Numerical Method - 2072

Q. 1 The sources of error are :-

1) Round off error :-

A computer can only represent a number approximately for eg: a number like $\frac{1}{3}$ may be represented as 0.333333 on a PC. Then the round off error in this case is $\frac{1}{3} - 0.333333 = 0.00000033$. Then there are other numbers that cannot be represented exactly. For eg: π and $\sqrt{2}$ are numbers that need to be approximated in computer calculations.

2) Truncation Error :-

Truncation Error is defined as the error caused by truncating a mathematical procedure for eg: the Maclaurin series for e^x is given as,

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

This series has an infinite number of terms but when using this series to calculate e^x, only a finite number of terms can be used. For eg: if one uses three terms to calculate e^x, then, $e^x \approx 1 + x + \frac{x^2}{2!}$, the truncation error for such an approximation is:

$$\text{Truncation Error} = e^x - \left(1 + x + \frac{x^2}{2!}\right) = \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots\cdots$$

Types of errors:

1) True Error :-

True error denoted by ε_t is the difference between the true value (also called the exact value) and the approximate value.

$$\text{True Error} = \text{True value} - \text{Approximate value}$$

2) Relative error :-

It is denoted by ε_r and is defined as the ratio between the true error & true value.

$$\varepsilon_r = \frac{\text{True error}}{\text{True value}}$$

3) Approximate error :-

It is denoted by ε_a & is defined as the difference between the present approximation & previous approximation. ie

ε_a = Present approx. − Previous approx.

4) Relative Approximate Error :-

It is denoted by ε_{ra} & defined as the ratio between the approximate error & the present approximation.

$$\varepsilon_{ra} = \frac{\text{Approx. error}}{\text{Present approx.}}$$

Soln :-

$f(x) = x^2 + 5.6x - 14 = 0$

Step 1: Guess $x = 0$ $\Rightarrow$ $f(x) = -14$

Step 2: Guess $x = 1$ $\Rightarrow$ $f(x) = -7.4$

Step 3: Guess $x = 2$ $\Rightarrow$ $f(x) = 1.2$

Step 4: Guess $x = 1.5$ $\Rightarrow$ $f(x) = -3.35$

Step 5: Guess $x = 1.8$ $\Rightarrow$ $f(x) = -0.68$

Step 6: Guess $x = 1.9$ $\Rightarrow$ $f(x) = 0.25$ (approx root)

Step 7: Guess $x = -1$ $\Rightarrow$ $f(x) = -18.6$

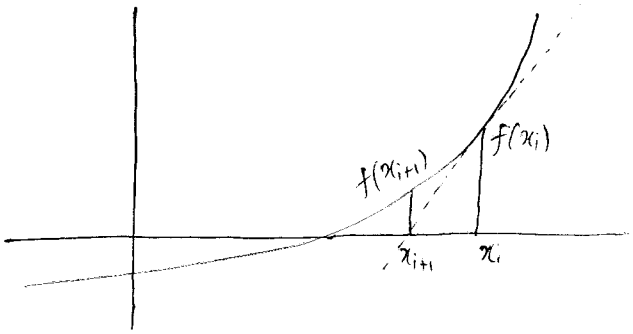
Step 8: Guess $x = -8$ $\Rightarrow$ $f(x) = 5.2$

Step 9: Guess $x = -7$ $\Rightarrow$ $f(x) = -4.2$

Step 10: Guess $x = -7.5$ $\Rightarrow$ $f(x) = 0.25$ (approx. root)

∴ The approx roots are: $x = 1.9$ & $x = -7.5$

Q.2 Newton-Raphson method is based on the principle that if the initial guess of the root of $f(x) = 0$ is at x_i, then if one draws the tangent to the curve at $f(x_i)$, the point x_{i+1} where the tangent crosses the x-axis is an improved estimate of the root.



The Newton Raphson formula is given by.

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

Convergence of Newton Raphson method :-

Let x_n be an estimate root of the given function $f(x)$. If x_n & x_{n+1} are close to each other, Then using Taylor's series expansion,

$$f(x_{n+1}) = f(x_n) + f'(x_n)(x_{n+1} - x_n) + \frac{f''(R)}{2}(x_{n+1} - x_n)^2 \quad \text{---} (1)$$

where R lies in interval x_n to x_{n+1} & hyghes terms have been dropped

Since $x_{n+1} = x_r$ is approximate root, so,

$$f(x_{n+1}) = 0$$

Then eqn (1) becomes.

$$0 = f(x) + f'(x)(x_r - x_n) + \frac{f''(R)}{2}(x_r - x_n)^2 \quad \text{---} (2)$$

As we know, from Newton Raphson formula,

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

Rearranging terms, we get,

$$f(x_n) = f'(x_n)(x_n - x_{n+1}) \quad \text{---} (3)$$

from (2) & (3),

$$0 = f'(x_n)(x_r - x_{n+1}) + \frac{f''(R)}{2}(x_r - x_n)^2 \quad \text{---} (4)$$

Let $e_{n+1} = x_r - x_{n+1}$

& $e_n = x_r - x_n$

Then eqn (4) becomes,

$$0 = f'(x_n)\, e_{n+1} + \frac{f''(R)}{2} e_n^2$$

$$\therefore e_{n+1} = -\frac{f''(R)\, e_n^2}{2 f'(x_n)} \quad \text{---} (5)$$

Eq (5) shows that Newton Raphson method converges quadratically.

Sol^n :-

$$f(x) = e^x - 4x^2 = 0 \quad \& \quad f'(x) = e^x - 8x$$

Let us assume $x_0 = 0$ & $\varepsilon = 0.01$

Iteration 1:

$f(x) =$ $f(x_0) = f(0) = 1$

$f'(x_0) = f'(0) = 1$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0 - \frac{1}{1} = -1$$

$$\left|\frac{x_1 - x_0}{x_1}\right| = \left|\frac{-1-0}{-1}\right| = 1 > \varepsilon$$

Iteration 2:

$x_0 = x_1 = -1$

$f(x_0) = f(-1) = -3.63$

$f'(x_0) = f'(-1) = 8.36$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = -1 - \frac{-3.63}{8.36} = -0.56$$

$$\left|\frac{x_1 - x_0}{x_1}\right| = \left|\frac{-0.56+1}{-0.56}\right| = 0.77 > \varepsilon$$

Iteration 3:

$x_0 = x_1 = -0.56$

$f(x_0) = f(-0.56) = -0.68$

$f'(x_0) = f'(-0.56) = 5.05$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = -0.56 - \frac{-0.68}{5.05} = -0.425$$

$$\left|\frac{x_1 - x_0}{x_1}\right| = \left|\frac{-0.425 + 0.56}{-0.425}\right| = 0.31 > \varepsilon$$

Iteration 4 :-

$x_0 = x_1 = -0.425$

$f(x_0) = f(-0.425) = -0.068$

$f'(x_0) = f'(-0.425) = 4.053$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = -0.425 - \frac{-0.068}{4.053} = -0.408$$

$$\left|\frac{x_1 - x_0}{x_1}\right| = \left|\frac{-0.408 + 0.425}{-0.408}\right| = 0.04 > \varepsilon$$

Iteration 5 :-

$x_0 = x_1 = -0.408$

$f(x_0) = f(-0.408) = -0.00087$

$f'(x_0) = f'(-0.408) = 3.9289$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = -0.408 - \frac{-0.00087}{3.9289} = -0.4077$$

$$\left|\frac{x_1 - x_0}{x_1}\right| = \left|\frac{-0.4077 + 0.408}{-0.4077}\right| = 0.00054 < \varepsilon$$

$\therefore$ Root $= x_1 = -0.4077$

Q.3 Soln :-

$x = 0.6$, $f(0.6) = ?$

$x_0 = 0.5$	$f_0 = -0.693$
$x_1 = 0.7$	$f_1 = -0.357$
$x_2 = 0.8$	$f_2 = -0.223$

From 2nd order Lagrange's interpolation, we have,

$$P_2(x) = f_0 l_0(x) + f_1 l_1(x) + f_2 l_2(x)$$

Now,

$$l_0(x) = \left(\frac{x - x_1}{x_0 - x_1}\right)\left(\frac{x - x_2}{x_0 - x_2}\right)$$

or $$l_0(0.6) = \left(\frac{0.6 - 0.7}{0.5 - 0.7}\right)\left(\frac{0.6 - 0.8}{0.5 - 0.8}\right) = \left(\frac{-0.1}{-0.2}\right) \times \left(\frac{-0.2}{-0.3}\right) = 0.333$$

$$l_1(x) = \left(\frac{x - x_0}{x_1 - x_0}\right)\left(\frac{x - x_2}{x_1 - x_2}\right)$$

$$\text{or } l_1(0.6) = \left(\frac{0.6 - 0.5}{0.7 - 0.5}\right)\left(\frac{0.6 - 0.8}{0.7 - 0.8}\right) = \left(\frac{0.1}{0.2}\right) \times \left(\frac{-0.2}{-0.1}\right) = 1$$

$$l_2(x) = \left(\frac{x - x_0}{x_2 - x_0}\right)\left(\frac{x - x_1}{x_2 - x_1}\right)$$

$$\text{or } l_2(0.6) = \left(\frac{0.6 - 0.5}{0.8 - 0.5}\right)\left(\frac{0.6 - 0.7}{0.8 - 0.7}\right) = \left(\frac{0.1}{0.3}\right) \times \left(\frac{-0.1}{0.1}\right) = -0.333$$

$$\therefore P_2(0.6) = f_0 l_0(0.6) + f_1 l_1(0.6) + f_2 l_2(0.6)$$

$$= -0.693 \times 0.333 - 0.357 \times 1 - 0.223 \times (-0.333)$$

$$= -0.231 - 0.357 + 0.074$$

$$= -0.513$$

Q.4 Soln:-

$x = 2.25$ $\quad f(2.25) = ?$

$x_0 = 1.4$ $\quad f_0 = 5.976$

$x_1 = 2.2$ $\quad f_1 = 28.972$

$x_2 = 3.0$ $\quad f_2 = 79$

Now,

From 2nd order Newton's divided difference interpolation, we have,

$$P_2(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1)$$

Now,

$$a_0 = f[x_0] = f_0 = 5.976$$

$$a_1 = f[x_1, x_0] = \frac{f_1 - f_0}{x_1 - x_0} = \frac{28.972 - 5.976}{2.2 - 1.4} = 28.745$$

$$a_2 = f[x_2, x_1, x_0] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0}$$

$$f[x_2, x_1] = \frac{f_2 - f_1}{x_2 - x_1} = \frac{79 - 28.972}{3 - 2.2} = 62.535$$

$$\therefore a_2 = \frac{62.535 - 28.745}{3 - 1.4} = 21.118$$

$\therefore P_2(2.25) = 5.976 + 28.745(2.25 - 1.4) + 21.118(2.25 - 1.4)(2.25 - 2.2)$

$= 5.976 + 24.433 + 0.897$

$= 31.306$

Q.5 Soln :-

Let $\varepsilon = 0.01$

$$\left.\begin{aligned} x_1 &= \frac{12 - x_2 - x_3}{10} \\ x_2 &= \frac{10 - x_1 + x_3}{10} \\ x_3 &= \frac{9 - x_1 + 2x_2}{10} \end{aligned}\right\} \quad ---(1)$$

Then from eqn (1)

Step 1: $x_1^{(1)} = \frac{12}{10} = 1.2$

$x_2^{(1)} = \frac{10 - 1.2 + 0}{10} = 0.88$

$x_3^{(1)} = \frac{9 - 1.2 + 2 \times 0.88}{10} = 0.956$

Step 2: $x_1^{(2)} = \frac{12 - 0.88 - 0.956}{10} = 1.016$

$x_2^{(2)} = \frac{10 - 1.016 + 0.956}{10} = 0.993$

$x_3^{(2)} = \frac{9 - 1.016 + 2 \times 0.993}{10} = 0.997$

Step 3: $x_1^{(3)} = \frac{12 - 0.993 - 0.997}{10} = 1.001$

$x_2^{(3)} = \frac{10 - 1.001 + 0.997}{10} = 0.999$

$x_3^{(3)} = \frac{9 - 1.001 + 2 \times 0.999}{10} = 0.999$

Now,

$$\left|\frac{x_1^{(3)} - x_1^{(2)}}{x_1^{(3)}}\right| = \left|\frac{1.001 - 1.016}{1.001}\right| = 0.01 \leq \varepsilon$$

$$\left|\frac{x_2^{(3)} - x_2^{(2)}}{x_2^{(3)}}\right| = \left|\frac{0.999 - 0.993}{0.999}\right| = 0.006 < \varepsilon$$

$$\left|\frac{x_3^{(3)} - x_3^{(2)}}{x_3^{(3)}}\right| = \left|\frac{0.999 - 0.997}{0.999}\right| = 0.002 < \varepsilon$$

$\therefore x_1 = 1.001 \approx 1$

$x_2 = 0.999 \approx 1$

$x_3 = 0.999 \approx 1$

Algorithm for Gauss-Seidel method :-

1. Read dimension of system of equations, say n
2. Read coefficients of matrix row-wise
3. Read elements of RHS vector
4. Read accuracy limit, say 0.001
5. for i = 1 to n
   new_x[i] = 0
   end for
6. for i = 1 to n
   sum = b[i]
   for j = 1 to n
   if (i ≠ j)
   sum = sum - a[i][j] * new_x[j]
   end for
   old_x[i] = new_x[i]
   new_x[i] = sum / a[i][i]
   $\varepsilon[i] = \left|\frac{new\_x[i] - old\_x[i]}{new\_x[i]}\right|$
   End for
7. for i = 1 to n
   if (ε[i] > 0.001)

goto step 6
end for
8. Display results in "new_x" vector

Q.6 Soln:-

$y' = (x^3 + xy^2)e^{(-x)}$; $y(0) = 1$ (ie $y = 1$ at $x = 0$)

$y(0.1) = ?$, $y(0.2) = ?$, $y(0.3) = ?$

Here,

$y' = (x^3 + xy^2)e^{(-x)}$

$y'' = \frac{dy'}{dx} = \frac{d}{dx}\left[(x^3 + xy^2)e^{-x}\right]$

$= (x^3 + xy^2)(-e^{-x}) + e^{-x}(3x^2 + x.2yy' + y^2.1)$

$= -e^{-x}(x^3 + xy^2) + e^{-x}(3x^2 + 2xyy' + y^2)$

$= e^{-x}(3x^2 + 2xyy' + y^2 - x^3 - xy^2)$

$= e^{-x}(3x^2 - x^3 + y^2 + 2xyy' - xy^2)$

$y''' = \frac{dy''}{dx} = \frac{d}{dx}\left[e^{-x}(3x^2 - x^3 + y^2 + 2xyy' - xy^2)\right]$

$= e^{-x}\frac{d}{dx}(3x^2 - x^3 + y^2 + 2xyy' - xy^2) + (3x^2 - x^3 + y^2 + 2xyy' - xy^2)(-e^{-x})$

$= e^{-x}\left[6x - 3x^2 + 2yy' + 2\{x(yy'' + y'^2) + yy'\}\right] - e^{-x}(3x^2 - x^3 + y^2 + 2xyy' - xy^2)$

$= e^{-x}\left[6x - 3x^2 + 2yy' + 2x(yy'' + y'^2) + yy' - 3x^2 + x^3 - y^2 - 2xyy' + xy^2\right]$

$= e^{-x}\left[x^3 - 6x^2 + 6x + 3yy' + 2x(yy'' + y'^2) - y^2 - 2xyy' + xy^2\right]$

Now, at $x = 0$ & $y = 1$,

$y' = 1$

$y'' = 1(0 - 0 + 1 + 0 - 0) = 1$

$y''' = 1(3 \times 1 \times 1 - 1^2) = 2$ } ---(1)

Now, The Taylor series is:

$y(x) = y(x_0) + (x - x_0)y'(x_0) + (x - x_0)^2\frac{y''(x_0)}{2!} + (x - x_0)^3\frac{y'''(x_0)}{3!}$ [neglecting higher terms]

---(2)

Using the values in eqn ① in eqn ②, we get

$$y(x) = 1 + (x-0).1 + (x-0)^2 . \frac{1}{2} + (x-0)^3 . \frac{2}{6.3}$$

or, $$y(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{3}$$

Now,

$$y(0.1) = 1 + 0.1 + \frac{0.1^2}{2} + \frac{0.1^3}{3} = 1.105$$

$$y(0.2) = 1 + 0.2 + \frac{0.2^2}{2} + \frac{0.2^3}{3} = 1.222$$

$$y(0.3) = 1 + 0.3 + \frac{0.3^2}{2} + \frac{0.3^3}{3} = 1.354$$

Q.7 Algorithm to obtain the solution of differential equation using Runge-Kutta method:-

1) Read initial values of x & y, say x_0 & y_0
2) Read the value at which functional value is required, say xp.
3) Read step size, say h
4) Set $y = x_0$, $y = y_0$
5) for $x = x_0$ to xp

```
m1 = f(x,y);
m2 = f(x + h/2, y + h/2 * m1);
m3 = f(x + h/2, y + h/2 * m2);
m4 = f(x + h, y + h * m3);
y = y + h/6 * (m1 + 2*m2 + 2*m3 + m4);
```

6) End for
7) Display functional value, y.

C-program:-

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define f(x,y) 2*(x)+(y)
```

```
int main()
{
    float x, xp, x0, y0, y, h, m1, m2, m3, m4;
    printf ("Enter initial values of x & y: \n");
    scanf("%f %f", &x0, &y0);
    printf ("Enter x at which function to be evaluated: \n");
    scanf("%f", &xp);
    printf ("Enter the step size: \n");
    scanf ("%f", &h);
    y = y0;
    x = x0;
    for (x = x0; x < xp; x = x+h)
    {
        m1 = f(x, y);
        m2 = f(x + 1/2.0 * h, y + 1/2.0 * h * m1);
        m3 = f(x + 1/2.0 * h, y + 1/2.0 * h * m2);
        m4 = f(x+h, y + h * m3);
        y = y + h/6 * (m1 + 2*m2 + 2*m3 + m4);
    }
    printf ("Function value at x = %f is %f \n", xp, y);
    getch();
    return 0;
}
```

Output -

Enter initial values of x & y:

0 1

Enter x at which function to be evaluated:

0.4

Enter the step size:

0.1

Function value at x = 0.4 is 1.075473