AMATH 583: High Performance Scientific Computing

Hao Li

Homework 3 April 28, 2023

1. Problem Introduction

In this problem, we are asked to implement six different loop order permutations of the general matrix multiplication (GEMM) operation, $C \leftarrow \alpha AB + \beta C$, using column-major ordering. The loop permutations are *ijk*, *jki*, *kij*, *jik*, *ikj*, and *kji*. These different permutations will exhibit distinct memory access patterns, which can affect the performance of the computation.

Additionally, we will explore the impact of compiler optimizations on the performance of our GEMM implementations for two specific loop permutations: *kij* and *jki*. We will measure the performance of these two permutations using different optimization levels provided by the compiler for square matrices of varying dimensions.

2. Results

2.1. **Float Precision.**

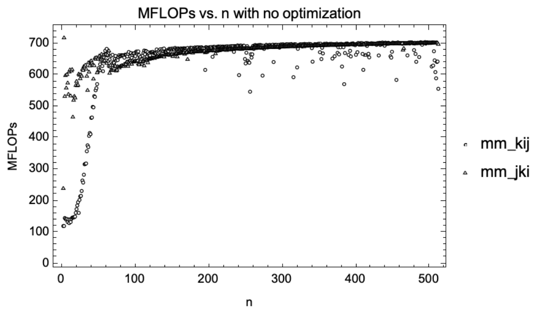

Figure 1. *FLOPs performance vs. $n = 2$ to 512 with no optimizer applied*

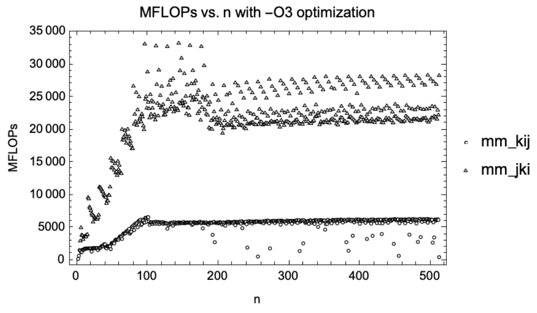

Figure 2. *FLOPs performance vs. $n = 2$ to 512 with -O3 optimizer applied*

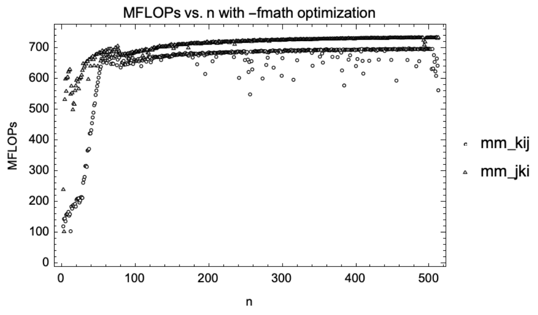

Figure 3. *FLOPs performance vs. $n = 2$ to 512 with -ffast-math optimizer applied*

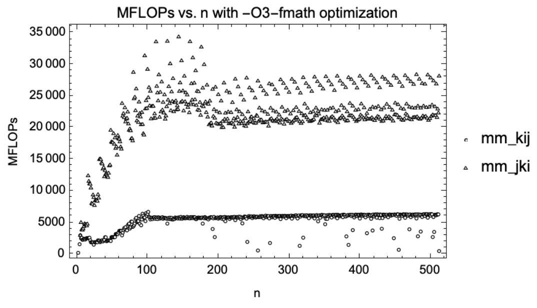

Figure 4. *FLOPs performance vs. $n = 2$ to 512 with -O3 & -ffast-math optimizer applied*

2.2. **Double Precision.**

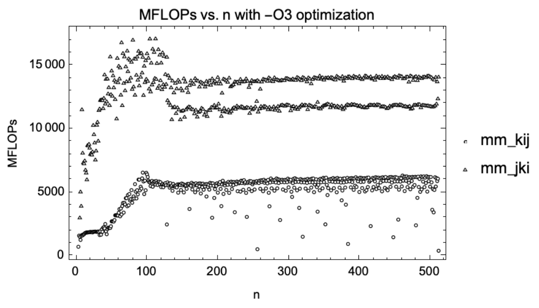

Figure 5. *FLOPs performance vs. $n = 2$ to 512 with* `-O3` *optimizer applied*

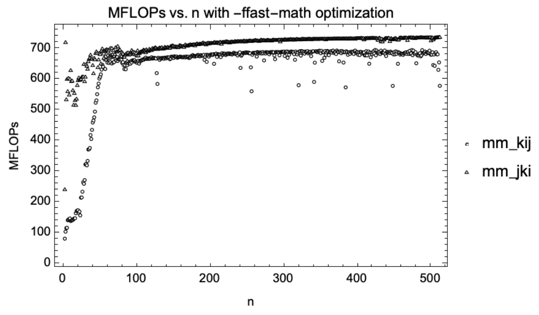

Figure 6. *FLOPs performance vs. $n = 2$ to 512 with* `-ffast-math` *optimizer applied*

3. Problem Introduction

In this problem, we are asked to implement a C++ template for the recursive Strassen matrix multiplication algorithm. We will then analyze the performance of our implementation by measuring the double precision performance for square matrices of even dimensions, ranging from $n = 2$ to $n = 512$. To obtain reliable performance results, we will run each test with ntrial repetitions, where ntrial ≥ 3, and calculate the average performance for each matrix dimension.

4. Results

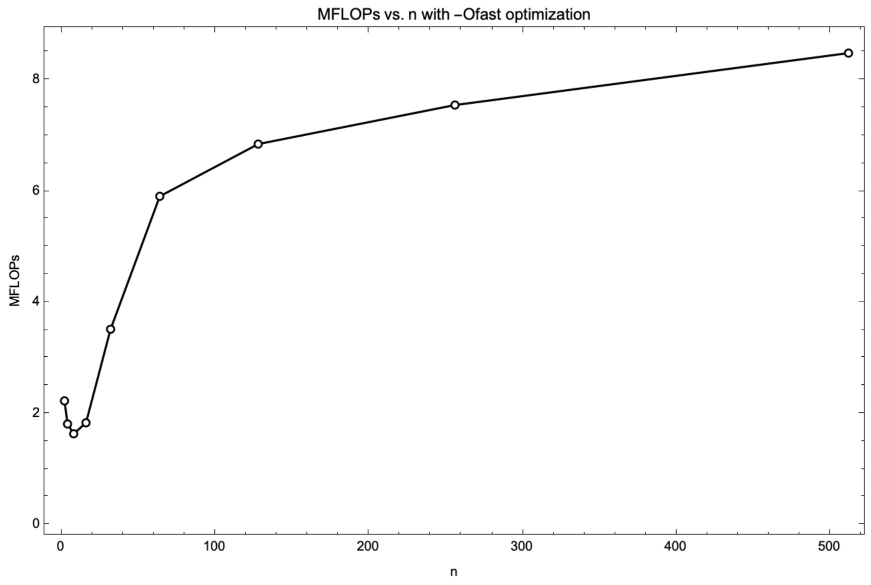

Figure 7. *FLOPs performance for Strassen algorithm vs. $n = 2$ to 512 with* `-Ofast` *optimizer applied*