Alok | 232827 | B.Sc Mathematics(H) - Second Year | Numerical Analysis

Practical 01: Bisection Method

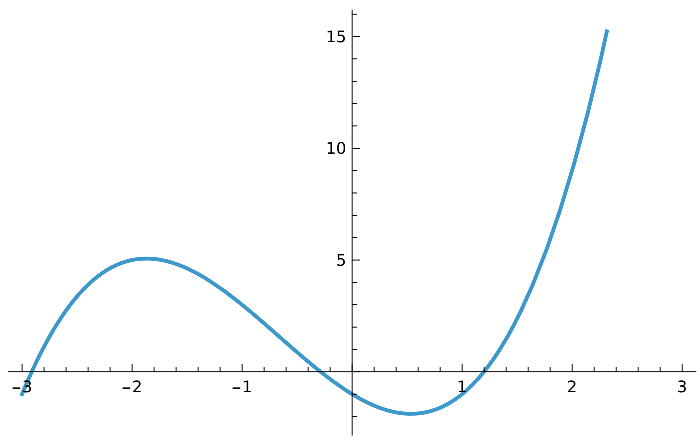
Bisection Method: Perform 10 Iteration and find the root to the expression.

In[10]:=
```
f[x_] = x^3 + 2 x^2 - 3 x - 1
Plot[f[x], {x, -3, 3}]
```

Out[10]=

$-1 - 3x + 2x^2 + x^3$

Out[11]=



In[24]:=
```
a = 1;
b = 2;
f[a]
f[b]
c = (a + b)/ 2.0
```

Out[26]=

-1

Out[27]=

9

Out[28]=

1.5

In[29]:=
```
For[i = 1, i ≤ 10, i++, If[f[a]*f[b] < 0, a = c , b = c];
c = (a + b)/2.0;
Print[c]];
```

```
1.75
1.625
1.5625
1.53125
1.51563
1.50781
1.50391
1.50195
1.50098
1.50049
```

Practical 02: Regula Falsi Method

In[68]:=
```
f[x_] = x^3 - 2.0
a = 1.0
b = 2.0
f[a]
f[b]
e = 0.00001
```

Out[68]= $-2. + x^3$

Out[69]= 1.

Out[70]= 2.

Out[71]= -1.

Out[72]= 6.

Out[73]= 0.00001

In[74]:=
```
p0 = N[2^(1/3)]
p1 = b - f[b] (b - a) / (f[b] - f[a])
```

Out[74]= 1.25992

Out[75]= 1.14286

In[76]:=
```
While[Abs[p1 - p0] > e, If[f[b]* f[p1] < 0, b = p1, a = p1];
p1 = b - (f[b] (b - a) / (f[b] - f[a]));
Print[p1]];
```

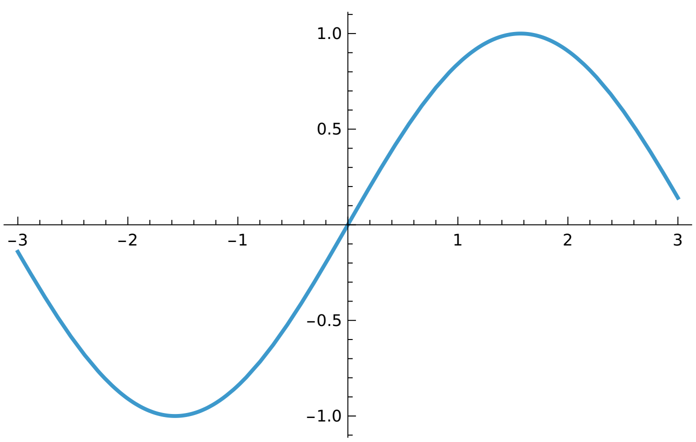
```
1.28994
1.25292
1.26159
1.25952
1.26002
1.2599
1.25993
```

Practical 03: Fixed Point Iteration

In[77]:=
```
h[x_] = Sin[x]
p0 = 0.05;
Plot[h[x], {x, -3, 3}]
```

Out[77]=
```
Sin[x]
```

Out[79]=



In[80]:=
```
For[i = 1, i ≤ 10, i++, p1 = h[p0];
p0 = p1;
Print[p1]]
```

```
0.0499792
0.0499584
0.0499376
0.0499168
0.0498961
0.0498754
0.0498547
0.0498341
0.0498135
0.0497929
```

Practical 04: Newton Method

In[81]:=

```
f[x_] = x^3 - 13
p = 2.351334
p0 = 1.0
p1 = p0 - f[p0]/f'[p0]
While[Abs[p1 - p] > e, p0 = p1;
p1 = p0 - f[p0]/f'[p0];
Print[p1]
]
```

Out[81]=

$-13 + x^3$

Out[82]=

```
2.35133
```

Out[83]=

```
1.
```

Out[84]=

```
5.
```

```
3.50667
2.69018
2.39222
2.35203
2.35133
```

In[86]:=
```
g[x_] = x^3 + 2 x^2 - 3 x - 1
p0 = 1.0
p1 = p0 - g[p0] / g'[p0]
```

Out[86]=

$-1 - 3 x + 2 x^2 + x^3$

Out[87]=

```
1.
```

Out[88]=

```
1.25
```

In[89]:=
```
For[i = 1, i ≤ 10, i++, p0 = p1;
p1 = p0 - g[p0] / g'[p0];
Print[p1]
]
```

```
1.20093
1.1987
1.19869
1.19869
1.19869
1.19869
1.19869
1.19869
1.19869
1.19869
```

Practical 05: Secant Method

In[90]:=

```
f[x_] = x^3 - 13
p = 2.351334
p0 = 2.0
p1 = 3.0
e = 0.005
p2 = p1 - f[p1](p1 - p0)/(f[p1]- f[p0])
While[Abs[p - p2] > e, p0 =p1; p1 = p2;
p2 = p1 - f[p1](p1 -p0)/(f[p1] - f[p0]);
Print[p2]
]
```

Out[90]=

$-13+x^3$

Out[91]=

2.35133

Out[92]=

2.

Out[93]=

3.

Out[94]=

0.005

Out[95]=

2.26316

2.33051

2.35214

Practical 06 : LU Decompositon

In[97]:=

```
M = {{1, 2, 3}, {2, 5, 7}, {2, 4, 1}}
{lu, p, c} = LUDecomposition[M]
MatrixForm[l = LowerTriangularize[lu, -1] + IdentityMatrix[3]]
```

Out[97]=

{{1, 2, 3}, {2, 5, 7}, {2, 4, 1}}

Out[98]=

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

Out[99]//MatrixForm=

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

In[100]:=

```
MatrixForm[u = UpperTriangularize[lu]]
```

Out[100]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & -5 \end{pmatrix}$$

Practical 07 : Gauss Jacobi Method

In[101]:=

```
3 x + y + z = 5
x + 4 y + z  = 4
x + 2 y - 3 z = 0
Array[x, 10]
```

Out[101]=

5

Out[102]=

4

Out[103]=

0

Out[104]=

{0, x[2], x[3], x[4], x[5], x[6], x[7], x[8], x[9], x[10]}

In[105]:=

```
Array[y, 10]
Array[z, 10]
```

Out[105]=

{0, y[2], y[3], y[4], y[5], y[6], y[7], y[8], y[9], y[10]}

Out[106]=

{0, z[2], z[3], z[4], z[5], z[6], z[7], z[8], z[9], z[10]}

In[107]:=

```
x[0] = 0
y[0] = 0
z[0] = 0
```

Out[107]=

0

Out[108]=

0

Out[109]=

0

In[111]:=

```
For[i = 2, i ≤ 10, i++, x[i] = (5 - y[i - 1] - z[i - 1]) / 3.0;
y[i] = (4 - x[i - 1] - z[i - 1]) / 4.0;
z[i] = (x [i - 1] + 2 y[i - 1]) / 3.0;
Print[" The Value of X is - ", x[i]];
Print[" The Value of Y is - ", y[i]] ;
Print[" The Value of Z is - ", z[i]]
Print["____________________"]
]
```

```
 The Value of X is – 1.66667
 The Value of Y is – 1.
 The Value of Z is – 0.
________________
 The Value of X is – 1.33333
 The Value of Y is – 0.583333
 The Value of Z is – 1.22222
________________
 The Value of X is – 1.06481
 The Value of Y is – 0.361111
 The Value of Z is – 0.833333
________________
 The Value of X is – 1.26852
 The Value of Y is – 0.525463
 The Value of Z is – 0.595679
________________
 The Value of X is – 1.29295
 The Value of Y is – 0.533951
 The Value of Z is – 0.773148
________________
 The Value of X is – 1.23097
 The Value of Y is – 0.483475
 The Value of Z is – 0.786951
________________
 The Value of X is – 1.24319
 The Value of Y is – 0.49552
 The Value of Z is – 0.732639
________________
 The Value of X is – 1.25728
 The Value of Y is – 0.506042
 The Value of Z is – 0.744744
________________
 The Value of X is – 1.24974
 The Value of Y is – 0.499494
 The Value of Z is – 0.756455
________________
```

Practical 08 : Gauss Seidel Method

In[22]:=
```
5 x + y + 2 z = 10
-3 x + 9 y + 4 z = -14
x + 2 y - 7 z = -33
```

Set: RowBox[{ Tag , StyleBox[TagBox[Plus , Function[Short[Slot[1], 5]]], ShowStringCharacters -> False], in , StyleBox[TagBox[RowBox[{RowBox[{ 5 , , x }], +, y , +, RowBox[{ 2 , , z }]}], Function[Short[Slot[1], 5]]], ShowStringCharacters -> False], is Protected.}]

Out[22]=
```
10
```

Set: RowBox[{ Tag , StyleBox[TagBox[Plus , Function[Short[Slot[1], 5]]], ShowStringCharacters -> False], in , StyleBox[TagBox[RowBox[{RowBox[{RowBox[{ – , 3 }], , x }], +, RowBox[{ 9 , , y }], +, RowBox[{ 4 , , z }]}], Function[Short[Slot[1], 5]]], ShowStringCharacters -> False], is Protected.}]

Out[23]=
```
-14
```

Set: RowBox[{ Tag , StyleBox[TagBox[Plus , Function[Short[Slot[1], 5]]], ShowStringCharacters -> False], in , StyleBox[TagBox[RowBox[{ x , +, RowBox[{ 2 , , y }], – , RowBox[{ 7 , , z }]}], Function[Short[Slot[1], 5]]], ShowStringCharacters -> False], is Protected.}]

Out[24]=
```
-33
```

In[25]:=
```
Array[x, 10]
Array[y, 10]
Array[z, 10]
```

Out[25]=
```
{0, x[2], x[3], x[4], x[5], x[6], x[7], x[8], x[9], x[10]}
```

Out[26]=
```
{0, y[2], y[3], y[4], y[5], y[6], y[7], y[8], y[9], y[10]}
```

Out[27]=
```
{0, z[2], z[3], z[4], z[5], z[6], z[7], z[8], z[9], z[10]}
```

In[28]:=
```
x[1] = 0
y[1] = 0
z[1] = 0
```

Out[28]=
```
0
```

Out[29]=
```
0
```

Out[30]=
```
0
```

In[33]:=
```
For[i = 2, i ≤ 10, i++,
 x[i] = (10 - y[i - 1] - 2 z[i - 1])/5.0;
 y[i] = (-14 + 3 x[i] - 4 z[i - 1])/9.0;
 z[i] = (33 + x[i] + 2 y[i])/7.0;
 Print["The value of x is ", x[i]];
 Print["The value of y is ", y[i]];
 Print["The value of z is ", z[i]];
  Print["------------------------------"]
]
```

```
The value of x is 2.
The value of y is -0.888889
The value of z is 4.74603
-----------------------------
The value of x is 0.279365
The value of y is -3.57178
The value of z is 3.73369
-----------------------------
The value of x is 1.22088
The value of y is -2.80801
The value of z is 4.08641
-----------------------------
The value of x is 0.927039
The value of y is -3.06272
The value of z is 3.97166
-----------------------------
The value of x is 1.02388
The value of y is -2.97944
The value of z is 4.00929
-----------------------------
The value of x is 0.992174
The value of y is -3.00674
The value of z is 3.99696
-----------------------------
The value of x is 1.00256
The value of y is -2.99779
The value of z is 4.001
-----------------------------
The value of x is 0.99916
The value of y is -3.00072
The value of z is 3.99967
-----------------------------
The value of x is 1.00028
The value of y is -2.99976
The value of z is 4.00011
-----------------------------
```

Practical 09 : Lagrange Interpolation

In[34]:=
```
Array[x1, 5]
```
Out[34]=
```
{x1[1], x1[2], x1[3], x1[4], x1[5]}
```

In[35]:=
```
x1[1] = 0
x1[2] = 1
x1[3] = 2
x1[4] = 4
x1[5] = 5
y1[1] = 0
y1[2] = 2
y1[3] = 5
y1[4] = 8
y1[5] = 4
```
Out[35]=
```
0
```
Out[36]=
```
1
```
Out[37]=
```
2
```
Out[38]=
```
4
```
Out[39]=
```
5
```
Out[40]=
```
0
```
Out[41]=
```
2
```
Out[42]=
```
5
```
Out[43]=
```
8
```
Out[44]=
```
4
```

```
j = 1
L1[x_] = Product[(x - x1[i]) / (x1 [1] - x1[i]), {i, 1, j - 1}] *
  Product[(x - x1[i]) / (x1[1] - x1[i]), {i, j + 1, 5}]
```

Out[45]=

1

Out[46]=

$$\frac{1}{40} (1 - x) (2 - x) (4 - x) (5 - x)$$

In[49]:=

```
j = 2
L2[x_] = Product[(x - x1[i]) / (x1[2] - x1[i]), {i, 1, j - 1}] *
  Product[(x - x1[i]) / (x1[2] - x1[i]), {i, j + 1, 5}]
```

Out[49]=

2

Out[50]=

$$\frac{1}{12} (2 - x) (4 - x) (5 - x) x$$

In[51]:=

```
j = 3
L3[x_] = Product[(x - x1[i]) / (x1[3] - x1[i]), {i, 1, j - 1}] *
  Product[(x - x1[i]) / (x1[3] - x1[i]) , {i, j + 1, 5}]
```

Out[51]=

3

Out[52]=

$$\frac{1}{12} (4 - x) (5 - x) (-1 + x) x$$

In[53]:=

```
P[x_] = y1[1] * L1[x] + y1[2] * L2[x] + y1[3] * L3[x]
```

Out[53]=

$$\frac{1}{6} (2 - x) (4 - x) (5 - x) x + \frac{5}{12} (4 - x) (5 - x) (-1 + x) x$$

In[54]:=

```
Simplify[P[x]]
```

Out[54]=

$$\frac{1}{12} x \left(-20 + 69 x - 28 x^2 + 3 x^3\right)$$

Practical 10 : Trapezoidal Rule

In[1]:=

```
Array[x, 9]
g[x_] = 1 / x
```

Out[1]= {x[1], x[2], x[3], x[4], x[5], x[6], x[7], x[8], x[9]}

Out[2]= $\frac{1}{x}$

```
In[3]:= a = 1
        b = 2
        h = N[1 / 8]
Out[3]= 1
Out[4]= 2
Out[5]= 0.125

In[6]:= x[1] = N[a]
        x[9] = N[b]
Out[6]= 1.
Out[7]= 2.

In[8]:= sum = (h / 2) (g[a] + g[b])
Out[8]= 0.09375

In[9]:= For[i = 2, i ≤ 8, i++, x[i] = x[i - 1] + h;
        sum = sum + (h / 2) (2 * g[x[i]])
        ]
        Print["Answer: ", N[sum]]
        Answer: 0.694122
```

Practical 11 : Euler's Method

```
In[11]:= f[r_, s_] = r + s
         h = .2
         p = .0
         q = 1.0
         n = (q - p) / h
Out[11]=
         r + s
Out[12]=
         0.2
Out[13]=
         0.
Out[14]=
         1.
Out[15]=
         5.
```

```
In[16]:= Array[r, 6]
         Array[s, 6]
Out[16]=
         {r[1], r[2], r[3], r[4], r[5], r[6]}
Out[17]=
         {s[1], s[2], s[3], s[4], s[5], s[6]}

In[21]:= r[1] = .0
         s[1] = 2.0
         For[i = 2, i ≤ 6, i++, r[i] = r[i - 1] + h;
         s[i] = s[i - 1] + h * (f[r [i - 1], s[i - 1]]);
         Print["r = ", r[i]];
         Print["s = ", s[i]];
         Print["______"]
         ]
Out[21]=
         0.
Out[22]=
         2.

         r = 0.2

         s = 2.4

         ______

         r = 0.4

         s = 2.92

         ______

         r = 0.6

         s = 3.584

         ______

         r = 0.8

         s = 4.4208

         ______

         r = 1.

         s = 5.46496

         ______
```

Practical 12 : Runge Kutta Method

```
In[1]:= f[p_,  q_] = 4 Exp[.8 p] - .5 q
        h = .1
        a = .0
        b = 1.
        n = (b - a)/h
```

Out[1]= $4 e^{0.8 p} - 0.5 q$

```
Out[2]= 0.1

Out[3]= 0.

Out[4]= 1.

Out[5]= 10.
```

```
In[6]:= Array[p, 11]
        Array[q, 11]

Out[6]= {p[1], p[2], p[3], p[4], p[5], p[6], p[7], p[8], p[9], p[10], p[11]}

Out[7]= {q[1], q[2], q[3], q[4], q[5], q[6], q[7], q[8], q[9], q[10], q[11]}
```

```
In[8]:= p[1] = 0.0
        q[1] = 2.0

Out[8]= 0.

Out[9]= 2.
```

```
In[10]:= For[i = 2, i ≤ 11, i++,
         p[i] = p[i - 1] + h;
         k1 = f[p[i - 1], q[i - 1]];
         k2 = f[p[i - 1] + h / 2, q[i - 1] + k1 * h / 2];
         k3 = f[p[i - 1] +  h/2, q[i - 1] + k2 * h / 2];
         k4 = f[p[i - 1] + h, q [i - 1 ] + k3*h];
         q[i] = q[i - 1] + (k1  + 2 * k2 + 2 * k3 + k4) *h / 6.0 ;
         Print["k1 = ", k1 " | k2", k2, " | k3", k3, " | k4 " , k4 ];
         Print["p = ", p[i], "q = ", q[i]]
         ]
```

k1 = 3. | k23.08824 | k33.08604 | k4 3.17885

p = 0.1q = 2.30879

k1 = 3.17875 | k23.27612 | k33.27369 | k4 3.37596

p = 0.2q = 2.63636

k1 = 3.37586 | k23.48303 | k33.48035 | k4 3.5928

p = 0.3q = 2.98462

k1 = 3.59269 | k23.71039 | k33.70745 | k4 3.83083

p = 0.4q = 3.35561

k1 = 3.83071 | k23.95975 | k33.95652 | k4 4.09167

p = 0.5q = 3.75152

k1 = 4.09154 | k24.23278 | k34.22925 | k4 4.37707

p = 0.6q = 4.17473

k1 = 4.37693 | k24.53132 | k34.52746 | k4 4.68895

p = 0.7q = 4.62779

k1 = 4.68879 | k24.85736 | k34.85315 | k4 5.02937

p = 0.8q = 5.11344

k1 = 5.0292 | k25.21306 | k35.20846 | k4 5.40059

p = 0.9q = 5.63466

k1 = 5.4004 | k25.60077 | k35.59576 | k4 5.80505

p = 1.q = 6.19463