Numerical Methods and Programming
Assignment-1

- Problem 1: Machine precision is defined as the largest number ϵ such that $1+\epsilon = 1$. Write a C program to calculate ϵ. Calculate it for both 'float' and 'double' data types.

- Problem 2: a) Write a C program to evaluate the function e^{-x} at x = 0.5 by summing the series:

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

  Sum the series until $|\frac{term}{sum}| < 10^{-8}$.
  b) Calculate the relative error by comparing with the exact value obtained from in-built library function 'exp'.
  c) Plot (log-log) relative error versus number of terms kept in the series.

- Problem 3: We know that roots of a quadratic equation of the form $ax^2 + bx + c = 0$ are $\frac{-b\pm\sqrt{b^2-4ac}}{2a}$
  Write a C program to evaluate the roots of such a quadratic equation with

  $$a = 1, b = 3000.001, c = 3$$

  Actual roots are $x_1 = -0.001$ and $x_2 = -3000$. Use single precision ('float') to represent real numbers.

- Problem 4: The n^{th} power of the number $\phi = \frac{\sqrt{5}-1}{2} \approx 0.61803398$ obeys a recursion relation

  $$\phi^{n+1} = \phi^{n-1} - \phi^n$$

  Starting with $\phi^0 = 1$ and $\phi^1 = 0.61803398$ evaluate the values upto $n = 20$ using the above recursion relation. Compare your results with ϕ^n evaluated using the in-built 'pow' function. Plot the relative error as a function of n.

- Problem 5: If divided differences are defined as

  $$f(x, x_0) = \frac{y - y_0}{x - x_0}$$
  $$f(x, x_0, x_1) = \frac{f(x, x_0) - f(x_0, x_1)}{x - x_1}$$
  $$f(x, x_0, x_1, x_2) = \frac{f(x, x_0, x_1) - f(x_0, x_1, x_2)}{x - x_2}$$

  a) prove that divided differences has the following symmetries with respect to exchange of arguments

  $$f(x_1, x_0) = f(x_0, x_1)$$
  $$f(x_2, x_1, x_0) = f(x_2, x_0, x_1)$$
  $$f(x_2, x_1, x_0) = f(x_1, x_2, x_0)$$
  $$f(x_3, x_2, x_1, x_0) = f(x_2, x_3, x_1, x_0)$$

b) show that y can be written as

$$\begin{aligned} y = y_0 &+ (x - x_0)f(x_0, x_1) \\ &+ (x - x_0)(x - x_1)f(x_0, x_1, x_2) \\ &+ (x - x_0)(x - x_1)(x - x_2)f(x_0, x_1, x_2, x_3) \end{aligned}$$

further by mathematical induction you can prove that **Newton's general interpolation formula** with divided differences is

$$\begin{aligned} y = y_0 &+ (x - x_0)f(x_0, x_1) \\ &+ (x - x_0)(x - x_1)f(x_0, x_1, x_2) \\ &+ (x - x_0)(x - x_1)(x - x_2)f(x_0, x_1, x_2, x_3) \\ &+ ... \\ &+ (x - x_0)(x - x_1)...(x - x_{n-1})f(x_0, x_1, x_2...x_{n-1}) \\ &+ (x - x_0)(x - x_1)...(x - x_n)f(x_0, x_1, x_2...x_n) \end{aligned} \tag{1}$$

- Problem 6: Write a C program to perform Newton's interpolation (Eq.1) and find $\log_{10} 323.5$ using the following table.

x	$\log_{10} x$
321.0	2.50651
322.0	2.50893
324.2	2.51081
325.0	2.51188

- Problem 7: If $y = P(x)$ is a polynomial of n^{th} degree which takes the values $y_0, y_1, y_2, ...y_n$ when x has the values $x_0, x_1, x_2, ...x_n$ respectively and $(n + 1)^{th}$ divided differences of this polynomial is given as

$$\begin{aligned} f(x_0, x_1, x_2, ..., x_n) = &\frac{y}{(x - x_0)(x - x_1)...(x - x_n)} \\ &+ \frac{y_0}{(x_0 - x)(x_0 - x_1)...(x_0 - x_n)} \\ &+ \frac{y_1}{(x_1 - x)(x_1 - x_0)...(x_1 - x_n)} \\ &+ ... \\ &+ \frac{y_n}{(x_n - x)(x_n - x_0)(x_n - x_1)...(x_1 - x_{n-1})} \end{aligned}$$

show that y can be written as

$$\begin{aligned} y = &\frac{(x - x_1)(x - x_2)...(x - x_n)}{(x_0 - x_1)(x_0 - x_2)...(x_0 - x_n)} y_0 \\ &+ \frac{(x - x_0)(x - x_2)...(x - x_n)}{(x_1 - x_0)(x_1 - x_2)...(x_1 - x_n)} y_1 \\ &+ \frac{(x - x_0)(x - x_1)(x - x_3)...(x - x_n)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)...(x_2 - x_n)} y_2 \\ &+ ... \\ &+ \frac{(x - x_0)(x - x_1)(x - x_3)...(x - x_{n-1})}{(x_n - x_0)(x_n - x_1)(x_n - x_3)...(x_n - x_{n-1})} y_n \end{aligned} \tag{2}$$

which is **Lagrange's interpolation formula**.

Hint: *You can use the fact that n^{th} divided difference of a polynomial of degree n is constant. Hence $(n+1)^{th}$ divided difference of a polynomial of degree n is zero.*

- Problem 8: a) Implement a C function to perform Gauss Elimination
  b) Use it to solve

$$0.0003x_1 + 3.0000x_2 = 2.0001$$

$$1.0000x_1 + 1.0000x_2 = 1.0000$$

  Use 'float' datatype for all real numbers. The exact solution is $x_1 = 1/3, x_2 = 2/3$
  c) Solve the above problem with pivotting.

- Problem 9: a) Extend the Gauss Elimination function to calculate LU decomposition of a matrix.
  b) Using LU decomposition, solve the system of equations:

$$7x_1 + 2x_2 - 3x_3 = -12$$

$$2x_1 + 5x_2 - 3x_3 = -20$$

$$x_1 - x_2 - 6x_3 = -26$$

- Problem 10: a) Solve the following set of equations by LU decomposition without pivoting

$$x_1 + 7x_2 - 4x_3 = -51$$

$$4x_1 - 4x_2 + 9x_3 = 62$$

$$12x_1 - x_2 + 3x_3 = 8$$

  b) Determine the matrix inverse. Check whether $[A][A]^{-1} = [I]$