# Optimization in High-Performance Computing: AoS vs SoA – Code and Execution Evidence

**Student Name:**  
Murali Krishna Chintha

**Course:**  
Algorithms and Data Structures (MSCS-532-B01) – Second Bi-term

**Instructor:**  
Vanessa Cooper

**Date:**  
08/17/2025

## Introduction

This document contains the source code, screenshots of execution, and repository link for the project *Optimization in High-Performance Computing: AoS vs SoA*. It complements the main APA report by providing direct implementation evidence and supporting artifacts.

## Python Source Code

Below is the implementation of optimization\_demo.py, which demonstrates the performance difference between Array of Structures (AoS) and Structure of Arrays (SoA) layouts using NumPy vectorization and Numba JIT compilation.

# === optimization\_demo.py ===  
# Baseline AoS implementation  
# Optimized SoA with NumPy  
# Optimized SoA with Numba  
# Benchmark harness and main entry  
  
# (Paste the complete Python file code here)

*Note.* In the final submission, ensure the code is pasted in full, formatted in monospace font (e.g., Courier New, 10 pt) for readability.

## Execution Screenshots

The following figures show the program’s runtime outputs and generated performance plots.

**Figure 1.** Execution output of the benchmark script.  
*(Insert terminal output screenshot here)*

**Figure 2.** Runtime performance of AoS vs SoA implementations.  
*(Insert speed\_vs\_N.png here)*

**Figure 3.** Relative speedup of SoA implementations over AoS baseline.  
*(Insert speedup\_vs\_N.png here)*

*Note.* Captions follow APA 7 formatting, with the figure label above and description below.

## GitHub Repository Link

The complete project code and files are hosted at:  
[Insert GitHub Repository Link Here]

## Observations

* The baseline AoS implementation was several orders of magnitude slower than the optimized SoA versions.
* The Numba JIT-compiled version provided the greatest speedup, aligning with theoretical expectations.
* Screenshots and plots confirm correctness and performance improvements discussed in the main report.