**Question 1.**

**Briefly discuss the strengths and weaknesses of bisection and Newton-Raphson algorithms in terms of reliability and efficiency of computations.**

The bisection method is very reliable because it always converges as long as you choose an interval where the function changes sign. It’s straightforward and doesn’t require knowing the derivative. However, its rate of convergence is only linear, so it can take many iterations to achieve high accuracy.

In contrast, the Newton–Raphson method converges much faster (quadratic convergence) when you start with a good initial guess. Its efficiency makes it attractive for problems where speed is important. On the downside, it is more sensitive to the choice of the starting value and requires the derivative of the function; if the derivative is zero or very small, or the initial guess is poor, the method might fail to converge or may converge to an unintended root.

**Question 2**

The standard Newton–Raphson method struggles here because the derivative f′(x)=2(x−1)\*e^(−(x−1)^2) is zero at x=1. When the iteration gets close to 1, dividing by a value near zero causes instability or huge corrective steps, meaning the method may either converge very slowly or even fail. The Modified Newton–Raphson method, which multiplies the correction term by the known multiplicity (here, 2), compensates for the flatness of the function near the root. This modification leads to much smoother and rapid convergence toward x=1, as demonstrated in the code output.

**Question 4**

**NumPy part**  
Using NumPy we form the matrices for any chosen value of a. By calculating the determinant (which comes out as 112 − 7a^2), we see that the singular cases are a = 4 and a= −4 . For these values, we compute the rank of A (which drops below 3) and compare it with the rank of the augmented matrix [A|B]. This systematic approach shows not only when A is singular but also helps assess whether the system has infinite solutions or is inconsistent.  
  
**SumPy part**

In the Sympy solution the matrix A is defined symbolically, and the determinant is computed exactly as 112−7a^2. Solving det(A)=0 returns a= 4 and a = −4. For each singular value, we substitute into A and B and then determine the rank of A and the augmented matrix [A|B]. This symbolic approach not only confirms the numeric findings but also demonstrates the power of symbolic computation when dealing with parametric matrices.