Numerical Method – 2070

Q.1. $f(x) = x^2 \sin x + e^{-x} - 3 = 0$

Let $x_1 = -2$, $x_2 = 2$ and $\varepsilon = 0.01$.

Iteration 1:

$f1 = f(x_1) = f(-2) = 4.249$

$f2 = f(x_2) = f(2) = -2.72$

Since, $f_1 \times f2 < 0$, so it brackets some roots.

$$x_0 = \frac{x_1 + x_2}{2} = \frac{-2+2}{2} = 0$$

$f_0 = f(x_0) = f(0) = -2$

Here, $f_1 \times f_0 < 0$, so,

set $x_2 = x_0 = 0$

set $f_2 = f_0 = -2$

$$\left|\frac{x_2 - x_1}{x_2}\right| = \left|\frac{0+2}{0}\right| = \infty > \varepsilon$$

Iteration 2:

$$x_0 = \frac{x_1 + x_2}{2} = \frac{-2+0}{2} = -1$$

$f_0 = f(x_0) = f(-1) = -0.3$

Since, $f_1 \times f_0 < 0$, so,

set, $x_2 = x_0 = -1$

$f_2 = f_0 = -0.3$

$$\left|\frac{x_2 - x_1}{x_2}\right| = \left|\frac{-1+2}{-1}\right| = 1 > \varepsilon.$$

Iteration 3:

$$x_0 = \frac{x_1 + x_2}{2} = \frac{-2-1}{2} = -1.5$$

$f_0 = f(x_0) = f(-1.5) = 1.42$

Since, $f_1 \times f_0 > 0$, so.

set. $x_1 = x_0 = -1.5$

$f_1 = f_0 = 1.42$

$$\left|\frac{x_2 - x_1}{x_2}\right| = \left|\frac{-1+1.5}{-1}\right| = 0.5 > \varepsilon$$

Iteration 4:

$$x_0 = \frac{x_1 + x_2}{2} = \frac{-1.5-1}{2} = -1.25$$

$f_0 = f(x_0) = f(-1.25) = 0.456$

Since, $f_1 \times f_0 > 0$, so

set $x_1 = x_0 = -1.25$

$f_1 = f_0 = 0.456$

$$\left|\frac{x_2 - x_1}{x_2}\right| = \left|\frac{-1+1.25}{-1}\right| = 0.25 > \varepsilon.$$

Iteration 5:

$$x_0 = \frac{x_1 + x_2}{2} = \frac{-1.25-1}{2} = -1.125$$

$f_0 = f(x_0) = f(-1.125) = 0.055$

Since $f_1 \times f_0 > 0$, so,

set $x_1 = x_0 = -1.125$

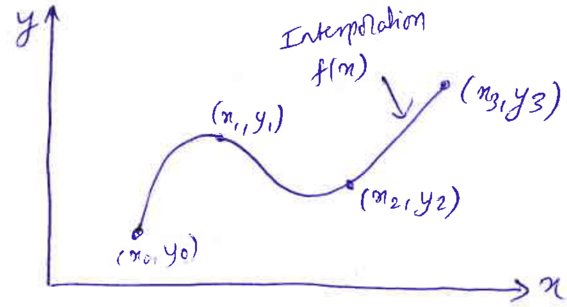
$f_1 = f_0 = 0.055$

$$\left|\frac{x_2 - x_1}{x_2}\right| = \left|\frac{-1+1.125}{-1}\right| = 0.125 > \varepsilon$$

and so on.

If $\left|\frac{x_2 - x_1}{x_2}\right| < \varepsilon$, then root $= \frac{x_1 + x_2}{2}$.

Q.2 Let $f(x)$ be continuous function may be used to represent the 'n+1' ~~point~~ data values with passing 'n+1' points. The process of computing the values of $f(x)$ or y for given x inside the given range is called Interpolation.



Soln:-

$h = 0.5$ & $x = 3.6$, $f(3.6) = ?$, $x_0 = 2$

$\because x = x_0 + sh \Rightarrow s = \dfrac{x - x_0}{h} = \dfrac{3.6 - 2}{0.5} = 3.2$

x_i	$f(x_i)$	$\Delta f(x_0)$	$\Delta^2 f(x_0)$
2	1.43		
		-0.4	
2.5	1.03		0.13
		-0.27	
3	0.76		0.11
		-0.16	
3.5	0.6		0.04
		-0.12	
4	0.48		0.03
		-0.09	
4.5	0.39		

Now, using 2nd order Newton's Forward difference formula,

$$f(x) = P_2(x) = P_2(x_0 + sh) = f(x_0) + \Delta f(x_0)s + \frac{\Delta^2 f(x_0)s(s-1)}{2!}$$

$$\therefore f(3.6) = P_2(2 + 3.2 \times 0.5) = 1.43 + (-0.4) \times 3.2 + \frac{0.13 \times 3.2 \times (3.2 - 1)}{2}$$

$$= 1.43 - 1.28 + 0.4576$$

$$= 0.6076$$

Q.3 Simpson's 1/3 Rule :-

As we know from Newton's Cotes formula,

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

By putting $n=2$ in eqn (1) & neglecting higher terms, we get,

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

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

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

$$\Rightarrow I = \int_{x_0}^{x_2} f(x)\,dx = \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right] \quad \text{—} \ (2)$$

This eqn (2) is called Simpson's 1/3 Rule

Soln:-

$$I = \int_{0.2}^{1.2} (x^2 + \ln x - \sin x)\,dx \ ; \quad h = 0.1$$

Here, $x_0 = 0.2$, $x_2 = 1.2$

$f(x) = x^2 + \ln x - \sin x$

$f(x_0) = f(0.2) = -1.57$

$f(x_1) = f(x_0 + h) = f(0.3) = -1.12$

$f(x_2) = f(1.2) = 1.6$

$$\therefore I = \int_{0.2}^{1.2} (x^2 + \ln x - \sin x)\,dx$$

$$= \frac{0.1}{3}\left[-1.57 + 4 \times (-1.12) + 1.6\right]$$

$$= -0.148$$

Q.4 Using Gauss Elimination

Arranging in the matrix form,

$$\begin{bmatrix} 1 & 10 & 1 \\ 10 & 1 & 1 \\ 1 & 1 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 24 \\ 15 \\ 33 \end{bmatrix}$$

$R_2 \rightarrow R_2 - 10R_3$

$$\begin{bmatrix} 1 & 10 & 1 \\ 0 & -9 & -99 \\ 1 & 1 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 24 \\ -315 \\ 33 \end{bmatrix}$$

$R_3 \rightarrow R_3 - R_1$

$$\begin{bmatrix} 1 & 10 & 1 \\ 0 & -9 & -99 \\ 0 & -9 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 24 \\ -315 \\ 9 \end{bmatrix}$$

$R_3 \rightarrow R_3 - R_2$

$$\begin{bmatrix} 1 & 10 & 1 \\ 0 & -9 & -99 \\ 0 & 0 & 108 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 24 \\ -315 \\ 324 \end{bmatrix}$$

The corresponding system of eq^n is:

$x_1 + 10x_2 + x_3 = 24$ — (1)

$-9x_2 - 99x_3 = -315$ — (2)

$108x_3 = 324$ — (1)

from (3), we get,

$x_3 = 3$

from (2),

$-9x_2 - 99 \times (3) = -315$

$\therefore x_2 = 2$

from (1),

$x_1 + 10 \times (2) + 3 = 24$

$\therefore x_1 = 1$

$$\therefore x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Q.5. Comparison betn Euler's & Huen's method:-

In Euler's method, the slope at (x_i, y_i) is used to estimate the value of $y(x_{i+1})$ as below:

$$y(x_{i+1}) = y(x_i) + m_1 h \quad ; \quad m_1 = f(x_i, y_i)$$

For better accuracy, we should choose small value of h. Therefore this method has slow convergence rate.

In Huen's method, we use the average of the slopes computed at the beginning and at the end of the interval. (ie average of slopes m1 & m2)
Using Huen's method, we can estimate the value of $y(x_{i+1})$ as below:

$$y(x_{i+1}) = y(x_i) + \frac{h}{2}(m_1 + m_2) \quad ; \quad m_1 = f(x_i, y_i) \ \& \ m_2 = f(x_{i+1}, y_{i+1})$$

Soln:-

$y' = 1 - 2x^2 y$ $\qquad x_0 = 0 \ , \ y_0 = 0$

$\therefore f(x,y) = 1 - 2x^2 y$ $\qquad h = 0.25$

Now, from Runge-Kutta method, we have,

$$m_1 = f(x_0, y_0) = 1 - 2x_0^2 y_0 = 1 - 2\times 0^2 \times 0 = 1$$

$$m_2 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{m_1 h}{2}\right) = f(0.125, 0.125) = 0.996.$$

$$m_3 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{m_2 h}{2}\right) = f(0.125, 0.1245) = 0.9961$$

$$m_4 = f(x_0 + h, y_0 + m_3 h) = f(0.25, 0.249) = 0.968$$

Hence,

$$y(1.5) = y_0 + \left(\frac{m_1 + 2m_2 + 2m_3 + m_4}{6}\right) h.$$

$$= 0 + \left(\frac{1 + 2\times 0.996 + 2\times 0.9961 + 0.968}{6}\right) \times 0.25$$

$$= 0.248$$

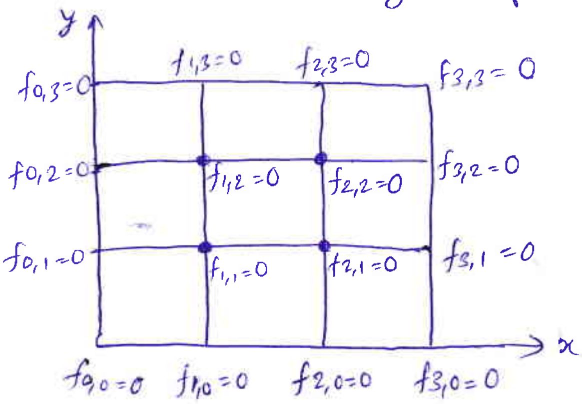
Q.6 Soln:-

Given, The Poisson's eqn : $\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 3x^2 y$; $0 \le x \le 1.5$ & $0 \le y \le 1.5$

$h = 0.5$

Here, $g(x,y) = 3x^2 y$

Lets divide the domain into grids of 3x3 as below:



we have,

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} = h^2 g_{i,j}$$

At $i=1$ & $j=1$,

$$f_{2,1} + f_{0,1} + f_{1,2} + f_{1,0} - 4f_{1,1} = 0.5^2 \times (3 \times 1^2 \times 1)$$

or, $f_{2,1} + f_{1,2} - 4f_{1,1} = 0.75$ —— (1)

At $i=2$ & $j=1$,

$$f_{3,1} + f_{1,1} + f_{2,2} + f_{2,0} - 4f_{2,1} = 0.5^2 \times (3 \times 2^2 \times 1)$$

or $f_{1,1} + f_{2,2} - 4f_{2,1} = 3$ —— (2)

At $i=1$ & $j=2$,

$$f_{2,2} + f_{0,2} + f_{1,3} + f_{1,1} - 4f_{1,2} = 0.5^2 \times (3 \times 1^2 \times 2)$$

or, $f_{2,2} + f_{1,1} - 4f_{1,2} = 1.5$ —— (3)

At $i=2$ & $j=2$,

$$f_{3,2} + f_{1,2} + f_{2,3} + f_{2,1} - 4f_{2,2} = 0.5^2 \times (3 \times 2^2 \times 2)$$

or, $f_{1,2} + f_{2,1} - 4f_{2,2} = 6$ —— (4)

Solving these eqns,

Using ④ in ①,

$6 + 4f_{2,2} - \cancel{f_{1,2}} + \cancel{f_{1,2}} - 4f_{1,1} = 0.75$

or, $4f_{2,2} - 4f_{1,1} = -5.25$ ---- ⓐ

Using ④ in ②,

$f_{1,1} + f_{2,2} - 4[6 + 4f_{2,2} - f_{1,2}] = 3$

or, $f_{1,1} + f_{2,2} - 24 - 16f_{2,2} + 4f_{1,2} = 3$

or, $-15f_{2,2} + f_{1,1} + 4f_{1,2} = 27$ ---- ⓑ

& we have eq^n ③,

$f_{2,2} + f_{1,1} - 4f_{1,2} = 1.5$ ---- ③

Solving eq^n ⓐ, ⓑ & ③, we get,

$f_{2,2} = -2.15625$

$f_{1,1} = -0.84375$

$f_{1,2} = -1.125$

Using these values in eq^n ①, we get,

$f_{2,1} = -1.5$

Q.7. Algorithm for Lagrange's Interpolation :-

1) Read number of points n
2) Read the value at which value is needed (say x)
3) Read available data points

```
for i = 1 to n
for j = 1 to n
if (j != i)
Lx[i] = Lx[i] * ((x - x[j]) / (x[i] - x[j]))
end if
end for
end for
```

4) for i = 0 to n

   v = v + fx[i] * Lx[i]

   end for

5) Print interpolation value v at x

6) Stop.

C-Program for Lagrange's Interpolation

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int n, i, j;
    float x, l, v = 0, ax[10], fx[10], Lx[10];
    printf("Enter the number of points : \n");
    scanf("%d", &n);
    printf("Enter the value of x : \n");
    scanf("%f", &x);
    for (i = 0; i<n; i++)
    {
        printf("Enter the value of x & fx at i = %d : \n", i);
        scanf("%f %f", &ax[i], &fx[i]);
    }
    for (i = 0; i<n; i++)
    {
        l = 1.0;
        for (j = 0; j<n; j++)
        {
            if (j != i)
                l = l * ((x - ax[j]) / (ax[i] - ax[j]));
        }
        Lx[i] = l;
    }
    for (i = 0; i<n; i++)
    {
        v = v + fx[i] * Lx[i];
    }
```

```
    printf ("Interpolation value = %f", v);
    getch();
    return 0;
}
```