

# Matrix-Matrix Product: Benchmarking Different Compilers and Parallel Computing

Georges Karagozian <sup>1</sup>

<sup>1</sup>Faculty of Science , University of Ontario Institute of Technology ,  
Oshawa, ON

October 2022

## 1 Introduction

This document is the first assignment in the MCSC6040 High Performance Computing course. The purpose of this assignment is performing matrix-matrix multiplication using different compilers and parallel programming with multiple threads in order to look at the computing time of each method used and compare the results.

## 2 Methods

The language used in this assignment was Fortran 90 with GNU and Intel compilers. Different combinations of these compilers were used with no optimization (O0) and most aggressive optimization (O3) as well as BLAS routine while keeping track of the time it takes to run the code. A python script was written to automate running these combinations and obtain the run time.

## 3 Results

Each case was run three times and the average time taken to run the code in seconds was recorded in table 1 below.

## 4 Conclusion

Matrix-matrix multiplication was performed using Fortran90 while utilizing different compilers (namely, GNU compiler gfortran and the Intel compiler ifort) with a different combination of optimization parameters. The results indicate that the Intel compiler is comparable to the GNU compiler without any optimization employed, but it's significantly

Table 1: Run time in seconds for the combination of compilers with optimization keys averaged over three runs.

<b>Optimization</b>	<b>GNU Compiler</b>	<b>Intel Compiler</b>
O3 with BLAS routine	0.836957333 s	0.85841266 s
O0 with BLAS routine	0.835368666 s	0.80245566 s
O3	4.115555 s	0.15917333 s
O0	11.66330067 s	12.53981933 s

faster with its built in optimization when the most aggressive optimization is selected, even faster than using the BLAS routine. Using the BLAS routine with either compiler and any optimization parameter yields similar results.