TRIBHUVAN UNIVERSITY

Institution of Science and Technology

Bachelor Level/Second Year/Third Semester/Science

Computer Science and Information Technology [CSc. 207]

(Numerical Method)

(New Course)

Full Marks: 60

Pass Marks: 24

Time: 3 hrs.

Candidates are required to give their answers in their own words as far as practicable.

The figures in the margin indicate full marks.

TU QUESTIONS-ANSWERS 2075

Group 'A'

Attempt any three questions:

(3×10 = 30)

1. **What is non-linear equation? Derive the required expression to calculate the root of non-linear.**

   $x^2 + \cos(X) - e^x - 2 = 0$

Sol^n: Those equations in which the variables are either of degree greater than one or less than one, but never one. In such systems output is not directly proportional to its input. Also such equations do not form or straight line in the graph.

As we know, the Taylor series is given by,

$$f(x) = f(x_0) + f'(x_0) \cdot h + \frac{f''(x_0)}{2!} h^2 + \frac{f'''(x_0)}{3!} h^3 + \ldots \qquad \ldots (i)$$

Where x_0 is the initial point and h is the interval such that $x = x_0 + h$.

neglecting higher order derivatives, the equation (i) becomes,

$$f(x) = f(x_0) + f'(x_0) \cdot h \qquad \ldots (ii)$$

And $\because f(x) = 0$ the above equation (ii) becomes

$$0 = f(x_0) + f'(x_0) \cdot h$$

Here, $h = x_2 - x_1$ $\because$ h is the interval between the points.

So, $0 = f(x_0) + f'(x_0) \cdot (x_1 - x_0)$

or, $$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

or, $$x_2 = x_1 - \frac{f(x_1)}{\frac{f(x_1) - f(x_0)}{x_1 - x_0}}$$

or, $$x_2 = x_1 = \frac{f(x_1) \cdot (x_1 - x_0)}{f(x_1) - f(x_0)}$$

or, $$\boxed{x_2 = x_1 - f(x_1) \cdot \frac{x_1 - x_0}{f(x_1) - f(x_0)}}$$

Generalizing,

$$x_{i+1} = x_i - f(x_i) \cdot \frac{(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}$$

Which is secant iterative expression to calculate the root of equation.

Let f(x): $x^2 + \cos(x) - e^{-x} - 2 = 0$

Using Trial and Error method for initial guesses.

x	-1	0	1	2	3
f(x)	-3.111	-2	-0.827	1.4485	5.9602

$\because f(1) = -0.827$ and $f(2) = 1.4485$ root lies between 1 and 2. Since scant method is non bucketing technique. The two initial assumptions for the iterative formula are: 2 and 3

$x_0 = 2$

$x_1 = 3$

Now using iterative formula of secant method;

1st iteration;

$x_0 = 2, x_1 = 3$

$$x_2 = x_1 - f(x_i) - \frac{f(x_1)(x_1 - x_0)}{f(x_1) - f(x_0)}$$

$$= 3 - \frac{5.9602(3-2)}{(5.9602 - 1.4485)}$$

$$= 1.678946$$

2nd iteration;

$$x_3 = x_2 - \frac{f(x_2)(x_2 - x_1)}{f(x_2) - f(x_1)}$$

$f(x_2) = 0.524350$

$f(x_1) = 59602$

$$x_3 = 1.678946 - \frac{0.524350(1.678946 - 3)}{0.524350 - 5.9602}$$

$$= 1.551515$$

3rd iteration;

$$x_4 = x_3 - f(x_3)\frac{(x_3 - x_2)}{f(x_3) - f(x_2)}$$

$f(x_3) = 0.214553$

$f(x_2) = 0.524350$

$$x_4 = -1.551515 - \frac{0.214553(1.551515 - 1.678946)}{0.214553 - 0.524350}$$

$x_4 = 1.46321$

4th iteration;

$$x_5 = x_4 - \frac{f(x_4)(x_4 - x_3)}{f(x_4) - f(x_3)}$$

$f(x_4) = 0.016982$

$f(x_3) = 0.214553$

$$x_5 = 1.469261 = \frac{0.016982(1.469261 - 1.551515)}{0.016982 - 0.214553}$$

$$= 1.462191.$$

$\because x_4$ and x_5 are same till 2 decimal place the root of the given equation correction up to decimal place is 1.46.

2. **What is matrix factorization? Factorize the given matrix A into LU using Dolittle algorithm and also solve Ax = b for given b using L and U matrices.**

$$A = \begin{bmatrix} 2 & 4 & -4 & 0 \\ 1 & 5 & -5 & -3 \\ 2 & 3 & 1 & 3 \\ 1 & 4 & -2 & 2 \end{bmatrix} \text{ and } b = \begin{bmatrix} 12 \\ 18 \\ 8 \\ 8 \end{bmatrix}$$

Solⁿ: As we know that, from LU Dolittle algorithm the matrix is decomposed into lower triangular and upper triangular matrices;

So, $[A] = [L]\,[U]$

$$\text{The coefficient matrix is } [A] = \begin{bmatrix} 2 & 4 & -4 & 0 \\ 1 & 5 & -5 & -3 \\ 2 & 3 & 1 & 3 \\ 1 & 4 & -2 & 2 \end{bmatrix}$$

Decomposing the given matrix as;

$$[A] = [L]\,[U] = \begin{bmatrix} 2 & 4 & -4 & 0 \\ 1 & 5 & -5 & -3 \\ 2 & 3 & 1 & 3 \\ 1 & 4 & -2 & 2 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

$$\begin{bmatrix} 2 & 4 & -4 & 0 \\ 1 & 5 & -5 & -3 \\ 2 & 3 & 1 & 3 \\ 1 & 4 & -2 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \ell_{21} & 1 & 0 & 0 \\ \ell_{31} & \ell_{32} & 1 & 0 \\ \ell_{41} & \ell_{42} & \ell_{43} & 1 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Now,

$1 \times u_{11} + 0 + 0 + 0 = 2,\qquad u_{11} = 2$

$1 \times u_{12} + 0 + 0 + 0 = 4,\qquad u_{11} = 4$

$1 \times u_{13} + 0 + 0 + 0 = -4,\qquad u_{11} = -4$

$1 \times u_{14} + 0 + 0 + 0 = 0,\qquad u_{11} = 0$

$\ell_{21} . u_{11} + 1 \times 0 + 0 + 0 = 1,\qquad \ell_{21} . 2 = 1,\qquad \ell_{21} = \frac{1}{2}$

$\ell_{21} . u_{12} + u_{22} + 0 + 0 = 5,\qquad \frac{1}{2} . 4 + u_{22} = 5\qquad u_{22} = 3$

$\ell_{21} . u_{14} + u_{24} + 0 + 0 = 3,\qquad \frac{1}{2} \times 0 + u_{24} + 0 + 0 = 3\qquad u_{24} = -3$

$\ell_{31} . u_{11} + 0 + 0 + 0 = 2,\qquad \ell_{31} \times 2 = 2,\qquad \ell_{31} = 1$

$\ell_{31} . u_{12} + \ell_{32} . u_{12} + 0 + 0 = 3,\quad 1 \times 4 + \ell_{32} \times 3 = 3,\quad \ell_{32} = -\frac{1}{3}$

$\ell_{31} . u_{12} + \ell_{32} . u_{12} + 0 + 0 = 3,\quad 1 \times 4 + \ell_{32} \times 3 = 3,\quad \ell_{32} = -\frac{1}{3}$

$\ell_{31} . u_{13} + \ell_{32} . u_{23} + u_{23} = 1, \quad 1 \times (-4) + \left(\frac{-1}{3}\right) \times (-3) + u_{33} = 1, \quad u_{33} = 4$

$\ell_{31} . u_{14} + \ell_{32} . u_{24} + u_{34} + 0 = 3, \quad 1 \times 0 + \left(\frac{-1}{3}\right) \times (-3) + u_{34} = 3, \quad u_{34} = 2$

$\ell_{41} . u_{11} = 1 \qquad u_{41} . 2 = 1, \qquad \ell_{41} = \frac{1}{2}$

$\ell_{41} . u_{12} + \ell_{42} . u_{22} = 4, \qquad \frac{1}{2} \times 4 + \ell_{42} \times 3 = 4, \qquad \ell_{42} = \frac{2}{3}$

$\ell_{41} . u_{13} + \ell_{42} . u_{23} + \ell_{43} . u_{33} = -2, \frac{1}{2} \times (-4) + \frac{2}{3} \times (-3) + \ell_{43} = -2, \ell_{43} = \frac{1}{2}$

$\ell_{41} . u_{124} + \ell_{42} . u_{24} + \ell_{43} . u_{34} + u_{44} = 2$

$\frac{1}{2} \times 0 + \frac{2}{3} \times (-3) + 2 \times 2 + u44 = 2$

$u_{44} = 3$

$\therefore$ The decomposed matrices are;

$$A = \begin{bmatrix} 2 & 4 & -4 & 0 \\ 1 & 5 & -5 & -3 \\ 2 & 3 & 1 & 3 \\ 1 & 4 & -2 & 2 \end{bmatrix} L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \\ 1 & -1/3 & 1 & 0 \\ 1/2 & 2/3 & 1/2 & 0 \end{bmatrix} u = \begin{bmatrix} 2 & 4 & -4 & 0 \\ 0 & 3 & -3 & -3 \\ 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

$$\text{Let } x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

To solve for X the matrix would be of the form,

[A] [X] = [B]

To find the value of [X], let we suppose for z matrix

$\therefore$ [L] [Z] = [C]

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \\ 1 & -1/3 & 1 & 0 \\ 1/2 & 2/3 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 21 \\ 22 \\ 23 \\ 24 \end{bmatrix} = \begin{bmatrix} 12 \\ 18 \\ 8 \\ 8 \end{bmatrix}$$

On solving, we get,

$z_1 = 12$

$\frac{1}{2} z_1 + z_2 = 18$

or, $\frac{1}{2} \times 12 + Z_2 = 18$

$\therefore \quad Z_2 = 12$

$Z_1 + \left(\frac{-1}{3}\right) Z_2 + Z_3 = 8$

or, $12 + \left(\frac{-1}{3}\right) \times 12 + 2_3 = 8$

$\therefore \quad Z_3 = 0$

$\frac{1}{2} Z_1 + \frac{2}{3} Z_2 + \frac{1}{2} Z_3 + 1Z_4 = 8$

or, $\frac{1}{2} \times 12 + \frac{2}{3} \times 12 + \frac{1}{2} \times 0 + 1Z_4 = 8$

or, $6 + 8 + 0 + 1 \times Z_4 = 8$

or, $14 + 0 + 1 \times Z_4 = 8$

or, $1Z_4 = -6$

$\therefore \quad Z_4 = -6$

$$\therefore \quad Z = \begin{bmatrix} 12 \\ 12 \\ 0 \\ -6 \end{bmatrix}$$

Now, [U] [X] = [Z]

$$\begin{bmatrix} 2 & 4 & -4 & 0 \\ 1 & 5 & -5 & -3 \\ 2 & 3 & 1 & 3 \\ 1 & 4 & -2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 12 \\ 0 \\ -6 \end{bmatrix}$$

Now, solving on these value, we get

$2x_1 + 4x_2 - 4x_3 = 12$ ...(i)

$3x_2 + 3x_3 - 3x_4 = 12$ ...(ii)

$4x_3 + 2x_4 - 0$ ...(iii)

$3x_4 = -6$

$\therefore \quad x_4 = -2$

Putting value of x_4 in equation (iii) we get,

$4x_3 + 2 \times -2 = 0$

$\therefore \quad x_3 = 1$

Putting value of x_3 and x_4 in equation (ii) we get

$3x_2 - 3 \times 1 \quad - 3 \times -2 = 12$

or, $3x_2 - 3 + 6 = 12$

or, $3x_2 + 3 = 12$

or, $3x_2 = 9$

$\therefore \quad x_2 = 3$

Putting value of x_2, x_3 and x_4 in equation (i) we get,

$2x_1 + 4 \times 3 - 4 \times 1 = 12$

or, $2x_1 + 12 - 4 = 12$

or, $2x_1 = 4$

$\therefore \quad x_1 = 2$

$$\therefore \quad X = \begin{bmatrix} 2 \\ 3 \\ 1 \\ -2 \end{bmatrix}$$

$$\text{Let } X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

To solve for X the matrix would be of the form,

[A] [X] = [B]

To find the value of [X], let us suppose for Z matrix

∴ [L] [Z] = [C]

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 \\ 1 & -1/3 & 1 & 0 \\ 1/2 & 2/3 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 18 \\ 8 \\ 8 \end{bmatrix}$$

On solving, we get,

$z_1 = 12$

$\frac{1}{2} z_1 + Z_2 = 18$

or, $\frac{1}{2} \times 12 + Z_2 = 18$

∴ $Z_2 = 12$

$Z_1 + \left(\frac{-1}{3}\right) Z_2 + Z_3 = 8$

or, $12 + \left(\frac{-1}{3}\right) \times 12 + Z_3 = 8$

or, $12 - 4 + Z_3 = 8$

∴ $Z_3 = 0$

$\frac{1}{2} Z_1 + \frac{2}{3} Z_2 + \frac{1}{2} Z_3 + 1Z_4 = 8$

or, $\frac{1}{2} \times 12 + \frac{2}{3} \times 12 + \frac{1}{2} \times 0 + 1Z_4 = 8$

or, $6 + 8 + 0 + 1 \times Z_4 = 8$

or, $14 + 0 + 1 \times Z_4 = 8$

or, $1Z_4 = -6$

∴ $Z_4 = 6$

$$\therefore \begin{bmatrix} 12 \\ 12 \\ 0 \\ -6 \end{bmatrix}$$

3. **What is initial value problem and boundary value problem? Write an algorithm and program to solve the boundary value problem using shooting method.**

Sol^n: The solution of an ordinary differential equation requires auxiliary conditions. For an n^{th} order equation, n conditions are required. If all the conditions are specified at the starting point, then the problem is called initial value problem.

For eg: Solving the equation;

$y' = x^2 + y^2$, given y (0) = 1.

If the conditions are known at different values of the independent variable, usually at the extreme points or boundaries of as system, then the problem is known as boundary-value problem. For eg:

Sowing the equation,y' = y, given y (0) + y (1) = 2

Algorithm for shooting method:

1. Start
2. Read Boundary say X_a, X_b, Y_a and Y_b
3. Read the point at which solution is needed, say X_p.
4. Rea accuracy limit, say E.
5. Convert higher order differential equation to system of differential equations.
6. Read value of h.
7. Approximate first approximation below:

   Set $X = X_a$

   $Y = Y_a$

   $g_1 = (Y_b - Y_a)/ (X_b - X_a)$

   Calculate $Y(X_b)$ by using Euler's method.

   Set $V_1 = Y$.

   If $(Y < Y_b)$

   $g_2 = 2g$,

   else

   $g_2 = g_1/2$

   Calculate y (X_b) by using Eulr's method set $V_2 = Y$
8. Compute new value of $y(X_b)$ as below:

   compute $g_3 = g_2 - \frac{V_2 - y_b}{v_2 - r_1}(g_2 - g_1)$

   Find y (X_b) by Euler's method,

   Computer error

   if (error < E)

   Display solution

   Goto step 9

   Else

   Set $v_1 = v_2$ $\quad V_2 = y (X_b)$

   Set $g_1 = g_2$ $\quad g_2 = y_3$

   Goto Step 8.
9. Terminate

Program to slove boundary value problem using shooting method.

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define f1(x, y, z) (z)
#define f2(x, y, z) 6 * (z)
int main ( )
{
float xa, xb, xa, Yb, x0, x0, X, Y, Z, xP, h, sol, ny, nz,
```

```
error, E, g [3], v[3], gs;
int i }
Print ("Enter BoundaryConditions/n");
Scanf ("%f%f%f%f", & Xa, &ya, &Xb, & Xb);
Print ("Enter the value of X for required value \n");
Scan f("%f", &h);
X = Xa;
Y = Ya;
g[1] = Z = (Yb - Ya)/Xa;
Print f ("g = % f\n", g[1]);
while (X < X ) b
{
  ny = y + (f1 (x, y, z)) * h;
  nz = z + (f2 (x, y, z)) * h;
  X = X + h;
  Y = nY;
  Z = nZ;
  if (X = = XP)
Sol = y;
{
v[1] = y;
If (Y < Yb)
  g[2] = z = 2 * g [1];
  else
  g[2] = z = 1/2 * g [1];
  Printf("g = %f\n", g[2]);
  white (X < Xb)
    {
    nY = y + (f1 (x, y, z)) * h;
    nZ = Z + (fz (x, y, z)) * h;
    X = X + h;
    Y = ny;
    n = ny;
    if (X = = XP)
    Sol = y ;
{
While (1)
    X = Xa;
    Y = Ya;
    gs = z = g[2] - (v[2] - yb)/(v[2] - v[1]) * (g[2] - g[1]);
    While (X < Xb)
    {
    ny = y + (f1 (x, y , z))* h;
    nz = z + (f2 (x, y , z))* h;
        X = X + n;
        Y = ny;
        Z = ny;
        if (X = = XP)
```

```
{
error = tabs (Y - Yb)/Y;
        v[1] = v[2]
        v[2] = y;
        g[1] = g [2];
        g[2] = gS;
        if (error < E)
        Print f("Y(%f) = %f", XP, Sol);
                break;
        {
        {
        getch ();
        return 0;
}
```

Group 'B'

Attempt any Eight questions: (5×8 = 40)

4. **Calculate a real negative root of following equation using Newton's method for polynomial.**

$x^4 + 2x^3 + 3x^2 + 4x = 5$

Sol^n: To find a root of the equation by using Newton's method we use the iterative formula as,

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Let $f(x) = x^4 + 2x^3 + 3x^2 + 4x - 5 = 0$

Now, $f'(x) = 4x^3 + 6x^2 + 6x + 4$

Using Trial and Error method to find initial assumption.

x	−4	−3	−2	−1
f(x)	155	37	−1	−7

Negative root lies between −3 and −2.

let $x_0 = -2$

$f(x_0) = -1$

$f'(x_0) = 4 \times (-2)^2 + 6 \times (-2)^2 + 6 \times (-2) + 4 = -16$

1st iteration:

$x_0 \quad = -2, f(x_0) = -1, f'(x_0) = -16$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

$$= -2 - \frac{(-1)}{(-16)} = -2.0625$$

2nd iteration:

$x_1 \quad = -2.0625, f(x_1) = 0.06007, f'(x_1) = 19.2$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

$$= -2.0625 - \frac{0.06007}{19.9521} = -2.0655$$

3rd iteration:

$x_2 \quad = -2.0655, f(x_2) = 0.11405, f'(x_2) = -18.043$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = -2.0591$$

4th iteration:

$x_3 = -2.0591$, $f(x_3) = 0.00075$, $f'(x_3) = -17.83$

$$x_4 = x_3 - \frac{f(x_3)}{f'(x_3)}$$

$= -2.0591$

$\therefore$ The value of x_3 and x_4 are same upto four decimal place; therefore, the real negative root is -2.0591.

5. **What is least squares approximation of fitting a function? How does it differ with polynomial interpolations? Explain with suitable example.**

Soln: Least squares approximation is a method of parameter estimation to estimate value of parameters of the model, based on the observed pairs of values and minimizing certain objective function

Lease square regression is a form of predictive modeling technique which investigates the relationship between a dependent and independent variables.

The objective function is error function which has to be minimized, since if all the residuals e_i error at data points (x_i, y_i) are zero, one may have found an equation in which all the points lie on a model. The error function is given as,

$\Sigma e_i = \Sigma(y_i - f(x_i))^2$, $i = 1, 2, \ldots n.$

the main aim would-be to minimize the given error function.

In polynomial interpolation we are given with some data points are we are supposed to find a curve which fits the input/output relationship perfectly. In case of interpolation, we don't have to worry about variance of the fitted curve. That means given the set of n data points (x_i, y_i) we look for the function f(x) that satisfies the relation $y_i = f(x_i)$ for all given data points.

When we do least square regression, we look for a function that minimizes some cost, usually sum of squares of errors. We don't require the function to have the exact values at given point; we just want a good approximation. general, we search for the function $y = f(x)$ that might not satisfy the relation $y_i = f(x_i)$ foray given data points, but the cost function $\Sigma(f(x_i) - y_i)^2$ will be the smallest possible of all the functions of given form.

Some Different method of polynomial interpolations are:

1. Lagrange interpolation
2. Newton's Divided interpolation are used for uneven space points interpolation.
3. Newton's forward interpolation
4. Newton's backward interpolation are used for even space points interpolation. Some other polynomial interpolation are: Spline interpolation, Guessian interpolation, Everetts interpolation.
5. Some different method for regression analysis which uses least squares approximate are:
   - Liner regression.
   - Exponential method for non-linear regression.
   - Polynomial model for non-linear regression.

6. **Find the lowest degree polynomial, which passes through the following points:**

fX	-2	-1	1	2	3	4
f(x)	-19	0	2	-3	-4	5

Using this polynomial estimate f(x) at x = 0

Sol^n: To find the polynomial we can use interpolation techniques

∴ The intervals of the given data points are not even. So we can use either Lagrange's interpolation or Newton's divided difference. But divided difference would not limit the order of polynomial. So we use Lagrange interpolation of 1st order as it has been asked to find the lowest degree polynomial. So, the lagrange interpolation of 1st order is given as;

$$P_1(x) = \frac{f_1(x - x_0)}{(x_1 - x_0)} + \frac{f_0(x - x_1)}{(x_0 - x_1)}$$

Here,

$x_0 = -1, \quad x_1 = 1$

$f_0 = 0, \quad f_1 = 2$

$$f_1(x) = \frac{2.(x-(-1))}{2} + \frac{0 \times (x-1)}{(-1-1)}$$

$$= \frac{2(x+1)}{2} + 0$$

$\boxed{P_1(x) = x + 1}$

The required polynomial of1st order is x + 1

$P(x) = x + 1$

Finding the value of f(x) at x = 0.

$P(0) = 0 + 1 = 1.$

∴ The value of f(x) at x = 0 is 1.

7. **Fit function of type y = a + bx for the following points using least square method.**

X	-1	1.2	2	2.7	3.6	4
f(x)	1	20	27	33	41	45

Sol^n:

x_1	y_1	x_i^2	x_iy_i
-1	1	1	-1
1.2	20	1.44	24
2	27	4	54
2.7	33	7.29	89.1
3.6	41	12.96	147.6
4	45	16	180
$\Sigma x_i = 12.5$	$\Sigma x_i = 167$	$\Sigma x_i^2 = 42.69$	$\Sigma x_iy_i = 493.7$

We know the form y = a + bx have the value of

$$b = \frac{n\Sigma x_iy_i - \Sigma x_i\Sigma y_i}{n\Sigma x_1^2 - (\Sigma x_i)^2}$$

$$a = \frac{\Sigma y_i}{n} - b\frac{\Sigma x_i}{n}$$

$$b = \frac{6 \times 493.7 - 12.5 \times 167}{6 \times 42.69 - (12.5)^2} = \frac{874.7}{99.89} = 8.75$$

$$a = \frac{167}{6} - 8.75 \times \frac{12.5}{6} = 9.6041$$

The equation y = a + bx is represented specifically as;
y = 9.6041 + 8.75 x.

8. **Calculate the integral value of the function x = 1.8 to x = 3.4 using Simpon's 1/3 rule.**

X	1.8	2.0	2.2	2.4	2.6	2.8	3.0	3.2	3.4
f(x)	0.003	0.778	1.632	2.566	3.579	4.672	4.672	4.097	8.429

Solⁿ: $$I = \int_{1.8}^{3.4} f(x)$$

And the value for I by using Simpson's $\frac{1}{3}$ rule can reapplied as;

$$I = \int_{1.8}^{3.4} f(x) = \frac{h}{3}\left[f(x_0) + 4 \sum_{\substack{i=1 \\ i = odd}}^{k-1} f(x_i) + 2 \sum_{\substack{i=1 \\ i = odd}}^{k-1} f(x_i) + f(x_n) \right]$$

Now, $h = \frac{b-a}{2}$ and b = 3.4 and a = 1.8

$$h = \frac{3.4 - 1.8}{2} = 0.8$$

$f(x_0) = 0.003$ $f(x_4) = 3.579$ $f(x_8) = 8.429$
$f(x_1) = 0.778$ $f(x_5) = 4.672$
$f(x_2) = 1.632$ $f(x_6) = 4.672$
$f(x_3) = 2.566$ $f(x_7) = 7.079$

$$I = \int_{1.8}^{3.4} f(x) = \frac{h}{3}[f(x_0) + 4\{f(x_1) + f(x_3) + f(x_5) + f(x_7)\} + 2\{f(x_2) + f(x_4) + f(x_6)\} + f(x_8)]$$

$$= \frac{0.8}{3}[0.003 + 4(0.778 + 2.566 + 4.672 + 7.097) + 2(1.632 + 3.579 + 4.672) + 8.429]$$

$$= 0.2667[0.003 + (15.113) + 2(9.883) + 8.429] = 0.2667[88.65]$$

$$I = 23.642955$$

The integral value from X = 1.8 to X = 3.4 for the provided f(x) is 23.642955.

9. **Evaluation the following integration using Romberg integration.**

$$\int_0^i \frac{\text{Sin } x}{x} dx$$

Solⁿ: The Romberg integrations expanded from with the dependencies is shown as;

$$\begin{tikzcd}
T_{0,0} \arrow[d] \arrow[dr] & & \\
T_{1,0} \arrow[r] \arrow[d] \arrow[dr] & T_{1,1} \arrow[dr] & \\
T_{2,0} \arrow[r] & T_{2,1} \arrow[r] & T_{2,2}
\end{tikzcd}$$

Using Trapzoidal rule for calculating T(0, 0)

$$T(0, 0) = \frac{h}{2}(f(x_0) + f(x_1))$$

$$= \frac{1-0}{2}$$

10. Solve the following set of equations using Gauss Seidel method.

$x + 2y + 3z = 4$
$6x - 4y + 5z = 10$
$5x + 2y + 2z = 25$

Solⁿ: Finding iterative formula for the given set of simultaneous equation.

$$x = \frac{25 - 2y - 2z}{5}$$

$$y = \frac{4 - 3z - x}{2}$$

$$z = \frac{10 + 4y - 6y}{5}$$

Iteration

Iteration	x	y	x
0	0	0	0
1	5	-0.5	-4.4
2	6.96	5.12	-2.256
3	3.854	3.457	0.1408
4	3.5608	0.008	-2.2665
5	6.0628	2.3683	-3.3807
6	5.404	4.36905	-0.9895
7	3.6481	1.6602	-1.095
8	4.7738	1.255	-2.724
9	5.5876	3.292	-2.071
10	4.511	2.851	-1.1324
11	4.312	1.5426	-1.94
12	5.158	2.331	-2.324
13	4.9972	2.9874	-1.60672
14	4.44	2.1900	-1.576
15	4.75	1.989	-2.108
16	5.0476	2.6382	-1.9465
17	4.72	2.5597	-1.6162

Similarly, the solution for the simultaneous equations are:

$$\left.\begin{array}{l} x = 4.72 \\ y2.5597 \\ z = -1.6162 \end{array}\right]$$

11. From the following differential equation estimate y(1) using RK 4th order method.

$\frac{dy}{dx} + 2x^2y = 4$ with $y(0) = 1$, [Take $h = 0.5$].

Solⁿ: $\frac{dy}{dx} + 2x^2y = 4$

or, $\frac{dy}{dx} = 4 - 2x^2y$

$f(x, y) = 4 - 2x^2 y, \quad x_0 = 0, y_0 = 1, h = 0.5$

we know that RK 4th order formula is given as;

$$y(x_n + h) = y_n + 1 = y_n + \frac{1}{6} h (m_1 + 2m_2 + 2m_3 + m_4)$$

$$y(x_n + h) = y_n + 1 - y_n + \frac{1}{6} h (m_1 + 2m_2 + 2m_3 + m_4)$$

Where,

$m_1 = f(x_i, y_i)$

$m_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hm_1)$

$m_3 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hm_2)$

$m_4 = f(x_i + h, y_i + hm_3)$

1st iteration:

$x_0 = 0, y_0 = 1, h = 0.5$

$$y(x_0 + h0) = y(0 + 0.5) = y(0.5) = y_0 + \frac{1}{6} . h (m_1 + 2m_2 + 2m_3 + m_4)$$

$m_1 = f(x_0, y_0) = f(0, 1) = 4$

$$m_2 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{h}{2} m_1\right) = f\left(0 + \frac{0.5}{2}, 1 + \frac{0.5}{2} \times 4\right)$$

$$= f(0.25, 2) = 4 - 2 \times 0.25^2 \times 2 = 3.75$$

$$m_3 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{m_2 h}{2}\right) = f\left(0 + 0.25, 1 + \frac{3.75 \times 0.5}{2}\right)$$

$$= f(0.25, 1.9375) = 3.7578$$

$$m_4 = f(x_0 + h, y_0 + m_3 h) = f(0 + 0.5, 1 + 3.7578 \times 0.5)$$

$$= f(0.5, 28789) = 2.5605$$

$$y_1 = 1 + \frac{1}{6} \times 0.5 (4 + 2 \times 3.75 + 2 \times 3.7578 + 2.5605)$$

$y_1 = 2.79800$

2nd iteration:

$x_1 = 0.5, y_1 = 2.798, h = 0.5$

$$y(y_1 + h) = y(0.5 + 0.5) = y(1) = y_2 = y_1 + \frac{h}{6} \cdot (m_1 + 2m_2 + 2m_3 + m_4)$$

$m_1 = f(x_1, y_1) = f(2.798) = 4 - 2 \times 0.5^2 \times 2.798 = 2.601$

$$m_2 = f\left(x_1 + \frac{h}{2}, y_1 + \frac{h}{2} m_1\right) = f\left(0.5 + \frac{0.5}{2}, 2.798 + 0.25 \times 2.601\right)$$

$$= f(0.75, 3.44825) = 0.1207$$

$$m_3 = f\left(x_1 + \frac{h}{2}, y_1 + \frac{h}{2} m_2\right) = f(0.75, 2.798 + 0.25 \times 0.1207)$$

$$= f(0.475, 2.8281) = 0.8183$$

$$m_4 = f(x_1 + h, y_1 + hm_3) = f(1 + 2.798 + 0.5 \times 0.8183)$$

$$= f(1, 3.2071) = -2.4142$$

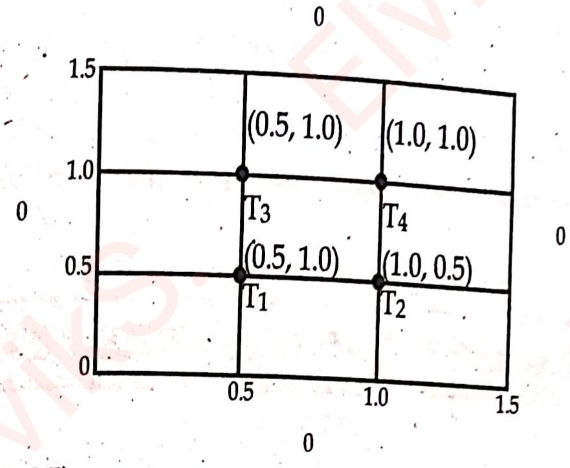
$$y_2 = y^{(1)} = 2.798 + \frac{0.5}{6} [2.601 + 2 \times 0.1207 + 2 \times 0.8183 - 2.4142]$$

$y(1) = 2.9700]$

$\therefore$ The value of given differential equation at $x = 1$ is 2.97.

12. Solve the Poison's equation $\nabla^2 f = 2xy$ over the square domain $0 \le x \le 1.5$, $0 \le y \le 1.5$ with $f = 0$ on the boundary and $h = 0.5$.

Sol^n: The given grid can be represented as;



For T_1 (0.5, 0.5):

$0 + T_3 + T_2 + 0 - 4T_1 = 2 \times 0.5 \times 0.5$

or, $-4T_1 + T_2 + T = 0.5$ ... (i)

For T_2 (1.0, 0.5):

$T_1 + T_4 + 0 + 0 - 4T_2 = 2 \times 1.0 \times 0.5$

or, $T_1 - 4T_2 + T_4 = 1$ ... (ii)

For T_3 (0.5, 1.0):

$0 + 0 + T_4 + T_1 - 4T_3 = 2 \times 0.5 \times 1.0$

or, $T_1 - 4T_3 + T_3 = 1$ ... (iii)

For T_4 (1.0, 1.0):

$T_3 + 0 + 0 + T_2 - 4T_4 = 2 \times 1 \times 1$

or, $T_2 - T_3 - 4T_4 = 2$ ... (iv)

The obtained equations are:

$-4T_1 + T_2 + T_3 + 0T_4 = 0.5$

$T_1 - 4T_2 + 0T_3 + T_4 = 1$

$T_1 + 0T_2 + 4T_3 + T_4 = 1$

$0T_1 - T_2 + T_3 - 4T_4 = 2$

On solving these equations we get,

$T_1 = -0.3958$

$T_2 = -0.5416$

$T_3 = -0.5416$

$T_4 = -0.7708$

TRIBHUVAN UNIVERSITY

Institution of Science and Technology

Bachelor Level/Second Year/Third Semester/Science — Full Marks: 60
Computer Science and Information Technology [CSc. 207] — Pass Marks: 24
(Numerical Method) — Time: 3 hrs.
(New Course)

Candidates are required to give their answers in their own words as far as practicable.

The figures in the margin indicate full marks.

TU QUESTIONS-ANSWERS 2077

Group 'A'

Attempt any three questions: **(3×10 = 30)**

1. Derive the formula for Newton Raphson Method. Solve the equation $x^2 + 4x - 9 = 0$ using Newton Raphson method. Assume error precision is 0.01. Discuss drawbacks of the Newton Raphson Method.

Ans: The Newton Raphson Method is referred to as one of the most commonly used techniques for finding the roots of given equations. It can be efficiently generalized to find solutions to a system of equations. Moreover, we can show that when we approach the root, the method is quadratically convergent. In this article, you will learn how to use the Newton Raphson method to find the roots or solutions of a given equation, and the geometric interpretation of this method.

Let x_0 be the approximate root of $f(x) = 0$ and let $x_1 = x_0 + h$ be the correct root. Then $f(x_1) = 0$

$\Rightarrow f(x_0 + h) = 0$....(1)

By expanding the above equation using Taylor's theorem, we get:

$f(x_0) + hf^1(x_0) + \ldots = 0$

$\Rightarrow h = -f(x_0) / f'(x_0)$

Therefore, $x_1 = x_0 - f(x_0)/ f'(x_0)$

Now, x_1 is the better approximation than x_0.

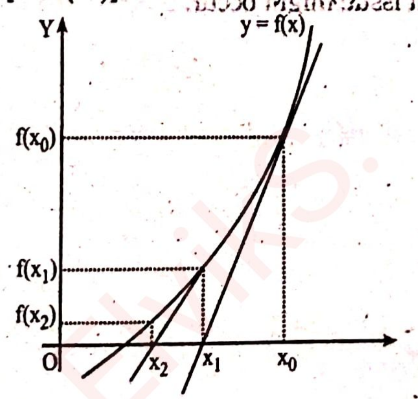
Similarly, the successive approximations $x_2, x_3, \ldots, x_{n+1}$ are given by

$x_{n+1}=x_n-f(x_n)/f'(x_n)$

This is called Newton Raphson formula.

Geometrical Interpretation of Newton Raphson Formula

The geometric meaning of Newton's Raphson method is that a tangent is drawn at the point [x0, f(x0)] to the curve y = f(x).



It cuts the x-axis at x_1, which will be a better approximation of the root. Now, drawing another tangent at $[x_1, f(x_1)]$, which cuts the x-axis at x_2, which is a still better approximation and the process can be continued till the desired accuracy is achieved.

Numerical part,

Solution:

$f(x) = x^2 + 4x - 9$

$f'(x) = 2x + 4$

Let us assume that initial guess is 4.0

Iteration 1

$x_0 = 4$ $\quad f(x) = 23$ $\quad f'(x) = 12$

$x_1 = 2.0833$

Error = 0.92

Iteration 2

$x_0 = 2.0833$ $\quad f(x) = 3.673$ $\quad f'(x) = 8.166$

$x_1 = 1.633$

Error = 0.275

Iteration 3

$x_0 = 1.633$ $\quad f(x) = 0.202$ $\quad f'(x) = 7.266$

$x_1 = 1.605$

Error = 0.017

Since error is less then specified limit

Root=1.605

Above solution can also be shown in table as below

Iteration	x	$f(x)$	$f'(x)$	Error
0	4.000	23.000	12.000	
1	2.083	3.674	8.167	0.920
2	1.634	0.202	7.267	0.275
3	1.606	0.001	7.211	0.017
4	1.606	0.000	7.211	0.000

Thus, root = 1.606

Drawbacks of the Newton Raphson Method

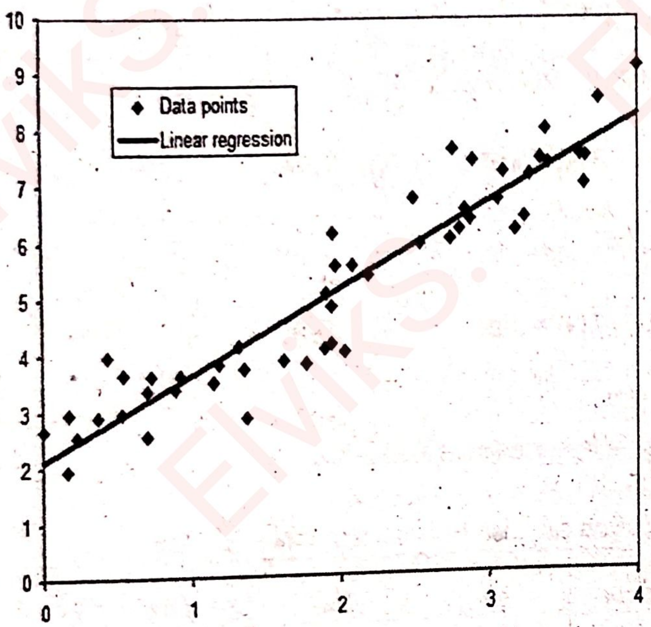
- It's convergence is not guaranteed. So, sometimes, for given equation and for given guess we may not get solution.
- Division by zero problem can occur.
- Root jumping might take place thereby not getting intended solution.
- Inflection point issue might occur.
- Symbolic derivative is required.
- In case of multiple roots, this method converges slowly.
- Near local maxima and local minima, due to oscillation, its convergence is slow.

2. How interpolation differs from regression? Write down algorithm and program for Lagrange interpolation.

Ans: Regression

Regression analysis is a family of methods in Statistics to determine relations in data. Those methods allow for evaluating the dependency of one variable on other variables. They are used to find trends in the analyzed data and quantify them. Regression analysis attempt to precisely predict the value of a dependent variable from the values of the independent variables. It also addresses measuring the degree of impact of each independent variable in the result.

The dependent variable is usually called the response or outcome. The independent variables are called features, and predictors, among others. This way, regression provides an equation to predict the response. For example, the value of the dependent variable is based on a particular combination of the feature values.



Interpolation

Interpolation is a family of methods studied in Numerical Analysis (NA), a field of Mathematics. NA is oriented to finding approximate solutions to mathematical problems. Those solutions should be as accurate as possible.

Let's assume that we have a set of points that correspond to a certain function. We don't know the exact expression of the function, but just those points. And we want to calculate the value of the function on some other points. Interpolation allows for approximating this function by using other simpler functions. The approximated function should pass through all the points in the previously known set. Then, it could be used to estimate other points not included in the set.

The main differences between the interpolation and regression methods:

Interpolation	Regression
1. Comes from Numerical Analysis	1. Comes from Statistics
2. Assumes that points in the dataset represent the values of a function accurately.	2. Accepts inexact values.
3. Assumes that one value of the independent variable just could be associated with one value of the dependent one.	3. Accepts that the same value of the independent variable could have several values of the dependent variable associated.

4. The approximated function should match exactly with all the dataset points.	4. The approximated function does not have to match exactly with the dataset points.
5. The error of the estimated values is bounded by some specific expression.	5. The error of the estimated values is bounded on average.
6. The approximated functions are frequently used for numerical integration and differentiation.	6. The approximated functions are frequently used for prediction and forecasting.

Algorithm: Lagrange Interpolation Method

1. Start
2. Read number of data (n)
3. Read data Xi and Yi for i=1 ton n
4. Read value of independent variables say xp
   whose corresponding value of dependent say yp is to be determined.
5. Initialize: yp = 0
6. For i = 1 to n
   Set p = 1
   For j =1 to n
     If i ≠ j then
       Calculate p = p * (xp - Xj)/(Xi - Xj)
     End If
   Next j
   Calculate yp = yp + p * Yi
   Next i
6. Display value of yp as interpolated value.
7. Stop

C Source Code: Lagrange Interpolation

```
#include<stdio.h>
#include<conio.h>
void main()
{
        float x[100], y[100], xp, yp=0, p;
        int i,j,n;
        clrscr();
        printf("Enter number of data: ");
        scanf("%d", &n);
        printf("Enter data:\n");
        for(i=1;i<=n;i++)
        {
            printf("x[%d] = ", i);
            scanf("%f", &x[i]);
            printf("y[%d] = ", i);
            scanf("%f", &y[i]);
        }
```

```
    printf("Enter interpolation point: ");
    scanf("%f", &xp);
    for(i=1;i<=n;i++)
    {
      p=1;
      for(j=1;j<=n;j++)
      {
            if(i!=j)
            {
                p = p* (xp - x[j])/(x[i] - x[j]);
            }
      }
      yp = yp + p * y[i];
    }
    printf("Interpolated value at %.3f is %.3f.", xp, yp);
    getch();
}
```

3. Why partial pivoting is used with Naïve Gauss Elimination method? Solve the following system of equations using Gauss Elimination method with partial pivoting. How Gauss Jordan method differs from Gauss elimination method?

$5x + y + 2z = 34$
$4y - 3z = 12$
$10x - 2y + z = -4$

Ans: The partial pivoting technique is used to avoid round off errors that could be caused when dividing every entry of a row by a pivot value that is relatively small in comparison to its remaining row entries.

In partial pivoting, for each new pivot column in turn, check whether there is an entry having a greater absolute value in that column below the current pivot row. If so, choose the entry among these having the maximum absolute value. (If two or more entries have the maximum absolute value, choose any one of those.) Then we switch rows to place the chosen entry into the pivot position before continuing the row reduction process.

Numerical part,

Method I :

In the process of converting given matrix into an Echelon form, the largest absolute value in a column, that is to be used for making Echelon form, are called, PIVOT for that column.

$$[A : B] = \left[\begin{array}{ccc:c} 5 & 1 & 2 & 34 \\ 0 & 4 & -3 & 12 \\ 10 & -2 & 1 & -4 \end{array}\right] \xrightarrow{R_1 \to R_3} \left[\begin{array}{ccc:c} \underline{10} & -2 & 1 & -4 \\ 0 & 4 & -3 & 12 \\ 5 & 1 & 2 & 34 \end{array}\right]$$

$$\xrightarrow{R_3 \to R_3 - \frac{1}{2}R_1} \left[\begin{array}{ccc:c} 10 & -2 & 1 & -4 \\ 0 & 4 & -3 & 12 \\ 0 & 2 & 3/2 & 36 \end{array}\right]$$

$$R_3 \rightarrow R_3 - \frac{1}{2}R_2 \left[\begin{array}{ccc:c} 10 & -2 & 1 & -4 \\ 0 & 4 & -3 & 12 \\ 0 & 2 & 3 & 30 \end{array}\right] \text{Echelon form}$$

Here 10 and 4 are the largest elements that are used in making of Echelon form. So, they can be taken as PIVOTS for C_1 and C_2 respectively

Method II :

$5x + y + 2z = 34$
$4y - 3z = 12$
$10x - 2y + z = -4$

Elements of C1 are |5|, |0|, |10| so largest element can be taken as 1st pivot for C1.

Hence, R1↔R3 ⇒

$10x - 2y + z = -4$ ....(1)
$4y - 3z = 12$ ....(2)
$5x + y + 2z = 34$ ....(3)

$R_3 \rightarrow R_3 - R_1/2$

⇒ $10x - 2y + z = -4$
$0x + 4y - 3z = 12$
$0x + 2y + 32z = 36$

Now elements of C_2: |4|, |2| so can be taken as 2nd Pivot for C_2

Using $R_3 \rightarrow R_3 - 1/2\, R_2$

⇒ $10x - 2y + z = -4$
$0x + 4y - 3z = 12$
$0x + 0y + 3z = 30$

Now, we have reached at Echelon form, so pivots are :

$x = 10, y = 4$

Gauss Jordan method differs from Gauss elimination method

Gauss Elimination Method	Gauss Jordan Method
1. In this method, the unknowns are eliminated successively and the system is reduced to an upper triangular system from which the unknowns are found by back substitution.	1. In this method, elimination of unknowns is performed by all equations not only from equations to follow. Thus the system ultimately reduces to a diagonal matrix form i.e. each equation involving only one unknowns.
2. Finding the solution of n simultaneous linear equation, the number of multiplications and divisions are of the order. n3/3. **For Example:** if n=5, the number of multiplications and divisions ISI elimination is approximately 42.	2. Finding the solution of n simultaneous linear equation, the number of multiplications and divisions are of the order. n3/2. **For Example:** if n=5, the number of multiplications and divisions are approximately 62.
3. It does not seem to be easier but requires about 50 percent fewer operations than Gauss Jordan Method.	3. It seems to be easier but requires about 50 percent fewer operations than Gauss elimination Method.
4. For large systems, Gauss Elimination Method is not preferred.	4. For large systems, Gauss Jordan Method is preferred to Gauss Elimination Method

Group 'B'

Attempt any Eight questions **(5×8 = 40)**

4. **Define the terms true error and relative error. Use Horner method to evaluate polynomial $2x^3-3x^2+5x-2$ at x=3 and write down its algorithm.**

Ans: Absolute Error

Absolute error is defined as the difference between a measured or derived value of a quantity and an actual value. The meaning of the absolute error depends on the quantity to be measured. Absolute errors are not enough because there is no information about the meaning of the error.

- At street distances, such as in large quantities, small errors in centimeters are negligible. When measuring the length of machine parts, the error of centimeters is large.
- In both cases the error is shown in centimeters, but the error in the second case is more important. When measuring distances between cities that are kilometers apart, errors of a few centimeters are negligible and irrelevant.
- Consider another case where the centimeter error when measuring a small mechanical part is a very significant error .Both errors are on the order of centimeters, but the second error is more serious than the first.

If x is the actual value of the quantity

x_0 is the measured value of the quantity, the absolute error value can be calculated using the following formula:

$\Delta x = x_0 - x$

Here, Δx is called the absolute error.

When considering multiple measurements, the **arithmetic mean** of the absolute error of each measurement should be the final absolute error.

Example of Absolute Error

Here are some examples of absolute mistakes in real life.

Suppose you want to measure the length of the eraser.

The actual length is 35mm and the measured length is 34.13mm.

Therefore, absolute error = actual length measurement - length = (35 - 34.13) mm = 0.87mm

Relative Error

Relative error is defined as the ratio of the absolute error of the measured value to the actual measured value.

- Using this method, you can determine the amount of absolute error with respect to the actual size of the measurement. If you don't know the actual measurement of the object, you can find the relative error from the measurement.
- Relative error indicates how good the measurement is in relation to the size of the object being measured.

- Note that the relative error is dimensionless. When writing relative errors, it is common to multiply the decimal error by 100 and express it as a percentage.

Relative Error Formula

Relative errors are calculated by the absolute error ratio and the actual amount value. If the absolute error is the measurement Δx, the actual value x0 is x and the relative error is -

$$\frac{(x0 - x)}{x} = \frac{(\Delta x)}{x}$$

Here, XR is a relative error.

Example of Relative Error

Here is an example of an actual relative error.

Suppose the actual length of the eraser is 35mm.

Here, the absolute error = (35 - 34.13) mm = 0.87 mm.

Therefore, relative error = $\frac{\text{absolute error}}{\text{actual length}} = \frac{0.87}{35} = 0.02485$

Numerical part,

$2x^3 - 3x^2 + 5x - 2$

We know that

a3 = 2, a2 = - 3, a1 = 5 and a0 = - 2

Now new sequence of constants can be determined by using recursive formula as below:

b3 = a3 = 2

b2 = a2 + b3*x = - 3 + 2* 3 = - 3 + 6 = 3

b1 = a1+ b2*x = 5 + 3 * 3 = 5 + 9 = 14

b0 = a0+ b1*x = - 2 + 14 * 3 = 40

Thus p(3) = 40

Horner method for polynomial evaluation algorithm

1. Start
2. Enter degree of polynomial, say n
3. Enter the value at which polynomial to be evaluated, x
4. Initially set $b_n=a_n$
5. While n>0

   $b_{n-1}=a_{n-1}+b_n*x$
6. End While
7. Display the value of b_0, which is the value of polynomial at x
8. Terminate

5. Construct Newton's forward difference table for the given data points and approximate the value of f(x) at x=1.1.

X	1.0	1.3	1.6	1.9	2.2
f(x)	0.7651977	0.6200860	0.4554022	0.2818186	0.1103623

Ans:

Here, $h = x_{i+1} - x_i = 0.3$

Since, $x = x_0 + sh \Rightarrow s = \dfrac{x - x_0}{h} = \dfrac{1.1 - 1.0}{0.3} = \dfrac{1}{3}$

x_i	$f(x_i)$	$\Delta f(x_0)$	$\Delta^2 f(x_0)$	$\Delta^3 f(x_0)$	$\Delta^4 f(x_0)$
1.0	0.7651977				
		-0.1451117			
1.3	0.6200860		-0.0195721		
		-0.1646838		0.0106723	
1.66	0.4554022		-0.0088998		0.0003548
		-0.1735836		0.0110271	
1.9	0.28181866		0.0021273		
		-0.1714563			
2.2	0.1103623				

Now the Newton forward divided-difference formula can be used with the forward divided differences that have a solid underscore in above table.

We know that,

$$p_n(x) = p_n(x_0 + sh) = f[x_0] + \sum_{k=1}^{n} \binom{s}{k} \Delta^k f(x_0)$$

Where,

$$\binom{s}{k} = \frac{s(s-1)\cdots(s-k+1)}{k!}$$

Thus,

$$p_4(x) = f[x_0] + \binom{s}{1}\Delta f(x_0) + \binom{s}{2}\Delta^2 f(x_0) + \binom{s}{3}\Delta^3 + \binom{s}{4}\Delta^4 f(x_0)$$

$$= f[x_0] + s\Delta f(x_0) + \frac{s(s-1)}{2!}\Delta^2 f(x_0) + \frac{s(s-1)(s-2)}{3!}\Delta^3 + \frac{s(s-1)(s-2)(s-3)}{4!}\Delta^4 f(x_0)$$

Hence,

$$p_4(1.1) = 0.7651977 + \frac{1}{3}(-0.1451117) + \frac{\frac{1}{3} \times -\frac{2}{3}}{2}(-0.0195721)$$

$$+ \frac{\frac{1}{3} \times -\frac{2}{3} \times -\frac{5}{3}}{6}(0.0106721) + \frac{\frac{1}{3} \times -\frac{2}{3} \times -\frac{5}{3} \times -\frac{8}{3}}{24}(0.0003548)$$

$$= 0.712$$

6. **Fit the curve y=ae^bx through the following data points.**

X	-4	-2	0	1	2	4
$y = f(x)$	0.57	1.32	4.12	6.65	11	30.3

Ans:

We know that exponential model is given by

$$y = ae^{bx}$$

$$\Rightarrow \log y = \log a + bx$$

This equation is similar in form to the linear equation $y = a + bx$. Thus we can evaluate parameters a and b by using the equation of linear regression model as below:

$$b = \frac{n\sum_{i=1}^{n} x_i \log y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} \log y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

And

$$\log a = \frac{\sum_{i=1}^{n} \log y_i}{n} - b\frac{\sum_{i=1}^{n} x_i}{n}$$

Now, calculate required summations as below

i	x_i	y_i	$\log y_i$	$x_i \log y_i$	x_i^2
1	-4	0.57	-0.562	2.248	14
2	-2	1.32	0.277	-0.555	4
3	0	4.12	1.415	0	0
4	1	6.65	1.894	1.894	1
5	2	11.0	2.398	4.796	4
6	4	30.3	3.411	13.645	16
n=6	$\sum x_i = 1$		$\sum \log y_i = 8.835$	$\sum x_i \log y_i = 22.03$	$\sum x_i^2 = 41$

Now,

$$b = \frac{6 \times 22.03 - 1 \times 8.835}{6 \times 41 - 1} = \frac{123.345}{245} = 0.503$$

$$\log a = \frac{\sum_{i=1}^{n} \log y_i}{n} - b\frac{\sum_{i=1}^{n} x_i}{n}$$

$$\log a = \frac{8.835}{6} - 0.503 \times \frac{1}{6} = 1.472 - 0.084 = 1.388$$

Since

$$\log a = 1.388$$

=> $a = e^{1.388} = 4.006$

Thus the regression formula then is

$$y = 4.006\, e^{0.503\, x}$$

7. Discuss the Doolittle LU decompose method for matrix factorization.

Ans: In numerical analysis and linear algebra, LU decomposition (where 'LU' stands for 'lower upper', and also called LU factorization) factors a matrix as the product of a lower triangular matrix and an upper triangular matrix. Computers usually solve square systems of linear equations using the LU decomposition, and it is also a key step when inverting a matrix, or computing the determinant of a matrix.

Coefficient matrix A of a system of linear equations can be decomposed into two triangular matrices L and U such that

$$[A]=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}=\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

If L has 1's on it's diagonal, then it is called a Doolittle factorization, thus Doolittle algorithms assumes that $\ell_{11}=1, \quad \ell_{22}=1 \ldots\ldots\ldots\ldots and \quad \ell_{nn}=1$

From above matrices,

$$a_{11}=\ell_{11}u_{11} \quad \Rightarrow u_{11}=a_{11} \qquad \because \ell_{11}=1$$

$$a_{12}=\ell_{11}u_{12} \quad \Rightarrow u_{12}=a_{12} \qquad \because \ell_{11}=1$$

$$\vdots$$

$$a_{1n}=\ell_{11}u_{1n} \quad \Rightarrow u_{1n}=a_{1n} \qquad \because \ell_{11}=1$$

$$a_{21}=\ell_{21}u_{11} \quad \Rightarrow \ell_{21}=\frac{a_{21}}{u_{11}}=\frac{a_{21}}{a_{11}}$$

$$a_{22}=\ell_{21}u_{12}+\ell_{22}u_{22} \quad \Rightarrow u_{22}=\frac{a_{22}-\ell_{21}u_{12}}{\ell_{22}}=a_{22}-\ell_{21}u_{12} \qquad \because \ell_{22}=1$$

$$\vdots$$

$$a_{2n}=\ell_{21}u_{1n}+\ell_{22}u_{2n} \quad \Rightarrow u_{2n}=\frac{a_{2n}-\ell_{21}u_{1n}}{\ell_{22}}=a_{2n}-\ell_{21}u_{1n} \qquad \because \ell_{22}=1$$

And

$$a_{n1}=\ell_{n1}u_{11} \quad \Rightarrow \ell_{n1}=\frac{a_{n1}}{u_{11}}=\frac{a_{n1}}{a_{11}}$$

$$a_{n2}=\ell_{n1}u_{12}+\ell_{n2}u_{22} \quad \Rightarrow \ell_{n2}=\frac{a_{n2}-\ell_{n1}u_{12}}{u_{22}}=\frac{1}{u_{22}}(a_{n2}-\ell_{n1}u_{12})$$

$$a_{n3}=\ell_{n1}u_{13}+\ell_{n2}u_{23}+\ell_{n3}u_{33} \quad \Rightarrow \ell_{n3}=\frac{a_{n3}-\ell_{n1}u_{13}-\ell_{n2}u_{23}}{u_{33}}=\frac{1}{u_{33}}(a_{n3}-\ell_{n1}u_{13}-\ell_{n2}u_{23})$$

$$\vdots$$

$$a_{nn}=\ell_{n1}u_{1n}+\ell_{n2}u_{2n}+\cdots\cdots+\ell_{nn}u_{nn} \quad \Rightarrow u_{nn}=\frac{a_{nn}-\ell_{n1}u_{1n}-\ell_{n2}u_{2n}\cdots\ell_{n,n-1}u_{n-1,n}}{\ell_{nn}}$$

$$\Rightarrow u_{nn}=(a_{nn}-\ell_{n1}u_{1n}-\ell_{n2}u_{2n}\cdots\ell_{n,n-1}u_{n-1,n})$$

Generalizing this we get,

If $i \leq j$

$$u_{ij}=a_{ij}-\sum_{k=1}^{i-1}l_{ik}u_{kj} \qquad j=1,2,3,\cdots\cdots\cdots\cdots,n$$

Where, $u_{11}=a_{11}, \quad u_{12}=a_{12},\cdots\cdots\cdots\cdots\cdots u_{1n}=a_{1n}$

and if $i > j$

$$\ell_{ij} = \frac{1}{u_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1}\ell_{ik}u_{kj}\right) \qquad j = 1, 2, \cdots\cdots\cdots, i-1$$

Where, $\ell_{11} = 1,\ \ \ell_{22} = 1, \cdots\cdots\cdots\cdots \ell_{nn} = 1 \quad and \quad \ell_{i1} = \dfrac{a_{i1}}{u_{11}}$

Algorithm for Matrix Factorization using Doolittle LU Decomposition

1. Start
2. Read Dimension of Matrix, say n
3. Read elements of matrix row-wise
4. Assign values to first row of U matrix as below

   For j=1 to n

   u[1][j]=a[1][j]

End for

5. Assign values to first row of L matrix as below

   For i=1 to n

   l [i][i]=1

   End for

6. Compute and assign values to first column of L matrix as below

   For i=2 to n

   l [i][1]=a[i][1]/u[1][1]

   End for

7. Compute and assign values to 2nd to nth rows of L and U matrix as below
8. For j=2 to n

   For i=2 to j

   $$u[i][j] = a[i][j] - \sum_{k=1}^{j-1}(l[i][k] * u[k][j])$$

   End for

   For i=j+1 to n

   $$l[i][j] = \frac{1}{u[j][j]}\left[a[i][j] - \sum_{k=1}^{j-1}(l[i][k] * u[k][j])\right]$$

   End for

End for

9. Display L and M matrix
10. Terminate

8. **Write down algorithm and program for the differenting continuous function using three point formula.**

Ans: Algorithm for Three-Point Backward difference Formula

1. Start
2. Read the value at which derivative is needed, say x
3. Read interval gap, say h
4. Calculate $f(x_i)$ & $f(x_{i+}h)$
5. Calculate $d = f'(x_i) = (f(x_i+h) - f(x_i-h))/\ 2h$
6. Display the value of derivative

7. Terminate

Program part,

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define f(x) 2*(x)*(x)+1
int main()
{
  float h,x,d;
  printf("Enter the value at which derivative is required\n");
  scanf("%f",&x);
  printf("Enter increment h\n");
  scanf("%f",&h);
  d=((f(x+h))-(f(x-h)))/(2*h);
  printf("Value of Derivative=%f\n",d);
  getch();
  return 0;
}
```

9. **How Simpson's 1/3 rule differs from Trapezoidal rule? Drive the formula for Simpson's 1/3 rule.**

Ans: Trapezoidal and Simpsons rule are two different methods for approximating the value of a definite integral. The Trapezoidal Rule is a polygonal rule that uses a sequence of trapezoids to approximate the value of a definite integral. The Simpsons Rule is a quadratic rule that uses a sequence of Simpsons quadrilaterals to approximate the value of a definite integral. Both methods produce an accurate result, but the Simpsons Rule is more accurate than the Trapezoidal Rule.

Simpson's rule is used for integration in the plane, while trapezoidal rule is used to calculate definite integrals. To be more precise, Simpson's rule is a particular case of Trapezoidal Rule where the number of subintervals n is 3. In this method we use left endpoint instead of midpoint and right endpoint values in the formula. Simpson's rule has higher accuracy compared to Trapezoidal Rule because it takes into account error term resulting from both lower and upper limits.

The Simpson's Rule is mainly used for approximating the value of definite integrals, whereas the Trapezoidal Rule is used for approximating the value of definite integrals as well as the value of definite sums. The Simpson's Rule is a more accurate approximation than the Trapezoidal Rule, but it can be more difficult to compute. In general, the Trapezoidal Rule is more commonly used than the Simpson's Rule, because it is simpler to compute and gives accurate results for a variety of functions.

Simpson's 1/3 Rule

The trapezoidal rule was based on approximating the integrand by a first order polynomial, and then integrating the polynomial over interval of integration.

Simpson's 1/3 rule is an extension of Trapezoidal rule where the integrand is approximated by a second order polynomial. The Simpson's 1/3 rule assumes n = 2. General quadrature formula for integration is given by

$$\int_a^b f(x)dx = \int_{x_0}^{x_0+nh} f(x)dx = nh\left[f(x_0)+\frac{n}{2}\Delta f(x_0)+\frac{1}{12}\left(2n^2-3n\right)\Delta^2 f(x_0)+\frac{1}{24}\left(n^3-4n^2+4n\right)\Delta^3 f(x_0)+\cdots\right]$$

Here, $h = \dfrac{b-a}{n}$

By putting n=2 in above relation and neglecting higher order forward differences, it can be written as

$$\int_{x_0}^{x_0+2h} f(x)dx = \int_{x_0}^{x_2} f(x)dx = 2h\left[f(x_0)+\Delta f(x_0)+\frac{1}{6}\Delta^2 f(x_0)\right]$$

$$= 2h\left[f(x_0)+\left(f(x_1)-f(x_0)\right)+\frac{1}{6}\left(f(x_0)-2f(x_1)+f(x_2)\right)\right]$$

$$= h\left[2f(x_0)+2\{f(x_1)-f(x_0)\}+\frac{1}{3}\left(f(x_2)-2f(x_1)+f(x_0)\right)\right]$$

$$\Rightarrow \quad I = \int_{x_0}^{x_2} f(x)dx = \frac{h}{3}\left[f(x_0)+4f(x_1)+f(x_2)\right]$$

This equation (2) is called Simpson's 1/3 rule.

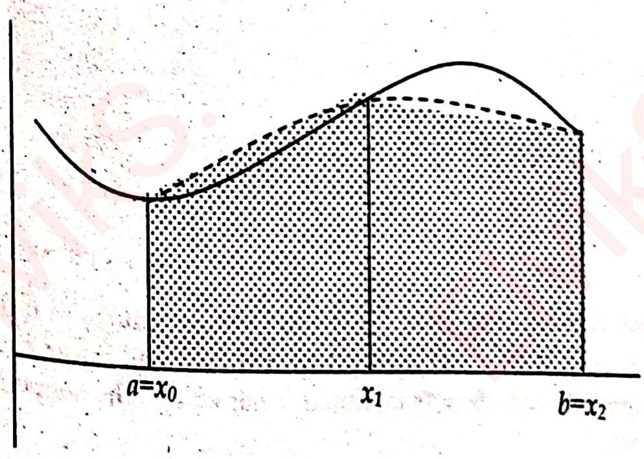


Figure: Geometrical Interpretation of Simpsons 1/3 rule

Algorithm

1. Start
2. Read value of lower & upper limit, say x_0 & $x2$
3. Set n=2
4. $h=(x_2-x_0)/n$
5. $x_1=x_0+h$
6. Calculate values $f(x_0)$, $f(x_1)$ and $f(x_2)$
7. Calculate the value of integration by using formula
$$v = \int_{x_0}^{x_2} f(x)dx = \frac{h}{3}\left[f(x_0)+4f(x_1)+f(x_2)\right]$$
8. Display the value of integration "v"
9. Terminate

10. **Approximate the solution of y'=2x+y, y(0)=1 using Euler's method with step size 0.1. Approximate the value of y(0.4).**

Ans: Here,

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

We know that,

$y(x_{i+1}) = y(x_i) + hf(x_i, y_i)$

Step 1

$x_0 = 0 \quad y_0 = 1$

=>

$$y(x_1) = y(0.1) = y(0) + 0.1 \times f(x_0, y_0)$$
$$= 1 + 0.1 \times 1 = 1.1$$

Step 2

$x_1 = 0.1 \quad y_0 = 1.1$

=>

$$y(x_2) = y(0.2) = y(0.1) + 0.1 \times f(x_1, y_1)$$
$$= 1.1 + 0.1 \times (2 \times 0.1 + 1.1) = 1.23$$

Step 3

$x_2 = 0.2 \quad y_2 = 1.23$

=>

$$y(x_3) = y(0.3) = y(0.2) + 0.1 \times f(x_2, y_2)$$
$$= 1.23 + 0.1 \times (2 \times 0.2 + 1.23) = 1.393$$

Step 4

$x_3 = 0.2 \quad y_3 = 1.23$

=>

$$y(x_4) = y(0.4) = y(0.3) + 0.1 \times f(x_3, y_3)$$
$$= 1.393 + 0.1 \times (2 \times 0.3 + 1.393) = 1.5923$$

Thus,

$y(0.4) = 1.5923$

11. **A plate of dimension 18cm x 18cm is subjected to temperatures as follows: left side at 100⁰c, right side at 200⁰c. Upper part at 50⁰c, and lower at 150⁰c. If square grid length of 6cmx6cm is assumed, what will be the temperature at the interior nodes?**

Ans: The nodes are shown in Figure below.

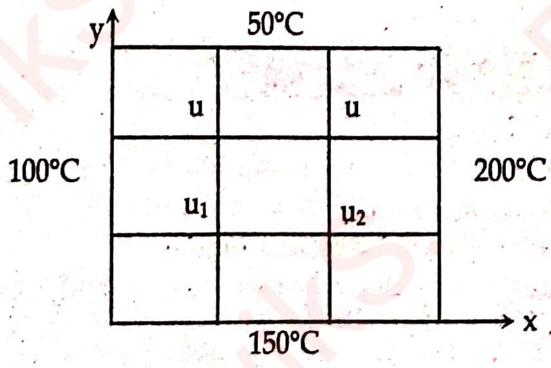


Figure: Plate with nodes

Now to get the temperature at the interior nodes we have to write Equation for all the combinations of i and j, $i = 1,\ldots, m-1; j = 1,\ldots, n-1$

For u_1

$$u_3 + u_2 + 100 + 150 - 4u_1 = 0 \quad \ldots\ldots(1)$$

$$\Rightarrow -4u_1 + u_2 + u_3 = -250$$

For u_2

$$u_4 + u_1 + 200 + 150 - 4u_2 = 0 \quad \ldots\ldots(2)$$

$$\Rightarrow u_1 - 4u_2 + u_4 = -350$$

For u_3

$$u_4 + u_1 + 100 + 50 - 4u_3 = 0 \quad \ldots\ldots(3)$$

$$\Rightarrow u_1 + u_4 - 4u_3 = -150$$

For u_4

$$u_2 + u_3 + 200 + 50 - 4u_4 = 0 \quad \ldots\ldots(4)$$

$$\Rightarrow u_2 + u_3 - 4u_4 = -250$$

By solving above 4 equations we get required solution.

12. How boundary value problems differs from initial value problems? Discuss shooting mthod for solving boundary value problem.

Ans: Initial Value Problems

Initial value problem does not require to specify the value at boundaries, instead it needs the value during initial condition. This usually apply for dynamic system that is changing over time as in Physics. An example, to solve a particle position under differential equation, we need the initial position and also initial velocity. Without these initial values, we cannot determine the final position from the equation.

Boundary Value Problems

In contrast, boundary value problems not necessarily used for dynamic system. Instead, it is very useful for a system that has space boundary. An example would be shape from shading problem in computer vision. To determine surface gradient from the PDE, one should impose boundary values on the region of interest.

Initial Value	Boundary Value
$y(t=0) = y_0$ $y'(t=0) = y_0'$	$y(t=0) = y_0$ $y(t=c) = y_1$ or $y'(t=0) = y_0'$ $y'(t=c) = y_1'$
There is only one unique solution.	There are many possible solutions.
Note: Both y and y' are evaluted at the at the same time/point.	Note: They are evaluated at different point of time.

Shooting mthod for solving boundary value problem

In this method given boundary value problem is first transformed into equivalent initial value problem and then it is solved by using any of the method used for solving initial value problem. Thus main steps involved in shooting method are:

- Transform boundary value problem into equivalent initial value problem
- Get solution of initial value problem by using any existing method
- Get solution of boundary value problem

Consider the boundary value problem

$$y'' = f(x, y, y') \qquad y(a) = u \qquad y(b) = v$$

Let $y' = z$; now we can obtain following set of two equations

$y' = z$

$z' = f(x, y, z)$

To solve above initial value problem, we need to have two conditions at $x = a$. We have given one condition $y(a) = u$. Let's guess another condition is $z(a) = g_1$. Here g_1 represents slope of $y(x)$ at $x = a$. Thus the problem can be written as system of two first order equations as below:

$y' = z \qquad y(a) = u$

$z' = f(x, y, z) \qquad z(a) = g_1$

Now, equation (1) can be solved by using any method for solving initial value problem until the solution at $x = b$ reaches to specified accuracy level. Suppose, first estimated value of $y(x)$ at $x = b$ is given by $y(b) = v_1$. If $v_1 = v$ then we are done and it is the required solution. Otherwise, we should repeat the same process by taking second guess say g_2. Suppose v_2 is the estimated value at $y(b)$ for second guess. If solution is not achieved from second guess, we can obtain better approximation by using linear interpolation as below:

$$\frac{g_3 - g_2}{v - v_2} = \frac{g_2 - g_1}{v_2 - v_1}$$

$$\Rightarrow \quad g_3 = g_2 - \frac{v_2 - v}{v_2 - v_1}(g_2 - g_1)$$

Now with $z(a) = g_3$, solution of $y(x)$ can be obtained. We have to repeat this process until solution with desired level of accuracy is achieved.

Algorithm

1. Start
2. Read Boundary conditions, say xa, xb, ya & yb
3. Read the point at which solution is needed, say xp
4. Read accuracy limit, say E
5. Convert higher order differential equation to system of differential equations
6. Read value of h
7. Approximate first approximation as below:

Set x=xa y=ya

g_1=(yb-ya)/(xb-xa)

Calculate y(xb) by using Euler's method

Set v_1=y

If(y<yb)

$g_2=2g_1$

else

$g_2= g_1/2$

Calculate y(xb) by using Euler's method

Set v_2=y

8. Compute new values of y(xb) as below

Compute $g_3 = g_2 - \frac{v_2 - y_b}{v_2 - v_1}(g_2 - g_1)$

Find $y(x_b)$ by using Euler's method

Compute error

if(error<E)

Display solution

Go to step 9

Else

Set v_1=v_2 v_2=y(xb)

Set g_1=g_2 g_2=g_3

Go to step 8

9. Terminate