List of topics for Numerical Methods Practical:

1. Bisection Method
2. Newton-Raphson Method
3. Secant Method
4. Gauss Jordan Method
5. Gauss Elimination Method
6. Matrix Inverse using Gauss Jordan Method
7. Power Method
8. Lagrange Interpolation
9. Least Squares Method of Curve Fitting:
    a. Linear form
    b. Exponential Form
    c. Quadratic Form / Polynomial (any degree)
10. Integration from discrete data using Trapezoidal Rule
11. Integration of a function using Simpson's 3/8 rule
12. Solution of 1st order ODE (IVP) using:
    a. Euler's method
    b. RK2 method
    c. RK4 method
    d. Simultaneous ODEs using RK4 method

Sections to be included in Report

- Theory / Principle / Background / Procedure
- Algorithm / Pseudocode
- Program Code
- Input / Output (at least 3 sets – should not be same with others)

Sample Algorithms / Pseudo-codes

Bisection Method

1. **Define** the non-linear equation as $f(x)$
2. **Input:**
    Initial interval (a, b)
    Error tolerance (E)
3. **If** $f(a) \times f(b) > 0$: Terminate with error message
5. **Repeat:**
    $c \leftarrow (a + b) / 2$
    If $f(c) \times f(a) > 0$
        $a \leftarrow c$
    Else
        $b \leftarrow c$
    End If
    While $|f(c)| > E$
6. **Print** root (c)
7. Stop

Least Square Method of Curve Fitting: Linear Form (y = a + bx)

1. Input number of data (n)
2. Initialize with 0:
   $sumX$, $sumX2$, $sumY$, $sumXY$
3. Compute the required sums ($\sum x, \sum x^2, \sum y, \sum xy$):
   For i = 1 to n
      Input x, y
      $sumX \leftarrow sumX + x$
      $sumX2 \leftarrow sumX2 + (x \cdot x)$
      $sumY \leftarrow sumY + y$
      $sumXY \leftarrow sumXY + (x \cdot y)$
   Next i
4. Compute the required constants of $y = a + bx$:
   $D1 \leftarrow (sumY \cdot sumX2 - sumXY \cdot sumX)$
   $D2 \leftarrow (n \cdot sumXY - sumX \cdot sumY)$
   $D3 \leftarrow (n \cdot sumX2 - sumX \cdot sumX)$
   $a \leftarrow D1/D3$
   $b \leftarrow D2/D3$
5. Print a and b
6. Stop

Gauss Jordan Method

1. Input number of unknowns (n)
2. Input the Augmented Coefficient Matrix (A)
3. Reduce matrix A to diagonal form:
   For j = 0 to n-1
      If $|A_{j,j}| < 0.00005$: Terminate with error message
      For i = 0 to n-1
         If $j \neq i$ then
            $Ratio \leftarrow A_{i,j}/A_{j,j}$
            For $k = j$ to n
               $A_{i,k} \leftarrow A_{i,k} - Ratio \times A_{j,k}$
            Next k
         End If
      Next i
   Next j
4. Obtain Solution Vector:
   For i = 0 to (n-1):
      $X_i \leftarrow A_{i,n}/A_{i,i}$
5. Output Solution Vector (X)
6. Stop

Power Method

1. Input:

    Order of square matrix (n)

    Square matrix of order n (A)

2. Initialize:

    Guess values to Eigen Vector ($X = [1, 1, ...]$):

    Error Tolerance ($E = 0.0005$)

    Iteration Threshold ($MAXITR = 100$)

4. Increment Count and check for threshold limit:

    $Count = Count + 1$

    If $Count > MAXITR$: Terminate with error message

5. Compute $Z \leftarrow A * X$:

```
For i = 0 to (n-1)
    Z_i ← 0
    For j = 0 to (n-1)
        Z_i ← Z_i + (A_{i,j} * X_j)
    Next j
Next i
```

5. Compute largest absolute from Z ($lambda$):

```
lambda = |Z_0|
For i = 1 to (n-1)
    If |Z_i| > lambda then lambda ← |Z_i|
Next i
```

6. Scale Z by $lambda$ ($Z \leftarrow Z/lambda$):

```
For i = 0 to (n-1)
    Z_i ← Z_i/lambda
Next i
```

7. Compute Error between Z and X ($D \leftarrow | Z - X |$):

```
For i = 0 to (n-1)
    D_i ← | Z_i - X_i |
Next i
```

8. Replace X by Z:

```
For i = 0 to (n-1)
    X_i ← Z_i
Next i
```

9. Compute largest error ($Dmax$):

```
Dmax ← D_0
For i = 1 to (n-1)
    If D_i > Dmax then Dmax ← D_i
Next i
```

10. If $Dmax > E$, then repeat from step 4.

11. Convert the Eigen Vector to normal form:

    $norm \leftarrow \sum_{i=0}^{n-1} X_i^2$

    $X \leftarrow X/\sqrt{norm}$

12. Output results:

    Print $lambda$ (the dominant Eigen Value)

    Print X (the Eigen Vector):

13. Stop

Integration from discrete data using Trapezoidal Rule

1. Input:
   Number of strips (n) [no. of data=n+1]
   Limits of integration (a, b)
   Y ordinates (Y_i for $i = 0$ to n)
2. Compute strip size:
   $h \leftarrow (b-a)/n$
3. Initialize: $sum \leftarrow Y_0 + Y_n$
4. For $i = 1$ to n-1
   $sum \leftarrow sum + 2 \times Y_i$
   Next i
5. $sum \leftarrow sum \times h/2$
6. Print sum
7. Stop

Integration of a given function using Simpson's 3/8 Rule

1. Define the integral as $f(x)$
2. Input:
   Limits of integration (a, b)
   Number of strips (n)
3. If (n mod $\mathbf{3} \neq 0$) then Output error message and terminate
4. Compute strip size: $h \leftarrow (b-a)/n$
5. Initialize: $sum \leftarrow f(a) + f(b)$
6. For $i = 1$ to n-1
   If (i mod $3 = 0$) $m \leftarrow 2$ else $m \leftarrow 3$
   $x \leftarrow a + i \cdot h$
   $sum \leftarrow sum + m \times f(x)$
   Next i
6. $sum \leftarrow sum \times 3h/8$
7. Print sum
8. Stop

Solution of 1st order ODE using RK4 / RK2/ Euler's Method

1. Define y' as $f(x, y)$
2. Input:
   Initial Conditions $(x_0, \ y_0)$
   Final value of x (x_n)
   Step size (h)
3. Determine step size: $h = (x_n - x_0)/n$
3. Initialize:
   $x \leftarrow x_0 \qquad y \leftarrow y_0$
4. For i=1 to n

$$k_1 \leftarrow h \cdot f(x, y)$$
$$k_2 \leftarrow h \cdot f\left(x + \frac{h}{2}, \quad y + \frac{k_1}{2}\right)$$
$$k_3 \leftarrow h \cdot f\left(x + \frac{h}{2}, \quad y + \frac{k_2}{2}\right)$$
$$k_4 \leftarrow h \cdot f(x + h, \ y + k_3)$$
$$k \leftarrow \frac{(k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4)}{6}$$

For RK2 method:

$$k_1 \leftarrow h \cdot f(x, y)$$
$$k_2 \leftarrow h \cdot f(x + h, y + k_1)$$
$$k \leftarrow \frac{k_1 + k_2}{2}$$

For Euler's Method:

$$k \leftarrow h \cdot f(x, y)$$

   $y \leftarrow y + k$
   $x \leftarrow x + h$
   Print (x, y)
   Next i
5. Stop