

1. Define the types of error in numerical calculation. Derive the formula for secant method and illustrate the method by figure.

There are different types of error in numerical calculation. Some of them are listed below:

Truncation error:- The error caused by the numerical method which is used to solve the problem is known as truncation error. For eg:- we can approximate e^x as

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!}$$

which is obtained by taking only the first four terms of the following Taylor series expansion

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \dots \dots \frac{x^n}{n!} + \dots \dots \dots$$

So, the truncation error that occurs in this case due to the use of Taylor's series is

$$\sum_{n=0}^{\infty} \frac{x^n}{n!}$$

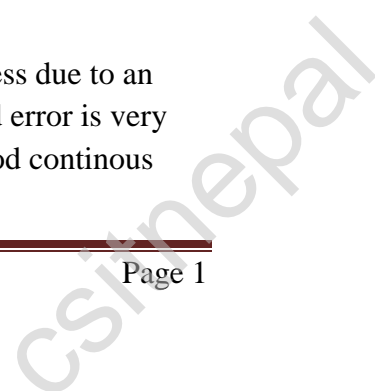
Round off error:- Every computer uses a fixed amount of memory to represent the numerical quantities such as integer and floating point number. If during some calculation, a number is generated which is larger than the allocated amount of memory then such number are stored by first approximating it to the nearest number that the computer can represent. The error occurs during this process is known as the round off error. For example, if a computer can represent only five digit point number then number like 362.688 is changed to 362.69
other examples are:-

$$\begin{aligned} 355.689 &\rightarrow 355.69 \\ 485.963 &\rightarrow 485.94 \end{aligned}$$

Error in original data:- When a real world problem are solved by using numerical methods, one must gather data to model the physical situation. But the data gather during this process are usually not exact due to various reason such as incorrect measurement or inaccurate measuring instruments; such kind of error are called error in original data.

Blunders:- Blunders are the mistakes that involve human, such human error can take many forms such as lack of problem understanding, wrong assumptions, making mistakes in computer programs, various mistakes in data input etc. Blunders can be avoid through a good understanding of problems and the numerical solution methods used as well as use of good programming tools and technique.

Propagated errors:- Errors that occurs in the succeeding step of a numerical process due to an occurrence of an earlier error is called a propagated error. Controlling propagated error is very important in numerical computation. If errors increase continuously as the method continuous



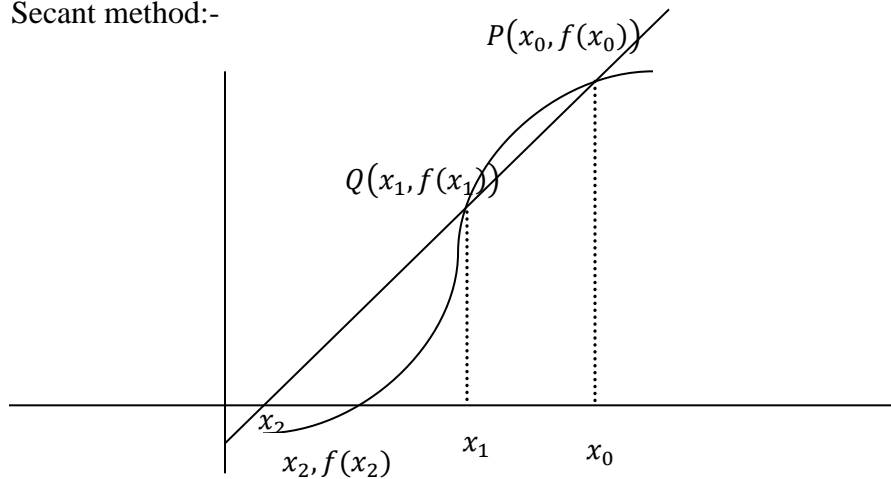
then it may grow out of control. Such numerical method is called unstable. If the errors made at early stages cancel out as the method continues, then such numerical method is said to be stable.

Absolute relative error:- If x_t denotes the true value of a numerical quantity and x_a denotes the approximate value, then we define absolute error e_a and relative error e_r as follows:

$$e_a = |x_t - x_a|$$

$$e_r = \frac{\text{absolute error}}{|\text{true value}|}$$

Secant method:-



Suppose that $f(x)$ is a continuous function whose root is ξ . Let x_0 & x_1 be two initial points which are sufficiently close to ξ . We join the points $p = (x_0, f(x_0))$ and $Q(x_1, f(x_1))$ by the secant PQ whose equation is

$$y - f(x_0) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} (x - x_0) \text{ --- (i)}$$

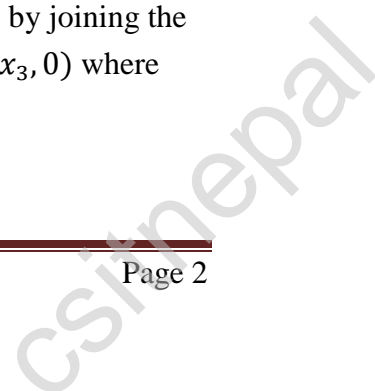
let $(x_2, 0)$ be the point of intersection of this secant PQ and the x-axis then we take the point x_2 as the first approximation of the root ξ . Since $(x_2, 0)$ lies on PQ, so from (i)

$$0 - f(x_0) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} (x_2 - x_0)$$

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

If $f(x_2) = 0$ then x_2 is the root of $f(x)$. If not, then we repeat the above process by joining the points $(x_1, f(x_1))$ & $(x_2, f(x_2))$ by secant whose intersection with the x-axis is $(x_3, 0)$ where

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



Numerical Method (2068 – Third Batch)

This point is taken as the second approximation of the root $\xi(x_i)$. In general, the n^{th} approximation of the root ξ is given by the formula

$$x_{n+1} = \frac{x_{n-1}f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}, n > 1$$

2. Define the linear least square approximations. Give the data set (x_i, y_i) as (20.5,765), (32.7,826), (51.0, 873), (73.2,942), (95.7,1032). Find the linear least square to fit given data

Least square approximation

Let $(x_i, y_i), i=1,2,\dots,n$ be a given set of 'n' pair of data points and let $y = f(x)$ be the curve that fits this data. Here $x = x_i$ the given value of the ordinate is y_i and the corresponding value on the fitting curve is $f(x_i)$. Then the error of approximation at $x = x_i$ is $e_i = y - f(x_i)$.

Let

$$\begin{aligned} S &= (y_1 - f(x_1))^2 + (y_2 - f(x_2))^2 + \dots + (y_n - f(x_n))^2 \\ &= e_1^2 + e_2^2 + \dots + e_n^2 \\ &= \sum_{i=1}^n e_i^2 \end{aligned}$$

Then the method of least squares approximation consists of finding an expression $y = f(x)$ such that the sum of the squares of the error is minimized.

Solution

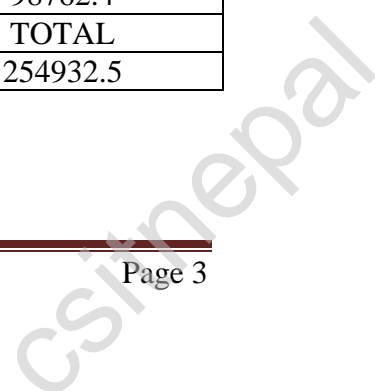
x_i	20.5	32.7	51.0	73.2	95.7
y_i	765	826	873	942	1032

Here we have $n=5$

x_i	y_i	x_i^2	$x_i y_i$
20.5	765	420.25	15682.5
32.7	826	1069.29	27010.2
51.0	873	2601	44523
73.2	942	5358.24	68954.4
95.7	1032	9158.49	98762.4
TOTAL	TOTAL	TOTAL	TOTAL
273.1	4438	18607.27	254932.5

So,

$$b = \frac{n \times \sum xy - \sum x \cdot \sum y}{n \times \sum x_i^2 - (\sum x_i y)^2}$$



$$\begin{aligned}
 &= \frac{5 \times 254932.5 - 273.1 \times 4438}{5 \times 18607.27 - (273.1)^2} \\
 &= -0.959629 \\
 a &= \frac{\sum y - \sum x}{n} \\
 &= \frac{4438 - (-0.959629) \times 273.1}{5} \\
 &= 835.18
 \end{aligned}$$

Hence, $y = a + bx = 835.18 - 0.959629x$

3. Evaluate $I = \int_0^1 e^{-x^2}$ using trapezoid rule with $n=10$. Also evaluate the same integral using Gaussian 3 point formula and compare the result

Solution

By using trapezoid rule

Here $a=0$ $b=1$ $n=10$ so

$h = \frac{b-a}{2} = \frac{1-0}{10} = 0.1$ so we get the following table

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$f(x)$	1	0.99	0.96	0.91	0.85	0.77	0.69	0.61	0.52	0.44	0.36

Therefore value of

$$\int_0^1 e^{-x^2} dx = \frac{h}{2} [1 + 0.36 + 2(0.99 + 0.96 + 0.91 + 0.85 + 0.77 + 0.69 + 0.61 + 0.52 + 0.44)]$$

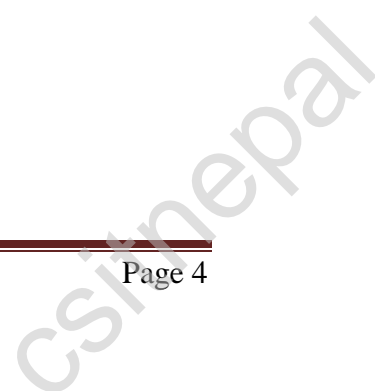
$$= \frac{0.1}{2} [1.36 + 2 * 6.74]$$

$$= 0.74$$

By using Gaussian 3 point formula

let

$$x = \frac{(b-a)y + b + a}{2}$$



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

$$= 0.5y + 0.5$$

Then limits of integration are changed from (0,1) to (-1,1) so, that

$$\int_0^1 e^{-x^2} dx = \frac{b-a}{2} \int_{-1}^1 e^{-(0.5y+0.5)^2} dy$$

Using Gaussian 3 point formula we get

$$\begin{aligned} & 0.5556 \times e^{-(0.5 \times (-0.7735) + 0.5)^2} + 0.88889 \times e^{-(0.5 \times 0 + 0.5)^2} + 0.55556 \times e^{-(0.5 \times 0.77460 + 0)} \\ & = 0.5556 \times 0.987 + 0.88889 \times 0.7788 + 0.55556 \times 0.9873 \\ & = 1.789 \end{aligned}$$

4. Solve the following system of linear equations using Gauss-elimination method (use partial pivoting if necessary):

$$\begin{aligned} 2x_2 + x_4 &= 0; 2x_1 + 2x_2 + 3x_3 + 2x_4 = -2; 4x_1 - 3x_2 + x_4 \\ &= -7; 6x_1 + x_2 - 6x_3 - 5x_4 = 6 \end{aligned}$$

The augmented form of the given equation is as follow:

$$\begin{bmatrix} 0 & 2 & 0 & 1 & 0 \\ 2 & 2 & 3 & 2 & -2 \\ 4 & -3 & 0 & 1 & -7 \\ 6 & 1 & -6 & -5 & 6 \end{bmatrix}$$

Interchanging 1st and 2nd row: $R_1 \leftrightarrow R_2$

$$\begin{bmatrix} 2 & 2 & 3 & 2 & -2 \\ 0 & 2 & 0 & 1 & 0 \\ 4 & -3 & 0 & 1 & -7 \\ 6 & 1 & -6 & -5 & 6 \end{bmatrix}$$

Eliminating x_1 from 2nd, 3rd and 4th row: $R_2 \rightarrow R_2; R_3 \rightarrow R_3 - 2R_1; R_4 \rightarrow R_4 - 3R_1$

$$\begin{bmatrix} 2 & 2 & 3 & 2 & -2 \\ 0 & 2 & 0 & 1 & 0 \\ 0 & -7 & -6 & -3 & -3 \\ 0 & -5 & -15 & -11 & 12 \end{bmatrix}$$

Eliminating x_2 from 3rd and 4th row: $R_3 \rightarrow R_3 + \frac{7}{2}R_2; R_4 \rightarrow R_4 + \frac{5}{2}R_2$

$$\begin{bmatrix} 2 & 2 & 3 & 2 & -2 \\ 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & -6 & \frac{1}{2} & -3 \\ 0 & 0 & -15 & -\frac{17}{2} & 12 \end{bmatrix}$$

Eliminating x_3 from 4th row: $R_4 \rightarrow R_4 - \frac{15}{6}R_3$

$$\begin{bmatrix} 2 & 2 & 3 & 2 & -2 \\ 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & -6 & \frac{1}{2} & -3 \\ 0 & 0 & 0 & -\frac{117}{12} & \frac{39}{2} \end{bmatrix}$$

Now, performing back substitution we get,

$$-\frac{117}{12}x_4 = \frac{39}{2} \Rightarrow x_4 = -2$$

$$-6x_3 + \frac{1}{2}x_4 = -3 \Rightarrow x_3 = \frac{1}{3}$$

$$2x_2 + x_4 = 0 \Rightarrow x_2 = 1$$

$$2x_1 + 2x_2 + 3x_3 + 2x_4 = -2 \Rightarrow x_1 = -\frac{1}{2}$$

Therefore, $x_1 = -\frac{1}{2}$; $x_2 = 1$; $x_3 = \frac{1}{3}$; $x_4 = -2$

OR 4. What do you mean by eigen-value eigen vectors problems? Find the largest eigen value correct to two significant digits and corresponding eigen vectors of the following matrix using power method.

$$A = \begin{bmatrix} 2 & 4 & 1 \\ 0 & 1 & 3 \\ 1 & 0 & 3 \end{bmatrix}$$

Eigen value and Eigen vector problems

Let A be an $n \times n$ matrix. A number $\lambda \in \mathbb{C}$ such that $Ax = \lambda x$. Every such non-zero matrix vector is called an eigen vector of A associated with the eigen value λ .

Solution

Here,

$$A = \begin{bmatrix} 2 & 4 & 1 \\ 0 & 1 & 3 \\ 1 & 0 & 3 \end{bmatrix}$$

Let the initial approximation to eigen vector be

$$\begin{aligned} x_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \text{ Then } x_1 = Ax_0 &= \begin{bmatrix} 2 & 4 & 1 \\ 0 & 1 & 3 \\ 1 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 2 + 4 + 1 \\ 0 + 1 + 3 \\ 1 + 0 + 3 \end{bmatrix} \\ &= \begin{bmatrix} 7 \\ 4 \\ 4 \end{bmatrix} \end{aligned}$$

Dividing x_1 by it's large variable by 7. Normalizing x_1 we get

$$\begin{aligned} &\begin{bmatrix} 1 \\ 0.57 \\ 0.57 \end{bmatrix} \\ x_2 = Ax_1 &= \begin{bmatrix} 2 & 4 & 1 \\ 0 & 1 & 3 \\ 1 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 0.57 \\ 0.57 \end{bmatrix} \\ &= \begin{bmatrix} 2 + 2.28 + 0.57 \\ 0.57 + 1.71 \\ 1 + 0 + 1.71 \end{bmatrix} \\ &= \begin{bmatrix} 4.85 \\ 2.28 \\ 2.71 \end{bmatrix} \end{aligned}$$

At this point, the approximation eigen value is 4.85 and the corresponding eigen vector is,

$$\begin{bmatrix} 1 \\ 0.47 \\ 0.55 \end{bmatrix}$$

5. Write an algorithm and computer program to solve system of linear equations using Gauss-Jordan method.

Algorithm for Gauss-Jordan method:

1. Normalize the first equation by dividing by its pivot element.
2. Eliminate variable x_1 from all other equations.
3. Normalize the second equation by dividing by its pivot element.

Numerical Method (2068 – Third Batch)

4. Eliminate variable x_2 from all the other equations above and below the normalized pivot equation.
5. Repeat this process until x_n is eliminated.
6. Obtain the solution as resulting vector b .

Computer program for Gauss-Jordan method:

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<math.h>
```

```
void main()
```

```
{    int n,i,j,k,l;
```

```
    float a[10][10],b[10],t;
```

```
    clrscr();
```

```
    printf("Give no. of variables ");
```

```
    scanf("%d", &n);
```

```
    for( i =0;i<= n; i++)
```

```
    {    for(j=0;j<n;j++)
```

```
        {    printf("%d ,%d  th coefficient wanted ", i+1 ,j+1);
```

```
            scanf("%f", &a[i][j]);
```

```
        }
```

```
        printf("%d th constant wanted ", i+1);
```

```
        scanf("%f", &b[i]);
```

```
    }
```

```
    for(i=0;i<= n; i++)
```

```
    {    t = fabs(a[i][i]);
```

```
        k = i ;
```

```
        for(j=i+1;j<n; j++)
```



```
if (fabs ( a[j][i] >t))
{
    k = j;
    t = fabs ( a[j][i] );
}

if( t< 1e - 5)
{
    printf(“Almost singular matrix, solution not possible\n”);
    exit(0);
}

if (i!= k)
for(l=0;l<n;l++)
{
    t = a[i][l];
    a[i][l] = a[k][l];
    a[k][l]= t;
    t = b[i] ;
    b[i]= b[l];
    b[l] = t;
}

t = a[i][i];
for(l=0;l<=n;l++) a[i][l]/=t;
b[i]/=t;
for( j=0;j<n;j++)
if( j!= i)
{
    t= a[i][i] ;
    for(l=0;l<n;l++)
        a[i][l]- = t*a[i][l];
```

```
        b[j]-= t*b[i];
    }
}
printf(" Solution are \n");
for( i=0 ; i< n ; i++)
printf(" x[%d]= %f \n",i, b[i]);
getch();
}
```

6. Apply Runge Kutta method of 2nd order and 4th order to find an approximate value of y when $x = 0.2$ given that $dy = x + y$ & $y(0) = 1$.

Here, $dy = x + y$; $y(0) = 1$

Let us assume $x_0 = 0, y_0 = 1$ & $h = 0.2$

For 2nd order:

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

$$m_2 = f(x_0 + h, y_0 + m_1 \times h) = f(0 + 0.2, 1 + 1 \times 0.2) = f(0.2, 1.2) = 1.4$$

Then,

$$y(0.2) = y(0) + \frac{m_1 + m_2}{2} \times h = 1 + \frac{1 + 1.4}{2} \times 0.2 = 1.24$$

For 4th order:

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

$$m_2 = f\left(x_0 + \frac{h}{2}, y_0 + m_1 \times \frac{h}{2}\right) = f\left(0 + \frac{0.2}{2}, 1 + 1 \times \frac{0.2}{2}\right) = f(0.1, 1.1) = 1.2$$

$$m_3 = f\left(x_0 + \frac{h}{2}, y_0 + m_2 \times \frac{h}{2}\right) = f\left(0 + \frac{0.2}{2}, 1 + 1.2 \times \frac{0.2}{2}\right) = f(0.1, 1.12) = 1.22$$

$$m_4 = f(x_0 + h, y_0 + m_3 \times h) = f(0 + 0.2, 1 + 1.22 \times 0.2) = f(0.2, 1.244) = 1.444$$

Then,

$$\begin{aligned}
 y(0.2) &= y(0) + \frac{m_1 + 2 \times m_2 + 2 \times m_3 + m_4}{6} \times h \\
 &= 1 + \frac{1 + 2 \times 1.2 + 2 \times 1.22 + 1.444}{6} \times 0.2 = 1.2428
 \end{aligned}$$

7. How can you solve Laplace's equation? Explain. The steady-state two dimensional heat flow in a metal plate is defined by $\frac{d^2T}{dx^2} + \frac{d^2y}{dy^2} = 0$.

A steel plate of size 30×30 cm is given. Two adjacent sides are placed at 100°C and other sides are held at 0°C . Find the temperature at interior points, assuming the grid size of 10×10 cm.

Difference equation to represent Laplace's equation:

Let $u = u(x, y)$ be a function of two independent variables x & y . Then by Taylor's formula:

$$u(x + h, y) = u(x, y) + hu_x(x, y) + \frac{h^2}{2!}u_{xx}(x, y) + \frac{h^3}{3!}u_{xxx}(x, y) + \dots \dots \dots (i)$$

$$u(x - h, y) = u(x, y) - hu_x(x, y) + \frac{h^2}{2!}u_{xx}(x, y) - \frac{h^3}{3!}u_{xxx}(x, y) + \dots \dots \dots (ii)$$

$$u(x, y + k) = u(x, y) + ku_y(x, y) + \frac{k^2}{2!}u_{yy}(x, y) + \frac{k^3}{3!}u_{yyy}(x, y) + \dots \dots \dots (iii)$$

$$u(x, y - k) = u(x, y) - ku_y(x, y) + \frac{k^2}{2!}u_{yy}(x, y) - \frac{k^3}{3!}u_{yyy}(x, y) + \dots \dots \dots (iv)$$

Adding equations (i) & (ii) and ignoring the terms containing h^4 and higher powers, we get

$$u(x + h, y) + u(x - h, y) = 2u(x, y) + h^2u_{xx}(x, y)$$

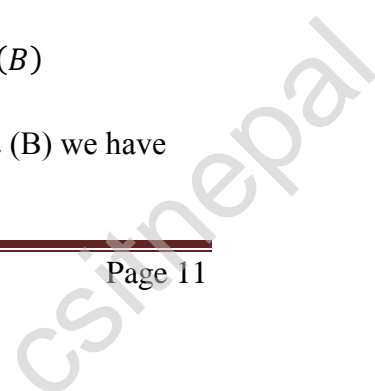
$$\text{or, } u_{xx}(x, y) = \frac{1}{h^2} [u(x + h, y) - 2u(x, y) + u(x - h, y)] \dots \dots \dots (A)$$

Adding equations (iii) & (iv) and ignoring the terms containing k^4 and higher powers, we get

$$u(x, y + k) + u(x, y - k) = 2u(x, y) + k^2u_{yy}(x, y)$$

$$\text{or, } u_{yy}(x, y) = \frac{1}{k^2} [u(x, y + k) - 2u(x, y) + u(x, y - k)] \dots \dots \dots (B)$$

Now if $u_{xx} + u_{yy} = 0$ is the given Laplace's equation, then from equation (A) & (B) we have



Numerical Method (2068 – Third Batch)

$$\frac{1}{h^2} [u(x+h, y) - 2u(x, y) + u(x-h, y)] + \frac{1}{k^2} [u(x, y+k) - 2u(x, y) + u(x, y-k)] = 0$$

Choosing $h = k$, we get

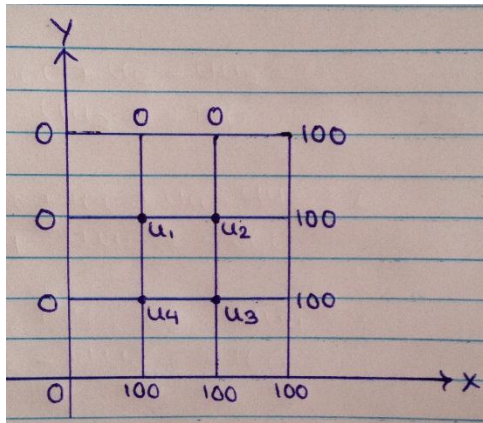
$$u(x+h, y) + u(x, y+h) + u(x-h, y) + u(x, y-h) - 4u(x, y) = 0$$

$$\therefore u(x, y) = \frac{1}{4} [u(x+h, y) + u(x, y+h) + u(x-h, y) + u(x, y-h)]$$

is the difference equation for Laplace's equation.

Numerical:

A steel plate is as follow:



From the difference equation for Laplace's equation, we have

$$0 + 0 + u_2 + u_4 - 4u_1 = 0 \Rightarrow -4u_1 + u_2 + u_4 = 0 \dots\dots\dots (i)$$

$$0 + 100 + u_1 + u_3 - 4u_2 = 0 \Rightarrow u_1 + u_3 - 4u_2 = -100 \dots\dots\dots (ii)$$

$$100 + 100 + u_2 + u_4 - 4u_3 = 0 \Rightarrow u_2 + u_4 - 4u_3 = -200 \dots\dots\dots (iii)$$

$$100 + 0 + u_1 + u_3 - 4u_4 = 0 \Rightarrow u_1 + u_3 - 4u_4 = -100 \dots\dots\dots (iv)$$

Solving the equations (i), (ii), (iii) & (iv), we get

$$u_1 = 25$$

$$u_2 = u_4 = 50$$

$$u_3 = 75$$