INTE 316 NUMERICAL ANALYSIS AND PROGRAMMING

CAT II

a) Use the algorithm below to solve $x^2 - x - 2 = 0$. Initial Guesses [1,3] Conduct three iterations

```
double regula (double c, int iterations, double tol)...
for ( int i = 0; i < iterations; i++)
{
    double x = ( fa*b - fb*a ) / ( fa - fb );
    double fx =  func( x, c );
    if ( fabs( fx ) < tol )
        return x;
    if ( fa * fx < 0 ) {
        b = x;
        fb = fx;   }
    else   {
        a = x;
        fa = fx;   }
}
return -1;
```

b) Using PYTHON show how the following is achieved **(PRACTICAL)**

i. Differentiation

ii. Numerical integration

iii. Curve Fitting

iv. Linear Regression

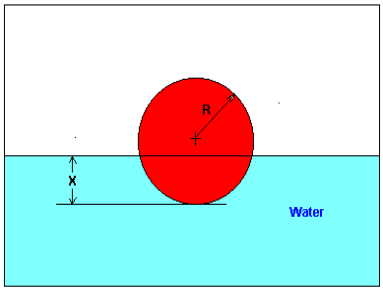
v. Spline Interpolation

c) A smart robotic agent with a laser Scanner is doing a quick quality check on holes drilled in a rectangular plate. The centers of the hole in the plate describes the path the arm needs to take, and the hole centers are located on a Cartesian coordinate system as shown

| **X (in)** | **2.00** | **4.25** | **5.25** | **7.81** | **9.20** | **10.60** |
|---|---|---|---|---|---|---|
| **Y(in)** | **7.2** | **7.1** | **6.0** | **5.0** | **3.5** | **5.0** |

If the laser scanner is traversing from **x=2.00 to x =4.25** in a linear path, what is the value of **y** at **x**= 4.0 using the linear spline formula , show how this problem can be solved using PYTHON (PRACTICAL)

d) You are working for 'DOWN THE TOILET COMPANY' that makes floats for ABC commodes. The floating ball has a specific gravity of 0.6 and has a radius of 5.5 cm. You are asked to find the depth to which the ball is submerged when floating in water.




The equation that gives the depth *x* in meters to which the ball is submerged under water is given by

$$f(x) = x^3 - 0.165x^2 + 3.993\times10^{-4}$$

Use the Newton's method of finding roots of equations to find

i. The depth 'x' to which the ball is submerged under water. Conduct three iterations to estimate the root of the above equation.

ii. The absolute relative approximate error at the end of each iteration

e)

Analyze the frequency components of a signal $s(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t)$ with $f_1 = 50$ Hz and $f_2 = 120$ Hz sampled at 1 kHz for 1 second.

Write a Python function to compute the Fast Fourier Transform (FFT) of the signal.

f) Explain the output of the following program

```
for n = 1:5
    x = n*0.1;
    z = myfunc2(x,2,3,7);
    fprintf('x = %4.2f  f(x) = %8.4f \r',x,z)
end
```

g) Write a program to show how the trapezoidal rule of integration works in PYTHON **(PRACTICAL)**

h) What is the output of the code below

```
x = [1 2 3 4 5 6];
y = [5.5 43.1 128 290.7 498.4 978.67];
p = polyfit(x,y,4)
x2 = 1:.1:6;
y2 = polyval(p,x2);
plot(x,y,'o',x2,y2)
grid on
```

I)

Given a set of data points $(x_i, y_i)$, you are to perform polynomial interpolation and approximation.

$$\begin{pmatrix} (1,1) \\ (2,4) \\ (3,9) \\ (4,16) \end{pmatrix}$$

**Questions:**

1. **Implementation**: Write a Python function to compute the coefficients of the Lagrange polynomial that interpolates the given data points.
2. **Implementation**: Write a Python function to implement Newton's Divided Difference method for polynomial interpolation.
3. **Algorithm Analysis**: Compare the interpolating polynomials obtained from Lagrange and Newton's methods. Discuss any differences.

J)

Find the eigenvalues and eigenvectors of the matrix:

$$\mathbf{A} = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 3 & -1 \\ 1 & -1 & 2 \end{pmatrix}$$

**Questions:**

1. **Implementation**: Write a Python function to compute the eigenvalues and eigenvectors of the matrix using the Power Iteration method.

2. **Implementation**: Write a Python function to compute the eigenvalues and eigenvectors using the QR Algorithm.

3. **Comparison**: Compare the eigenvalues and eigenvectors obtained from the Power Iteration method and the QR Algorithm. Discuss any differences.

K)

Write a Python function to implement the Gradient Descent method for minimizing the given function. Use a learning rate of 0.1 and initial guess $(x_0, y_0) = (0, 0)$.

$$f(x, y) = x^2 + y^2 - xy + x - y + 1.$$