1. Bisection Method

```
Bisection[f_, a0_, b0_, tol_, n_] :=
 Module[{a = a0, b = b0, c, i = 1}, Print ["    n    a    b    c    f(c)"];
  While [ i ≤ n , c = (a + b) / 2;
   Print [NumberForm[i, 5], NumberForm [a, {10, 8}],
    NumberForm[b, {10, 8}], NumberForm [c, {10, 8}], NumberForm[f[c], {10, 8}]];
   If [f[a] * f[c] < 0 , a = c, b = c];
   If [Abs [a - b] < tol, {Print["The solution is:", NumberForm[c, 16]];
     Return[]}];
   i++];
   Print["The maximum iteration failed and the solution without accuracy is:",
   NumberForm[c, 16]];]
f1[x] := x^3 + 2 x^2 - 3 x - 1;
f2[x] := x^3 + 2 x^2 - 3 x - 3;
f3[x] := sin[x];
Bisection[f1, 0, 2, 10^(-7), 50]
```

```
    n    a    b    c    f(c)
10.000000002.000000001.00000000f1[1.00000000]
20.000000002.000000001.00000000f1[1.00000000]
30.000000002.000000001.00000000f1[1.00000000]
40.000000002.000000001.00000000f1[1.00000000]
50.000000002.000000001.00000000f1[1.00000000]
60.000000002.000000001.00000000f1[1.00000000]
70.000000002.000000001.00000000f1[1.00000000]
80.000000002.000000001.00000000f1[1.00000000]
90.000000002.000000001.00000000f1[1.00000000]
100.000000002.000000001.00000000f1[1.00000000]
110.000000002.000000001.00000000f1[1.00000000]
120.000000002.000000001.00000000f1[1.00000000]
130.000000002.000000001.00000000f1[1.00000000]
140.000000002.000000001.00000000f1[1.00000000]
150.000000002.000000001.00000000f1[1.00000000]
160.000000002.000000001.00000000f1[1.00000000]
170.000000002.000000001.00000000f1[1.00000000]
180.000000002.000000001.00000000f1[1.00000000]
190.000000002.000000001.00000000f1[1.00000000]
200.000000002.000000001.00000000f1[1.00000000]
    n    a    b    c    f(c)
10.000000002.000000001.00000000f1[1.00000000]
20.000000002.000000001.00000000f1[1.00000000]
30.000000002.000000001.00000000f1[1.00000000]
40.000000002.000000001.00000000f1[1.00000000]
```

```
50.000000002.000000001.00000000f1[1.00000000]
60.000000002.000000001.00000000f1[1.00000000]
70.000000002.000000001.00000000f1[1.00000000]
80.000000002.000000001.00000000f1[1.00000000]
90.000000002.000000001.00000000f1[1.00000000]
100.000000002.000000001.00000000f1[1.00000000]
110.000000002.000000001.00000000f1[1.00000000]
120.000000002.000000001.00000000f1[1.00000000]
130.000000002.000000001.00000000f1[1.00000000]
140.000000002.000000001.00000000f1[1.00000000]
150.000000002.000000001.00000000f1[1.00000000]
160.000000002.000000001.00000000f1[1.00000000]
170.000000002.000000001.00000000f1[1.00000000]
180.000000002.000000001.00000000f1[1.00000000]
190.000000002.000000001.00000000f1[1.00000000]
200.000000002.000000001.00000000f1[1.00000000]
210.000000002.000000001.00000000f1[1.00000000]
220.000000002.000000001.00000000f1[1.00000000]
230.000000002.000000001.00000000f1[1.00000000]
240.000000002.000000001.00000000f1[1.00000000]
250.000000002.000000001.00000000f1[1.00000000]
260.000000002.000000001.00000000f1[1.00000000]
270.000000002.000000001.00000000f1[1.00000000]
280.000000002.000000001.00000000f1[1.00000000]
290.000000002.000000001.00000000f1[1.00000000]
300.000000002.000000001.00000000f1[1.00000000]
310.000000002.000000001.00000000f1[1.00000000]
320.000000002.000000001.00000000f1[1.00000000]
330.000000002.000000001.00000000f1[1.00000000]
340.000000002.000000001.00000000f1[1.00000000]
350.000000002.000000001.00000000f1[1.00000000]
360.000000002.000000001.00000000f1[1.00000000]
370.000000002.000000001.00000000f1[1.00000000]
380.000000002.000000001.00000000f1[1.00000000]
390.000000002.000000001.00000000f1[1.00000000]
400.000000002.000000001.00000000f1[1.00000000]
410.000000002.000000001.00000000f1[1.00000000]
420.000000002.000000001.00000000f1[1.00000000]
430.000000002.000000001.00000000f1[1.00000000]
```

```
440.000000002.000000001.00000000f1[1.00000000]
450.000000002.000000001.00000000f1[1.00000000]
460.000000002.000000001.00000000f1[1.00000000]
470.000000002.000000001.00000000f1[1.00000000]
480.000000002.000000001.00000000f1[1.00000000]
490.000000002.000000001.00000000f1[1.00000000]
500.000000002.000000001.00000000f1[1.00000000]
The maximum iteration failed and the solution without accuracy is:1
210.000000002.000000001.00000000f1[1.00000000]
220.000000002.000000001.00000000f1[1.00000000]
230.000000002.000000001.00000000f1[1.00000000]
240.000000002.000000001.00000000f1[1.00000000]
250.000000002.000000001.00000000f1[1.00000000]
260.000000002.000000001.00000000f1[1.00000000]
270.000000002.000000001.00000000f1[1.00000000]
280.000000002.000000001.00000000f1[1.00000000]
290.000000002.000000001.00000000f1[1.00000000]
300.000000002.000000001.00000000f1[1.00000000]
310.000000002.000000001.00000000f1[1.00000000]
320.000000002.000000001.00000000f1[1.00000000]
330.000000002.000000001.00000000f1[1.00000000]
340.000000002.000000001.00000000f1[1.00000000]
350.000000002.000000001.00000000f1[1.00000000]
360.000000002.000000001.00000000f1[1.00000000]
370.000000002.000000001.00000000f1[1.00000000]
380.000000002.000000001.00000000f1[1.00000000]
390.000000002.000000001.00000000f1[1.00000000]
400.000000002.000000001.00000000f1[1.00000000]
410.000000002.000000001.00000000f1[1.00000000]
420.000000002.000000001.00000000f1[1.00000000]
430.000000002.000000001.00000000f1[1.00000000]
440.000000002.000000001.00000000f1[1.00000000]
450.000000002.000000001.00000000f1[1.00000000]
460.000000002.000000001.00000000f1[1.00000000]
470.000000002.000000001.00000000f1[1.00000000]
480.000000002.000000001.00000000f1[1.00000000]
490.000000002.000000001.00000000f1[1.00000000]
500.000000002.000000001.00000000f1[1.00000000]
510.000000002.000000001.00000000f1[1.00000000]
```

520.000000002.000000001.00000000f1[1.00000000]

530.000000002.000000001.00000000f1[1.00000000]

540.000000002.000000001.00000000f1[1.00000000]

550.000000002.000000001.00000000f1[1.00000000]

560.000000002.000000001.00000000f1[1.00000000]

570.000000002.000000001.00000000f1[1.00000000]

580.000000002.000000001.00000000f1[1.00000000]

590.000000002.000000001.00000000f1[1.00000000]

600.000000002.000000001.00000000f1[1.00000000]

610.000000002.000000001.00000000f1[1.00000000]

620.000000002.000000001.00000000f1[1.00000000]

630.000000002.000000001.00000000f1[1.00000000]

640.000000002.000000001.00000000f1[1.00000000]

650.000000002.000000001.00000000f1[1.00000000]

660.000000002.000000001.00000000f1[1.00000000]

670.000000002.000000001.00000000f1[1.00000000]

680.000000002.000000001.00000000f1[1.00000000]

690.000000002.000000001.00000000f1[1.00000000]

700.000000002.000000001.00000000f1[1.00000000]

710.000000002.000000001.00000000f1[1.00000000]

720.000000002.000000001.00000000f1[1.00000000]

730.000000002.000000001.00000000f1[1.00000000]

740.000000002.000000001.00000000f1[1.00000000]

750.000000002.000000001.00000000f1[1.00000000]

760.000000002.000000001.00000000f1[1.00000000]

770.000000002.000000001.00000000f1[1.00000000]

780.000000002.000000001.00000000f1[1.00000000]

790.000000002.000000001.00000000f1[1.00000000]

800.000000002.000000001.00000000f1[1.00000000]

810.000000002.000000001.00000000f1[1.00000000]

820.000000002.000000001.00000000f1[1.00000000]

830.000000002.000000001.00000000f1[1.00000000]

840.000000002.000000001.00000000f1[1.00000000]

850.000000002.000000001.00000000f1[1.00000000]

860.000000002.000000001.00000000f1[1.00000000]

870.000000002.000000001.00000000f1[1.00000000]

880.000000002.000000001.00000000f1[1.00000000]

890.000000002.000000001.00000000f1[1.00000000]

900.000000002.000000001.00000000f1[1.00000000]

```
910.000000002.000000001.00000000f1[1.00000000]
920.000000002.000000001.00000000f1[1.00000000]
930.000000002.000000001.00000000f1[1.00000000]
940.000000002.000000001.00000000f1[1.00000000]
950.000000002.000000001.00000000f1[1.00000000]
960.000000002.000000001.00000000f1[1.00000000]
970.000000002.000000001.00000000f1[1.00000000]
980.000000002.000000001.00000000f1[1.00000000]
990.000000002.000000001.00000000f1[1.00000000]
1000.000000002.000000001.00000000f1[1.00000000]
The maximum iteration failed and the solution without accuracy is:1
```

```
Bisection[a0_, b0_, m_] := Module[{a = N[a0], b = N[b0]}, c = (a + b) / 2;
   k = 0;
   While[k < m && ((b - a) / 2) > 10 (-7),
    If[Sign[f[b]] == Sign[f[c]], b = c, a = c]; c = (a + b) / 2;
    k = k + 1;];
   Print["c= ", NumberForm[c, 16]];
   Print["f[c]= " NumberForm[f[c], 16]];];

f[x_] = x^3 + 2 x^2 - 3 x - 1;
Bisection[1, 2, 30]

f[x_] = x^3 + 2 x^2 - 3 x - 3;
Bisection[1, 2, 30]

f[x_] = sin[x];
Bisection[3, 4, 30]
```

c= 1.198691243771464

f[c]= $\left(1.559712359266996\times 10^{-9}\right)$

c= 1.460504870396107

f[c]= $\left(3.487127031576165\times 10^{-9}\right)$

c= 3.5

f[c]= sin[3.5]

2. Secant Method

```
SecantMethod[x0_, x1_, max_] := Module[{}, k = 1; p0 = N[x0];
  p1 = N[x1];
  p2 = p1;
  p1 = p0;
  Print["n    x(n-1)     x(n)     x(n+1)     f(x(n+1))"];
  While[(k < max && Abs [f[p2]] > 5 * 10^(-7)),
   p0 = p1;
   p1 = p2;
   p2 = p1 - (f[p1] (p1 - p0) / (f[p1] - f[p0]));
   k = k + 1;
   Print [k - 1, PaddedForm [N[p0, 16], {10, 6}], PaddedForm [N[p1, 16], {10, 6}],
    PaddedForm [N[p2, 16], {10, 6}], PaddedForm [N[f[p2]], {10, 6}]]];
  Print["p", k, "=", NumberForm [p2, 11]];
  Print ["f[p", k, "]=", NumberForm [f[p2], 11]];]
```

```
f[x_] := x^3 - 2 * x - 5;
SecantMethod[3, 2, 50]
```

n	x(n−1)	x(n)	x(n+1)	f(x(n+1))
1	3.000000	2.000000	2.058824	−0.390800
2	2.000000	2.058824	2.096559	0.022428
3	2.058824	2.096559	2.094511	−0.000457
4	2.096559	2.094511	2.094551	-5.157852×10^{-7}
5	2.094511	2.094551	2.094551	1.188472×10^{-11}

p6=2.0945514815

f[p6]=$1.1884715434\times10^{-11}$

```
f[x_] := Cos[x] - x;
SecantMethod[0, 1, 50]
```

n	x(n−1)	x(n)	x(n+1)	f(x(n+1))
1	0.000000	1.000000	0.685073	0.089299
2	1.000000	0.685073	0.736299	0.004660
3	0.685073	0.736299	0.739119	−0.000057
4	0.736299	0.739119	0.739085	3.529262×10^{-8}

p5=0.73908511213

f[p5]=$3.5292622824\times10^{-8}$

3. Regula-Falsi

```
Regularfalsi[a0_, b0_, tol_, m_] :=
  Module[{a = N[a0], b = N[b0]}, If[f[a] * f[b] > 0, Print["Interval is not correct"];
     Exit[];, Print["    n    a    b    c    f(c)"];
     c = (a * f [b] - b * f[a]) / (f[b] - f[a]);
      k = 0;
     While [k < m && Abs [f[c]] > tol, If [Sign [f[b]] == Sign [f[c]], tempa = a;
        tempb = b;
       b = c, tempa = a;
       tempb = b;
       a = c];
      c = (a * f [b] - b * f[a]) / (f[b] - f[a]);
      k = k + 1 ;
      Print[PaddedForm[k, 5], PaddedForm[N[tempa, 16], {10, 6}], PaddedForm[N[tempb,
         16], {10, 6}], PaddedForm[N[c, 16], {10, 6}], PaddedForm[N[f[c]], {10, 6}]]];
     If[Abs[f[c]] > tol, Print["After", k,
       " iterations result without accuracy is : "];
      Print["c = " , NumberForm[c, 16]];
      Print["f[c] = ", NumberForm[f[c], 16]],
      Print["Accuracy is acheived and result is : "];
      Print["c = ", NumberForm[c, 16]];
      Print["f[c]=", NumberForm[f[c], 16]]];];];

f[x_] = x^3 + 2 * x^2 + - 3 * x - 1;
Regularfalsi[1, 2, 10^-(12), 50]
```

n	a	b	c	f(c)
1	1.000000	2.000000	1.151744	−0.274401
2	1.100000	2.000000	1.176841	−0.130743
3	1.151744	2.000000	1.188628	−0.060876
4	1.176841	2.000000	1.194079	−0.028041
5	1.188628	2.000000	1.196582	−0.012852
6	1.194079	2.000000	1.197728	−0.005877
7	1.196582	2.000000	1.198251	−0.002685
8	1.197728	2.000000	1.198490	−0.001226
9	1.198251	2.000000	1.198600	−0.000560
10	1.198490	2.000000	1.198649	−0.000255
11	1.198600	2.000000	1.198672	−0.000117
12	1.198649	2.000000	1.198683	−0.000053
13	1.198672	2.000000	1.198687	−0.000024
14	1.198683	2.000000	1.198689	−0.000011
15	1.198687	2.000000	1.198690	-5.059540×10^{-6}
16	1.198689	2.000000	1.198691	-2.309250×10^{-6}
17	1.198690	2.000000	1.198691	-1.053976×10^{-6}
18	1.198691	2.000000	1.198691	-4.810502×10^{-7}
19	1.198691	2.000000	1.198691	-2.195584×10^{-7}
20	1.198691	2.000000	1.198691	-1.002097×10^{-7}
21	1.198691	2.000000	1.198691	-4.573716×10^{-8}
22	1.198691	2.000000	1.198691	-2.087511×10^{-8}
23	1.198691	2.000000	1.198691	-9.527708×10^{-9}
24	1.198691	2.000000	1.198691	-4.348584×10^{-9}
25	1.198691	2.000000	1.198691	-1.984758×10^{-9}
26	1.198691	2.000000	1.198691	-9.058736×10^{-10}
27	1.198691	2.000000	1.198691	-4.134550×10^{-10}
28	1.198691	2.000000	1.198691	-1.887051×10^{-10}
29	1.198691	2.000000	1.198691	-8.612933×10^{-11}
30	1.198691	2.000000	1.198691	-3.930900×10^{-11}
31	1.198691	2.000000	1.198691	-1.794032×10^{-11}
32	1.198691	2.000000	1.198691	-8.189893×10^{-12}
33	1.198691	2.000000	1.198691	-3.736567×10^{-12}
34	1.198691	2.000000	1.198691	-1.705303×10^{-12}
35	1.198691	2.000000	1.198691	-7.780443×10^{-13}

Accuracy is acheived and result is :

c1.19869124351587

f[c]=$-7.780442956573097\times10^{-13}$

4. Newton-Raphson Method

```
NewtonRapshon[x0_, max_, err_] := Module[{}, k = 0; p0 = N[x0];
  p1 = p0;
  Print["n    x0    f[x0]    f'[x0]    x1"];
  While[(k < max && Abs[f[p1]] > err),
    p0 = p1;
    If[f'[p0] == 0, Print["p0 is not correct"]; Exit[];,
     p1 = p0 - f[p0] / f'[p0];
     k = k + 1;];
    Print[k, PaddedForm[N[p0, 16], {10, 6}], PaddedForm[N[f[p0], 16], {10, 6}],
     PaddedForm[N[f'[p0], 16], {10, 6}], PaddedForm[N[p1], {10, 6}]];]
   Print["P after ", k, " iteration =", NumberForm[p1, 16]];
  Print["f[p]=", NumberForm[f[p1], 16]];]
```

```
f[x_] := x^3 + 2 x^2 - 3 x - 1;
NewtonRapshon[2, 13, 10^-8];
```

```
n    x0    f[x0]    f'[x0]    x1
1    2.000000    9.000000   17.000000    1.470588
2    1.470588    2.093833    9.370242    1.247133
3    1.247133    0.308997    6.654550    1.200699
4    1.200699    0.012279    6.127827    1.198695
5    1.198695    0.000022    6.105388    1.198691
P after 5 iteration =1.19869124352843
```

f[p]=$7.59046159259924\times10^{-11}$

5. Gauss Elimination Method

In[4]:=
```
Gausselim[A0_] := Module[{a = N[A0]}, Print[MatrixForm[a]];
  size = Dimensions[a];
  n = size[[1]];
  m = size[[2]];
  For[i = 1, i ≤ n - 1, i = i + 1,
   For[k = i + 1, k ≤ n, k = k + 1, factor = a[[k, i]] / a[[i, i]];
     For[p = i, p ≤ m, p = p + 1, a[[k, p]] = a[[k, p]] - factor * a[[i, p]];];];];
  Print[MatrixForm[a]];
  ClearAll[x, i];
  x[n] = a[[n, m]] / a[[n, n]];
  Print[x[n]];
  For[i = n - 1, i ≥ 1, i = i - 1, s = 0;
   For[j = i + 1, j ≤ n, j = j + 1, s = s + a[[i, j]] * x[j];];
   x[i] = (a[[i, m]] - s) / (a[[i, i]]);
   Print[x[i]];];]
```

In[5]:=
```
a = {{1, 2, 3, 1}, {2, 6, 10, 0}, {3, 14, 28, -8}};
Gausselim[a]
```

$$\begin{pmatrix} 1. & 2. & 3. & 1. \\ 2. & 6. & 10. & 0. \\ 3. & 14. & 28. & -8. \end{pmatrix}$$

$$\begin{pmatrix} 1. & 2. & 3. & 1. \\ 0. & 2. & 4. & -2. \\ 0. & 0. & 3. & -3. \end{pmatrix}$$

-1.

1.

2.

6. Gauss-Jordan Method

In[27]:=

```
(ClearA11[a, n, m, q, max, w, i, r, v, j, k, p, temp];)
a = {{3, 2, -4, 3}, {2, 3, 3, 15}, {5, -3, 1, 14}};
Print[MatrixForm[a]];
size = Dimensions[a];
n = size[[1]];
m = size[[2]];
For[i = 1, i ≤ n, i = i + 1,
  maxtemp = Max[a[[i ;; n, i]]];
  position = Position[a[[i ;; n, i]], maxtemp];
  position = First[First[position]];
  position = position + (i - 1);
  a[[{i, position}]] = a[[{position, i}]];
  temp = a[[i, i]];
  For[p = 1, p ≤ m, p = p + 1,
   a[[i, p]] = a[[i, p]] / temp];
  For[j = 1, j ≤ n, j = j + 1,
   If[i != j, factor = a[[j, i]] / a[[i, i]];
     For[k = 1, k ≤ m, k = k + 1, a[[j, k]] = a[[j, k]] - a[[i, k]] * factor];];];];
Print[MatrixForm[a]]
```

$$\begin{pmatrix} 3 & 2 & -4 & 3 \\ 2 & 3 & 3 & 15 \\ 5 & -3 & 1 & 14 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

7. Gauss-Jacobi Method

```
In[41]:= Gaussjacobi[A0_, B0_, X0_, max_] :=
  Module[{A = N[A0], B = N[B0], i, j, k = 0, n = Length[X0], X = X0, Xold = X0},
   Print["X", 0, "=" X];
   While[k < max,
    For[i = 1, i ≤ n, i = i + 1,
     X[[i]] =
      (B[[i]] - Sum[A[[i, j]] * Xold[[j]], {j, 1, i - 1}] - Sum[A[[i, j]] * Xold[[j]], {j, i + 1, n}]) / A[[i, i]]];
    Print["X", k + 1, "=", NumberForm[X, 10]];
    If[Max[Abs[X - Xold]] < 5 * 10^-6,
     Print["Solution with convergence tolerance of 5*10^-6 = ",
      NumberForm[X, 10]];
     Break[];,
     Xold = X;
     k = k + 1;];];]

In[42]:= A0 = {{4, 2, -1}, {2, 4, 1}, {-1, 1, 4}}; B0 = {{1}, {-1}, {1}}; X0 = {{0}, {0}, {0}};
Gaussjacobi[A0, B0, X0, 50]
```

```
X0{{0}, {0}, {0}}
X1={{0.25}, {-0.25}, {0.25}}
X2={{0.4375}, {-0.4375}, {0.375}}
X3={{0.5625}, {-0.5625}, {0.46875}}
X4={{0.6484375}, {-0.6484375}, {0.53125}}
X5={{0.70703125}, {-0.70703125}, {0.57421875}}
X6={{0.7470703125}, {-0.7470703125}, {0.603515625}}
X7={{0.7744140625}, {-0.7744140625}, {0.6235351563}}
X8={{0.7930908203}, {-0.7930908203}, {0.6372070313}}
X9={{0.805847168}, {-0.805847168}, {0.6465454102}}
X10={{0.8145599365}, {-0.8145599365}, {0.652923584}}
X11={{0.8205108643}, {-0.8205108643}, {0.6572799683}}
X12={{0.8245754242}, {-0.8245754242}, {0.6602554321}}
X13={{0.8273515701}, {-0.8273515701}, {0.6622877121}}
X14={{0.8292477131}, {-0.8292477131}, {0.6636757851}}
X15={{0.8305428028}, {-0.8305428028}, {0.6646238565}}
X16={{0.8314273655}, {-0.8314273655}, {0.6652714014}}
X17={{0.8320315331}, {-0.8320315331}, {0.6657136828}}
X18={{0.8324441873}, {-0.8324441873}, {0.6660157666}}
X19={{0.8327260353}, {-0.8327260353}, {0.6662220936}}
X20={{0.832918541}, {-0.832918541}, {0.6663630176}}
X21={{0.8330500249}, {-0.8330500249}, {0.6664592705}}
X22={{0.8331398301}, {-0.8331398301}, {0.6665250125}}
X23={{0.8332011682}, {-0.8332011682}, {0.666569915}}
X24={{0.8332430628}, {-0.8332430628}, {0.6666005841}}
X25={{0.8332716774}, {-0.8332716774}, {0.6666215314}}
X26={{0.8332912216}, {-0.8332912216}, {0.6666358387}}
X27={{0.8333045705}, {-0.8333045705}, {0.6666456108}}
X28={{0.8333136879}, {-0.8333136879}, {0.6666522852}}
X29={{0.8333199153}, {-0.8333199153}, {0.666656844}}
X30={{0.8333241686}, {-0.8333241686}, {0.6666599576}}
```

Solution with convergence tolerance of $5*10^{-6}$ =
{{0.8333241686}, {-0.8333241686}, {0.6666599576}}

8. Gauss-Seidel Method

```
In[11]:= GaussSeidal[A0_, B0_, X0_, max_] :=
  Module[{A = N[A0], B = N[B0], i, j, k = 0, n = Length[X0], X = X0, Xold = X0},
   Print["X", 0, "=" X];
   While[k < max,
    For[i = 1, i ≤ n, i = i + 1,
     X[[i]] = (B[[i]] - Sum[A[[i, j]] * X[[j]], {j, 1, i-1}] - Sum[A[[i, j]] * Xold[[j]], {j, i+1, n}]) / A[[i, i]]];
    Print["X", k + 1, "=", NumberForm[X, 10]];
    If[Max[Abs[X - Xold]] < 5 * 10^-6,
     Print["Solution with convergence tolerance of 5*10^-6 = ",
      NumberForm[X, 10]];
     Break[];,
     Xold = X;
     k = k + 1;];];]
```

In[14]:= $A0 = \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix}$; $B0 = \begin{pmatrix} 2 \\ 4 \\ 10 \end{pmatrix}$; $X0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$;

```
GaussSeidal[A0, B0, X0, 20]

X0{{0}, {0}, {0}}

X1={{0.5}, {1.125}, {2.78125}}

X2={{0.78125}, {1.890625}, {2.97265625}}

X3={{0.97265625}, {1.986328125}, {2.996582031}}

X4={{0.9965820313}, {1.998291016}, {2.999572754}}

X5={{0.9995727539}, {1.999786377}, {2.999946594}}

X6={{0.9999465942}, {1.999973297}, {2.999993324}}

X7={{0.9999933243}, {1.999996662}, {2.999999166}}

X8={{0.9999991655}, {1.999999583}, {2.999999896}}

X9={{0.9999998957}, {1.999999948}, {2.999999987}}

Solution with convergence tolerance of 5*10^-6 =
 {{0.9999998957}, {1.999999948}, {2.999999987}}
```

9. Lagrange Interpolation

```
In[13]:= Lagrange1[x_, f_, y_] := Module[{s = 0, m, p = 1, prod},
   m = Length[x];
   For[i = 1, i ≤ m, i = i + 1,
    p = 1;
    For[j = 1, j ≤ m, j = j + 1,
     If[j ≠ i,
       prod = (y - x[[j]]) / (x[[i]] - x[[j]]); p = p * prod;];];
    s = s + p * f[[i]] ;];
   Print["Function value at ", y "is ", s];
   Print["Absolute error = ", Abs[s - Exp[y]]];]
```

```
In[17]:= x = {-1, 0, 1};
f = {Exp[-1], 1, Exp[1]};
Lagrange1[x, f, 0.5]
```

Function value at 0.5 is 1.72337

Absolute error = 0.0746495

```
In[20]:= Lagrange1[x, f, -0.7]
```

Function value at -0.7 is 0.443469

Absolute error = 0.0531166

```
In[21]:= Lagrange1[x, f, 0.3]
```

Function value at 0.3 is 1.40144

Absolute error = 0.0515788

10. Newton's Divided Difference Interpolation

```
NthDividedDiff[x0_, f0_, start_, end_] := Module[{x = x0, f = f0, i = start, j = end, ans},
   If[i == j, Return[f[[i]]],
     ans = (NthDividedDiff[x, f, i + 1, j] - NthDividedDiff[x, f, i, j - 1]) /
        (x[[j]] - x[[i]]);
     Return[ans]];];

NewtonDDPoly[x0_, f0_] := Module[{x1 = x0, f = f0, n, P, k, j},
   n = Length[x1];
   P[y_] = 0;
   For[i = 1, i ≤ n, i++,
    prod[y_] = 1;
    For[k = 1, k ≤ i - 1, k++, prod[y_] = prod[y] * (y - x1[[k]])];
    P[y_] = P[y] + NthDividedDiff[x1, f, 1, i] * prod[y]];
   Return[P[y]];];
nodes = {0, 1, 3};
values = {1, 3, 55};
NewtonPoly[y_] = NewtonDDPoly[nodes, values];
NewtonPoly[y]
NewtonPoly[y_] = Simplify[NewtonPoly[y]];
NewtonPoly[y]
NewtonPoly[2]
```

$1 + 2 y + 8 (-1 + y) y$

$1 - 6 y + 8 y^2$

21

11. Trapezoidal Rule

```
TrapezoidalRule[a_, b_, f_] := Module[{}, k = ((b - a) / 2) (f[a] + f[b]);
  Print["Integral Value is : ", k];]
  k
f[x_] := 1 / (1 + x)

N[TrapezoidalRule[0, 1, f]]

                          3
Integral Value is : -
                          4
```

12. Simpson's Rule

```
(*Aim: To approximate the value of integrals ∫₀¹xdx,
∫₀¹e^-x dx and ∫₀¹1/(1+x^2) dx using Simpson Rule*)
(*Programming*)
SimpsonRule[a_, b_, f_] := Module[{}, k = ((b - a) / 6) * (f[a] + 4 f[(a + b) / 2] + f[b]);
  Print["Integral Value is : ", k];]

f[x_] := x

SimpsonRule[0, 1, f]
```

Integral Value is : $\frac{1}{2}$

```
f[x_] := E^(-x)

SimpsonRule[0, 1, f]
```

Integral Value is : $\frac{1}{6}\left(1+\frac{1}{e}+\frac{4}{\sqrt{e}}\right)$

```
f[x_] := 1 / (1 + x^2)

SimpsonRule[0, 1, f]
```

Integral Value is : $\frac{47}{60}$

13. Euler's Method

In[3]:=
```
eulerMethod[a_, b_, h_, f_, y0_] := Module[
 {n, xi, yi, OutputDetails},
 n = (b - a)/h;
 xi = Table[a + h(j - 1), {j, 1, n + 1}];
 yi = Table[0, {n + 1}];
 yi[[1]] = y0;
 OutputDetails = {{0, xi[[1]], y0}};

 For[i = 1, i ≤ n, i++,
  yi[[i + 1]] = yi[[i]] + h*f[xi[[i]], yi[[i]]];
  OutputDetails = Append[OutputDetails, {i, xi[[i + 1]], yi[[i + 1]]}]
 ];

 Grid[
  Prepend[
   Transpose[{Range[0, n], xi, yi}],
   {"i", "xi", "yi"}
  ],
  Frame → All,
  Alignment → Right
 ]
];



f[x_, y_] := 2*x + y;
eulerMethod[0, 1, 0.2, f, 1]
```

Out[5]=

i	xi	yi
0	0.	1
1	0.2	1.2
2	0.4	1.52
3	0.6	1.984
4	0.8	2.6208
5	1.	3.46496

14. Runge-Kutta Method

```
In[28]:= RungeKutta4thOrder[a0_, b0_, h0_, f_, y0_] := Module[
  {a = a0, b = b0, n, h = h0, xi, yi, k1, k2, k3, k4, OutputDetails},
  n = (b - a)/h;
  xi = Table[a + (j - 1) h, {j, 1, n + 1}];
  yi = Table[0, {n + 1}];
  yi[[1]] = y0;
  OutputDetails = {{0, xi[[1]], y0}};

  For[i = 1, i ≤ n, i++,
   k1 = h * f[xi[[i]], yi[[i]]];
   k2 = h * f[xi[[i]] + h/2, yi[[i]] + k1/2];
   k3 = h * f[xi[[i]] + h/2, yi[[i]] + k2/2];
   k4 = h * f[xi[[i]] + h, yi[[i]] + k3];

   yi[[i + 1]] = yi[[i]] + (k1 + 2*k2 + 2*k3 + k4)/6;
   OutputDetails = Append[OutputDetails, {i, N[xi[[i + 1]]], N[yi[[i + 1]]]}];
  ];

  Print[
   NumberForm[
    TableForm[OutputDetails, TableHeadings → {None, {"i", "xi", "yi"}}],
    6
   ]
  ];
 ];



 f[x_, y_] := 2*x + y;



 Print["f(x,y)=", f[x, y]];
 yi = RungeKutta4thOrder[0, 1, 0.2, f, 1];
```

$f(x, y) = 2x + y$

i	xi	yi
0	0.	1
1	0.2	1.2642
2	0.4	1.67545
3	0.6	2.26632
4	0.8	3.07656
5	1.	4.15475