\frac{x_2 - x_1}{x_2}\right| \times 100 \\ &= \left|\frac{0.06238 - 0.06238}{0.06238}\right| \times 100 \\ &= 0\%\end{aligned}$$

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

Advantages and Drawbacks of Newton Raphson Method

http://numericalmethods.eng.usf.edu

Advantages

- Converges fast (quadratic convergence), if it converges.
- Requires only one guess

Drawbacks

1. Divergence at inflection points

   Selection of the initial guess or an iteration value of the root that is close to the inflection point of the function $f(x)$ may start diverging away from the root in ther Newton-Raphson method.

For example, to find the root of the equation $f(x) = (x-1)^3 + 0.512 = 0$

The Newton-Raphson method reduces to

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

Table 1 shows the iterated values of the root of the equation.

The root starts to diverge at Iteration 6 because the previous estimate of 0.92589 is close to the inflection point of $x = 1$.

Eventually after 12 more iterations the root converges to the exact value of $x = 0.2.$

Drawbacks – Inflection Points

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
…	…
18	0.2000

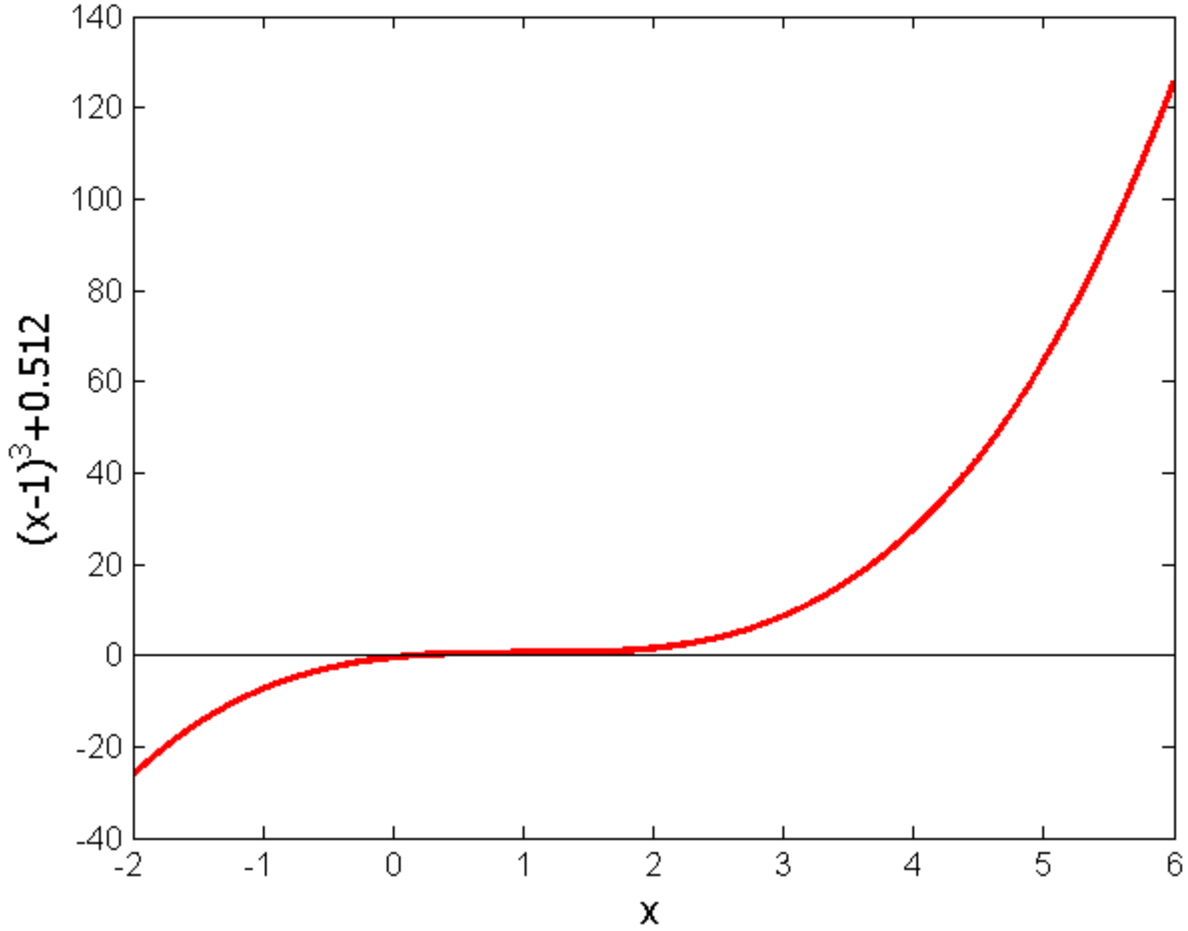


Figure 8 Divergence at inflection point for

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

Drawbacks – Division by Zero

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 $x_0 = 0.02$, the denominator will equal zero.

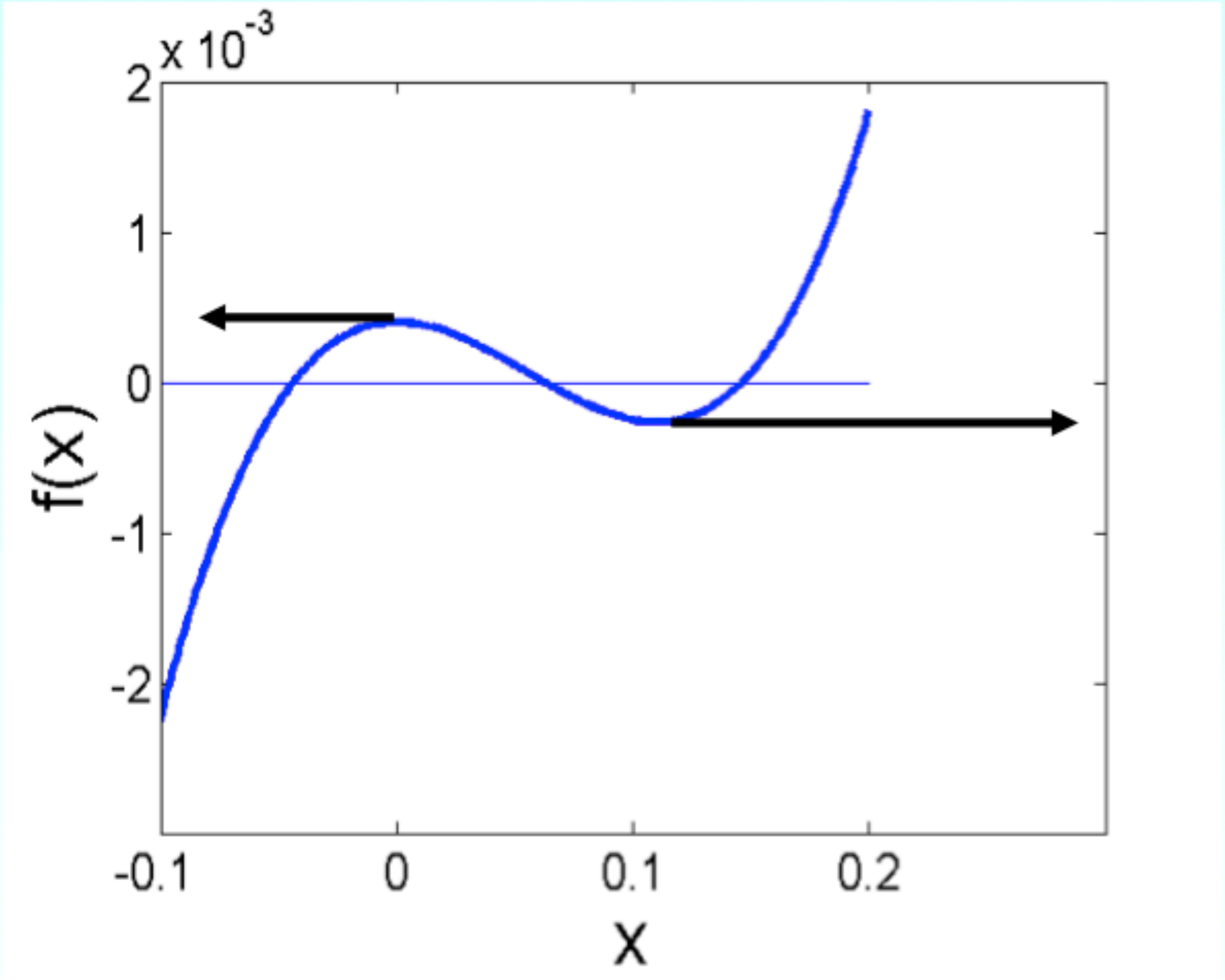


Figure 9 Pitfall of division by zero or near a zero number

Drawbacks – Oscillations near local maximum and minimum

3. Oscillations near local maximum and minimum

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.

Drawbacks – Oscillations near local maximum and minimum

Table 3 Oscillations near local maxima and mimima 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

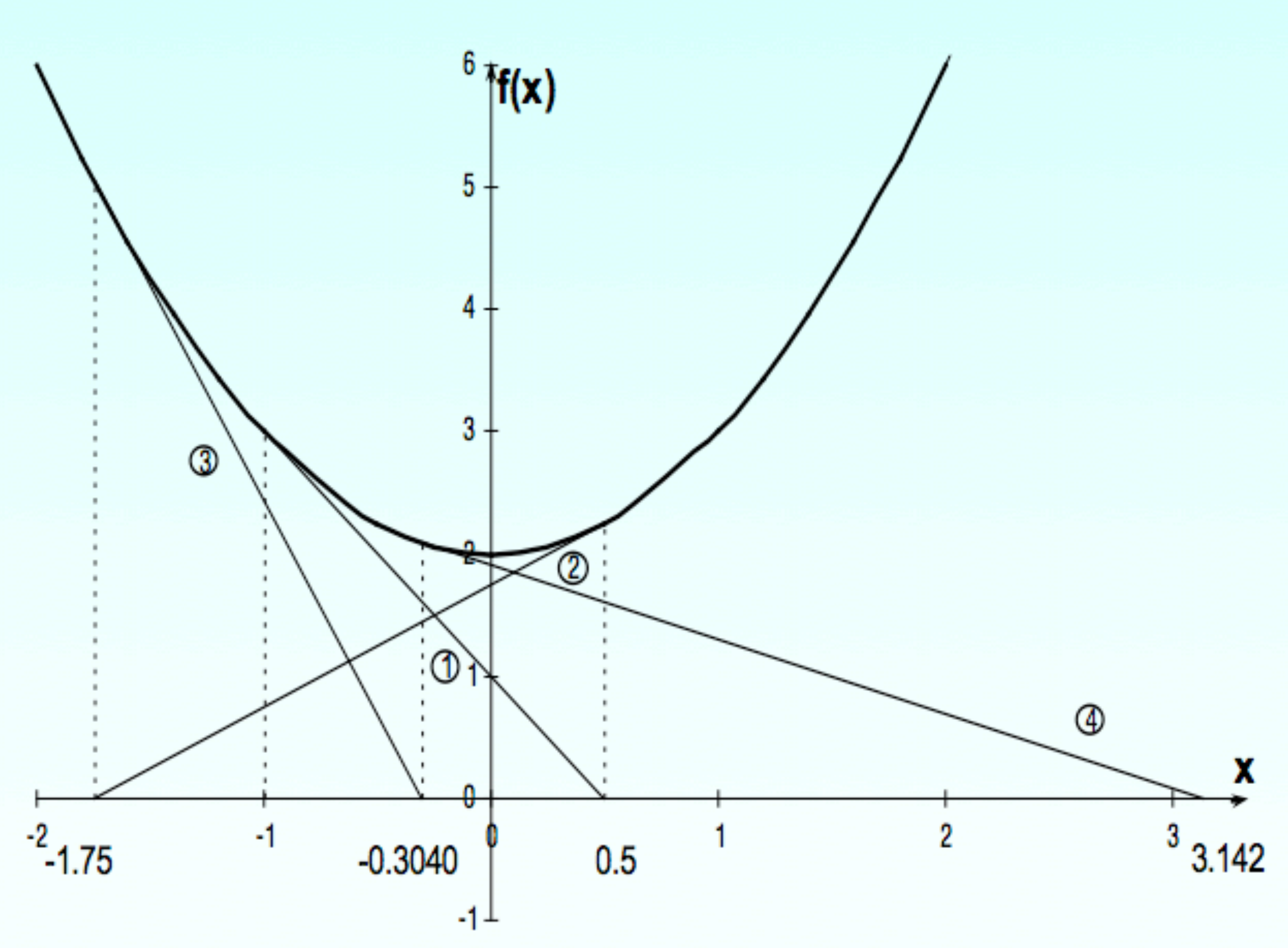


Figure 10 Oscillations around local minima for $f(x) = x^2 + 2$.

Drawbacks – Root Jumping

4. Root Jumping

In some cases 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

$$f(x) = \sin x = 0$$

Choose

$$x_0 = 2.4\pi = 7.539822$$

It will converge to

$$x = 0$$

instead of

$$x = 2\pi = 6.2831853$$

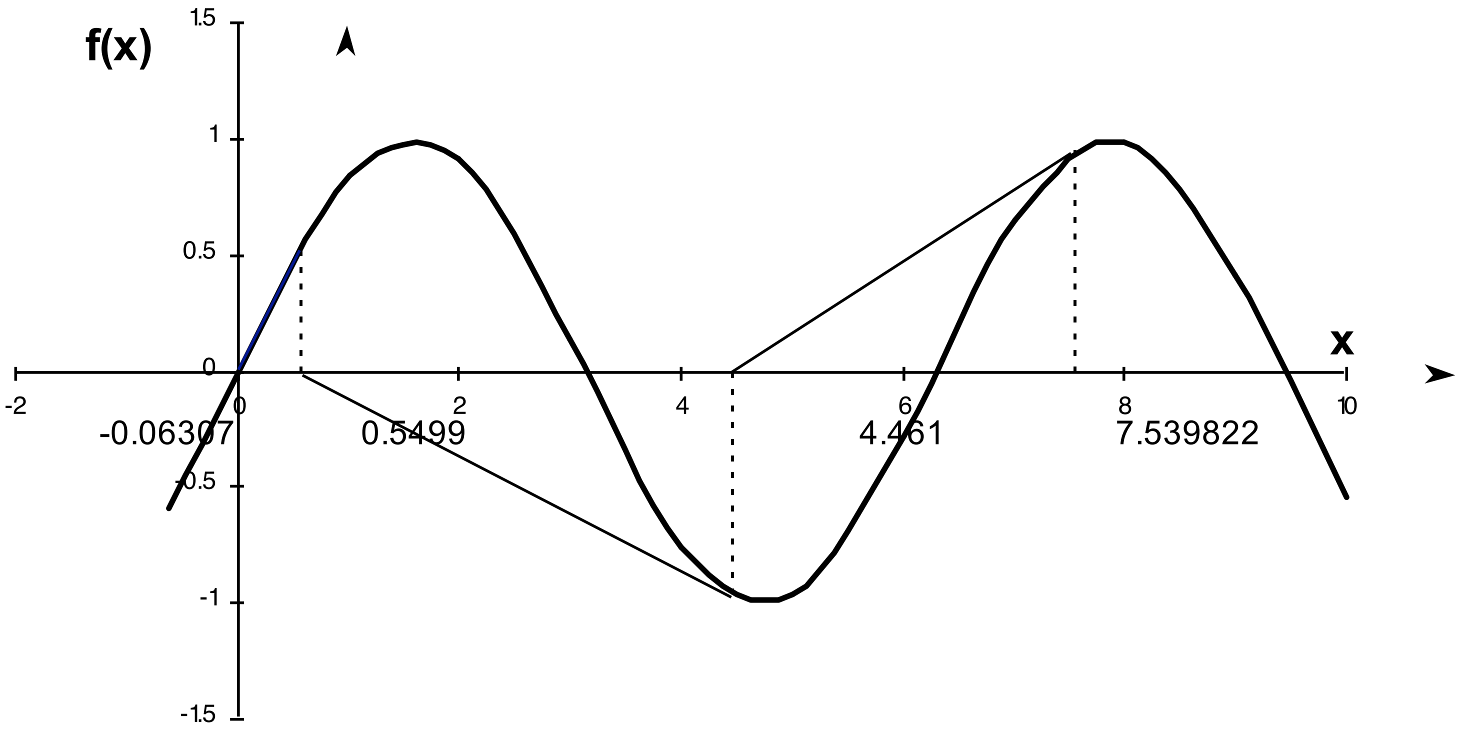


Figure 11 Root jumping from intended location of root for $f(x) = \sin x = 0$.

Additional Resources

For all resources on this topic such as digital audiovisual lectures, primers, textbook chapters, multiple-choice tests, worksheets in MATLAB, MATHEMATICA, MathCad and MAPLE, blogs, related physical problems, please visit

http://numericalmethods.eng.usf.edu/topics/newton_raphson.html

THE END

Secant Method

Major: All Engineering Majors

Authors: Autar Kaw, Jai Paul

http://numericalmethods.eng.usf.edu

Transforming Numerical Methods Education for STEM Undergraduates

Secant Method

http://numericalmethods.eng.usf.edu

Secant Method – Derivation

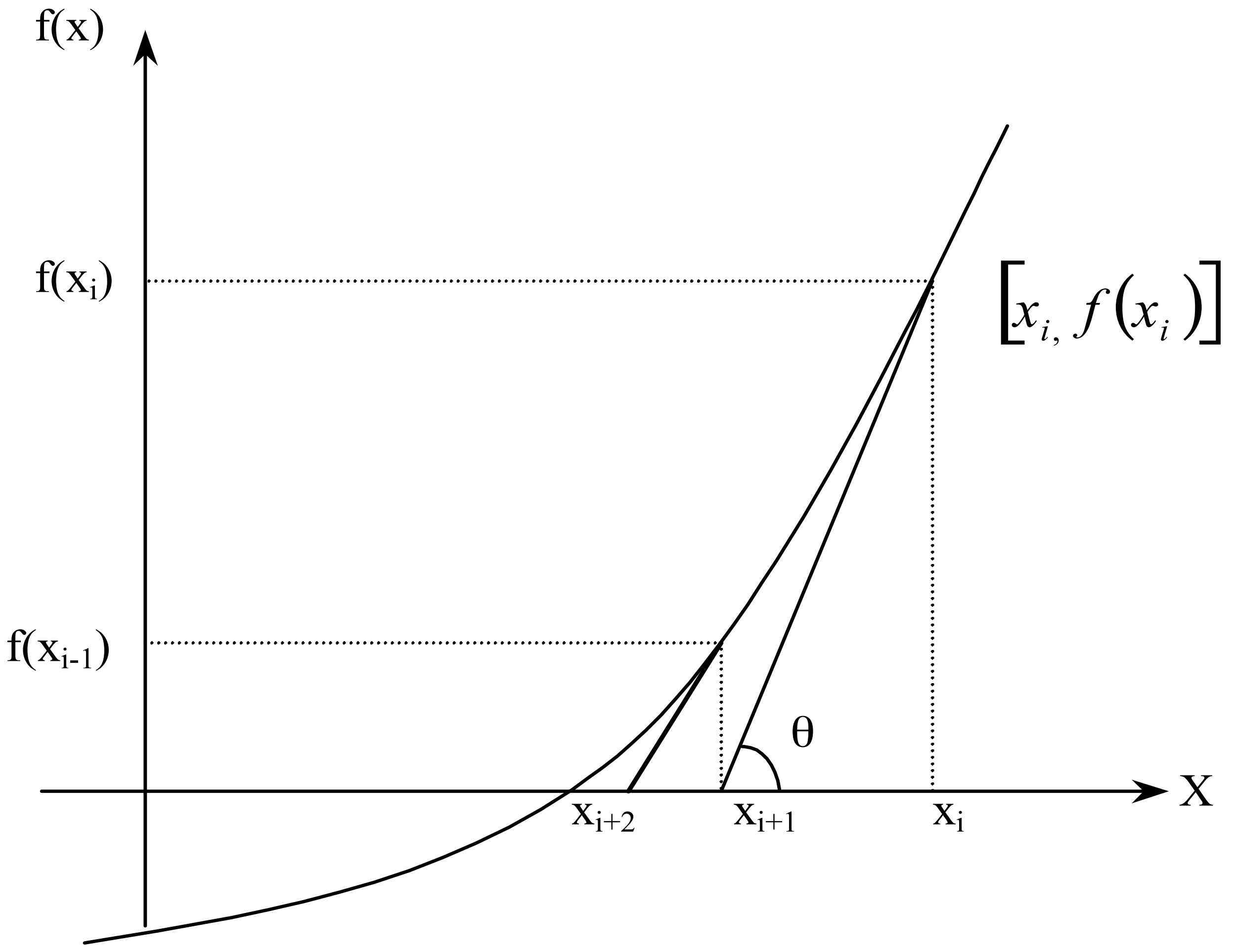


Figure 1 Geometrical illustration of the Newton-Raphson method.

Newton's Method

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

Approximate the derivative

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \qquad (2)$$

Substituting Equation (2) into Equation (1) gives the Secant method

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

Algorithm for Secant Method

Step 1

Calculate the next estimate of the root from two initial guesses

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

Find the absolute relative approximate error

$$\left|\in_a\right| = \left|\frac{x_{i+1} - x_i}{x_{i+1}}\right| \times 100$$

Step 2

Find if the absolute relative approximate error is greater than the prespecified relative error tolerance.

If so, replace x_i with the newly calculated value x_{i+1}, go back to step 1, else stop the algorithm.

Also check if the number of iterations has exceeded the maximum number of iterations.

Example 1

You are working for ‘DOWN THE TOILET COMPANY’ that makes floats for ABC commodes. The floating ball has a specific gravity of 0.6 and has a radius of 5.5 cm. You are asked to find the depth to which the ball is submerged when floating in water.

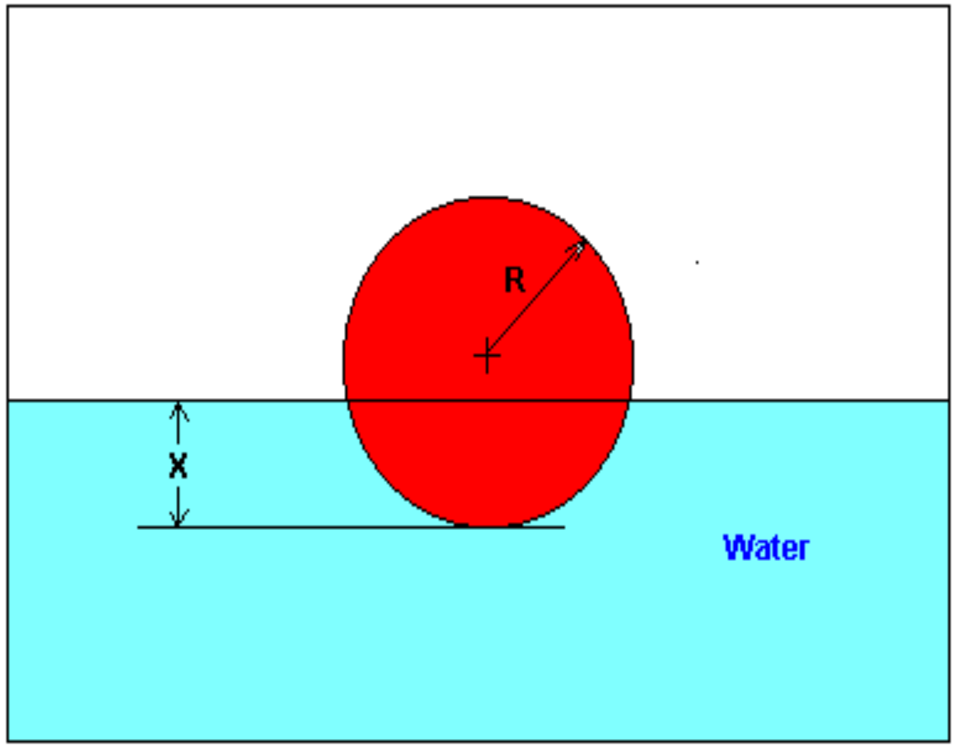


Figure 3 Floating Ball Problem.

Example 1 Cont.

The equation that gives the depth x to which the ball is submerged under water is given by

$$f(x) = x^3 - 0.165x^2 + 3.993\times10^{-4}$$

Use the Secant method of finding roots of equations to find the depth x to which the ball is submerged under water.

- Conduct three iterations to estimate the root of the above equation.
- Find the absolute relative approximate error and the number of significant digits at least correct at the end of each iteration.

Example 1 Cont.

Solution

To aid in the understanding of how this method works to find the root of an equation, the graph of f(x) is shown to the right,

where

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

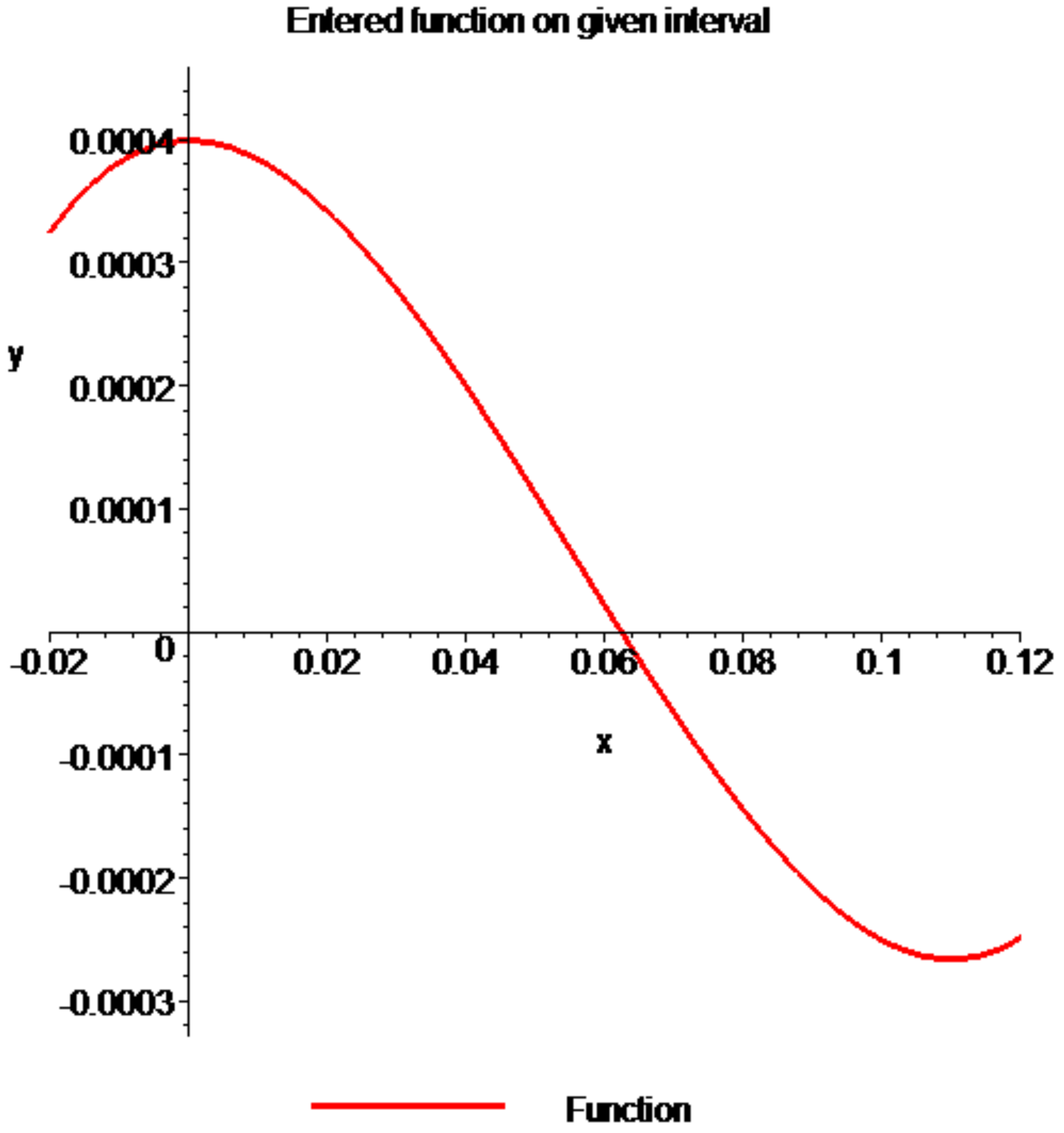


Figure 4 Graph of the function f(x).

Example 1 Cont.

Let us assume the initial guesses of the root of $f(x)=0$ as $x_{-1}=0.02$ and $x_0=0.05.$

Iteration 1

The estimate of the root is

$$x_1 = x_0 - \frac{f(x_0)(x_0 - x_{-1})}{f(x_0) - f(x_{-1})}$$

$$= 0.05 - \frac{\left(0.05^3 - 0.165(0.05)^2 + 3.993\times10^{-4}\right)(0.05-0.02)}{\left(0.05^3 - 0.165(0.05)^2 + 3.993\times10^{-4}\right) - \left(0.02^3 - 0.165(0.02)^2 + 3.993\times10^{-4}\right)}$$

$$= 0.06461$$

Example 1 Cont.

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

$$\left|\in_a\right| = \left|\frac{x_1 - x_0}{x_1}\right| \times 100$$

$$= \left|\frac{0.06461 - 0.05}{0.06461}\right| \times 100$$

$$= 22.62\%$$

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

Example 1 Cont.

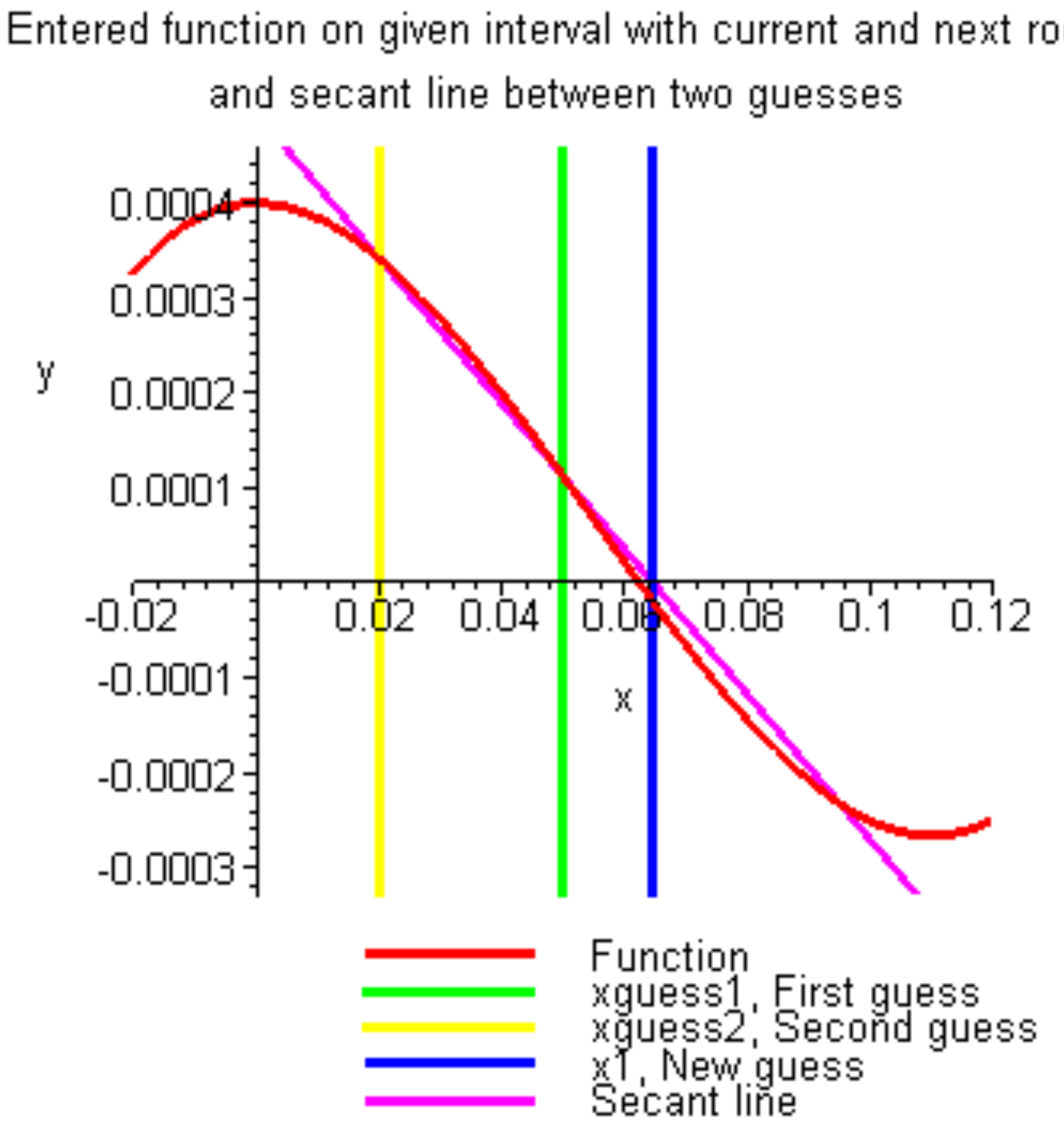


Figure 5 Graph of results of Iteration 1.

Example 1 Cont.

Iteration 2

The estimate of the root is

$$x_2 = x_1 - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)}$$

$$= 0.06461 - \frac{\left(0.06461^3 - 0.165(0.06461)^2 + 3.993\times10^{-4}\right)(0.06461 - 0.05)}{\left(0.06461^3 - 0.165(0.06461)^2 + 3.993\times10^{-4}\right) - \left(0.05^3 - 0.165(0.05)^2 + 3.993\times10^{-4}\right)}$$

$$= 0.06241$$

Example 1 Cont.

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

$$\left|\in_a\right| = \left|\frac{x_2 - x_1}{x_2}\right| \times 100$$

$$= \left|\frac{0.06241 - 0.06461}{0.06241}\right| \times 100$$

$$= 3.525\%$$

The number of significant digits at least correct is 1, as you need an absolute relative approximate error of 5% or less.

Example 1 Cont.

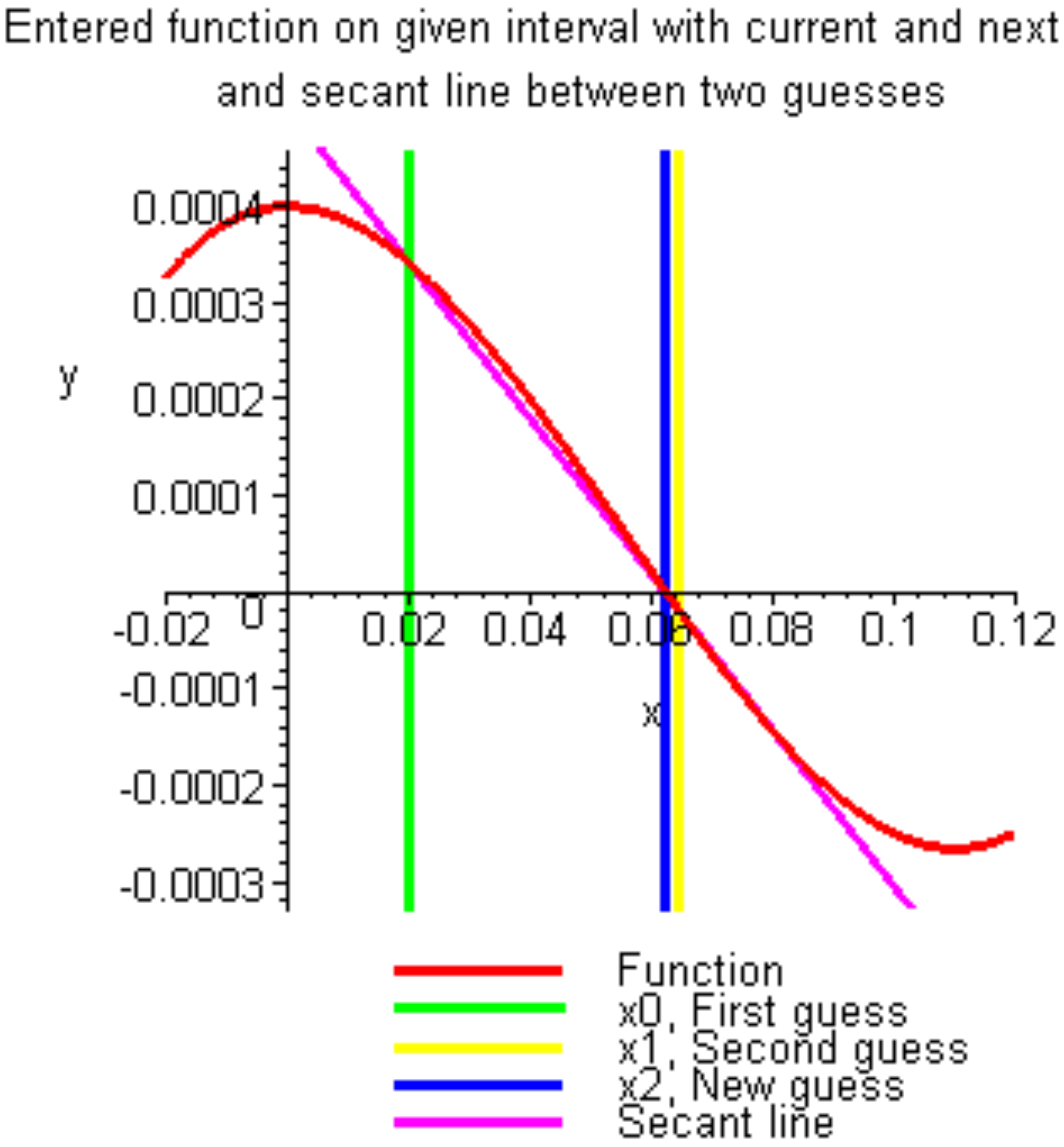


Figure 6 Graph of results of Iteration 2.

Example 1 Cont.

Iteration 3

The estimate of the root is

$$x_3 = x_2 - \frac{f(x_2)(x_2 - x_1)}{f(x_2) - f(x_1)}$$

$$= 0.06241 - \frac{\left(0.06241^3 - 0.165(0.06241)^2 + 3.993\times10^{-4}\right)(0.06241 - 0.06461)}{\left(0.06241^3 - 0.165(0.06241)^2 + 3.993\times10^{-4}\right) - \left(0.05^3 - 0.165(0.06461)^2 + 3.993\times10^{-4}\right)}$$

$$= 0.06238$$

Example 1 Cont.

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

$$|\in_a| = \left|\frac{x_3 - x_2}{x_3}\right| \times 100$$

$$= \left|\frac{0.06238 - 0.06241}{0.06238}\right| \times 100$$

$$= 0.0595\%$$

The number of significant digits at least correct is 5, as you need an absolute relative approximate error of 0.5% or less.

Iteration #3

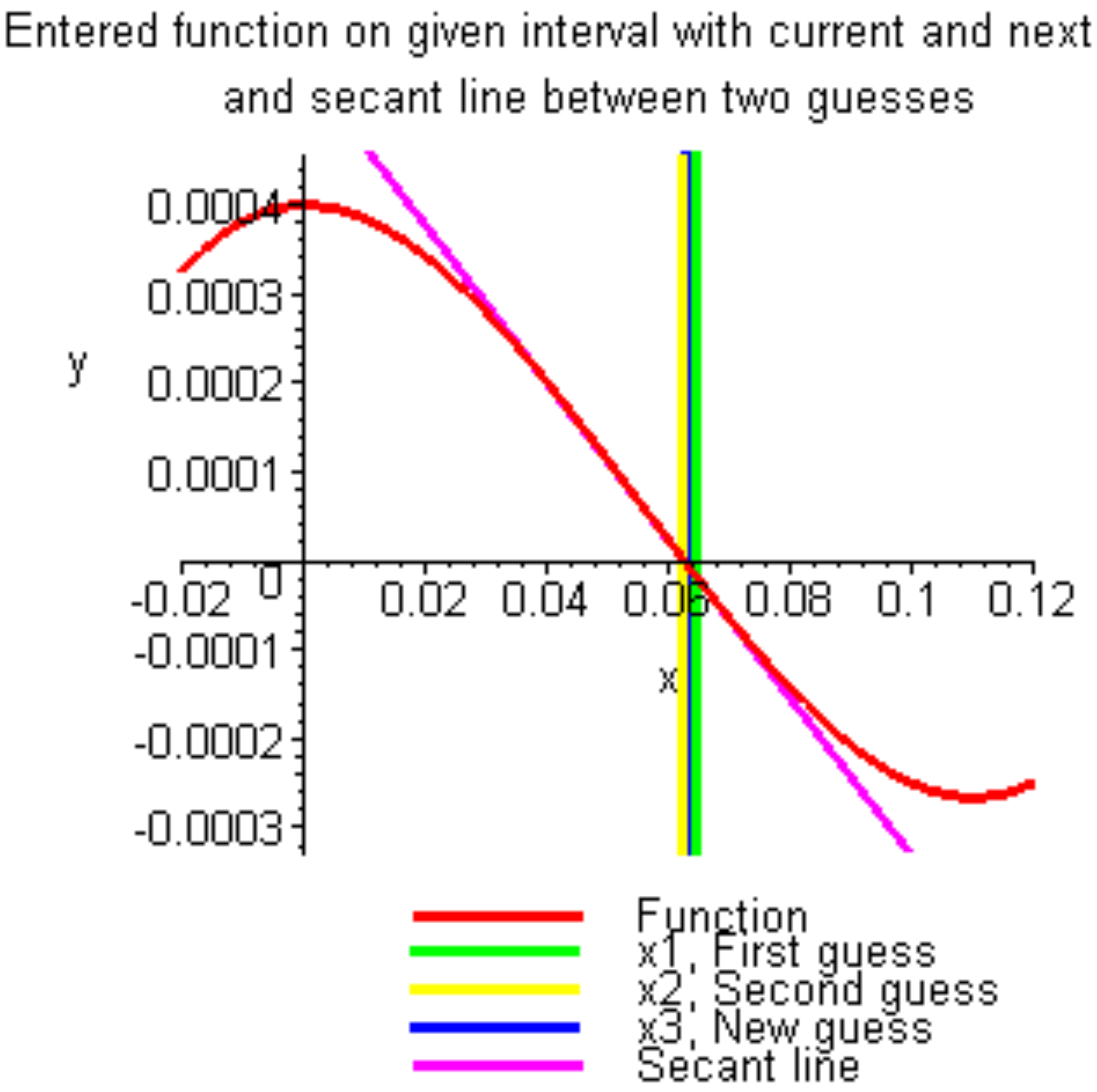
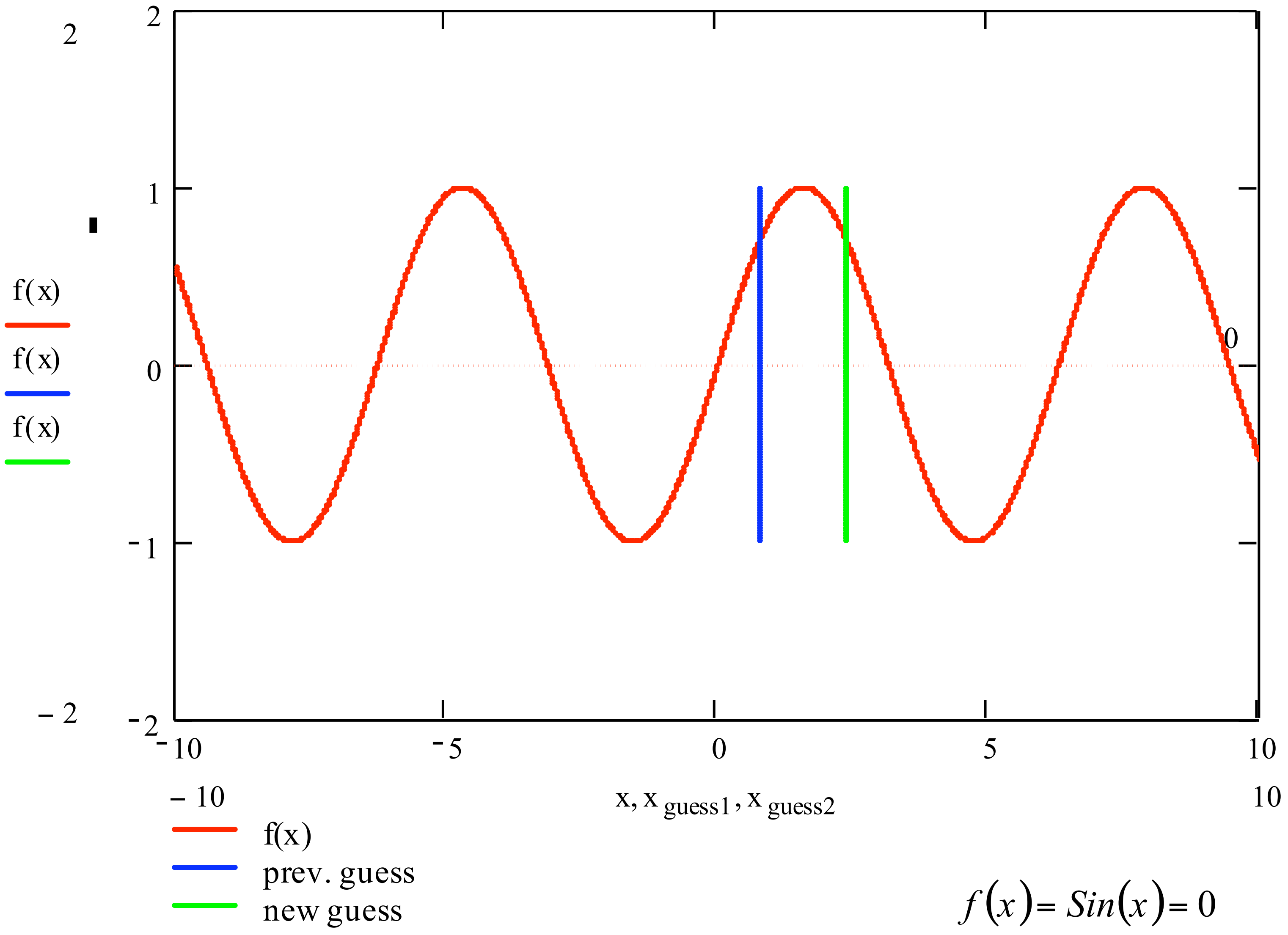


Figure 7 Graph of results of Iteration 3.

Advantages

- Converges fast, if it converges
- Requires two guesses that do not need to bracket the root

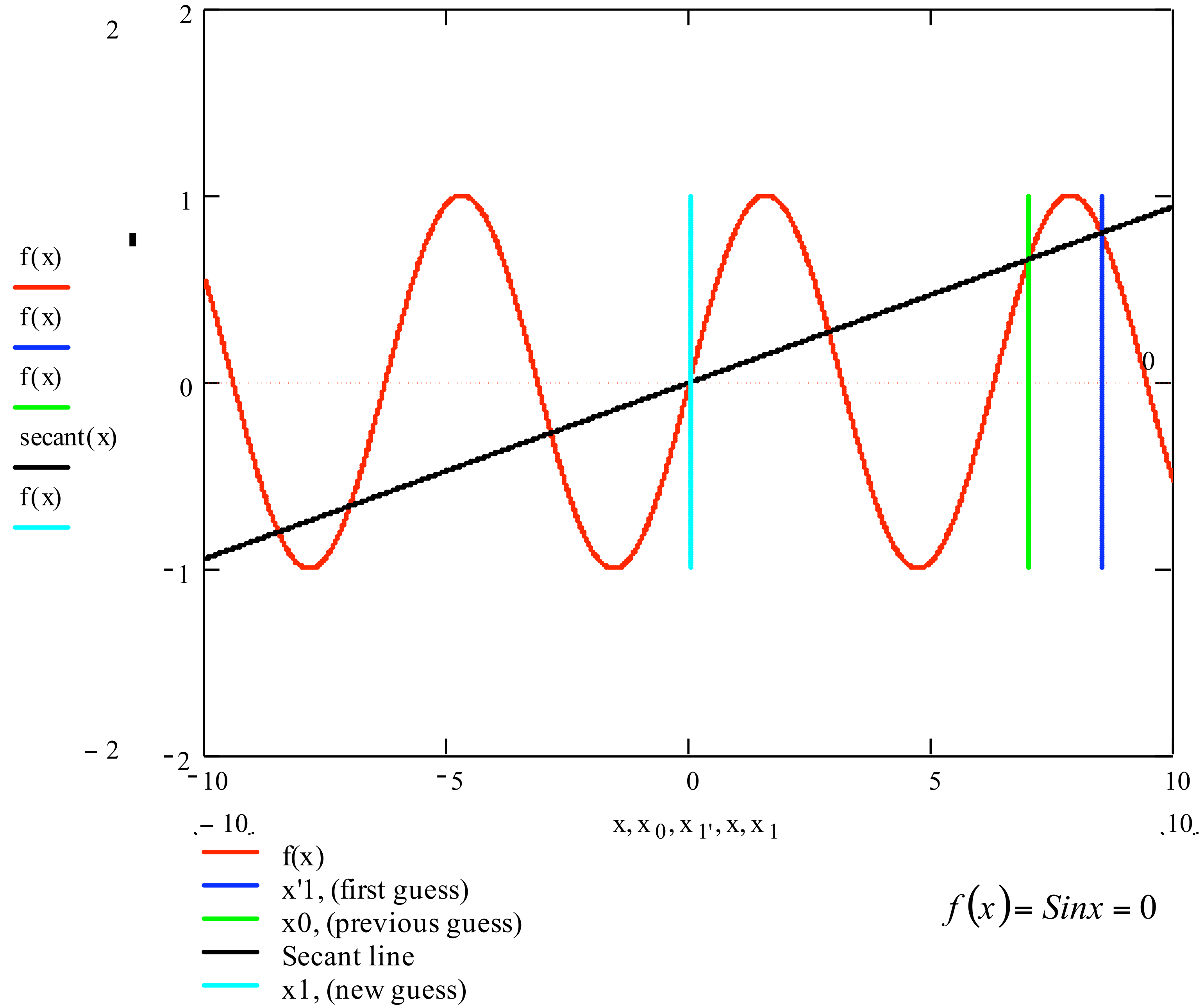
Drawbacks



$f(x) = Sin(x) = 0$

Division by zero

Drawbacks (continued)



$$f(x) = Sinx = 0$$

Root Jumping

Additional Resources

For all resources on this topic such as digital audiovisual lectures, primers, textbook chapters, multiple-choice tests, worksheets in MATLAB, MATHEMATICA, MathCad and MAPLE, blogs, related physical problems, please visit

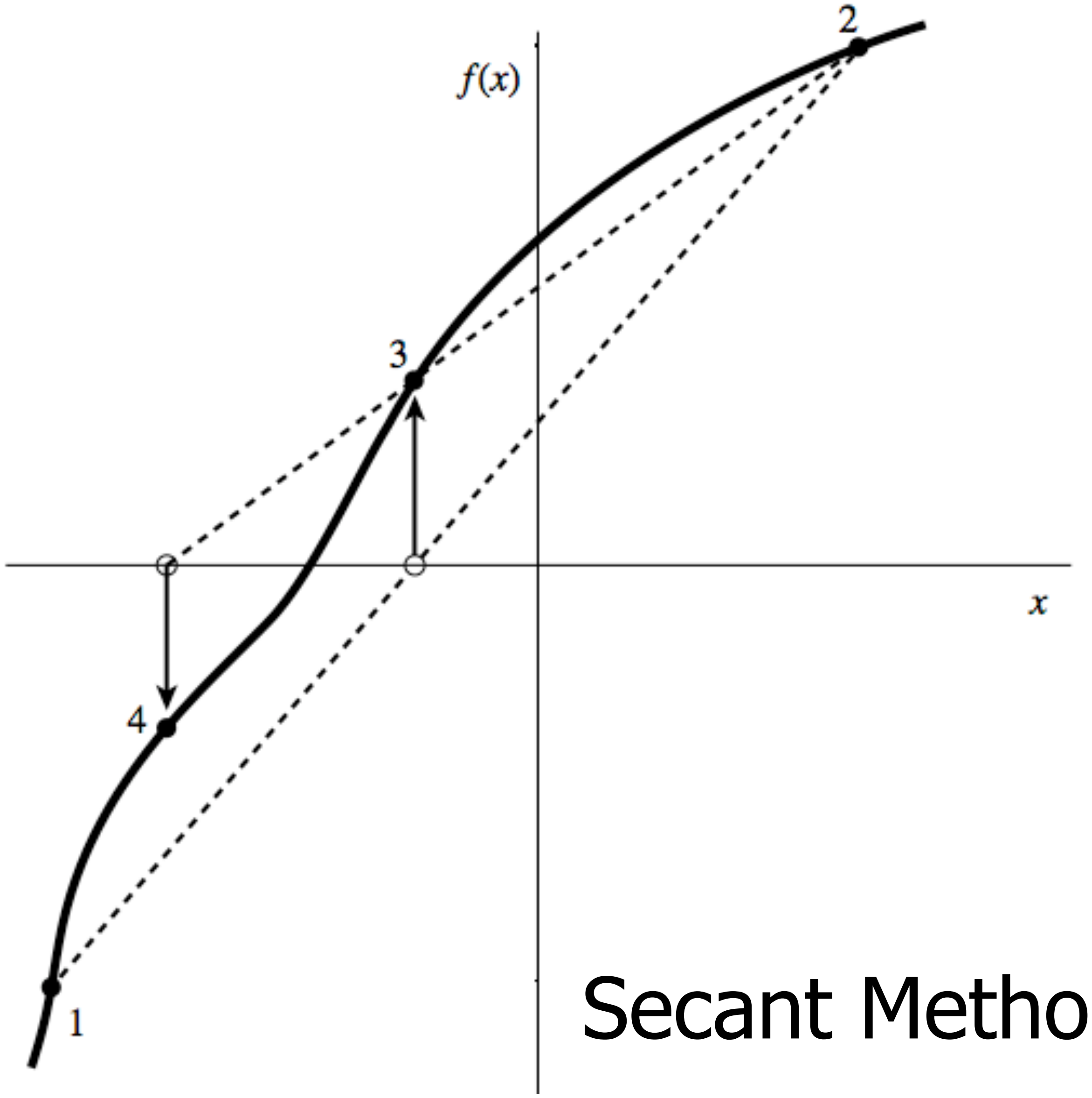
http://numericalmethods.eng.usf.edu/topics/secant_method.html

THE END

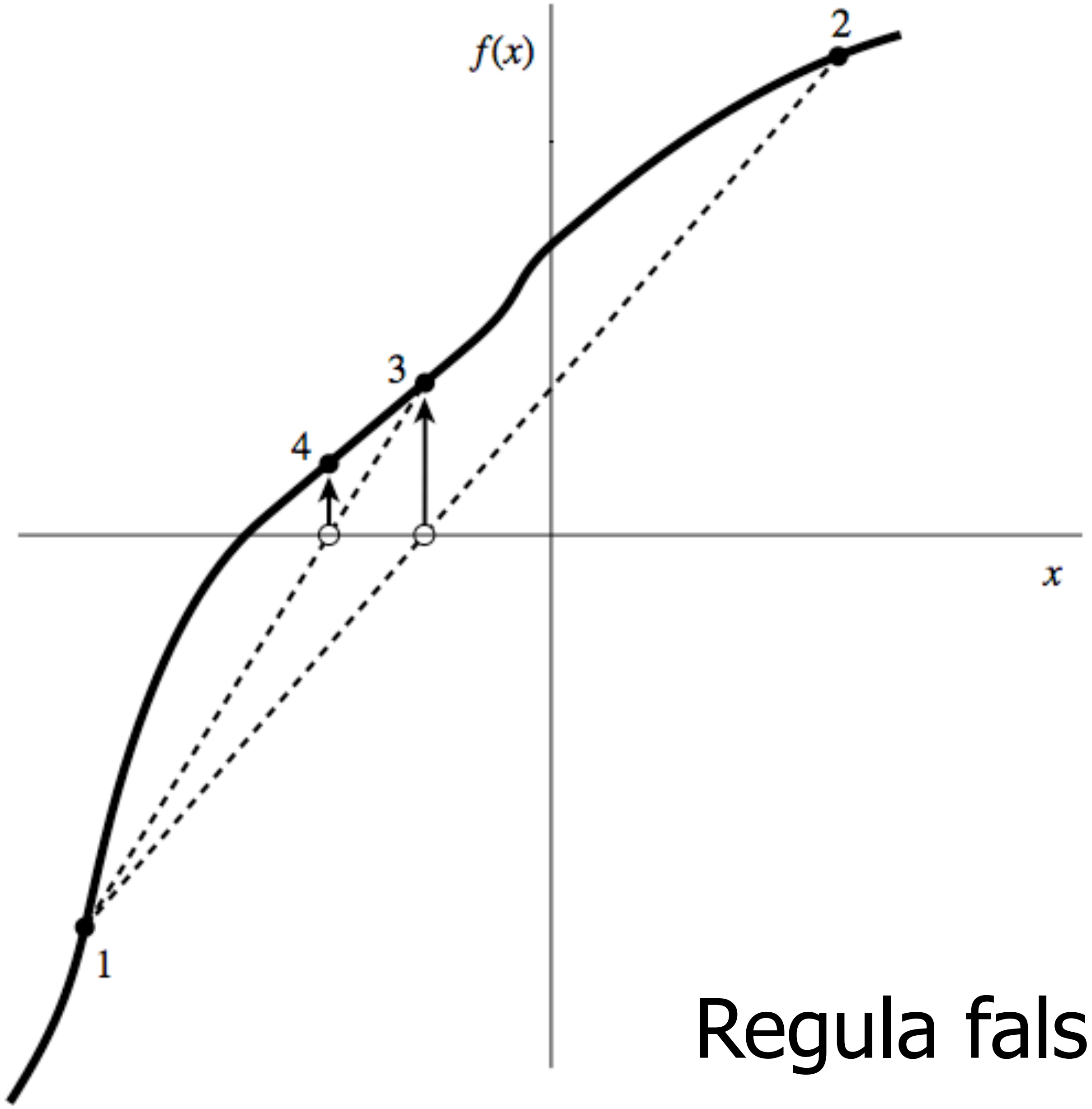
Regula Falsi
(False Postion Method)

- similar to secant method (linear approximation)
- keeps the point x_i of prior estimate to estimate the new value with opposite sign (secant method: uses newly evaluted x_{i+1})
- brackets the root

Regula Falsi
(False Postion)



Secant Method



Regula falsi

Algorithm for Regula Falsi Method

Step 0

choose two inital guesses x_{-1} and x_0 bracketing the root

assume

$$f(x_{-1}) < f(x_0)$$

Step 1

Calculate the next estimate of the root from two initial guesses

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

if $f(x_{i+1}) > 0$ then $x_i = x_{i+1}$
else replace appropriete limits

Step 2

Find the absolute relative approximate error

$$\epsilon = \left| \frac{x_{i-1} - x_i}{x_{i-1}} \right|$$

If the absolute relative approximate error is greater than the prespecified relative error tolerance go back to step 1 else stop.

Also check if the number of iterations has exceeded the maximum number of iterations.

Advantages

- always converges (unlike secant method)

Disvantages

- root must be bracketed in the initial guess
- converges slower than Secant method

Resources

- Based on: http://numericalmethods.eng.usf.edu
  von Autar Kaw, Jai Paul

  - Recommended:
    Numerical Recipes (2nd/3rd Edition). Press et al.,
    Cambridge University Press
    http://www.nr.com/oldverswitcher.html

Root finding

Return by 9:15 a.m. tomorrow

Free Training

- Write a program code to calculate the root $x_\star$ (i.e., with $f(x_\star) = 0$) of some given function f(x) using the four methods presented in the lecture:

  1. Bisection method
  2. Regula falsi (Interpolation)
  3. Newton-Raphson
  4. Secant method

- Solve the quadratic equation $x^2 + x - 1 = 0$ using these four methods. Plot the graph first to find proper starting values for the iteration.

Assignment for the Afternoon / Homework

- **Exercise 1, 8 points**: Convergence (I).
  Solve the equation $f(x) = \cos(x) - \frac{1}{4}$ with your program using $x_a = 0$ and $x_b = \pi/2$ (for methods 1, 2 and 4) and $x_a = \pi/2$ (for Newton-Raphson) as initial values. Calculate and write out both the absolute true error ($|e_n| = |x_n - x_*|$) and the absolute true relative error $|f_n| = |x_n - x_*|/|x_*|$ for the first 20 iterations.

- **Exercise 2, 6 points**: Convergence (II).
  Plot $|e_n/e_0|$, where e_0 is the error for $n = 0$, versus the iteration step n from the data in Ex. 1 in a logarithmic plot. Compare your results to the expected convergence behaviour of these methods.

1. Bisection method
2. Regula falsi (Interpolation)
3. Newton-Raphson
4. Secant method

- Solve the quadratic equation $x^2 + x - 1 = 0$ using these four methods. Plot the graph first to find proper starting values for the iteration.

Assignment for the Afternoon / Homework

- **Exercise 1, 8 points**: Convergence (I).
  Solve the equation $f(x) = \cos(x) - \frac{1}{4}$ with your program using $x_a = 0$ and $x_b = \pi/2$ (for methods 1, 2 and 4) and $x_a = \pi/2$ (for Newton-Raphson) as initial values. Calculate and write out both the absolute true error ($|e_n| = |x_n - x_*|$) and the absolute true relative error $|f_n| = |x_n - x_*|/|x_*|$ for the first 20 iterations.

- **Exercise 2, 6 points**: Convergence (II).
  Plot $|e_n/e_0|$, where e_0 is the error for $n = 0$, versus the iteration step n from the data in Ex. 1 in a logarithmic plot. Compare your results to the expected convergence behaviour of these methods.

- **Exercise 3, 6 points**: Double-well potential
  $f(x) = 0.1\,x^4 - 4\,x^2 - 10$. Double-well potentials play an important role in quantum mechanics and molecular dynamics to describe the motion of a particle in the force field of two others. Determine the root of $f(x)$ with the four root-finding methods. Start the iteration with the initial values $x_a = 2.5$ and $x_b = 7.0$ (use $x_a = 2.5$ for the Newton-Raphson algorithm). Discuss the results of the four methods in terms of their convergence behaviour. Give an example in which the bisection method cannot be applied or would give a wrong result (i.e., not converge to a root of $f(x)$).