بسم ا الر ن الرحيم

DEPARTMENT OF CIVIL ENGINEERING – KFUPM

Numerical and Statistical Methods in Civil Engineering

CE 318- 51- 2011-2012 (111)

Computer Lab. Sessions NO. 06 & 07

Subj.: ***Mathematica*** **software to perform numerical inegrations & EXCEL Sheet to determine the roots of a nonlinear equation**

DATE: Oct. 24,'11

Objective: Objective: to determine roots of nonlinear equations using available computer softwares (e.g. Excel and *Mathematica* software©).

---

1. Use the ***Mathematica software*** to perform *numerical integration* of the following functions.

$$f(x) = x \cos x \quad \text{for inegral limits } x_l = 1 \ \textit{and} \ x_u = 4.$$

Compare the *numerical results* with *exact solution* using integartion by parts formula (i.e. $\int u\, dv = vu - \int v\, du$).

2. Use the ***EXCEL Solver*** ***and Mathematica software to find the root of the following functions*** **(to a tolerance of 1×10^{-5}):**

(a) $f(x) = x^3 - 3x^2 + 2x - 0.375$ (root in vicinity of 1.0).

(b) $f(x) = \sin(x) + \cos(1+x^2) - 1$ (find only first *positive* root).

Procedure for Lab.-Report Evaluation:

1. Start working in the assigned session, then complete your computer works *preferably* within the session *or* shortly afterwards using the same computing machine on which you may save your work for future use (if necessary).
2. Submit for evaluation your summary of organized computer work assignment in the beginning of the next lab.
3. Your report *should* include: i)**Introduction** expalining the work undertaken and its main objectives; ii) Clear outline of the numerical procedure(s) used; iii) **Print-out** of the work completed; and iv) **Summary** and conclusions.

Supporting Notes:
Numerical Example using Mathematica:

Background on "Gauss-Legendre Quadrature (Numerical Integration):

To approximate the integral

$$\int_{-1}^{1} f[x]\, dx \approx \sum_{k=1}^{n} w_{n,k}\, f[x_{n,k}]$$

by *sampling* $f[x]$ at the n *unequally spaced abscissas* $x_{n,1}, x_{n,2}, \cdots, x_{n,n}$, where the corresponding weights are $w_{n,1}, w_{n,2}, \cdots, w_{n,n}$. The abscissas and weights are obtained from a table of values. The method is attributed to Johann Carl Friedrich Gauss (1777-1855) and Adrien-Marie Legendre (1752-1833).

Example 1:

Use the **Gauss-Legendre quadrature** rules for ***n*** **= 2, 3, and 4** points to compute numerical approximations for

$$I = \int_{-1}^{1} e^{-x^2/2}\, dx$$

Solution 1:

First, enter the formula $\text{Exp}\left[\frac{-x^2}{2}\right]$ or $E^{-x^2/2}$ or $e^{-x^2/2}$.

```
f[x_] = e^(-x^2/2);
Print["f[x] = ", f[x]];
```

$f[x] = e^{-\frac{x^2}{2}}$

Solution using Gauss-Legendre quadrature with **n = 2**:

```
w1 =  1.0 ;
w2 =  1.0 ;
x1 = -0.577350269189625 ;
x2 =  0.577350269189625 ;

Q2 = w1 f[x1] + w2 f[x2]
```

```
1.69296
```

Solution using Gauss-Legendre quadrature with **n = 3**:

```
w1 =  0.555555555555556 ;
w2 =  0.888888888888889 ;
w3 =  0.555555555555556 ;
x1 = -0.774596669241483 ;
x2 =  0.0 ;
x3 =  0.774596669241483 ;

Q3 = w1 f[x1] + w2 f[x2] + w3 f[x3]
```

1.71202

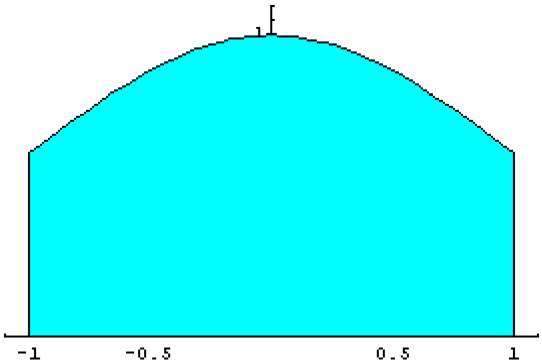
Solution using Gauss-Legendre quadrature with **n = 4**:

```
w1 =  0.347854845137453 ;
w2 =  0.652145154862546 ;
w3 =  0.652145154862546 ;
w4 =  0.347854845137453 ;
x1 = -0.861136311594053 ;
x2 = -0.339981043584856 ;
x3 =  0.339981043584856 ;
x4 =  0.861136311594053 ;

Q4 =  w1 f[x1] + w2 f[x2] + w3 f[x3] + w4 f[x4]
```

1.71122

```
line1 = Graphics[{Line[{{-1, 0}, {-1, f[-1]}}]}];
line2 = Graphics[{Line[{{1, 0}, {1, f[1]}}]}];
Needs["Graphics`Colors`"];
Needs["Graphics`FilledPlot`"];
graph = Plot[f[x], {x, -1, 1},
        PlotRange → {{-1.1, 1.1}, {0, 1.1}}];
gr1 = FilledPlot[{f[x]}, {x, -1, 1},
        Fills → {{{1, Axis}, Cyan}}];
Show[graph, gr1, line1, line2];
Print["The area under y = ", f[x]];
Print["is approximately ", Q4];
```



The area under y = $e^{-\frac{x^2}{2}}$

is approximately 1.71122

The shifted Gauss-Legendre rule for [a,b]. To approximate the integral $\int_a^b f[t]\, dt$ use the change of variable $t = \frac{a+b}{2} + \frac{b-a}{2} x$ and $dt = \frac{b-a}{2} dx$. Then use $g[x] = f\left[\frac{a+b}{2} + \frac{b-a}{2} x\right]$ and apply the Gauss-Legendre rules for $\frac{b-a}{2} \int_{-1}^{1} g[x]\, dx$.

Exercise 3: Use the shifted Gauss-Legendre rules for n = 3 points to approximate the integrals

Illustrate the comparisons for the integral $\int_0^1 e^{-x^2/2} dx$, $\int_1^2 e^{-x^2/2} dx$ and $\int_2^3 e^{-x^2/2} dx$.

Solution. Enter the abscissas and weights. Copy them from Exercise 1 and make sure they are activated!

```
w1 =  0.555555555555556 ;
w2 =  0.888888888888889 ;
w3 =  0.555555555555556 ;
x1 = -0.774596669241483 ;
x2 =  0.0 ;
x3 =  0.774596669241483 ;
```

Exercise 3 (a: Find the integral over [0,1]

```
a = 0 ;
b = 1 ;
```

$$g[x\_] = f\left[\frac{a+b}{2} + \frac{b-a}{2} x\right];$$

$$Q3 = \frac{b-a}{2} (w1\, g[x1] + w2\, g[x2] + w3\, g[x3])$$

```
0.855626
```

Compare with *Mathematica*'s calculation.

```
v1 = NIntegrate[f[x], {x, a, b}]
v1 - Q3

0.855624
```

-2.00215×10^{-6}

Exercise 3 (b): Find the integral over [1,2]

```
a = 1 ;
b = 2 ;
```

$$g[x\_] = f\left[\frac{a+b}{2} + \frac{b-a}{2} x\right];$$

$$Q3 = \frac{b-a}{2} (w1\, g[x1] + w2\, g[x2] + w3\, g[x3])$$

```
0.34066
```

Compare with *Mathematica*'s calculation.

```
v2 = NIntegrate[f[x], {x, a, b}]
v2 - Q3

0.340664

3.40802 × 10^-6
```

Exercise 3 (c):

Find the integral over [2, 3]

```
a = 2;
b = 3;
```

$$g[x\_] = f\left[\frac{a+b}{2} + \frac{b-a}{2}\,x\right];$$

$$Q3 = \frac{b-a}{2}\,(w1\,g[x1] + w2\,g[x2] + w3\,g[x3])$$

```
0.053644
```

Compare with *Mathematica*'s calculation.

```
v3 = NIntegrate[f[x], {x, a, b}]
v3 - Q3

0.0536424

- 1.58428 × 10^-6
```

What famous numbers do you recognize in the following list ?

$$N\left[\frac{1}{\sqrt{2\pi}}\{v1, v2, v3\}\right]$$

```
{0.341345, 0.135905, 0.0214002}
```

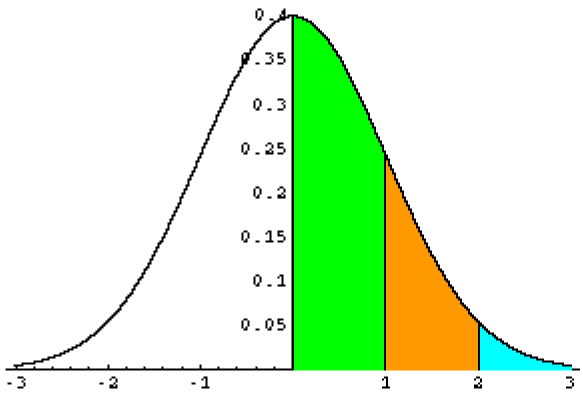
Or perhaps the following list ?

$$N\left[\frac{1}{\sqrt{2\pi}}\{v1, v2 + v1, v3 + v1 + v2\}\right]$$

```
{0.341345, 0.47725, 0.49865}
```

$$g[x\_] = \frac{1}{\sqrt{2\pi}}\,f[x];$$

```
line0 = Graphics[{Line[{{0, 0}, {0, g[0]}}]}];
line1 = Graphics[{Line[{{1, 0}, {1, g[1]}}]}];
line2 = Graphics[{Line[{{2, 0}, {2, g[2]}}]}];
gr0 = Plot[g[x], {x, -3, 0},
  PlotRange → {{-3.1, 3.1}, {0, 0.4}}];
gr1 = FilledPlot[{g[x]}, {x, 0, 1},
  Fills → {{{1, Axis}, Green}}];
gr2 = FilledPlot[{g[x]}, {x, 1, 2},
  Fills → {{{1, Axis}, Orange}}];
gr3 = FilledPlot[{g[x]}, {x, 2, 3},
  Fills → {{{1, Axis}, Cyan}}];
Show[gr0, gr1, gr2, gr3, line0, line1, line2];
```



Exercise 4: Investigate the truncation error bound formulas for the Gauss-Legendre quadrature rules of n = 2, 3, and 4 points.

Use the integral $\int_{-1}^{1} e^{-x^2/2}\,dx$ for the investigation

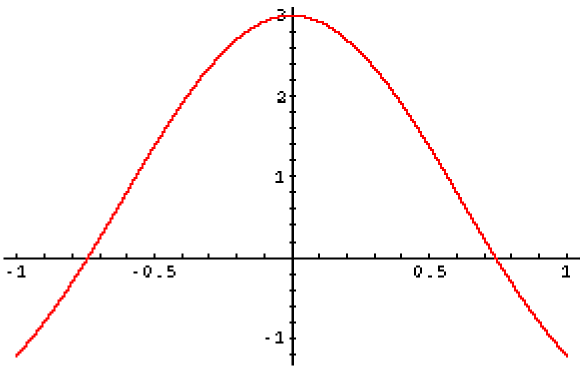
Solution. Use the quadrature values obtained in Exercise 1.

Q2 = 1.692963449781228922
Q3 = 1.712020245201909766
Q4 = 1.711224504599488499

And use the "true" numerical value of the integral as found in Exercise 2
v = 1.7112487837842944

Use symbolic differentiation, and a graph to determine the bound $M4 = \max_{-1 \le x \le 1} | f^{(4)}[x] |$.

```
f4[x_] = Expand[D[f[x], {x, 4}]];
M4 = Max[{Abs[f4[-1]], Abs[f4[0]], Abs[f4[1]]}];
Plot[f4[x], {x, -1, 1}, PlotStyle → Red];
Print["f''''[x] = ", f4[x]];
Print["M4 = ", M4];
```

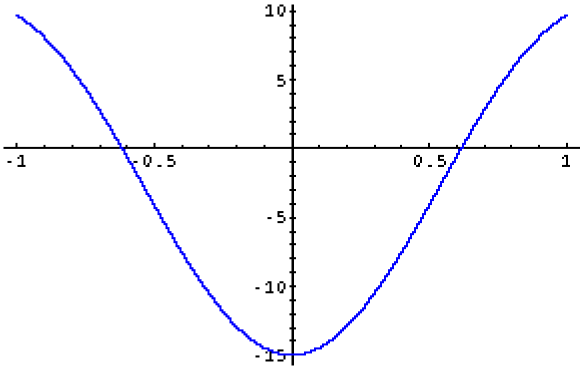


$f''''[x] = 3E^{-\frac{x^2}{2}} - 6E^{-\frac{x^2}{2}} x^2 + E^{-\frac{x^2}{2}} x^4$

M4 = 3

Use symbolic differentiation, and a graph to determine the bound $M6 = \max_{-1 \le x \le 1} | f^{(6)}[x] |$.

```
f6[x_] = Expand[D[f[x], {x, 6}]];
Plot[f6[x], {x, -1, 1}, PlotStyle → Blue];
M6 = Max[{Abs[f6[-1]], Abs[f6[0]], Abs[f6[1]]}];
Print["f''''''[x] = ", f6[x]];
Print["M6 = ", M6];
```

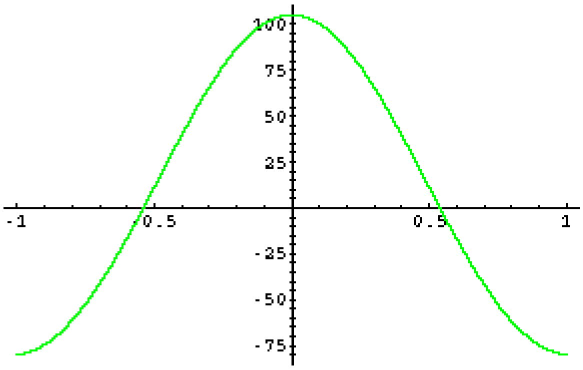


$$f''''''[x] = -15 E^{-\frac{x^2}{2}} + 45 E^{-\frac{x^2}{2}} x^2 - 15 E^{-\frac{x^2}{2}} x^4 + E^{-\frac{x^2}{2}} x^6$$

M6 = 15

Use symbolic differentiation, and a graph to determine the bound $M8 = \max_{-1 \le x \le 1} |f^{(8)}[x]|$.

```
f8[x_] = Expand[D[f[x], {x, 8}]];
Plot[f8[x], {x, -1, 1}, PlotStyle → Green];
M8 = Max[{Abs[f8[-1]], Abs[f8[0]], Abs[f8[1]]}];
Print["f''''''''[x] = ", f8[x]];
Print["M8 = ", M8];
```



$$f''''''''[x] = 105 E^{-\frac{x^2}{2}} - 420 E^{-\frac{x^2}{2}} x^2 + 210 E^{-\frac{x^2}{2}} x^4 - 28 E^{-\frac{x^2}{2}} x^6 + E^{-\frac{x^2}{2}} x^8$$

M8 = 105

Now compare the actual error and error bounds for the quadrature rules.

For n = 2, $|E_2(f)| \le \frac{M4}{135}$.

```
EQ2 = v - Q2
EB2 = M4/135.
```

0.0182853

0.0222222

For n = 3, $|E_3(f)| \le \frac{M6}{15750}$.

```
EQ3 = v - Q3
EB3 = M6/15750.
```

-0.000771461

0.000952381

For n = 4, $|E_4(f)| \le \frac{M8}{3472875}$.

```
EQ4 = v - Q4
EB4 = M8/3472875.
```

0.0000242792

0.0000302343

****

Using *Mathematica* to find roots of equations and to solve a system of equations.

Example 1. Find all the real solutions to the cubic equation $x^3 + 4x^2 - 10 = 0$.

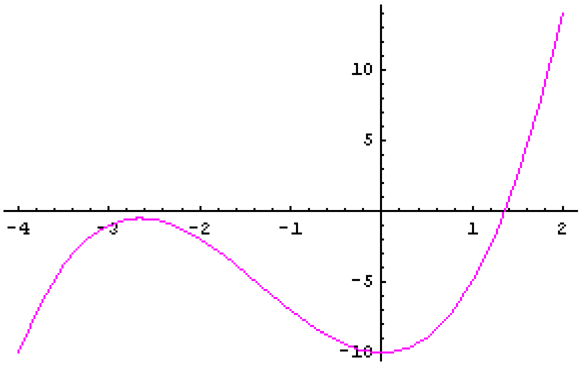
Solution 1.

```
f[x_] = x^3 + 4 x^2 - 10;
Print["f[x] = ", f[x]];
```

f[x] = $-10 + 4x^2 + x^3$

Plot the function.

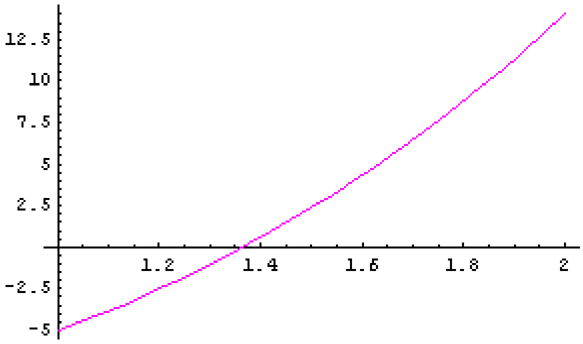
```
Needs["Graphics`Colors`"];
Plot[f[x], {x, -4, 2}, PlotStyle → Magenta];
Print["y = f[x] = ", f[x]];
```



y = f[x] = $-10 + 4x^2 + x^3$

There appears to be only one real root which lies in the interval [1,2].

```
Plot[f[x], {x, 1, 2}, PlotStyle → Magenta];
Print["y = f[x] = ", f[x]];
```



y = f[x] = $-10 + 4x^2 + x^3$

Call the **Bisection** subroutine on the interval [1,2] using 10 iterations

```
Bisection[1, 2, 10];
```

k	a_k	c_k	b_k	$f[c_k]$
0	1.	1.5	2.	2.375
1	1.	1.25	1.5	-1.796875
2	1.25	1.375	1.5	0.162109375
3	1.25	1.3125	1.375	-0.848388671875
4	1.3125	1.34375	1.375	-0.350982666015625
5	1.34375	1.359375	1.375	-0.0964088439941406
6	1.359375	1.3671875	1.375	0.03235578536987305
7	1.359375	1.36328125	1.3671875	-0.03214997053146362
8	1.36328125	1.365234375	1.3671875	0.00007202476263046265
9	1.36328125	1.3642578125	1.365234375	-0.01604669075459242
10	1.3642578125	1.36474609375	1.365234375	-0.007989262812770903

c = 1.36474609375
Δc = ±0.000488281
f[c] = -0.007989262812770903

After 10 iterations, the interval has been reduced to [a,b] where

```
Print["a = ", NumberForm[a, 16]];
Print["b = ", NumberForm[b, 16]];
Print[""];
Print["[a, b] = [", a, ", ", b, "]"];
```

a = 1.3642578125
b = 1.365234375

[a, b] = [1.36426, 1.36523]

The root lies somewhere in the interval [a,b] the width of which is

```
Print["b-a = ", NumberForm[b - a, 16] ];
b-a = 0.0009765625
```

The reported root is alleged to be

```
Print["c = ", NumberForm[c, 16] ];
c = 1.36474609375
```

The accuracy we can guarantee is one half of the interval width.

Print$\left["\frac{b-a}{2} = ", \text{NumberForm}\left[\frac{b-a}{2}, 16\right]\right];$

$\frac{b-a}{2} = 0.00048828125$

Ext ra Notes (roots finding using Mathematica):

The Bisection Method

Background. The bisection method is one of the bracketing methods for finding roots of equations.

Implementation. Given a function f(x) and an interval which might contain a root, perform a predetermined number of iterations using the bisection method.

Limitations. Investigate the result of applying the bisection method over an interval where there is a discontinuity. Apply the bisection method for a function using an interval where there are distinct roots. Apply the bisection method over a "large" interval.

Program for the Bisection Method

```
Bisection[a0_, b0_, m_] :=
 Module[{},
  a = N[a0];
  b = N[b0];
  c = (a + b)/2;
  k = 0;
  output = {{k, a, c, b, f[c]}};
  While[k < m,
   If[Sign[f[b]] == Sign[f[c]],
    b = c, a = c; ];
   c = (a + b)/2;
   k = k + 1;
   output = Append[output, {k, a, c, b, f[c]}]; ];
  Print[NumberForm[TableForm[output,
     TableHeadings -> {None, {"k", "ak", "ck", "bk", "f[ck]"}}], 16]];
  Print["   c   = ", NumberForm[c, 16] ];
  Print[" Δc   = ±", (b - a)/2];
  Print["f[c]  = ", NumberForm[f[c], 16] ]; ]
```

Example 1. Find all the real solutions to the cubic equation $x^3 + 4x^2 - 10 = 0$.

Concise Program for the Bisection Method

```
Bisection[a0_, b0_, m_] :=
 Module[{a = N[a0], b = N[b0]},
  c = (a + b)/2;
  k = 0;
  While[k < m,
   If[Sign[f[b]] == Sign[f[c]],
    b = c, a = c; ];
   c = (a + b)/2;
   k = k + 1; ];
  Print["   c   = ", NumberForm[c, 16] ];
  Print["  Δc   = ±", (b - a)/2 ];
  Print["f[c] = ", NumberForm[f[c], 16] ]; ];
```

Now test the example to see if it still works. Use the last case in Example 1 given above and compare with the previous results.

```
Bisection[1, 2, 30];
   c   = 1.365230013150722
  Δc   = ±4.65661 × 10^-10
f[c] = -4.349217874732858 × 10^-9
```

Reducing the Computational Load for the Bisection Method

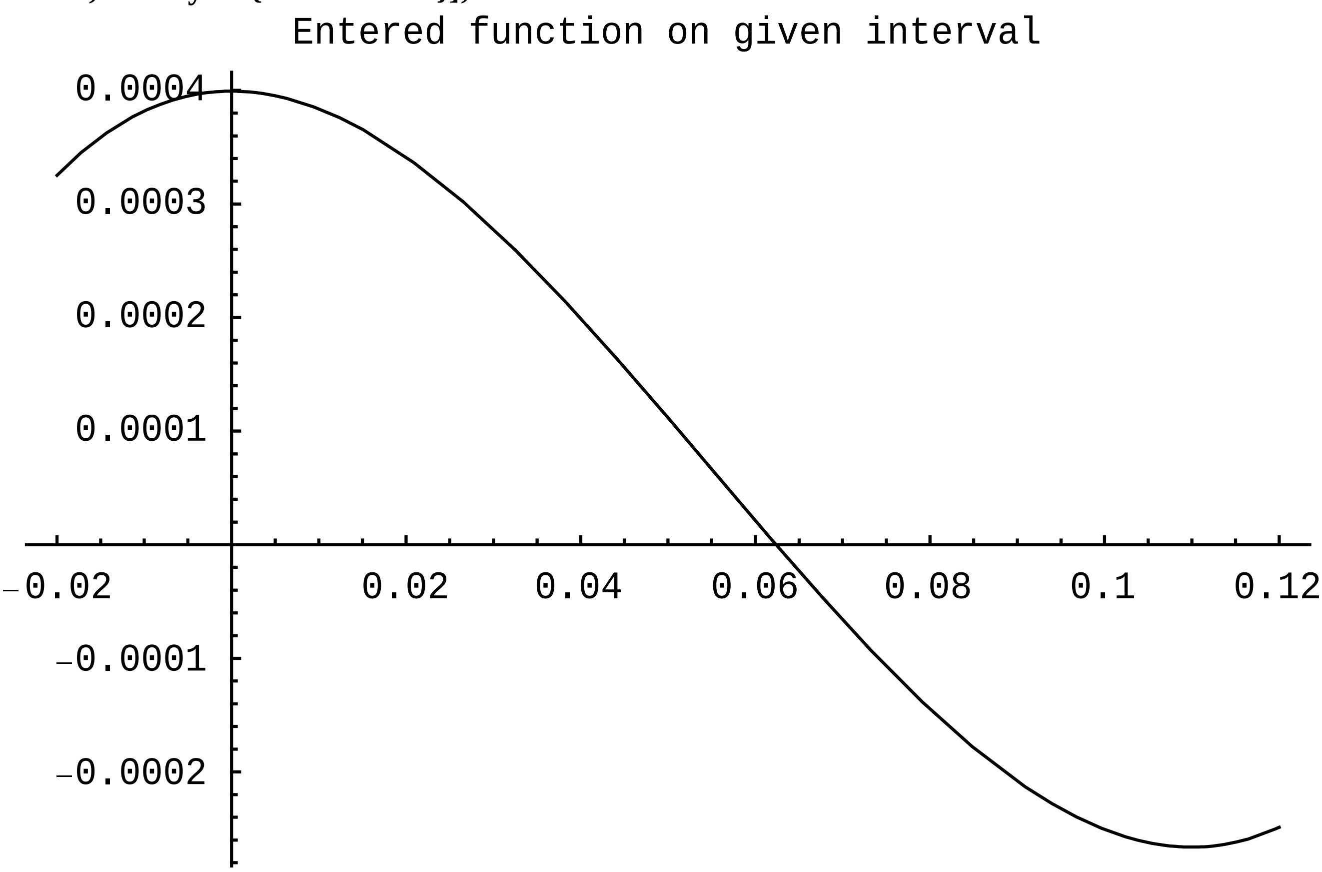
The following program uses fewer computations in the bisection method and is the traditional way to do it. Can you determine how many fewer functional evaluations are used?

```
Bisection[a0_, b0_, m_] :=
Module[{a = N[a0], b = N[b0]},
  c = (a + b)/2;
  Yb = f[b];
  Yc = f[c];
  k = 0;
  While[k < m,
   If[Sign[Yb] == Sign[Yc],
    b = c;
    Yb = Yc,
    a = c; ];
   c = (a + b)/2;
   Yc = f[c];
   k = k + 1; ];
 Print["   c   = ", NumberForm[c, 16] ];
 Print["  Δc   = ±", (b - a)/2 ];
 Print["f[c] = ", NumberForm[Yc, 16] ]; ]
```

Extra Notes:

Abstract: **This simulation shows how the bisection method for finding roots of an equation f[x]=0 works.**

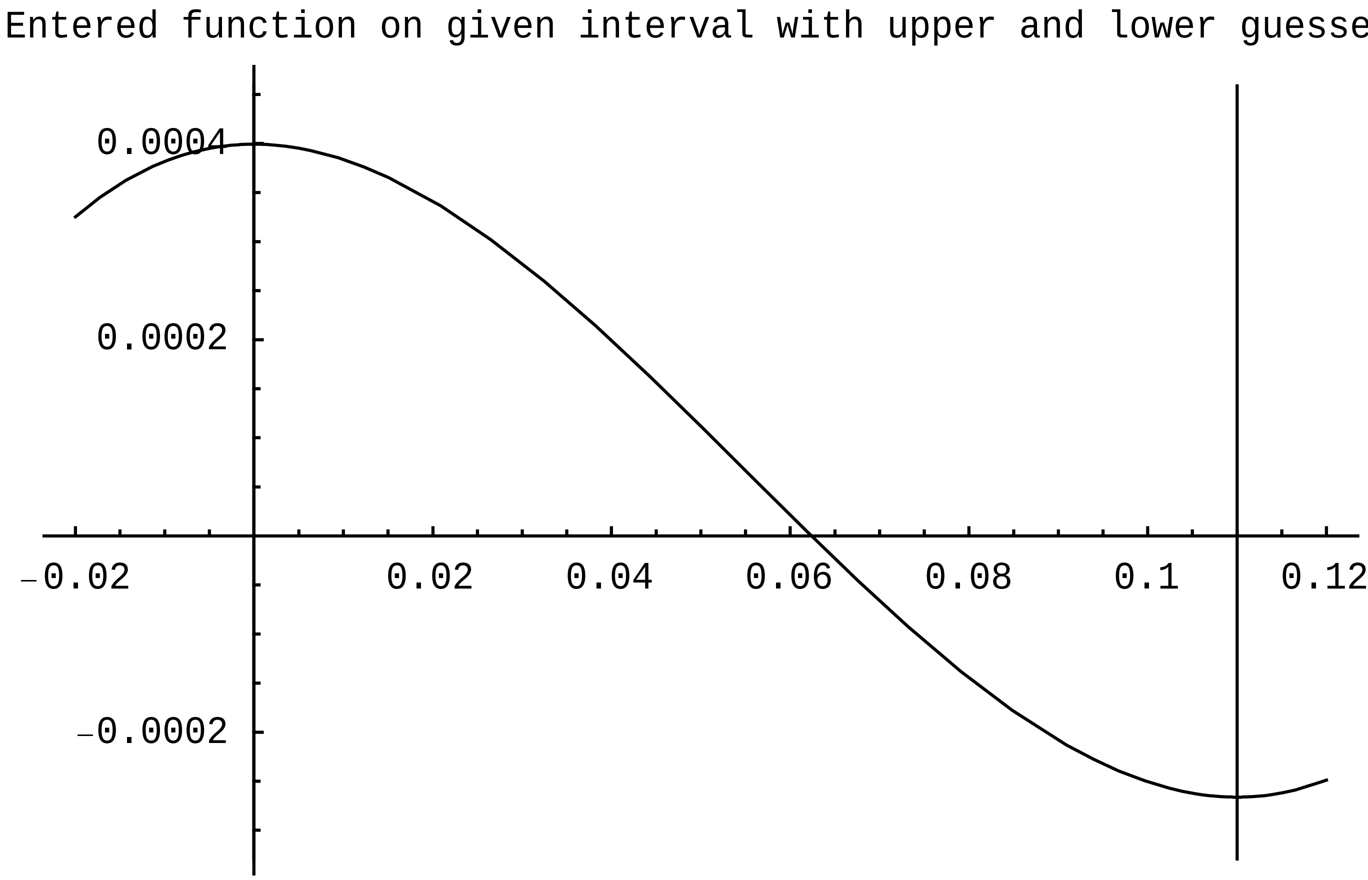
INPUTS: Enter the Following
Function in f[x]=0
f[x_]:=x^3-0.165*x^2+3.993*10^-4
Range of 'x' you want to see the function
x_b=-0.02;
x_e=0.12;
curve=Plot[f[x],{x,x_b,x_e},PlotLabel→"Entered function on given interval",TextStyle→{FontSize→11}];



Lower initial guess
x_l=0.0;
Upper initial guess
x_u=0.11;
SOLUTION
maxi=f[x_b];
mini = f[x_b];
step = (x_e-x_b)/10;
Do[If[f[i]> maxi,maxi=f[i]];If[f[i]< mini,mini=f[i]], {i,x_b,x_e,step}];
tot=maxi-mini;
mini=mini-0.1*tot;
maxi = maxi + 0.1*tot;
Check first if the lower and upper guesses bracket the root of the equation
f[x_l]
0.0003993
f[x_u]
-0.0002662
%*%%

-1.06294×10^{-7}

Show[Graphics[Line[{{x_u,maxi},{x_u,mini}}]],curve,Graphics[Line[{{x_l,maxi},{x_l,mini}}]], Axes→True,PlotLabel→"Entered function on given interval with upper and lower guesses",TextStyle→{FontSize→11}];



Iteration 1

New estimate of root

$x_r=(x_u+x_l)/2$

0.055

Finding the value of the function at the lower and upper guesses and the estimated root

f[x_l]

0.0003993

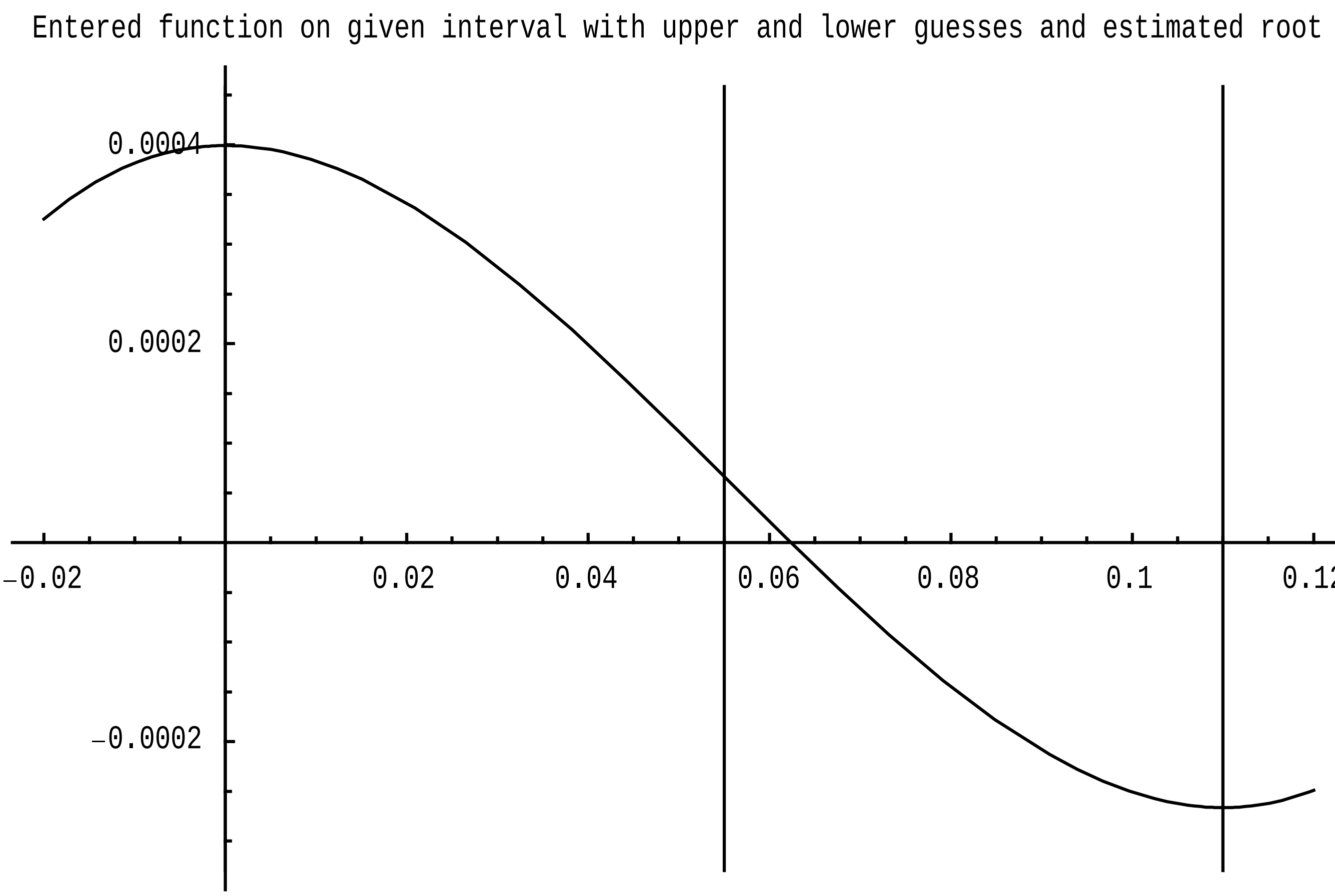
f[x_u]

-0.0002662

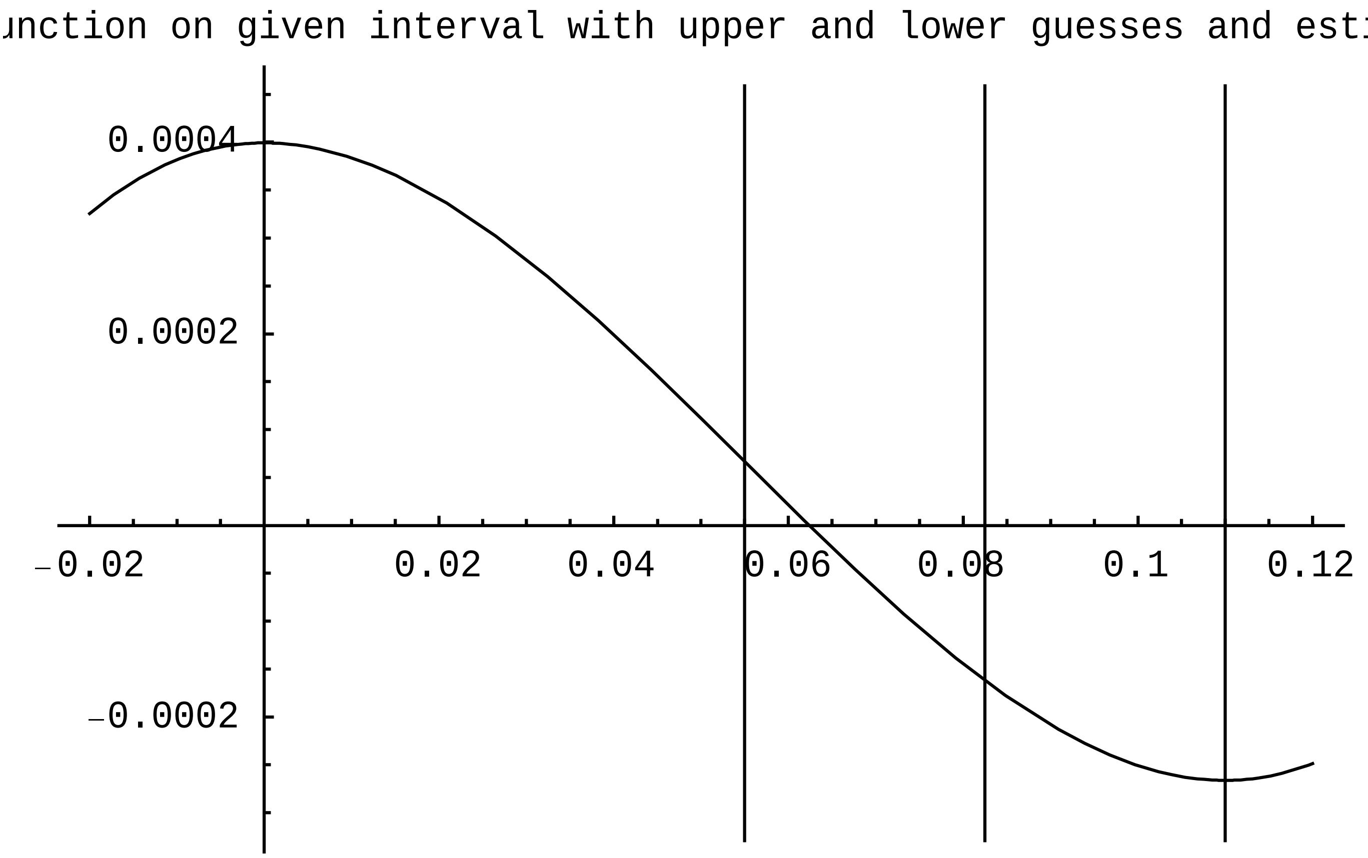
f[x_r]

0.00006655

Show[Graphics[Line[{{x_u,maxi},{x_u,mini}}]],curve,Graphics[Line[{{x_l,maxi},{x_l,mini}}]], Graphics[Line[{{x_r,maxi},{x_r,mini}}]],Axes→True,PlotLabel→"Entered function on given interval with upper and lower guesses and estimated root",TextStyle→{FontSize→9}];



Check the interval between which the root lies. Does it lie in (x_l,x_r) or (x_r,x_u)?
If[f[x_r]*f[x_u]≤0,x_l=x_r,x_u=x_r];
x_u
0.11
x_l
0.055
Store the value of the previous guess in $x_p$to calculate absolute relative approximate error.
x_p=x_r;
Iteration 2
New estimate of root
x_r=(x_l+x_u)/2
0.0825
Finding the value of the function at the lower and upper guesses and the estimated root
f[x_l]
0.00006655
f[x_u]
-0.0002662
f[x_r]
-0.000162216
Absolute relative approximate error, Abs[∊a].
∊a=Abs[(x_r-x_p)/x_r*100]
33.3333
Show[Graphics[Line[{{x_u,maxi},{x_u,mini}}]],curve,Graphics[Line[{{x_l,maxi},{x_l,mini}}]], Graphics[Line[{{x_r,maxi},{x_r,mini}}]],Axes→True,PlotLabel→"Entered function on given interval with upper and lower guesses and estimated root",TextStyle→{FontSize→11}];



Check the interval between which the root lies. Does it lie in (x_l,x_r) or (x_r,x_u)?

If[f[x_r]*f[x_u]≤0,x_l=x_r,x_u=x_r];

x_u

0.0825

x_l

0.055

Store the value of the previous guess in $x_p$to calculate absolute relative approximate error.

x_p=x_r;

Iteration 3

New estimate of root

x_r=(x_l+x_u)/2

0.06875

Finding the value of the function at the lower and upper guesses and the estimated root

f[x_l]

0.00006655

f[x_u]

-0.000162216

f[x_r]

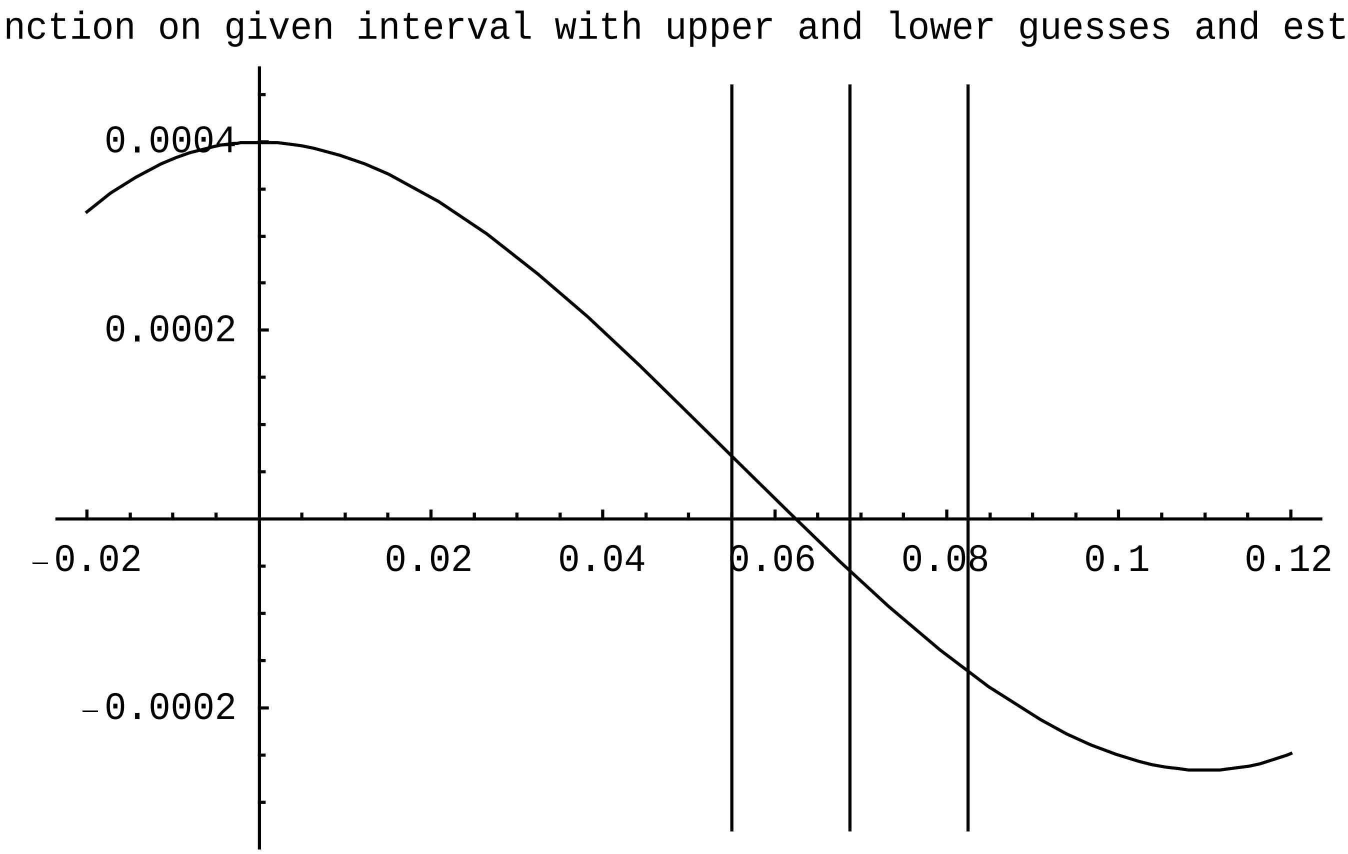
-0.0000556316

Absolute relative approximate error, Abs[ϵa].

ϵa=Abs[(x_r-x_p)/x_r*100]

20.

Show[Graphics[Line[{{x_u,maxi},{x_u,mini}}]],curve,Graphics[Line[{{x_l,maxi},{x_l,mini}}]], Graphics[Line[{{x_r,maxi},{x_r,mini}}]],Axes→True,PlotLabel→"Entered function on given interval with upper and lower guesses and estimated root",TextStyle→{FontSize→11}];



Check the interval between which the root lies. Does it lie in (x_l,x_r) or (x_r,x_u)?

If[f[x_r]*f[x_u]≤0,x_l=x_r,x_u=x_r];

x_u

0.06875

x_l

0.055

Store the value of the previous guess in $x_p$to calculate absolute relative approximate error.

x_p=x_r;

Iteration 4

New estimate of root

x_r=(x_l+x_u)/2

0.061875

Finding the value of the function at the lower and upper guesses and the estimated root

f[x_l]

0.00006655

f[x_u]

-0.0000556316

f[x_r]

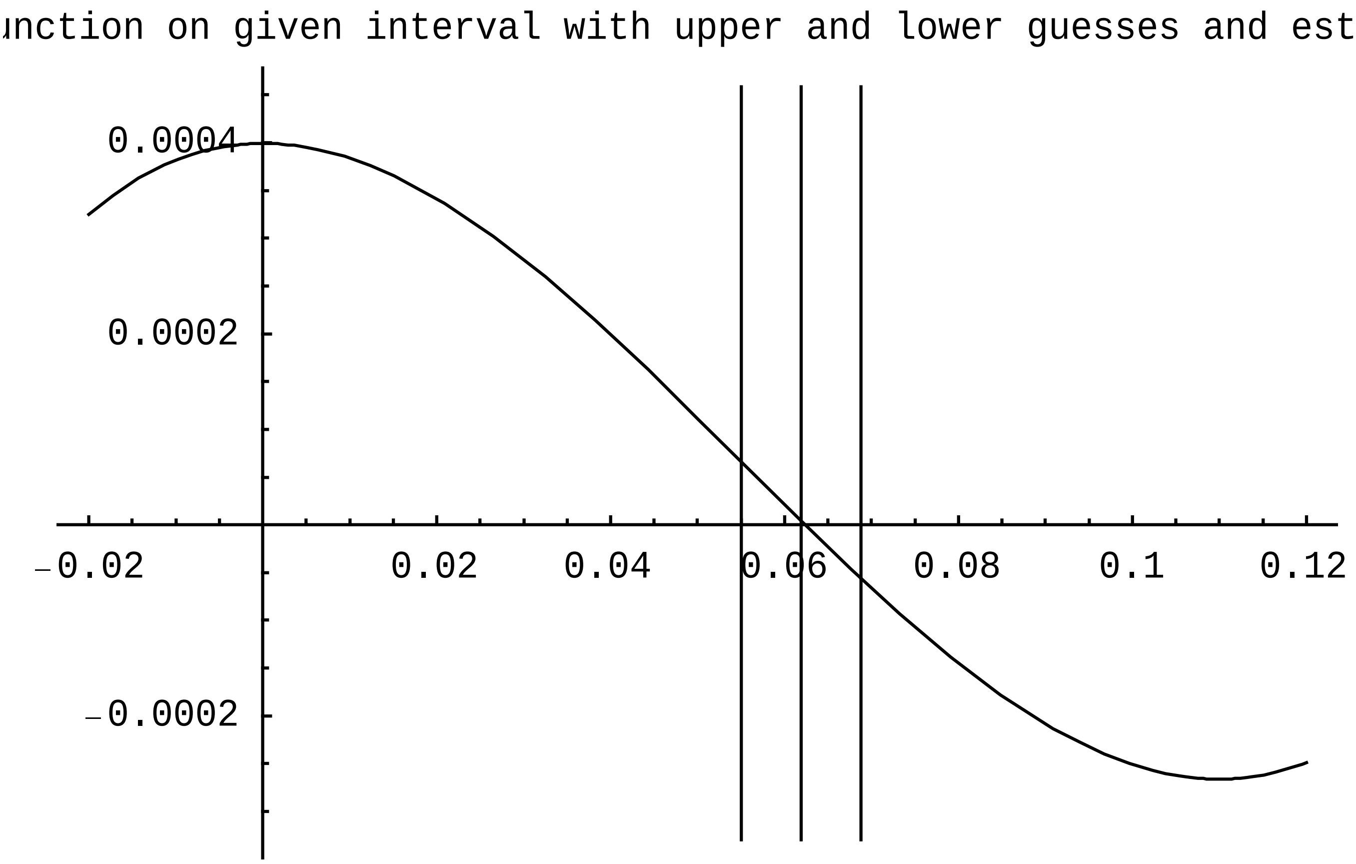
4.48433×10^{-6}

Absolute relative approximate error, Abs[ϵa].

ϵa=Abs[(x_r-x_p)/x_r*100]

11.1111

Show[Graphics[Line[{{x_u,maxi},{x_u,mini}}]],curve,Graphics[Line[{{x_l,maxi},{x_l,mini}}]], Graphics[Line[{{x_r,maxi},{x_r,mini}}]],Axes→True,PlotLabel→"Entered function on given interval with upper and lower guesses and estimated root",TextStyle→{FontSize→11}];



Check the interval between which the root lies. Does it lie in (x_l,x_r) or (x_r,x_u)?

If[f[x_r]*f[x_u]≤0,x_l=x_r,x_u=x_r];

x_u

0.06875

x_l

0.061875

Store the value of the previous guess in $x_p$to calculate absolute relative approximate error.

x_p=x_r;

Iteration 5

New estimate of root

x_r=(x_l+x_u)/2

0.0653125

Finding the value of the function at the lower and upper guesses and the estimated root

f[x_l]

4.48433×10^{-6}

f[x_u]

-0.0000556316

f[x_r]

-0.0000259392

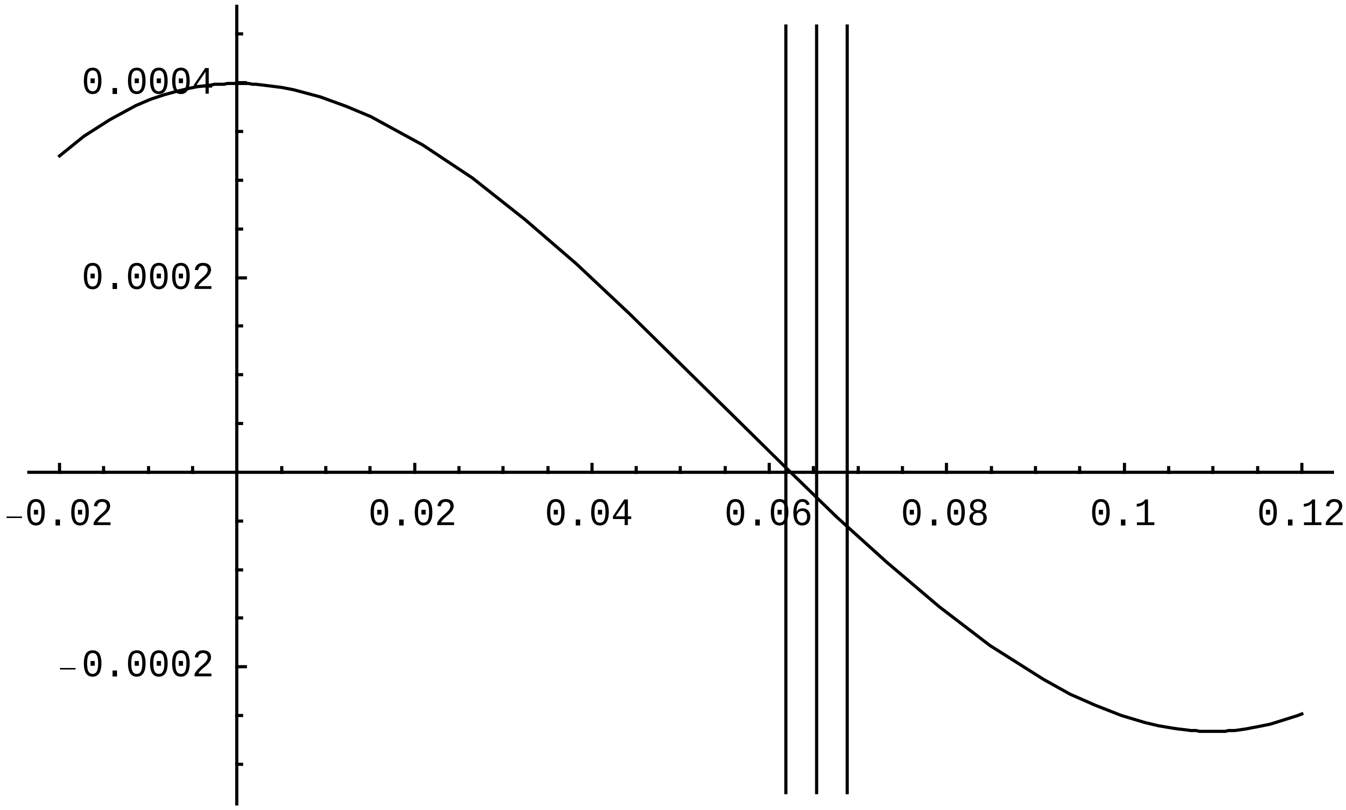
Absolute relative approximate error, Abs[∊a].

∊a=Abs[(x_r-x_p)/x_r*100]

5.26316

Show[Graphics[Line[{{x_u,maxi},{x_u,mini}}]],curve,Graphics[Line[{{x_l,maxi},{x_l,mini}}]], Graphics[Line[{{x_r,maxi},{x_r,mini}}]],Axes→True,PlotLabel→"Entered function on given interval with upper and lower guesses and estimated root",TextStyle→{FontSize→11}];

unction on given interval with upper and lower guesses and esti



Check the interval between which the root lies. Does it lie in (x_l,x_r) or (x_r,x_u)?

If[f[x_r]*f[x_u]≤0,x_l=x_r,x_u=x_r];

x_u

0.0653125

x_l

0.061875

Store the value of the previous guess in $x_p$to calculate absolute relative approximate error.

x_p=x_r;