Report on Implemented Tasks

Introduction

The purpose of this work was to implement and test two fundamental concepts in data analysis and programming: 1. Visualizing mathematical functions along with noisy, real-world-like data points. 2. Determining whether a numerical sequence is monotonic. Both tasks were executed in Python using libraries such as matplotlib and numpy.

Task 1: Visualization of a Sine Curve with Noise-Corrupted Data

Objective: To generate a sine curve and compare it with observed data that contains noise, simulating real-world measurements.

Methodology:
- The true function was defined as $y = \sin(2\pi x)$, evaluated on the interval [0,1] with step size 0.01.
- A set of sample points was chosen at intervals of 0.1.
- Instead of perfectly following the sine function, the sample values were manually corrupted with noise to mimic measurement errors.
- A plot was generated using matplotlib that displayed the true sine curve and the noisy observed data points.

Results: The plot shows the smooth sine curve as the underlying mathematical truth and the blue circular markers as noisy observed data.

Conclusion: The visualization highlights the discrepancy between theoretical models and noisy empirical data.

Task 2: Monotonicity Check of Arrays

Objective: To implement a function that checks whether a given sequence of numbers is monotonic (entirely non-increasing or non-decreasing).

Methodology:
- A function is_monotonic(array) was defined.
- It iterates through the array and counts increases and decreases.
- If both increases and decreases are found, the array is not monotonic.
- Otherwise, it is monotonic.
- The function was tested on three arrays:
  - Increasing sequence
  - Decreasing sequence
  - Mixed sequence

Results:
- [1, 2, 3, 3, 3, 4, 5, 6, 7] → Monotonic
- [7, 6, 5, 4, 3, 3, 3, 2, 1] → Monotonic
- [1, 5, 4, 3, 3, 3, 2, 6, 7] → Not Monotonic

Conclusion: The function correctly identified monotonic and non-monotonic sequences.

General Conclusion

This notebook demonstrated two essential computational tasks: 1. Data visualization with noise, providing practical insight into real-world measurement challenges. 2. Sequence monotonicity detection, a fundamental algorithmic concept useful in mathematical analysis and data processing.

Both tasks were implemented effectively in Python, showing strong applicability in scientific computing and programming.