Numerical Methods Practical File

By:

Name: keshav sharma

Course: B.Sc. Physics Hons. 3rd year (5th Semester)

Class Roll Number: 5362

University Roll Number: 2305567033

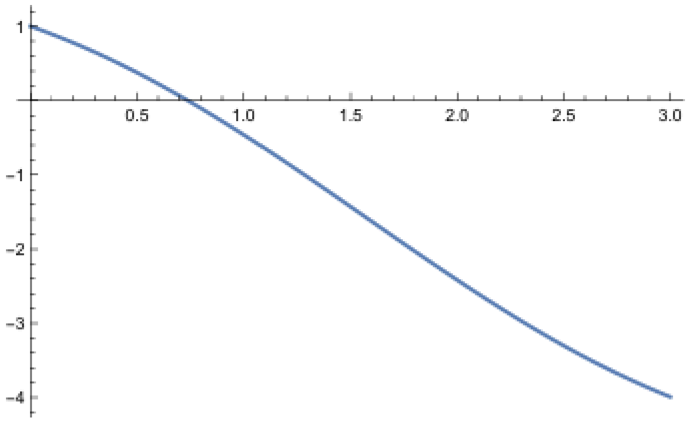
INDEX

Bisection Method

Question 1. Perform five iterations of the bisection method to obtain a root of the equation f (x) = cosx - x. Also, plot the function in the interval [0, 3].

```
Bisection[a0_, b0_, m_] := Module[{a = N[a0], b = N[b0], c, k}, c = (a + b) / 2;
   k = 1;
   While[k < m, If[Sign[f[b]] == Sign[f[c]], b = c, a = c];
    c = (a + b) / 2;
    Print[k, "th iteration is ", NumberForm[c, 16]];
    k = k + 1;];
   Print["Root after ", k, " iterations is = ", NumberForm[c, 16]];
   Print["f[c] = ", NumberForm[f[c], 16]];
  ];

f[x_] := Cos[x] - x;

Bisection[0, 1, 5];

Plot[f[x], {x, 0, 3}]
```

```
1th iteration is 0.75
2th iteration is 0.625
3th iteration is 0.6875
4th iteration is 0.71875
Root after 5 iterations is = 0.71875
f[c] = 0.03387937241806649
```



Question 2. Using the Bisection method find the root of the function f (x) = x3 - 2 in the interval [1, 2] with an error tolerance of 10^-4.

```
ClearAll;

f[x_] := x^3 - 2;

Bisection[a0_, b0_, error_, f_] := Module[{a = N[a0], b = N[b0], c, k = 1}, c = (a + b) / 2;
   While[Abs[b - a] > 2 * error, If[Sign[f[b]] == Sign[f[c]], b = c, a = c];
    c = (a + b) / 2;
    Print[k, "th iteration is = ", NumberForm[c, 16]];
    k++;];
   Print["Root after ", k, " iterations is = ", NumberForm[c, 16]];
   Print["f[c] = ", NumberForm[f[c], 16]];
  ];

error = 10^-4;
a = 1;
b = 2;

Bisection[a, b, error, f];
```

```
1th iteration is = 1.25
2th iteration is = 1.375
3th iteration is = 1.3125
4th iteration is = 1.28125
5th iteration is = 1.265625
6th iteration is = 1.2578125
7th iteration is = 1.26171875
8th iteration is = 1.259765625
9th iteration is = 1.2607421875
10th iteration is = 1.26025390625
11th iteration is = 1.260009765625
12th iteration is = 1.2598876953125
13th iteration is = 1.25994873046875
Root after 14 iterations is = 1.25994873046875
f[c] = 0.0001318234124028095
```

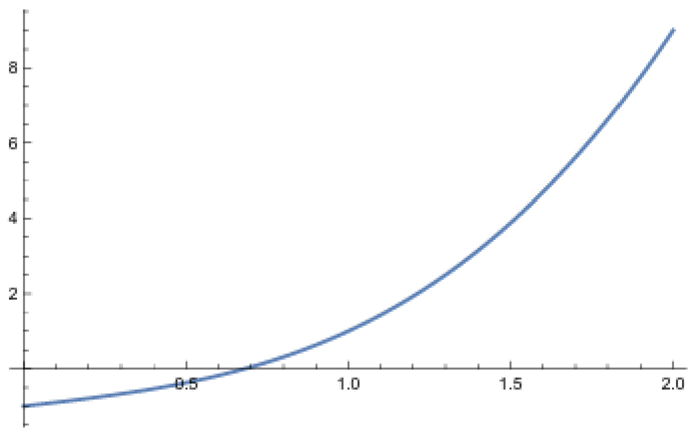
Secant Method

Question 3. Perform ten iterations of the secant method to obtain a root of the equation f (x) = x3 + x - 1. Also, plot the function in the interval [0, 2]

```
ClearAll;
f[x_] := x^3 + x - 1;

SecantMethod[x0_, x1_, max_] := Module[{p0 = N[x0], p1 = N[x1], p2, k = 0}, While[k < max,
    If[f[p1] == f[p0],
     Print["Division by zero - Secant method cannot proceed."];
     Return[];
    ];
    p2 = p1 - f[p1] (p1 - p0) / (f[p1] - f[p0]);
    Print["Root after ", k, " iteration is = ", NumberForm[p2, 16]];
    p0 = p1;
    p1 = p2;
    k++;];
   Print["Final approximation c = ", NumberForm[p2, 16]
   ];
   Print["f[c] = ", NumberForm[f[p2], 16]];
  ];

SecantMethod[0, 1, 9];

Plot[f[x], {x, 0, 2}]
```

```
Root after 0 iteration is = 0.5
Root after 1 iteration is = 0.6363636363636363
Root after 2 iteration is = 0.6900523560209423
Root after 3 iteration is = 0.6820204196481856
Root after 4 iteration is = 0.6823257814098928
Root after 5 iteration is = 0.6823278043590257
Root after 6 iteration is = 0.6823278038280184
Root after 7 iteration is = 0.6823278038280193
Root after 8 iteration is = 0.6823278038280193
Final approximation c = 0.6823278038280193
```
f[c] = $-1.110223024625157 \times 10^{-16}$

Out[•]=



Question 4. Find a real root of the equation f (x) = x3 - 2 using secant method with absolute error tolerence 10^-8.

```
ClearAll;

SecantMethod[x0_, x1_, error_, f_] := Module[{p0 = N[x0], p1 = N[x1], p2, k = 1},
  While[Abs[(f[p1] (p1 - p0)) / (f[p1] - f[p0])] > error,
   p2 = p1 - (f[p1] (p1 - p0)) / (f[p1] - f[p0]);
   Print["root after ", k, " iteration is ", NumberForm[p2, 16]];
   p0 = p1;
   p1 = p2;
   k++;];
  Print["Final Root = ", NumberForm[p1, 16]];
  Print["f[root] = ", NumberForm[f[p1], 16]];]
SecantMethod[x0, x1, error, f];
```

```
root after 1 iteration is 0.4871416534984858

root after 2 iteration is 0.5837796851369866

root after 3 iteration is 0.5673864490804865

root after 4 iteration is 0.5671425603070896

root after 5 iteration is 0.5671432904419066

Final Root = 0.5671432904419066
```

f[root] = $5.034095362788094 \times 10^{-11}$

Regula Falsi Method

Question 5. Write a program to find the real root of the following equation using regula falsi method with tolernce of 10^-6

f (x) = x3 - x - 1

```
ClearAll[RegulaFalsi];

RegulaFalsi[f_, a0_, b0_, error_, max_] :=
 Module[{a = N[a0], b = N[b0], c, fa, fb, fc, k = 1}, fa = f[a];
  fb = f[b];
  If[fa fb > 0,
   Print["The function has the same sign at the endpoints. Choose another interval."];
   Return[];
  ];
  While[k ≤ max, c = b - fb (b - a) / (fb - fa);
   fc = f[c];
   Print["Iteration ", k, ":  c = ", NumberForm[c, 16], "   f(c) = ", NumberForm[fc, 16]
   ];
   If[Abs[fc] < error, Print["Converged to root at iteration ", k];
    Return[c];
   ];
   If[fa fc < 0, b = c; fb = fc, a = c; fa = fc];
   k++;];
  Print["Stopped after max iterations. Final approximation = ", NumberForm[c, 16]
  ];
  Return[c];
 ]
f[x_] := x^3 - x - 1;
RegulaFalsi[f, 1, 2, 10^-6, 20]
```

```
Iteration 1:  c = 1.166666666666667   f(c) = -0.5787037037037033
Iteration 2:  c = 1.253112033195021   f(c) = -0.2853630296393199
Iteration 3:  c = 1.293437401918683   f(c) = -0.1295420928219719
Iteration 4:  c = 1.311281021487234   f(c) = -0.05658848726924948
Iteration 5:  c = 1.318988503566463   f(c) = -0.02430374718359696
Iteration 6:  c = 1.322282717465796   f(c) = -0.01036185006965229
Iteration 7:  c = 1.323684293855161   f(c) = -0.004403949880785074
Iteration 8:  c = 1.324279461731951   f(c) = -0.001869258374370908
Iteration 9:  c = 1.324531986580092   f(c) = -0.0007929591932531732
Iteration 10:  c = 1.324639093308037   f(c) = -0.0003363010300609925
Iteration 11:  c = 1.32468451516667   f(c) = -0.0001426137451630005
Iteration 12:  c = 1.324703776471376   f(c) = -0.00006047499488426311
Iteration 13:  c = 1.324711944079721   f(c) = -0.00002564379829328445
Iteration 14:  c = 1.324715407452098   f(c) = -0.00001087390403009536
```
Iteration 15: c = 1.324716876044874   f(c) = $-4.610916005676202\times10^{-6}$

Iteration 16: c = 1.324717498779053   f(c) = $-1.955187088675814\times10^{-6}$

Iteration 17: c = 1.324717762839675   f(c) = $-8.29066130414446\times10^{-7}$

```
Converged to root at iteration 17
```

Out[8]= 1.32472

Newton Raphson Method

Question 6. Find a root of the equation f (x) = x3 - 4 x + 2 using the Newton-Raphson method with absolute error tolerance 10^-8

```
ClearAll[NewtonRaphson];

NewtonRaphson[f_, x0_, error_, max_] := Module[{x = N[x0], xnew, k = 1, df},
  df[x_] := D[f[t], t] /. t → x;
  While[k ≤ max, xnew = x - f[x] / df[x];
   Print["Iteration ", k, ":  x = ", NumberForm[xnew, 16]];
   If[Abs[xnew - x] < error, Print["Converged to root at iteration ", k];
    Return[xnew];];
   x = xnew;
   k++;];
  Print["Stopped after max iterations. Final approximation = ", NumberForm[x, 16]];
  Return[x];]
f[x_] := x^3 - 4 x + 2;
NewtonRaphson[f, 0.5, 10^-8, 20]
```

```
Iteration 1:  x = 0.5384615384615384
Iteration 2:  x = 0.539188599680093
Iteration 3:  x = 0.5391888728108506
Iteration 4:  x = 0.5391888728108891
Converged to root at iteration 4
```

Out[14]=

```
0.539189
```

Question 7. Write apogram toshow LU Decomposition for

$$A = \begin{pmatrix} 2 & 1 & 4 \\ 3 & 4 & -1 \\ 1 & 2 & 3 \end{pmatrix}$$

```
ClearAll;

MyLUDecomposition[A0_, n_] := Module[{A = A0, i, p, m, L, U}, U = A0;
    L = IdentityMatrix[n];
    Print[MatrixForm[L], MatrixForm[U], " = ", MatrixForm[A0]];
    For[p = 1, p ≤ n - 1, p++,
     For[i = p + 1, i ≤ n, i++, m = A[[i, p]] / A[[p, p]];
       L[[i, p]] = m;
       A[[i]] = A[[i]] - m A[[p]];
       U = A;
       Print[MatrixForm[L], MatrixForm[U], " = ", MatrixForm[A0]];
      ];
    ];
    Print["L = ", MatrixForm[L]];
    Print["U = ", MatrixForm[U]];
   ];

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

MyLUDecomposition[A, 3];
```

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

$$\begin{pmatrix} 1 & 0 & 0 \\ \frac{3}{2} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 4 \\ 0 & \frac{5}{2} & -7 \\ 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 4 \\ 3 & 4 & -1 \\ 1 & 2 & 3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ \frac{3}{2} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 4 \\ 0 & \frac{5}{2} & -7 \\ 0 & \frac{3}{2} & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 4 \\ 3 & 4 & -1 \\ 1 & 2 & 3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ \frac{3}{2} & 1 & 0 \\ \frac{1}{2} & \frac{3}{5} & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 4 \\ 0 & \frac{5}{2} & -7 \\ 0 & 0 & \frac{26}{5} \end{pmatrix} = \begin{pmatrix} 2 & 1 & 4 \\ 3 & 4 & -1 \\ 1 & 2 & 3 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{3}{2} & 1 & 0 \\ \frac{1}{2} & \frac{3}{5} & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} 2 & 1 & 4 \\ 0 & \frac{5}{2} & -7 \\ 0 & 0 & \frac{26}{5} \end{pmatrix}$$

Question 8. Use the Gauss Jacobi iteration method to solve the system of equations
2x1 - x2 = 7
-x1 + 2 x2 - x3 = 1
-x2 + 2 x3 = 1
using 12 iterations and with initial guess X0 = (0, 0, 0) ^T

```
ClearAll;

GaussJacobi[A0_, B0_, X0_, max_] :=
  Module[{A = N[A0], B = N[B0], i, j, k = 0, n = Length[X0],
    X = X0, Xk = X0},
   Print[Subscript["X", 0], " = ", MatrixForm[X]
   ];
   While[k < max,
    For[i = 1, i ≤ n, i++, X[[i]] = (B[[i]] - Sum[A[[i, j]] × Xk[[j]], {j, 1, n}]) / A[[i, i]] + Xk[[i]];
    ];
    Print[Subscript["X", k + 1], " = ", MatrixForm[X]
    ];
    Xk = X;
    k++;
   ];
   Print["Number of iterations performed = ", max];
   Return[X];
  ];

A = {{2, -1, 0}, {-1, 2, -1}, {0, -1, 2}};

B = {{7}, {1}, {1}};

X0 = {{0}, {0}, {0}};

GaussJacobi[A, B, X0, 12]
```

$$X_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$X_1 = \begin{pmatrix} 3.5 \\ 0.5 \\ 0.5 \end{pmatrix}$$

$$X_2 = \begin{pmatrix} 3.75 \\ 2.5 \\ 0.75 \end{pmatrix}$$

$$X_3 = \begin{pmatrix} 4.75 \\ 2.75 \\ 1.75 \end{pmatrix}$$

$$X_4 = \begin{pmatrix} 4.875 \\ 3.75 \\ 1.875 \end{pmatrix}$$

$$X_5 = \begin{pmatrix} 5.375 \\ 3.875 \\ 2.375 \end{pmatrix}$$

$$X_6 = \begin{pmatrix} 5.4375 \\ 4.375 \\ 2.4375 \end{pmatrix}$$

$$X_7 = \begin{pmatrix} 5.6875 \\ 4.4375 \\ 2.6875 \end{pmatrix}$$

$$X_8 = \begin{pmatrix} 5.71875 \\ 4.6875 \\ 2.71875 \end{pmatrix}$$

$$X_9 = \begin{pmatrix} 5.84375 \\ 4.71875 \\ 2.84375 \end{pmatrix}$$

$$X_{10} = \begin{pmatrix} 5.85938 \\ 4.84375 \\ 2.85938 \end{pmatrix}$$

$$X_{11} = \begin{pmatrix} 5.92188 \\ 4.85938 \\ 2.92188 \end{pmatrix}$$

$$X_{12} = \begin{pmatrix} 5.92969 \\ 4.92188 \\ 2.92969 \end{pmatrix}$$

Number of iterations performed = 12

Out[•]=

{{5.92969}, {4.92188}, {2.92969}}

Question 9. Write a program to solve a system of linear equations using the Gauss-Seidel iterative method

$$2 x1 - x2 = 7$$

$$-x1 + 2 x2 - x3 = 1$$

$$-x2 + 2 x3 = 1$$

```
ClearAll;

GaussSeidel[A_, B_, X0_, max_] := Module[{n = Length[X0], X = X0, Xold, i, k},
   Print[Subscript["X", 0], " = ", MatrixForm[X]];
   For[k = 1, k ≤ max, k++, Xold = X;
    For[i = 1, i ≤ n, i++,
     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[Subscript["X", k], " = ", MatrixForm[X]];
   ];
   Print["Number of iterations performed = ", max];
   Return[X];
  ];

A = {{2, -1, 0}, {-1, 2, -1}, {0, -1, 2}};
B = {7, 1, 1};
X0 = {0, 0, 0};

GaussSeidel[A, B, X0, 10]
```

$$X_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$X_1 = \begin{pmatrix} 3.5 \\ 0.5 \\ 0.5 \end{pmatrix}$$

$$X_2 = \begin{pmatrix} 3.75 \\ 2.5 \\ 0.75 \end{pmatrix}$$

$$X_3 = \begin{pmatrix} 4.75 \\ 2.75 \\ 1.75 \end{pmatrix}$$

$$X_4 = \begin{pmatrix} 4.875 \\ 3.75 \\ 1.875 \end{pmatrix}$$

$$X_5 = \begin{pmatrix} 5.375 \\ 3.875 \\ 2.375 \end{pmatrix}$$

$$X_6 = \begin{pmatrix} 5.4375 \\ 4.375 \\ 2.4375 \end{pmatrix}$$

$$X_7 = \begin{pmatrix} 5.6875 \\ 4.4375 \\ 2.6875 \end{pmatrix}$$

$$X_8 = \begin{pmatrix} 5.71875 \\ 4.6875 \\ 2.71875 \end{pmatrix}$$

$$X_9 = \begin{pmatrix} 5.84375 \\ 4.71875 \\ 2.84375 \end{pmatrix}$$

$$X_{10} = \begin{pmatrix} 5.85938 \\ 4.84375 \\ 2.85938 \end{pmatrix}$$

$$X_{11} = \begin{pmatrix} 5.92188 \\ 4.85938 \\ 2.92188 \end{pmatrix}$$

$$X_{12} = \begin{pmatrix} 5.92969 \\ 4.92188 \\ 2.92969 \end{pmatrix}$$

Number of iterations performed = 12

Out[•]=

{{5.92969}, {4.92188}, {2.92969}}

Question 10. Find the unique polynomial of degree 2 or less such that
f (0) = 1, f (2) = 5, f (3) = 10
using Lagrange interpolation.

```
ClearAll;

Lagrange[x0_, f0_] := Module[{xi = x0, fi = f0, n, m, polynomial}, n = Length[xi];
   m = Length[fi];
   If[n ≠ m, Print["List of points and function values are not of same size"];
    Return[];];
   For[i = 1, i ≤ n, i++,
    L[i, x_] = (Product[(x - xi[[j]]) / (xi[[i]] - xi[[j]]), {j, 1, i - 1}]) *
       (Product[(x - xi[[j]]) / (xi[[i]] - xi[[j]]), {j, i + 1, n}]);
   ];
   polynomial[x_] = Sum[L[k, x] * fi[[k]], {k, 1, n}];
   Return[polynomial[x]];
  ];

nodes = {0, 2, 3};
values = {1, 5, 10};

lagrangepolynomial[x_] = Lagrange[nodes, values];
lagrangepolynomial[x_] = Simplify[lagrangepolynomial[x]
  ];

Print["Lagrange Polynomial = ", lagrangepolynomial[x]];
```

Lagrange Polynomial = $1 + x^2$

Question 11. Using Newton’ s interpolation formula,
find the polynomial of degree 2 or less that satisfies f (1) = 2, f (2) = 5, f (3) = 7.

```
ClearAll[NewtonInterp];

NewtonInterp[x0_List, f0_List] := Module[{n = Length[x0], dd, poly},
    (*Create a divided difference table*)dd = Table[0, {n}, {n}];
    Do[dd[[i, 1]] = f0[[i]], {i, 1, n}];
    Do[dd[[i, j]] = (dd[[i + 1, j - 1]] - dd[[i, j - 1]]) / (x0[[i + j - 1]] - x0[[i]]),
     {j, 2, n}, {i, 1, n - j + 1}];
    poly[x_] := Sum[dd[[1, k]] * Product[(x - x0[[m]]), {m, 1, k - 1}], {k, 1, n}];
    Simplify[poly[x]]
   ];
nodes = {1, 2, 3};
values = {2, 5, 7};

P[x_] = NewtonInterp[nodes, values];
Print["Newton Interpolating Polynomial = ", P[x]];
```

Newton Interpolating Polynomial = $\frac{1}{2}\left(-4+9x-x^2\right)$

Question 12. Evaluate the definite integral

$\int_0^2 (x2 + 1)\,dx$ using the Trapezoidal Rule with 4 subintervals

```
ClearAll[TrapezoidalRule];

TrapezoidalRule[f_, a_, b_, n_] :=
  Module[{h, sum = 0, x, i},
   h = (b - a) / n;
   For[i = 1, i ≤ n - 1, i++, x = a + i h;
    sum = sum + f[x];
   ];
   Return[(h / 2) (f[a] + 2 sum + f[b])];
  ];
f[x_] := x^2 + 1;

TrapezoidalRule[f, 0, 2, 4]
```

Out[69]=

$\frac{19}{4}$

Question 13. Evaluate the definite integral

$\int_0^4 (x3 + \sin\, x)\, dx$ **usig Simpsons rule with 8 subinervals**

```
ClearAll[SimpsonsRule];

SimpsonsRule[f_, a_, b_, n_] := Module[{h, sum1 = 0, sum2 = 0, x, i},
   If[OddQ[n], Print["n must be even for Simpson's Rule"];
    Return[];
   ];
   h = (b - a) / n;
   For[i = 1, i ≤ n - 1, i += 2, x = a + i h;
    sum1 = sum1 + f[x];
   ];
   For[i = 2, i ≤ n - 2, i += 2, x = a + i h;
    sum2 = sum2 + f[x];
   ];
   Return[(h / 3) (f[a] + 4 sum1 + 2 sum2 + f[b])];
  ];

f[x_] := x^3 + Sin[x];

SimpsonsRule[f, 0, 4, 8]
```

Out[73]=

$$\frac{1}{6}\left(64 + 2\,(36 + \mathrm{Sin}[1] + \mathrm{Sin}[2] + \mathrm{Sin}[3]) + 4\left(62 + \mathrm{Sin}\left[\frac{1}{2}\right] + \mathrm{Sin}\left[\frac{3}{2}\right] + \mathrm{Sin}\left[\frac{5}{2}\right] + \mathrm{Sin}\left[\frac{7}{2}\right]\right) + \mathrm{Sin}[4]\right)$$

Question 14. Use Euler's method to approximate the solution of the initial value problem

dx / dy = x + y , y (0) = 1 on the interval 0 ≤ x ≤ 1 using a step size h = 0.1.

```
ClearAll[EulerMethod];

EulerMethod[f_, x0_, y0_, h_, n_] := Module[
  {x = x0, y = y0, i}, Print["Iteration 0:  x = ", x, "    y = ", y];
  For[i = 1, i ≤ n, i++, y = y + h * f[x, y];
   x = x + h;
   Print["Iteration ", i, ":  x = ", x, "    y = ", y];
  ];
  Return[y];
 ]
f[x_, y_] := x + y;

EulerMethod[f, 0, 1, 0.1, 10]
```

```
Iteration 0:  x = 0   y = 1
Iteration 1:  x = 0.1   y = 1.1
Iteration 2:  x = 0.2   y = 1.22
Iteration 3:  x = 0.3   y = 1.362
Iteration 4:  x = 0.4   y = 1.5282
Iteration 5:  x = 0.5   y = 1.72102
Iteration 6:  x = 0.6   y = 1.94312
Iteration 7:  x = 0.7   y = 2.19743
Iteration 8:  x = 0.8   y = 2.48718
Iteration 9:  x = 0.9   y = 2.8159
Iteration 10:  x = 1.   y = 3.18748
```

Out[79]=

3.18748

Question 15. Use the fourth – order Runge-Kutta (RK4) method to approximate the solution of the initial value problem

dx / dy = x2 – y, y (1) = 2 on the interval 1 ≤ x ≤ 2 using a step size h = 0.1.

```
ClearAll[RK4];

RK4[f_, x0_, y0_, h_, n_] := Module[{x = x0, y = y0, k1, k2, k3, k4, i},
   Print["Iteration 0:  x = ", x, "   y = ", y];
   For[i = 1, i ≤ n, i++,
    k1 = h * f[x, y];
    k2 = h * f[x + h / 2, y + k1 / 2];
    k3 = h * f[x + h / 2, y + k2 / 2];
    k4 = h * f[x + h, y + k3];
    y = y + (k1 + 2 k2 + 2 k3 + k4) / 6;
    x = x + h;
    Print["Iteration ", i, ":  x = ", x, "   y = ", y];];
   Return[y];
  ];
f[x_, y_] := x^2 - y;

RK4[f, 1, 2, 0.1, 10]
```

```
Iteration 0:  x = 1   y = 2
Iteration 1:  x = 1.1   y = 1.91484
Iteration 2:  x = 1.2   y = 1.85873
Iteration 3:  x = 1.3   y = 1.83082
Iteration 4:  x = 1.4   y = 1.83032
Iteration 5:  x = 1.5   y = 1.85653
Iteration 6:  x = 1.6   y = 1.90881
Iteration 7:  x = 1.7   y = 1.98659
Iteration 8:  x = 1.8   y = 2.08933
Iteration 9:  x = 1.9   y = 2.21657
Iteration 10:  x = 2.   y = 2.36788
Out[83]=
2.36788
```