Laboratory 1

Leyre Remartínez Villuendas (839995)
Gonzalo Valero Domingo (842392)

October 3, 2025

1 Objective

The aim of this laboratory is to evaluate the performance of two approaches for multiplying matrices of size NxN (using the command time): a naïve implementation versus an optimized implementation using Eigen Mathematical library. The comparison will be carried out across different matrix sizes, and under varying experimental conditions, such as different compiler levels and scenarios with and without CPU stress, in order to analyze their impact on execution time and efficiency.

2 Tests Design and Results

The performance has been evaluated through three different experiments. Each test was conducted on square matrices of different sizes, specifically 2, 10, 100, 250, 500, 750, 1000, 1250, and every measurement was repeated 5 times to reduce the influence of outliers and obtain more reliable averages:

2.1 Time type comparison

Evaluation of both implementations considering the three types of execution time (real, user and system).

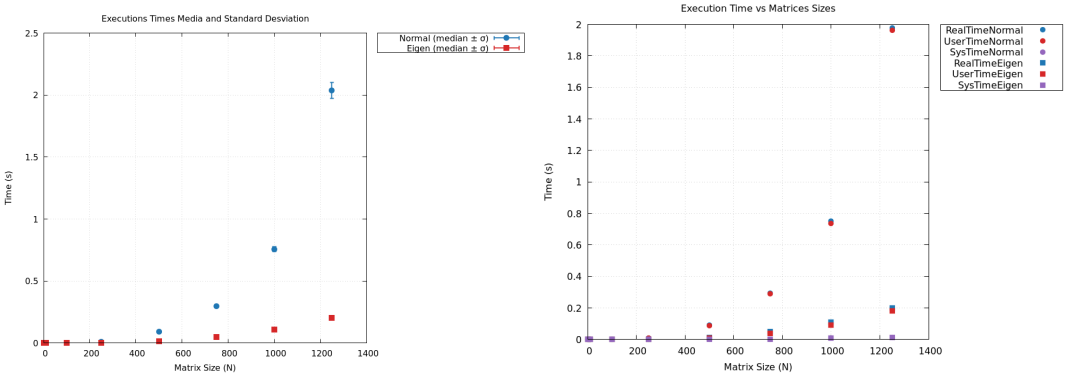

Figure 1: Graphs comparing mean an standard deviation in real time (left) and comparing different types of times execution (right).

For the naïve implementation, the real time grew rapidly reaching more than 2 seconds for N = 1250. In contrast, the Eigen implementation showed a much slower growth, with real time of only 0.21 seconds for the same matrix size. A clear difference is also observed in the standard deviation, where the eigen implementation shows a greater stability than the naïve one.

2.2 Compiler optimization comparison

Analysis of real execution time under different compiler optimization levels (-O1, -O2, -O3)

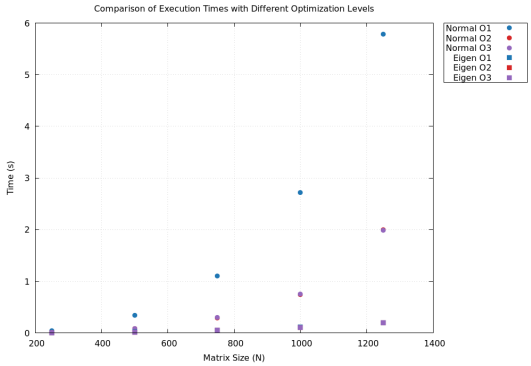

Figure 2: Graphs comparing time with different optimization's levels on both programs.

Figure 2 shows that the naïve implementation benefits more from higher compiler optimization levels than the Eigen implementation, since the Eigen Library is already highly optimized and further compiler optimizations have little to no impact on its performance.

2.3 Stress testing

Repetition of the first experiment while stressing the CPU to assess how system load affects performance.

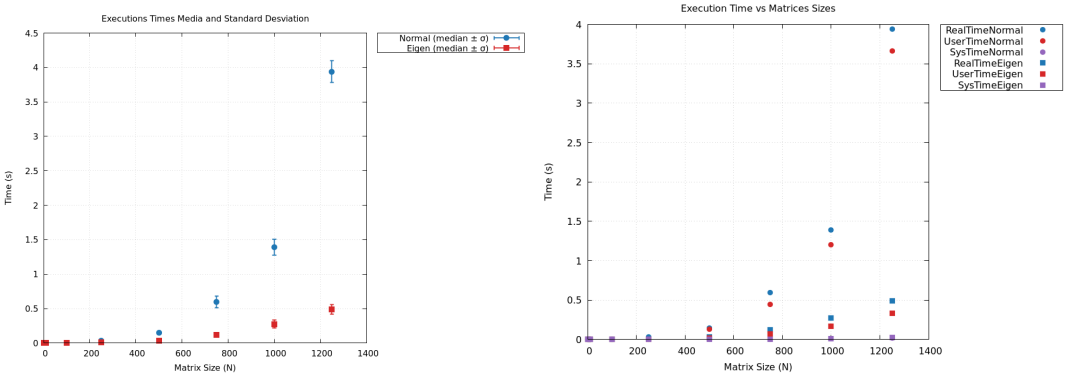

Figure 3: Graphs comparing with stress mean an standard deviation in real time (left) and comparing different types of times execution (right).

Figure 3 reveals a 100% increase in execution time for matrices of 1250 size due to the lack of resources avaiible on the CPU. This is evident form the fact that, in the initial experiment, real time and user time were nearly identical, reflecting minimal interference from other processes. However, under CPU stress, the real time is higher than the user time, indicating that the program experiences delays due to contention for processing resources.