**Experiment No.**: 01

**Experiment Name:** Introduction to Python and Error Analysis.

**Theory:** Error analysis helps us understand how close an approximate value is to the true or exact value. There are three common types of errors:

• Absolute Error (AE):

The difference between the true value and the approximate value.

**AE = | True Value – Approximate Value |**

• Relative Error (RE):

Measures the error in relation to the size of the true value.

**𝑅𝐸 = 𝐴𝐸 / | 𝑇𝑟𝑢𝑒 𝑉𝑎𝑙𝑢𝑒 |**

• Percentage Error (PE):

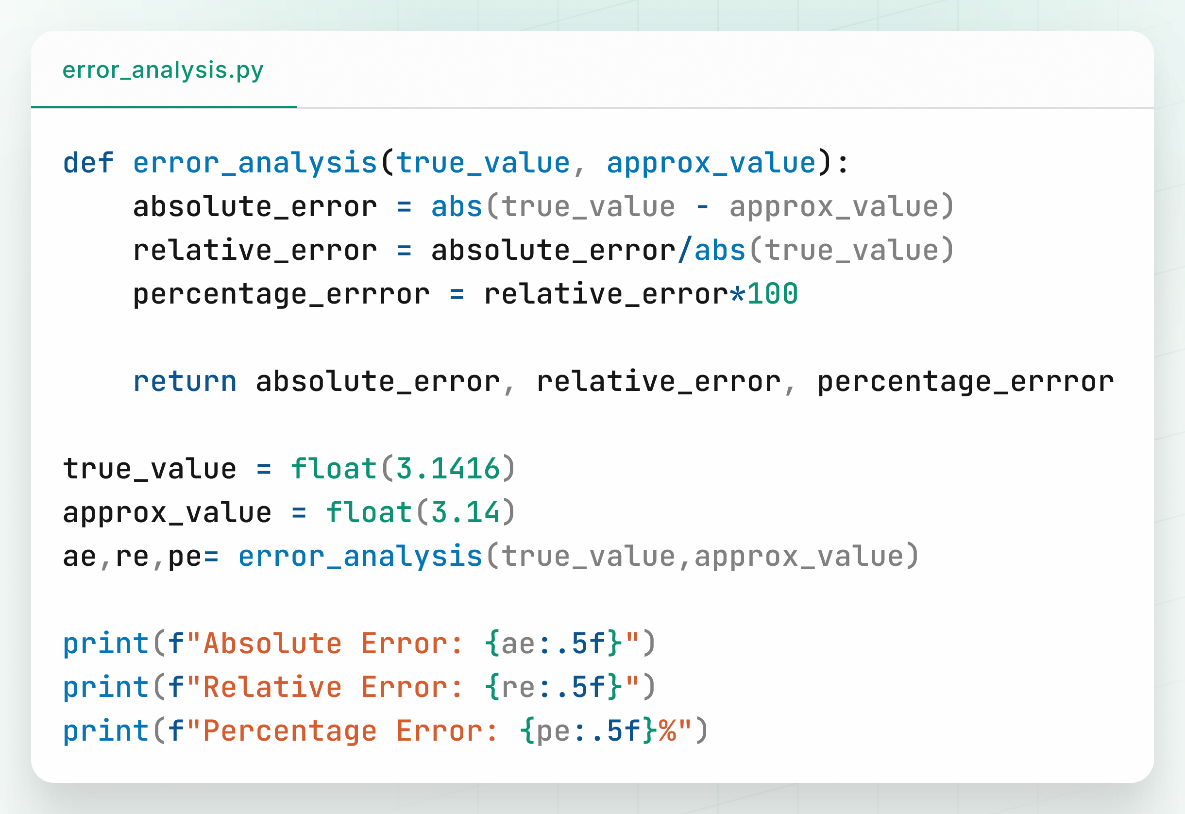
Represents the relative error as a percentage.

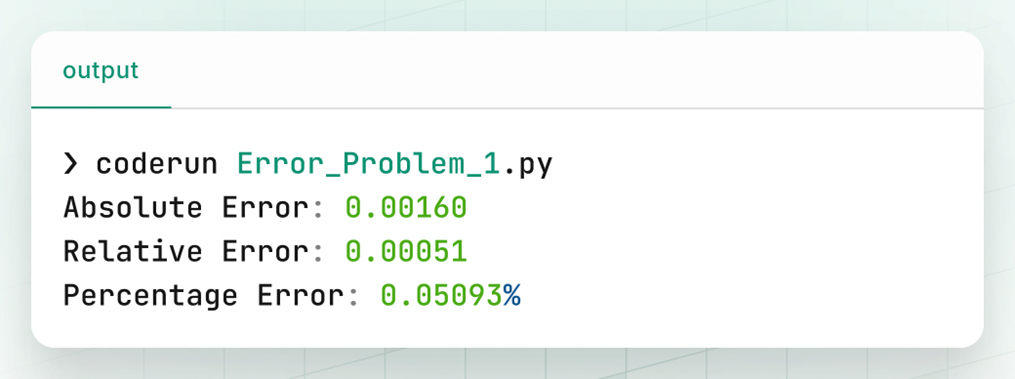
PE = RE × 100

These measures help in evaluating how accurate an approximation is and guide improvements in

numerical techniques.

**Program:**



**Output:**

**Discussion & Conclusion:**

In this experiment, we calculated the absolute, relative, and percentage errors between a true value (3.1416) and an approximate value (3.14). The results showed a small error, indicating the approximation is quite close to the true value. Understanding these errors helps in evaluating the quality of approximations and deciding if they are acceptable for practical use. This foundational knowledge is essential for numerical methods and scientific computations.

**Experiment:** 02

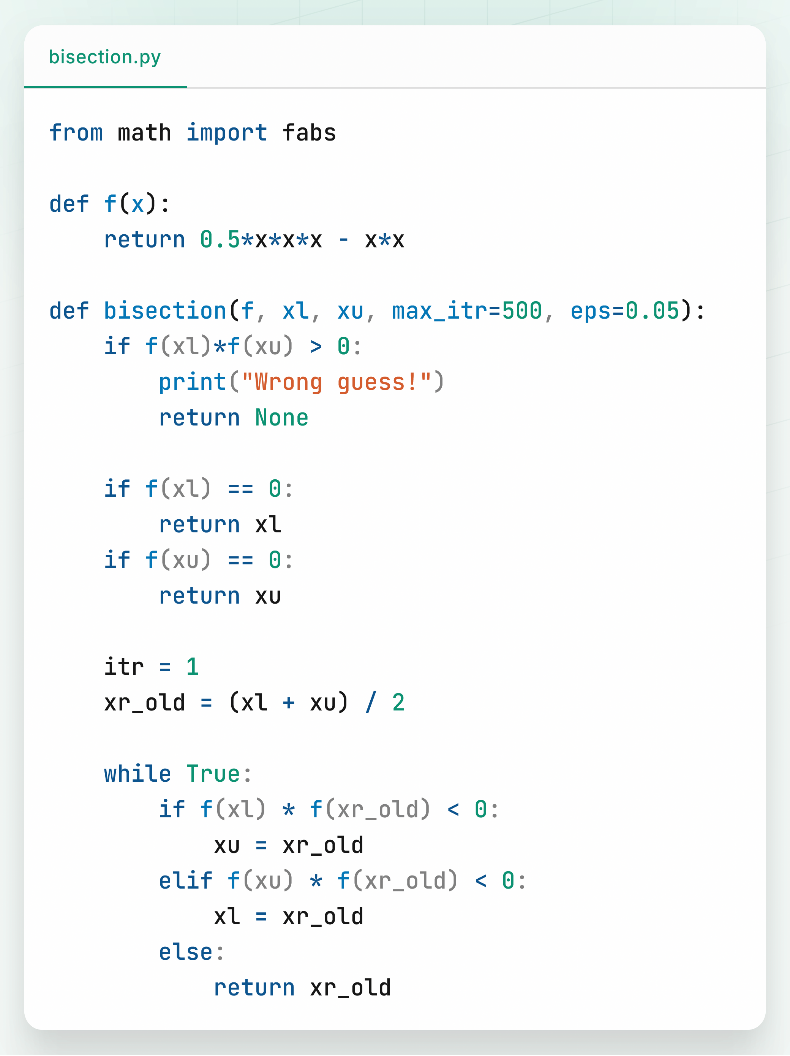
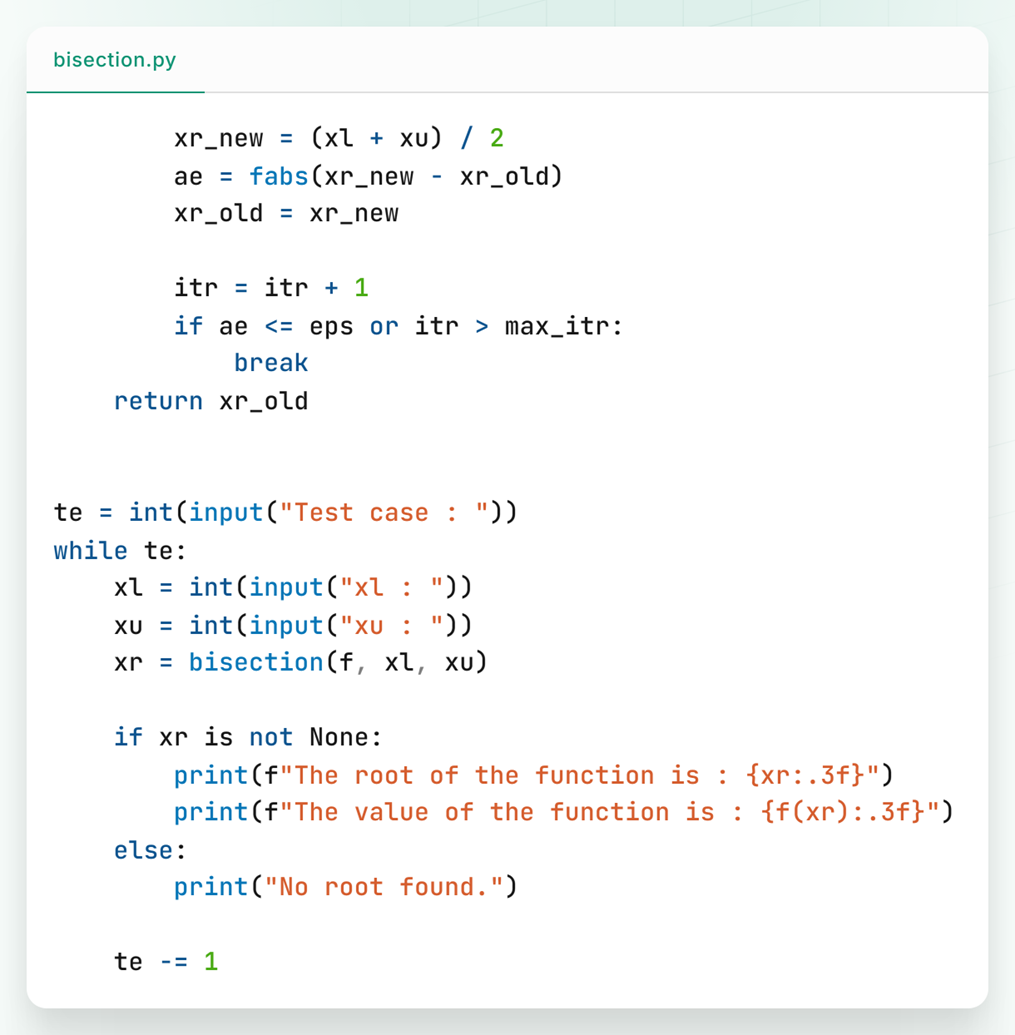
**Experiment Name:** Implementation of Bisection Method for Solving Non-Linear Equation

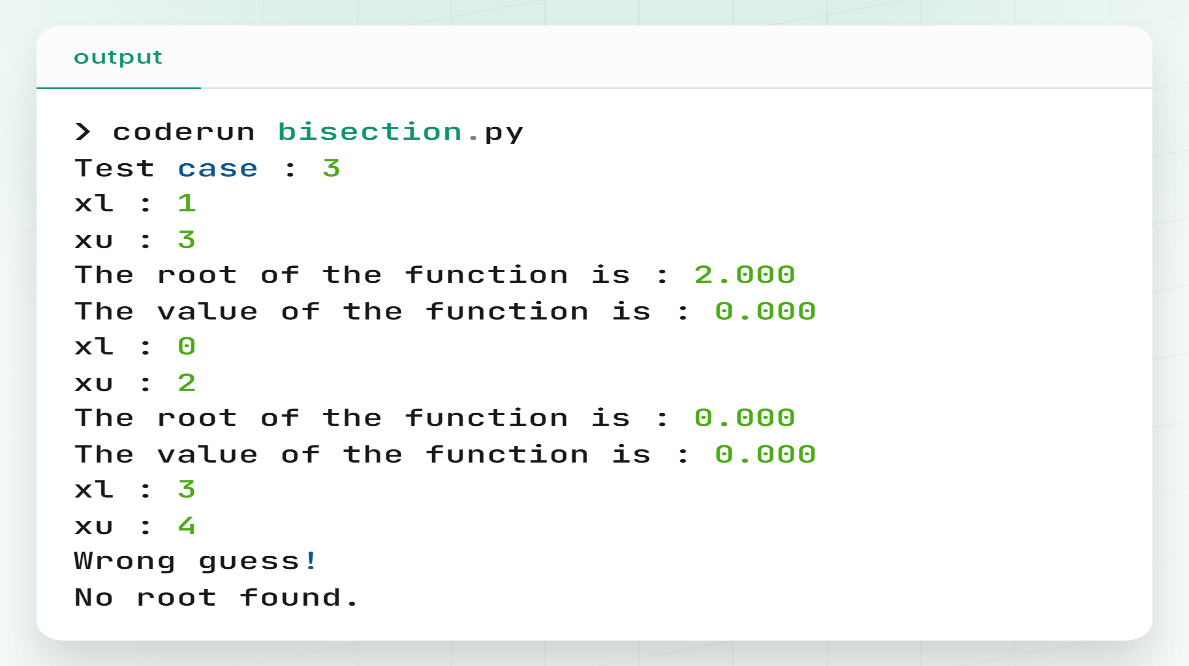
**Theory:**

The bisection method is a root-finding technique that repeatedly divides an interval in half and selects the subinterval where the function changes sign, ensuring the root lies within. It requires two initial guesses that bracket the root (function values at these points have opposite signs). The method converges by narrowing down the interval until the root is approximated within a desired tolerance. The midpoint of the interval is calculated using the formula:

X r = (𝑥𝑙+𝑥𝑢) / 2

The process continues until the approximate error is within a specified tolerance or a maximum number of iterations is reached. The method is simple, stable, and ensures convergence if the initial interval is correctly chosen.

**Program:**

**Output:**

**Discussion & Conclusion:**

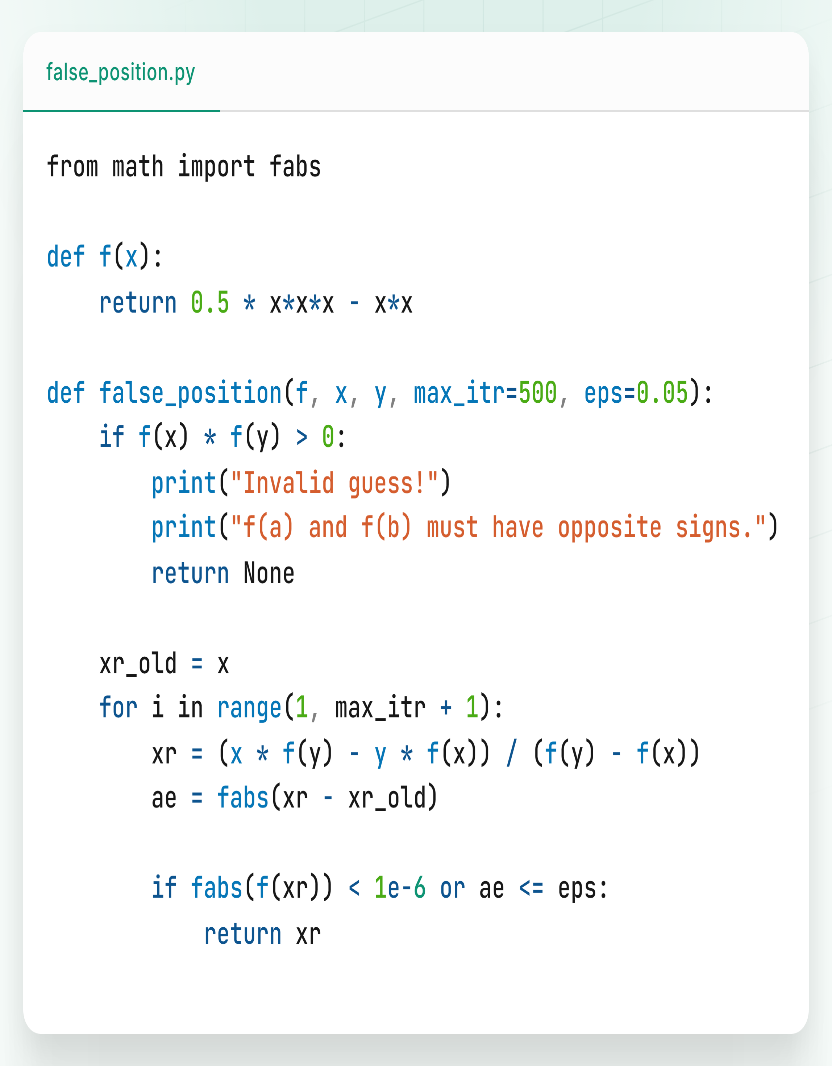
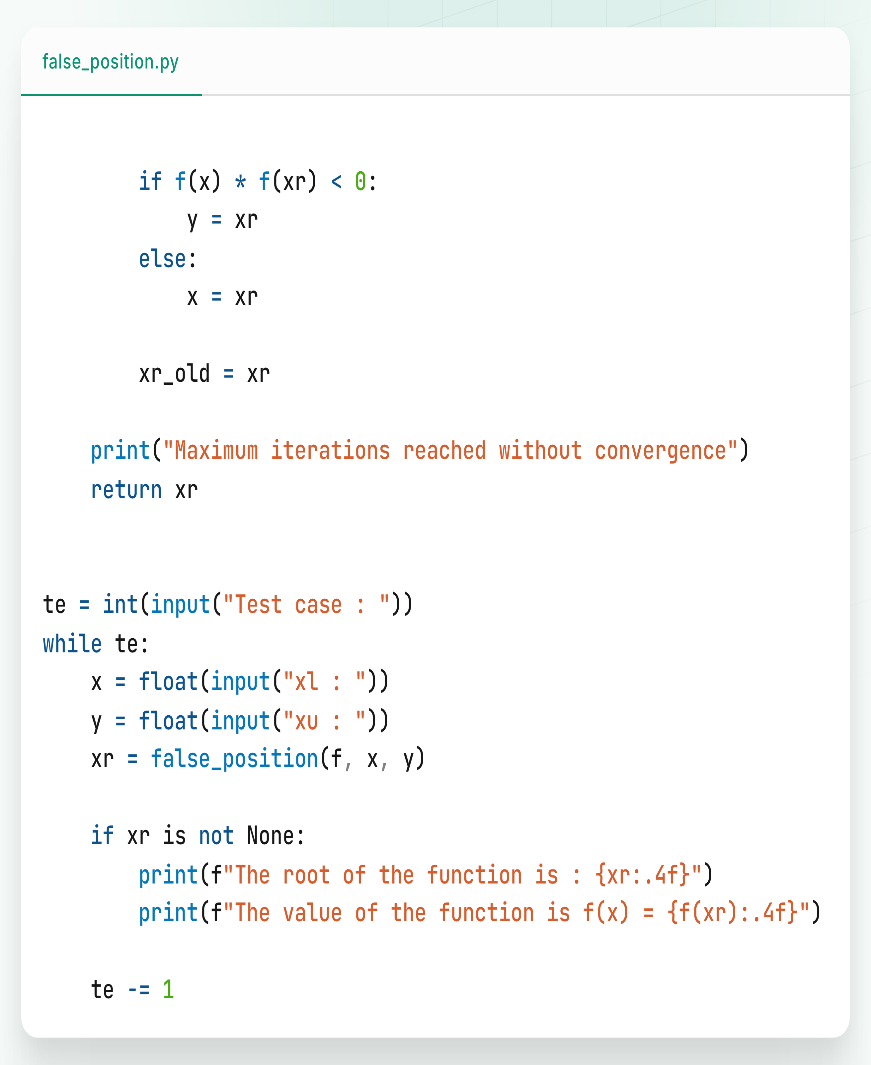
This experiment implemented the bisection method to find roots of the function ( f(x) = 0.5x^3 - x^2 ). By iteratively halving the interval and checking the sign of the function, the method successfully approximated the root. The method is simple, reliable, and guarantees convergence if the initial guesses bracket a root. However, it can be slower compared to other methods. This experiment demonstrated the practical application of the bisection method and its effectiveness in solving non-linear equations.

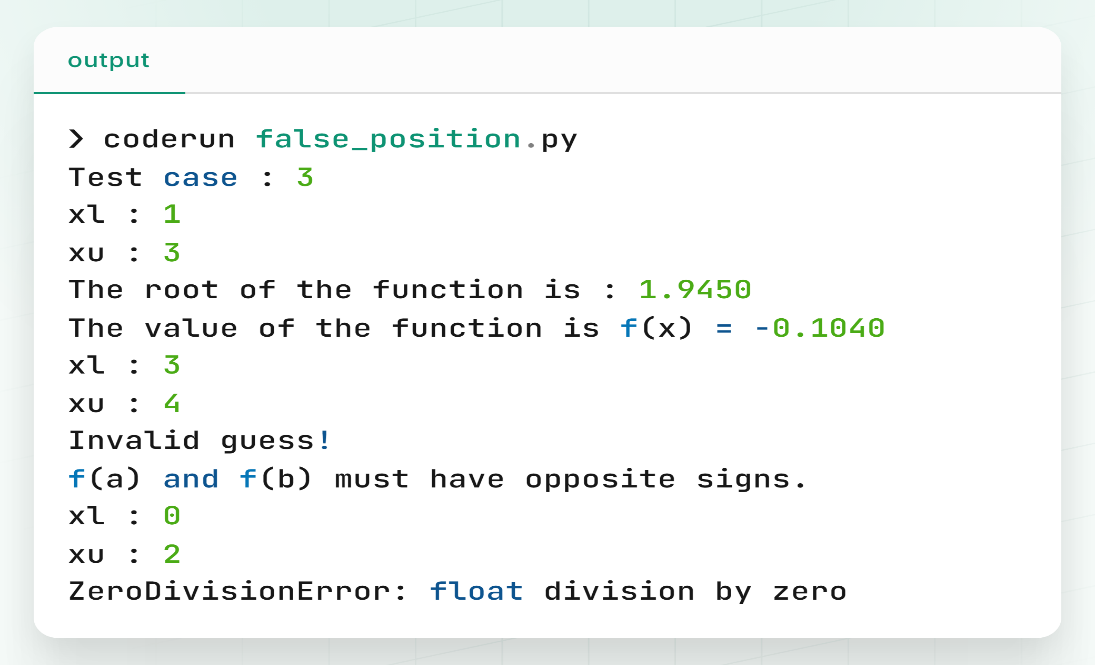
**Experiment No.:** 03

**Experiment Name:** Implementation of False Position Method for Solving Non-LinearEquations.

**Theory:**

The false position (or regula falsi) method is another root-finding technique that improves on the bisection method by using a secant line to approximate the root. It uses two points that bracket the root and calculates the root estimate by linear interpolation. Like bisection, it requires the function values at the initial points to have opposite signs. This method often converges faster than bisection but still guarantees convergence under proper conditions .

**Program:**

**Output:**

**Discussion & Conclusion:**

In this experiment, the false position method was implemented to find roots of the same function. By using linear interpolation between two points, the method provided a root approximation more efficiently than simple bisection. The experiment showed that false position can converge faster while maintaining reliability. It is a useful method when faster convergence is desired without sacrificing the guarantee of finding a root. This experiment highlighted the balance between speed and reliability in numerical root-finding methods.