

Lab3

March 3, 2024

In today's session, we continued what we worked on last class by working with NumPy arrays, and then we also worked with for and while loops.

0.1 Additional Notes from the Last Session

- While printing outputs at the end of code, we can use `print(f"—")`. The `f` before the quotes tells Python that it is a formatted string, and when using variables in this string, we can use curly brackets. For example, instead of saying `print("x =" vector[0])`, we can say `print(f"x = {vector[0]}")`.
- We can reverse lists by doing `[::-1]`.

0.2 Introduction to NumPy Arrays

Why use these? - NumPy arrays are faster to use than Python lists, and are overall more convenient to use. Some elementary functions for these arrays are creating arrays and matrices, and then manipulating them using functions like `sum()`, `mean()`, or `max()`. There are also functions that return stats about these different arrays and matrices, such as `size` or `shape`.

Two functions that are very useful are `np.arange()` and `np.linspace()`. These two functions allow the user to create array vectors across a defined range. The input for `np.arange` is (starting, ending (not including), step size). The input for `np.linspace` is (starting, ending, amount of elements).

Array slicing is how we navigate through different elements in an array. The syntax for slicing is `A[start:end:stride]`.

0.3 Algorithms

3 Different Control Structures: For loops, While loops, and if/else statements.