Assignment no 1 on Numerical method Lab

1. Develop a C program to implement Bisection Method. Test your program to find a root of the following equation:

$$xsinx + cosx = 0 \quad x\ in\ (0, \pi)$$

Example 1 Find a root of cos(x) - x * exp(x) = 0
Example 2 Find a root of x^4-x-10 = 0
Example 3 Find a root of x-exp(-x) = 0
Example 4 Find a root of exp(-x) * (x^2-5x+2) + 1= 0
Example 5 Find a root of x-sin(x)-(1/2)= 0
Example 6 Find a root of exp(-x) = 3log(x)

The solution needs to be correct up to the 6^{th} place after the decimal point. Display the output in a tabular form with the following information.

i (iteration count)	a (lower bound of interval)	b (upper bound)	m (mid point)	f(m)	Abs. error	Order of convergence

Modify the above program to implement the method of False Position / Regula Falsi on the same equation.

Make a comparative assessment of the two methods on the basis of the results you obtain.

2. Write a C program to implement Fixed Point Iteration. Apply the method on the following:

$$e^x - 4x^2 = 0 \quad x\ in\ (4,5)$$

$$x = \mp\sqrt{e^x/4}$$

Keep check on whether the condition for convergence is satisfied through your program. Display the output in a tabular form with the following information.

i	x_i	$\lvert g'(x_i)\rvert$	$f(x_i)$	Abs. error	Order of convergence

3. Develop a C program to implement Newton-Raphson method. Test your program on the following:

$$e^x = 2x + 1$$

The solution needs to be correct up to 4th place after the decimal point. Display the result in the following form:

i	x_i	$f(x_i)$	Abs. error	Order of convergence

4. Develop a C program to implement Secant method. Test the program on the following problem:

$$\frac{|f(x) * f''(x)|}{\{f'(x)\}^2} < 1$$

5. The function $x^3 - 2x^2 - 4x + 8$ has double root at x=2. How would you change your program to apply Newton-Raphson method to solve the above equation at x=2. The solution needs to be correct up to 3 decimal places. Take an initial guess of $x_0 = 1.2$.

6. Develop a C program to implement Gaussian Elimination method. Test your program on the following system of linear equations:

$$\begin{aligned} x_1 + x_2 - x_3 + x_4 &= 2 \\ 2x_1 + x_2 + x_3 - 3x_4 &= 1 \\ 3x_1 - x_2 - x_3 + x_4 &= 2 \\ 5x_1 + x_2 + 3x_3 - 2x_4 &= 7 \end{aligned}$$

After elimination of each variable, display the augmented coefficient matrix. Incorporate pivoting technique in the program.

7. Develop a C program to implement Gauss-Seidel iterative method. Test your program on the following:

$$\begin{aligned} 5x_1 - x_2 + x_3 &= 10 \\ 2x_1 + 8x_2 - x_3 &= 11 \\ -x_1 + x_2 + 4x_3 &= 3 \end{aligned}$$

The solution needs to correct up to the 3rd place after the decimal point. Display the output in the following tabular form:

K (no. of iteration)	x_1	x_2	x_3	$max_i\{x_i^k - x_i^{k-1}\}$

$$x_1 = 2.000005, x_2 = 0.999999, x_3 = 1.000002, Abs.error = 0.000032$$

8. Develop a C program to find the inverse of a non-singular matrix by Gauss-Jordan elimination method. Arrange for verification of the product of the matrix and its inverse. Test your program on the following:

$$\begin{matrix} 1 & 5 & 3 \\ 1 & 3 & 2 \\ 2 & 4 & -6 \end{matrix}$$

9. Develop a C program to implement Euler's method for solution of first order differential equations. Test your program on the following:

$$\frac{dy}{dx} = 2x^2 + 2y \; with \; y(0) = 1$$

Solution is required over the interval $0 \leq x \leq 1$ *with* $h = 0.1$
The exact solution is $y = 1.5e^{2x} - x^2 - x - 0.5$

Produce the output of your program in the following format:

x	*y computed*	*y actual*	*Abs. error*

Modify your program to implement the Modified Euler's method. Produce the output to compare the performances of the Euler's method, Modified Euler's method and the actual solution.

10. Develop a C program to implement Trapezoidal rule for numerical integration. In each iteration, the program computes the integral by doubling the number of intervals considered in the last iteration. The program terminates after achieving the desired precision. Test yur program on the following:

$$\int_0^1 {}^1/_{(1+x)}\, dx$$

The computed value of the integral needs to be correct up to the 4th place after the decimal point in the absolute error.

11. Develop a C program to implement Simpson's 1/3rd rule. Test your program on the above problem.