**Name:** Ananya Kumari

**Roll no:** 2111739

**Course:** B.Sc. Mathematical Sciences

**Subject:** Numerical Methods Practical File

**Submitted To:** Mr. Dhanpal Singh & Mr. Panjabi Singh

# Practical 1(Bisection Method)

Ques 1. Find out the roots of the function f(x)=Cos(x) in the interval [0,2] using bisection method. Compute the approx value of the root after 14 iterations.

Solution:

f[x\_] := Cos[x] x0 = 0.0;

x1 = 2.0;

n = 14

14

If[f[x0] \* f[x1] > 0, Print[

"These Values do not fit in the given condition for IVT. Please change the values"], For[i = 1, i ≤ n, i ++, a = (x0 + x1) / 2; Print[ i, "th iteration value is: ", a];

If[f[x0] \* f[a] < 0, x1 = a, x0 = a];];]

1th iteration value is: 1. 2th iteration value is: 1.5 3th iteration value is: 1.75 4th iteration value is: 1.625

5th iteration value is: 1.5625 6th iteration value is: 1.59375 7th iteration value is: 1.57813 8th iteration value is: 1.57031 9th iteration value is: 1.57422 10th iteration value is: 1.57227 11th iteration value is: 1.57129 12th iteration value is: 1.5708 13th iteration value is: 1.57056 14th iteration value is: 1.57068

Ques 2. Find out the roots of the function f(x)=Cos(x) in the interval [0,2] using bisection method. Maximum iteration allowed are 20. Maximum error bound is 0.0001. Terminate the loop if any

**2** *Practical 1 (Bisection Method)[1].nb*

condition is satisfied. Solution:

f[x\_] := Cos[x] e = 0.0001;

x0 = 0.0;

x1 = 2.0;

n = 20;

If[f[x0] \* f[x1] > 0, Print[

"These values do not fit the given condition of IVT. Please change the values"], For[i = 1, i ≤ n, i ++, a = (x0 + x1) / 2;

If[Abs[(x1 - x0) / 2] < e, Return[a], Print[i "th iteration value is: ", a]; Print["Estimated error in", i, " th iteration is: ", (x1 - x0) / 2]; If[f[x0] \* f[a] < 0, x1 = a, x0 = a]]];

Print["Root is: ", a];

Print["Estimated error in ", i, " the iteration is: ", (x1 - x0) / 2]]

*Practical 1 (Bisection Method)[1].nb* **3**

th iteration value is: 1.

Estimated error in1 th iteration is: 1. 2 th iteration value is: 1.5

Estimated error in2 th iteration is: 0.5 3 th iteration value is: 1.75

Estimated error in3 th iteration is: 0.25 4 th iteration value is: 1.625

Estimated error in4 th iteration is: 0.125 5 th iteration value is: 1.5625

Estimated error in5 th iteration is: 0.0625 6 th iteration value is: 1.59375

Estimated error in6 th iteration is: 0.03125 7 th iteration value is: 1.57813

Estimated error in7 th iteration is: 0.015625 8 th iteration value is: 1.57031

Estimated error in8 th iteration is: 0.0078125 9 th iteration value is: 1.57422

Estimated error in9 th iteration is: 0.00390625 10 th iteration value is: 1.57227

Estimated error in10 th iteration is: 0.00195313 11 th iteration value is: 1.57129

Estimated error in11 th iteration is: 0.000976563 12 th iteration value is: 1.5708

Estimated error in12 th iteration is: 0.000488281 13 th iteration value is: 1.57056

Estimated error in13 th iteration is: 0.000244141 14 th iteration value is: 1.57068

Estimated error in14 th iteration is: 0.00012207

Return[1.57074]

# Practical 2 (Regula-Falsi)

Q1. Find approximate root of Cos(x) in the interval (0,2) in 10 iterations.

Solution:

f[x\_] := Cos[x] x0 = 0;

x1 = 2;

e = 0.000001;

iterations = 10;

Iff[x0] \* f[x1] > 0,

Print["These values do not satisfy the IVP so change the initial value."], Forn = 2, n ≤ iterations, n ++,

xn-2 \* f[xn-1] - xn-1 \* f[xn-2]

xn = N 

**;**

f[xn-1] - f[xn-2]

If[Abs[xn - xn-1] < e , Return[xn]]; Print[n - 1, "th iteration is: ", xn];

Print ["Estimated error: ", Abs[xn - xn-1]]; Print["Approximate Root ", xn]

1th iteration is: 1.41228 Estimated error: 0.587717 2th iteration is: 1.57391 Estimated error: 0.161623 3th iteration is: 1.57078 Estimated error: 0.0031228 4th iteration is: 1.5708

Estimated error: 0.0000128049

Return[1.5708]

Approximate Root 1.5708

Q2. Find approximate root of following equations using the secant method:

1. 3 - 3 *x*2 + 2 *x* + 5 = 0
2. Cos(x) - xe*x* = 0
3. *e*-*x* = *x*

**2** *Practical 2 (Regula Falsi).nb*

Solution:

1. 3 - 3 *x*2 + 2 *x* + 5 = 0

f[x\_] := x^ 3 - 3 x^ 2 + 2 x + 5 x0 = - 1;

x1 = 0;

e = 0.000001;

iterations = 10;

Iff[x0] \* f[x1] > 0,

Print["These values do not satisfy the IVP so change the initial value."], Forn = 2, n ≤ iterations, n ++,

xn-2 \* f[xn-1] - xn-1 \* f[xn-2]

xn = N 

**;**

f[xn-1] - f[xn-2]

If[Abs[xn - xn-1] < e , Return[xn]]; Print[n - 1, "th iteration is: ", xn];

Print ["Estimated error: ", Abs[xn - xn-1]]; Print["Approximate Root ", xn]

Plot[f[x], {x, - 3, 3}]

1th iteration is: -0.833333 Estimated error: 0.833333 2th iteration is: -0.962567 Estimated error: 0.129234 3th iteration is: -0.901757 Estimated error: 0.0608094 4th iteration is: -0.904081 Estimated error: 0.00232406 5th iteration is: -0.904161

Estimated error: 0.0000794928

Return[- 0.904161]

Approximate Root -0.904161

10

-3

-2

-1

1

2

3

-10

-20

-30

-40

-50

*Practical 2 (Regula Falsi).nb* **3**

1. Cos(x) - xe*x* = 0

f[x\_] := Cos[x] - x x

x0 = - 2;

x1 = - 1;

e = 0.000001;

iterations = 10;

Iff[x0] \* f[x1] > 0,

Print["These values do not satisfy the IVP so change the initial value."], Forn = 2, n ≤ iterations, n ++,

xn-2 \* f[xn-1] - xn-1 \* f[xn-2]

xn = N 

**;**

f[xn-1] - f[xn-2]

If[Abs[xn - xn-1] < e , Return[xn]]; Print[n - 1, "th iteration is: ", xn];

Print ["Estimated error: ", Abs[xn - xn-1]]; Print["Approximate Root ", xn]

Plot[f[x], {x, - 3, 3}]

1th iteration is: -1.86193 Estimated error: 0.861932 2th iteration is: -1.86407 Estimated error: 0.00214255 3th iteration is: -1.864

Estimated error: 0.0000795107

Return[- 1.864]

Approximate Root -1.864

-3

-2

-1

1

2

3

-5

-10

-15

-20

-25

-30

**4** *Practical 2 (Regula Falsi).nb*

1. *e*-*x* = *x*

f[x\_] := -x - x x0 = 0;

x1 = 1;

e = 0.000001;

iterations = 10;

Iff[x0] \* f[x1] > 0,

Print["These values do not satisfy the IVP so change the initial value."], Forn = 2, n ≤ iterations, n ++,

xn-2 \* f[xn-1] - xn-1 \* f[xn-2]

xn = N 

**;**

f[xn-1] - f[xn-2]

If[Abs[xn - xn-1] < e , Return[xn]]; Print[n - 1, "th iteration is: ", xn];

Print ["Estimated error: ", Abs[xn - xn-1]]; Print["Approximate Root ", xn]

Plot[f[x], {x, - 1, 1}]

1th iteration is: 0.6127 Estimated error: 0.3873 2th iteration is: 0.563838 Estimated error: 0.0488614 3th iteration is: 0.56717

Estimated error: 0.00333197 4th iteration is: 0.567143 Estimated error: 0.0000270518 Return[0.567143]

Approximate Root 0.567143

|  |  |
| --- | --- |
| 3  1  2  1 |  |
| .0 -0.5 | 0.5 1.0 |

# Practical 2 (Secant Method)

Ques 1. Solve *x*3 - 2x - 5 = 0 Solution:

f[x\_] := x^ 3 - 2 x - 5 x0 = 2;

x1 = 3;

e = 0.000001;

iterations = 10;

Iff[x0] \* f[x1] > 0,

Print["These values do not satisfy the IVP so change the initial value."], Forn = 2, n ≤ iterations, n ++,

**xn-1 - xn-2**

xn = N  xn-1 - f[xn-1] \*

**;**

f[xn-1] - f[xn-2]

If[Abs[xn - xn-1] < e , Return[xn]]; Print[n - 1, "th iteration is: ", xn];

Print ["Estimated error: ", Abs[xn - xn-1]]; Print["Approximate Root ", xn]

1th iteration is: 2.05882 Estimated error: 0.941176 2th iteration is: 2.08126 Estimated error: 0.0224401 3th iteration is: 2.09482 Estimated error: 0.0135605 4th iteration is: 2.09455 Estimated error: 0.000274715 5th iteration is: 2.09455

Estimated error: 2.05019 × 10-6

Return[2.09455]

Approximate Root 2.09455

Q2. Find approximate root of following equations using the secant method:

1. 3 - 3 *x*2 + 2 *x* + 5 = 0
2. Cos(x) - xe*x* = 0
3. *e*-*x* = *x*

Solution:

1. 3 - 3 *x*2 + 2 *x* + 5 = 0

g[x\_] := x^ 3 - 3 x^ 2 + 2 x + 5 x0 = - 1;

x1 = 0;

e = 0.000001;

iterations = 10;

Ifg[x0] \* g[x1] > 0,

Print["These values do not satisfy the IVP so change the initial value."], Forn = 2, n ≤ iterations, n ++,

**xn-1 - xn-2**

xn = N  xn-1 - g[xn-1] \*

**;**

g[xn-1] - g[xn-2]

If[Abs[xn - xn-1] < e , Return[xn]]; Print[n - 1, "th iteration is: ", xn];

Print ["Estimated error: ", Abs[xn - xn-1]]; Print["Approximate Root ", xn]

Plot[g[x], {x, - 3, 3}]

1th iteration is: -0.833333 Estimated error: 0.833333 2th iteration is: -0.962567 Estimated error: 0.129234 3th iteration is: -0.901757 Estimated error: 0.0608094 4th iteration is: -0.904081 Estimated error: 0.00232406 5th iteration is: -0.904161

Estimated error: 0.0000794928

Return[- 0.904161]

Approximate Root -0.904161

10

-3

-2

-1

1

2

3

-10

-20

-30

-40

-50

1. Cos(x) - xe*x* = 0

k[x\_] := Cos[x] - x e^(x)

x0 = 1;

x1 = 2;

e = 0.000001;

iterations = 10;

Ifk[x0] \* k[x1] > 0,

Print["These values do not satisfy the IVP so change the initial value."], Forn = 2, n ≤ iterations, n ++,

**xn-1 - xn-2**

xn = N  xn-1 - k[xn-1] \*

**;**

k[xn-1] - k[xn-2]

If[Abs[xn - xn-1] < e , Return[xn]]; Print[n - 1, "th iteration is: ", xn];

Print ["Estimated error: ", Abs[xn - xn-1]]; Print["Approximate Root ", xn]

Plot[k[x], {x, 1, 2}]

1th iteration is: 1.5649 Estimated error: 0.435096 2th iteration is: 1.57098 Estimated error: 0.00607467 3th iteration is: 1.5708 Estimated error: 0.000182263 Return[1.5708]

Approximate Root 1.5708

0.4

0.2

1.2

1.4

1.6

1.8

2.0

0.2

0.4

1. *e*-*x* = *x*

k[x\_] := -x - x x0 = 0;

x1 = 1;

e = 0.000001;

iterations = 10;

Ifk[x0] \* k[x1] > 0,

Print["These values do not satisfy the IVP so change the initial value."], Forn = 2, n ≤ iterations, n ++,

**xn-1 - xn-2**

xn = N  xn-1 - k[xn-1] \*

**;**

k[xn-1] - k[xn-2]

If[Abs[xn - xn-1] < e , Return[xn]]; Print[n - 1, "th iteration is: ", xn];

Print ["Estimated error: ", Abs[xn - xn-1]]; Print["Approximate Root ", xn]

Plot[k[x], {x, - 6, 2}]

1th iteration is: 0.6127 Estimated error: 0.3873 2th iteration is: 0.563838 Estimated error: 0.0488614 3th iteration is: 0.56717

Estimated error: 0.00333197 4th iteration is: 0.567143 Estimated error: 0.0000270518 Return[0.567143]

Approximate Root 0.567143

-6 -4 -2 2

200

150

100

50

# Practical 3 (Newton-Raphson Method)

Q1. Find approximate root of the equation 4 - *x* - 10 = 0 Solution:

f[x\_] := x^ 4 - x - 10 x0 = 2;

e = 5 \* 10-5;

iterations = 5;

Forn = 1, n ≤ iterations, n ++,

f[xn-1]

xn = Nxn-1 -

**;**

f'[xn-1]

If[Abs[xn - xn-1] < e, Return[xn]]; Print[n, "th iteration's value is: ", xn];

Print["Estimated error is: ", Abs[xn - xn-1]];

Print["Final approximate root is: ", xn] Plot[f[x], {x, - 1, 3}]

1th iteration's value is: 1.87097 Estimated error is: 0.129032

2th iteration's value is: 1.85578 Estimated error is: 0.015187

3th iteration's value is: 1.85558 Estimated error is: 0.000196141 Return[1.85558]

Final approximate root is: 1.85558

|  |  |
| --- | --- |
| 40  -  30  20  10 |  |
| 1  -10 | 1 2 3 |

Q2. Find the approximate root of the following:

1. 3x - Cos(x) - 1
2. Cosx - xe*x*=0
3. *e*-*x* = *x*

Solution:

1. 3x - Cos(x) - 1

f[x\_] := 3 x - Cos[x] - 1 x0 = 1;

e = 5 \* 10-5;

iterations = 5;

Forn = 1, n ≤ iterations, n ++,

f[xn-1]

xn = Nxn-1 -

**;**

f'[xn-1]

If[Abs[xn - xn-1] < e, Return[xn]]; Print[n, "th iteration's value is: ", xn];

Print["Estimated error is: ", Abs[xn - xn-1]];

Print["Final approximate root is: ", xn] Plot[f[x], {x, - 1, 3}]

1th iteration's value is: 0.620016 Estimated error is: 0.379984

2th iteration's value is: 0.607121 Estimated error is: 0.0128953 Return[0.607102]

Final approximate root is: 0.607102

|  |  |
| --- | --- |
| 8  -  6  4  2 |  |
| 1  -2  -4 | 1 2 3 |

Solution:

*Practical\_3\_(Newton\_Raphson\_Method)[1].nb* **3**

1. Cosx - xe*x*=0

g[x\_] := Cos[x] - x \* x

x0 = 1;

err = 5 \* 10-5;

iterations = 5;

Forn = 1, n ≤ iterations, n ++,

g[xn-1]

xn = Nxn-1 -

**;**

g'[xn-1]

If[Abs[xn - xn-1] < err, Return[xn]]; Print[n, "th iteration's value is: ", xn];

Print["Estimated error is: ", Abs[xn - xn-1]];

Print["Final approximate root is: ", xn] Plot[g[x], {x, - 1, 3}]

1th iteration's value is: 0.653079 Estimated error is: 0.346921

2th iteration's value is: 0.531343 Estimated error is: 0.121736

3th iteration's value is: 0.51791 Estimated error is: 0.0134335

4th iteration's value is: 0.517757 Estimated error is: 0.00015253 Return[0.517757]

Final approximate root is: 0.517757

-1

1

2

3

-10

-20

-30

-40

-50

-60

1. *e*-*x* =

Solution:

g[x\_] := -x - x x0 = 1;

err = 5 \* 10-5;

iterations = 5;

Forn = 1, n ≤ iterations, n ++,

g[xn-1]

xn = Nxn-1 -

**;**

g'[xn-1]

If[Abs[xn - xn-1] < err, Return[xn]]; Print[n, "th iteration's value is: ", xn];

Print["Estimated error is: ", Abs[xn - xn-1]];

Print["Final approximate root is: ", xn] Plot[g[x], {x, - 1, 3}]

1th iteration's value is: 0.537883 Estimated error is: 0.462117

2th iteration's value is: 0.566987 Estimated error is: 0.0291041

3th iteration's value is: 0.567143 Estimated error is: 0.000156295 Return[0.567143]

Final approximate root is: 0.567143

3

2

1

-1

1

2

3

-1

-2

-3

# Practical 4

Gauss Jordan Method

Q1. Solve the given system of equations using Gauss Jordan Method:

y + z = 2 2x + 3z = 5

x + y + z = 3 Solution:

A = {{0, 1, 1, 2}, {2, 0, 3, 5}, {1, 1, 1, 3}}

{{0, 1, 1, 2}, {2, 0, 3, 5}, {1, 1, 1, 3}}

A // MatrixForm

0 1 1 2

2 0 3 5

1 1 1 3

RowReduce[A] // MatrixForm

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 |

Solve[{x  1, y  1, z  1}, {x, y, z}]

{{x  1, y  1, z  1}}

Gauss Elimination Method

Q1. Solve the given system of equations using Gauss Jordan Method:

2x + y + z = 4

3x + 5y + 2z = 15

2x + y + 4z = 18 Solution:

A = {{2, 1, 1}, {3, 5, 2}, {2, 1, 4}};

A // MatrixForm

B = {4, 15, 18};

B // MatrixForm m1 = Length[A]; m2 = Length[B];

x = Table[0, {m1}];

If[m1 ≠ m2, Print["This system of equation can not be solved"], Table[AppendTo[A 〚i〛, B〚i〛], {i, m1}];

Print["[A|B]=", A // MatrixForm];

For[i = 1, i ≤ m1 - 1, i ++, s = Abs[A〚i, i〛]; c = i;

For[j = i + 1, j ≤ m1, j ++, If[Abs[A〚j, i〛] > s, s = A〚j, i〛; c = j;]];

For[k = 1, k ≤ m1 + 1, k ++, d[k] = A〚i, k〛; A〚i, k〛 = A〚c, k〛;

A〚c, k〛 = d[k]];

Print["step", i, A // MatrixForm];

For[j = i + 1, j ≤ m1, j ++, m = A〚j, i〛/ A〚i, i〛;

For[k = 1, k ≤ m1 + 1, k ++, A〚j, k〛 = A〚j, k〛- (m \* A〚i, k〛)];]; Print[A // MatrixForm];];

For[i = 0, i ≤ m1 - 1, i ++,

x〚m1 - i〛 = (A〚m1 - i, m1 + 1〛- Sum[A〚m1 - i, j〛\* x〚j〛, {j, m1 - i + 1, m1}]) /

A〚m1 - i, m1 - i〛;]; Print["x =", x // MatrixForm];]

2 1 1

3 5 2

2 1 4

4

15

18

*Practical\_4\_(Gauss\_Jordan\_and\_Elimination\_Method)[1].nb* **3**

[A|B]=

step1

2 1 1 4

3 5 2 15

2 1 4 18

3 5 2 15

2 1 1 4

2 1 4 18

3 5 2 15

0 - 7 - 1 -6

3 3

0 - 7 8 8

3 3

3 5 2 15

step2

0 - 7

3

0 - 7

3

- 1 -6

3

8 8

3

|  |  |  |
| --- | --- | --- |
| 3 5 | 2 | 15 |
| 0 - 7 - 1 -6 | | |
| 3 | 3 |  |
| 0 0 | 3 | 14 |

- 9

7

x = 40

21

14

3

# Practical 5

Gauss Jacobi Method

Q1. Find the given system of given system of equations.

1. 27 *x*1 + 6 *x*2 - *x*3 = 85, 6 *x*1 + 15 *x*2 + 2 *x*3 = 72, *x*1 + *x*2 + 54 *x*3 = 110
2. 5 *x*1 + 2 *x*2 + *x*3 = 10, 3 *x*1 + 7 *x*2 + 4 *x*3 = 21, *x*1 + *x*2 + 9 *x*3 = 12
3. 10 *x*1 - 2 *x*2 + *x*3 = 12, *x*1 + 9 *x*2 - *x*3 = 10, 2 *x*1 - *x*2 + 11 *x*3 = 20

Solution:

i)

n = 3; (\* number of variables \*)

a = {{27, 6, - 1}, {6, 15, 2}, {1, 1, 54}};

MatrixForm[a]

x = {0, 0, 0} (\* initial value of x01,x02,x03 \*)

b = {85, 72, 110}

For[k = 1, k ≤ 5, k ++, For[i = 1, i ≤ n, i ++,

x〚i〛 = (b〚i〛- Sum[a〚i, j〛\* x〚j〛, {j, 1, i - 1}] - Sum[a〚i, j〛\* x〚j〛, {j, i + 1, n}]) /

a〚i, i〛];

For[m = 1, m ≤ n, m ++, x〚m〛 = N[x〚m〛]]]

For[p = 1, p ≤ n, p ++, Print["x[", p, "]=", x〚p〛]]

27 6 - 1

6 15 2

1 1 54

{0, 0, 0}

{85, 72, 110}

x[1]=2.42548 x[2]=3.57302 x[3]=1.92595

Solution.

ii)

n = 3; (\* number of variables \*)

a = {{5, 2, 1}, {3, 7, 4}, {1, 1, 9}};

MatrixForm[a]

x = {0, 0, 0} (\* initial value of x01,x02,x03 \*)

b = {10, 21, 12}

For[k = 1, k ≤ 5, k ++, For[i = 1, i ≤ n, i ++,

x〚i〛 = (b〚i〛- Sum[a〚i, j〛\* x〚j〛, {j, 1, i - 1}] - Sum[a〚i, j〛\* x〚j〛, {j, i + 1, n}]) /

a〚i, i〛];

For[m = 1, m ≤ n, m ++, x〚m〛 = N[x〚m〛]]]

For[p = 1, p ≤ n, p ++, Print["x[", p, "]=", x〚p〛]]

5 2 1

3 7 4

1 1 9

{0, 0, 0}

{10, 21, 12}

x[1]=0.999005 x[2]=2.00009 x[3]=1.0001

Solution:

iii)

n = 3; (\* number of variables \*)

a = {{10, - 2, 1}, {1, 9, - 1}, {2, - 1, 11}};

MatrixForm[a]

x = {0, 0, 0} (\* initial value of x01,x02,x03 \*)

b = {12, 10, 20}

For[k = 1, k ≤ 5, k ++, For[i = 1, i ≤ n, i ++,

x〚i〛 = (b〚i〛- Sum[a〚i, j〛\* x〚j〛, {j, 1, i - 1}] - Sum[a〚i, j〛\* x〚j〛, {j, i + 1, n}]) /

a〚i, i〛];

For[m = 1, m ≤ n, m ++, x〚m〛 = N[x〚m〛]]]

For[p = 1, p ≤ n, p ++, Print["x[", p, "]=", x〚p〛]]

10 - 2 1

1 9 - 1

2 - 1 11

{0, 0, 0}

{12, 10, 20}

x[1]=1.26241 x[2]=1.15906 x[3]=1.69402

Gauss Seidel Method

Q1. Solve the following equations using Gauss Seidel Method:

1. 27 *x*1 + 6 *x*2 - *x*3 = 85, 6 *x*1 + 15 *x*2 + 2 *x*3 = 72, *x*1 + *x*2 + 54 *x*3 = 110
2. 5 *x*1 + 2 *x*2 + *x*3 = 10, 3 *x*1 + 7 *x*2 + 4 *x*3 = 21, *x*1 + *x*2 + 9 *x*3 = 12
3. 10 *x*1 - 2 *x*2 + *x*3 = 12, *x*1 + 9 *x*2 - *x*3 = 10, 2 *x*1 - *x*2 + 11 *x*3 = 20

Solution:

i)

n = 3; (\* number of variables \*)

a = {{27, 6, - 1}, {6, 15, 2}, {1, 1, 54}};

MatrixForm[a]

x = {0, 0, 0} (\* initial value of x01,x02,x03 \*)

b = {85, 72, 110}

For[k = 1, k ≤ 5, k ++, For[i = 1, i ≤ n, i ++,

y〚i〛 = (b〚i〛- Sum[a〚i, j〛\* y〚j〛, {j, 1, i - 1}] - Sum[a〚i, j〛\* x〚j〛, {j, i + 1, n}]) /

a〚i, i〛];

For[m = 1, m ≤ n, m ++, x〚m〛 = N[y〚m〛]]]

For[p = 1, p ≤ n, p ++, Print["x[", p, "]=", x〚p〛]]

27 6 - 1

6 15 2

1 1 54

{0, 0, 0}

{85, 72, 110}

x[1]=2.42548 x[2]=3.57302 x[3]=1.92595

Solution. ii)

n = 3; (\* number of variables \*)

a = {{5, 2, 1}, {3, 7, 4}, {1, 1, 9}};

MatrixForm[a]

x = {0, 0, 0} (\* initial value of x01,x02,x03 \*)

b = {10, 21, 12}

For[k = 1, k ≤ 5, k ++, For[i = 1, i ≤ n, i ++,

y〚i〛 = (b〚i〛- Sum[a〚i, j〛\* y〚j〛, {j, 1, i - 1}] - Sum[a〚i, j〛\* x〚j〛, {j, i + 1, n}]) /

a〚i, i〛];

For[m = 1, m ≤ n, m ++, x〚m〛 = N[y〚m〛]]]

For[p = 1, p ≤ n, p ++, Print["x[", p, "]=", x〚p〛]]

5 2 1

3 7 4

1 1 9

{0, 0, 0}

{10, 21, 12}

x[1]=0.999005 x[2]=2.00009 x[3]=1.0001

Solution:

iii)

n = 3; (\* number of variables \*)

a = {{10, - 2, 1}, {1, 9, - 1}, {2, - 1, 11}};

MatrixForm[a]

x = {0, 0, 0} (\* initial value of x01,x02,x03 \*)

b = {12, 10, 20}

For[k = 1, k ≤ 5, k ++, For[i = 1, i ≤ n, i ++,

y〚i〛 = (b〚i〛- Sum[a〚i, j〛\* y〚j〛, {j, 1, i - 1}] - Sum[a〚i, j〛\* x〚j〛, {j, i + 1, n}]) /

a〚i, i〛];

For[m = 1, m ≤ n, m ++, x〚m〛 = N[y〚m〛]]]

For[p = 1, p ≤ n, p ++, Print["x[", p, "]=", x〚p〛]]

10 - 2 1

1 9 - 1

2 - 1 11

{0, 0, 0}

{12, 10, 20}

x[1]=1.26241 x[2]=1.15906 x[3]=1.69402

Newton Interpolation

Q1. For the data (3,293), (5,508) , (6,585), (9,764). Using Newton interpolation, find interpolation polynomial p(x) and also find p(5.6).

points = {{3, 293}, {5, 508}, {6, 585}, {9, 764}}

n = Length[points] y = points〚All, 1〛 f = points〚All, 2〛

dd[k\_] := Sum[(f〚i〛 / Product[If[Equal[j, i], 1, (y〚i〛- y〚j〛)], {j, 1, k}]), {i, 1, k}];

p[x\_] = Sum[(dd[i] \* Product[If[i ≤ j, 1, x - y〚j〛], {j, 1, i - 1}]), {i, 1, n}] Simplify[p[x]]

Evaluate[p[5.6]]

{{3, 293}, {5, 508}, {6, 585}, {9, 764}}

4

{3, 5, 6, 9}

{293, 508, 585, 764}

293 +

215

2

61

(- 3 + x) - 

6

35

(- 5 + x) (- 3 + x) + 

36

(- 6 + x) (- 5 + x) (- 3 + x)

1

 (- 9702 + 9003 x - 856 x2 + 35 x3)

36

556.033

Lagrange Interpolation

Q1. For the data (-1, 5), (0, 1), (1, 1), (2,11) find the interpolation polynomial P(x) and find P(1.5) using langrange interpolation. Solution:

xi = {- 1, 0, 1, 2}

fi = {5, 1, 1, 11}

n = Length[xi]

Fork = 1, k ≤ n, k ++,

Lk[x\_] =

**k-1**

 (x - xi〚j〛) / (xi〚k〛 - xi〚j〛)

**j=1**

**n**

**\* **

**j=k+1**

(x - xi〚j〛) / (xi〚k〛 - xi〚j〛) 

**n**

p[x\_] = Σ Lk[x] \* fi〚k〛;

**k=1**

Print["Lagrange Interpolation Polynomial P[x] = ", p[x]]

Print["Simplify Lagrange Interpolation Polynomial P[x] = ", Simplify[p[x]]] Print["Value of P[x] at x = 1.5 is: ", p[1.5]]

{- 1, 0, 1, 2}

{5, 1, 1, 11}

4

Lagrange Interpolation Polynomial P[x] =

5 1

- (1 - x) (2 - x) x +

6 2

1

(1 - x) (2 - x) (1 + x) +

2

11

(2 - x) x (1 + x) + 

6

(-1 + x) x (1 + x)

Simplify Lagrange Interpolation Polynomial P[x] = 1 - 3 x + 2 x2 + x3 Value of P[x] at x = 1.5 is: 4.375

Trapezoidal Rule

Q1. Use trapezoidal rule to integrate f(x) = ex2 from 0 to 2 for n = 10.

Solution:

f[x\_] = Exp[x ^ 2];

a = 0;

b = 2;

n = 10;

h = (b - a) / n;

app = N[(h / 2) \* (f[a] + 2 \* Sum[f[i], {i, a + h, b - h, h}] + f[b])]; Ex = N[Integrate[f[x], {x, 0, 2}]];

Print["The exact value of f[x] is: ", Ex] Print["The approximate value of f[x] is: ", app] Print["The error is: ", Abs[Ex - app]]

The exact value of f[x] is: 16.4526

The approximate value of f[x] is: 17.1702 The error is: 0.717582

Q2. Using the trapezoidal rule, evaluate the following:

1 x2

i)∫

0 1+x3

 for n = 5.

6 1

ii) ∫

0 1+x2

x for n = 6.

iii) ∫ 0.6-x2 x for n = 6.

0

Solution:

1 x2

i)∫

0 1+x3

 for n = 5.

f[x\_] = x^ 2 / (1 + x^ 3);

a = 0;

b = 1;

n = 5;

h = (b - a) / n;

app = N[(h / 2) \* (f[a] + 2 \* Sum[f[i], {i, a + h, b - h, h}] + f[b])]; Ex = N[Integrate[f[x], {x, 0, 1}]];

Print["The exact value of f[x] is: ", Ex] Print["The approximate value of f[x] is: ", app] Print["The error is: ", Abs[Ex - app]]

The exact value of f[x] is: 0.231049

The approximate value of f[x] is: 0.231878 The error is: 0.000829247

6 1

ii) ∫

0 1+x2

x for n = 6.

f[x\_] = 1 / (1 + x^ 2);

a = 0;

b = 6;

n = 6;

h = (b - a) / n;

app = N[(h / 2) \* (f[a] + 2 \* Sum[f[i], {i, a + h, b - h, h}] + f[b])]; Ex = N[Integrate[f[x], {x, 0, 6}]];

Print["The exact value of f[x] is: ", Ex] Print["The approximate value of f[x] is: ", app] Print["The error is: ", Abs[Ex - app]]

The exact value of f[x] is: 1.40565

The approximate value of f[x] is: 1.4108 The error is: 0.00515093

iii) ∫ 0.6-x2 x for n = 6.

0

f[x\_] = Exp[(- x) ^ 2];

a = 0;

b = 0.6;

n = 6;

h = (b - a) / n;

app = N[(h / 2) \* (f[a] + 2 \* Sum[f[i], {i, a + h, b - h, h}] + f[b])]; Ex = N[Integrate[f[x], {x, 0, 0.6}]];

Print["The exact value of f[x] is: ", Ex] Print["The approximate value of f[x] is: ", app] Print["The error is: ", Abs[Ex - app]]

The exact value of f[x] is: 0.680492

The approximate value of f[x] is: 0.681924 The error is: 0.00143156

Simpson’s 1/3 Rule

Q1. Using Simpson’s 1/3 Rule integrate f (x) = x2

1 x3

+

from 0 to 1 for n

= 4.

Solution:

f[x\_] = x^ 2 / (1 + x^ 3);

a = 0;

b = 1;

n = 4;

h = (b - a) / n;

app = N[(h / 3) \* (f[a] + 4 \* Sum[f[i], {i, a + h, b - h, 2 h}] +

2 \* Sum[f[i], {i, a + 2 h, b - 2 h, 2 h}] + f[b])];

Ex = N[Integrate[f[x], {x, 0, 1}]]; Print["The exact value of f[x] is: ", Ex]

Print["The approximate value of f[x] is: ", app] Print["The error is: ", Abs[Ex - app]]

The exact value of f[x] is: 0.231049

The approximate value of f[x] is: 0.231085 The error is: 0.0000355959

Q2. Using simpson’s 1/3 rule, evaluate the following:

1 1

i)∫

0 x2+6 x+10

x for n = 10.

6 1

ii) ∫

0 1+x2

x for n = 6.

iii) ∫ 0.6-x2 x for n = 6.

0

Solution:

1 1

i)∫

0 x2+6 x+10

x for n = 10.

f[x\_] = 1 / (x^ 2 + 6 x + 10);

a = 0;

b = 1;

n = 10;

h = (b - a) / n;

app = N[(h / 3) \* (f[a] + 4 \* Sum[f[i], {i, a + h, b - h, 2 h}] +

2 \* Sum[f[i], {i, a + 2 h, b - 2 h, 2 h}] + f[b])];

Ex = N[Integrate[f[x], {x, 0, 1}]]; Print["The exact value of f[x] is: ", Ex]

Print["The approximate value of f[x] is: ", app] Print["The error is: ", Abs[Ex - app]]

The exact value of f[x] is: 0.0767719

The approximate value of f[x] is: 0.0767719 The error is: 2.23635 × 10-8

6 1

ii) ∫

0 1+x2

x for n = 6.

f[x\_] = 1 / (1 + x^ 2);

a = 0;

b = 6;

n = 6;

h = (b - a) / n;

app = N[(h / 3) \* (f[a] + 4 \* Sum[f[i], {i, a + h, b - h, 2 h}] +

2 \* Sum[f[i], {i, a + 2 h, b - 2 h, 2 h}] + f[b])];

Ex = N[Integrate[f[x], {x, 0, 6}]]; Print["The exact value of f[x] is: ", Ex]

Print["The approximate value of f[x] is: ", app] Print["The error is: ", Abs[Ex - app]]

The exact value of f[x] is: 1.40565

The approximate value of f[x] is: 1.36617 The error is: 0.0394742

iii) ∫ 0.6-x2 x for n = 6.

0

f[x\_] = Exp[(- x) ^ 2];

a = 0;

b = 0.6;

n = 6;

h = (b - a) / n;

app = N[(h / 3) \* (f[a] + 4 \* Sum[f[i], {i, a + h, b - h, 2 h}] +

2 \* Sum[f[i], {i, a + 2 h, b - 2 h, 2 h}] + f[b])];

Ex = N[Integrate[f[x], {x, 0, 0.6}]]; Print["The exact value of f[x] is: ", Ex]

Print["The approximate value of f[x] is: ", app] Print["The error is: ", Abs[Ex - app]]

The exact value of f[x] is: 0.680492

The approximate value of f[x] is: 0.680499 The error is: 7.00825 × 10-6

# Practical 8 (Euler’s Method)

Q1. Using Euler’s method find an approximate value of y corresponding x = 1, for the first order ODE f(x,y) = x + y and y = 1 at x = 0.

Solution:

f[x\_, y\_] := x + y;

a = 0;

b = 1;

n = 10;

y[0] = 1;

h = (b - a) / n;

For[i = 0, i ≤ n, i ++, x[i] = a + h \* i; y[i + 1] = y[i] + h \* f[x[i], y[i]];

Print["Value at x[", i, "]=", x[i], " is ", N[y[i]]]]

Value at x[0]=0 is 1.

1

Value at x[1]=

Value at x[2]= Value at x[3]= Value at x[4]= Value at x[5]= Value at x[6]= Value at x[7]= Value at x[8]= Value at x[9]=

is 1.1

10

1

is 1.22

5

3

is 1.362

10

2

is 1.5282

5

1

is 1.72102

2

3

is 1.94312

5

7

is 2.19743

10

4

is 2.48718

5

9

is 2.8159

10

Value at x[10]=1 is 3.18748

Q2.

1. Using Euler’s method find an approximate value of y corresponding x = 0.1, for the first order ODE f(x,y) = (y-x)/(x+y) and y = 1 at x = 0.
2. Using Euler’s method find an approximate value of y corresponding x = 0.4, for the first order ODE f(x,y) = y + *ex* and y = 0 at x = 0.
3. Using Euler’s method find an approximate value of y corresponding x = 1.2, for the first order ODE f(x,y) =log(x+y) and y

= 2 at x = 0.

Solution:

1. Using Euler’s method find an approximate value of y corresponding x = 0.1, for the first order ODE f(x,y) = (y-x)/(x+y) and y = 1 at x = 0.

f[x\_, y\_] := (y - x) / (x + y);

a = 0;

b = 0.1;

n = 10;

y[0] = 1;

h = (b - a) / n;

For[i = 0, i ≤ n, i ++, x[i] = a + h \* i; y[i + 1] = y[i] + h \* f[x[i], y[i]];

Print["Value at x[", i, "]=", x[i], " is ", N[y[i]]]]

Value at x[0]=0. is 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Value | at | x[1]=0.01 | is | 1.01 |
| Value | at | x[2]=0.02 | is | 1.0198 |
| Value | at | x[3]=0.03 | is | 1.02942 |
| Value | at | x[4]=0.04 | is | 1.03885 |
| Value | at | x[5]=0.05 | is | 1.04811 |
| Value | at | x[6]=0.06 | is | 1.0572 |
| Value | at | x[7]=0.07 | is | 1.06613 |
| Value | at | x[8]=0.08 | is | 1.07489 |
| Value | at | x[9]=0.09 | is | 1.08351 |
| Value | at | x[10]=0.1 | is | 1.09198 |

1. Using Euler’s method find an approximate value of y

*Practical 8 (Euler's Method)[1].nb* **3**

corresponding x = 0.4, for the first order ODE f(x,y) = y + *ex* and y = 0 at x = 0.

f[x\_, y\_] := y + x;

a = 0;

b = 0.4;

n = 10;

y[0] = 0;

h = (b - a) / n;

For[i = 0, i ≤ n, i ++, x[i] = a + h \* i; y[i + 1] = y[i] + h \* f[x[i], y[i]];

Print["Value at x[", i, "]=", x[i], " is ", N[y[i]]]]

Value at x[0]=0. is 0. Value at x[1]=0.04 is 0.04

Value at x[2]=0.08 is 0.0832324 Value at x[3]=0.12 is 0.129893 Value at x[4]=0.16 is 0.180189 Value at x[5]=0.2 is 0.234337 Value at x[6]=0.24 is 0.292566 Value at x[7]=0.28 is 0.355119 Value at x[8]=0.32 is 0.422249 Value at x[9]=0.36 is 0.494224 Value at x[10]=0.4 is 0.571326

1. Using Euler’s method find an approximate value of y corresponding x = 1.2, for the first order ODE f(x,y) =log(x+y) and y

= 2 at x = 0.

f[x\_, y\_] := Log[x + y];

a = 0;

b = 1.2;

n = 10;

y[0] = 2;

h = (b - a) / n;

For[i = 0, i ≤ n, i ++, x[i] = a + h \* i; y[i + 1] = y[i] + h \* f[x[i], y[i]];

Print["Value at x[", i, "]=", x[i], " is ", N[y[i]]]]

Value at x[0]=0. is 2.

Value at x[1]=0.12 is 2.08318 Value at x[2]=0.24 is 2.17797 Value at x[3]=0.36 is 2.28392 Value at x[4]=0.48 is 2.40059 Value at x[5]=0.6 is 2.52755 Value at x[6]=0.72 is 2.66438 Value at x[7]=0.84 is 2.81068 Value at x[8]=0.96 is 2.96607 Value at x[9]=1.08 is 3.13018 Value at x[10]=1.2 is 3.30268