

# GPU Matrix Algorithms - Preliminary Problem Statement

Hansen, Oliver Juhl Friis  
oljh@itu.dk

Heidemann, William Skou  
wihe@itu.dk

February 2nd, 2024

## Purpose

With this Bachelor's project we wish to gain an understanding of a new tool and improve our current skill set. We wish to learn about the advantages and disadvantages of utilizing the GPU for calculations in parallel. We want to be able to use the GPU using the CUDA C language, developed by NVIDIA. The technical purpose for the project is to achieve a performance increase of matrix algorithms by utilizing the GPU.

## Project Plan

Specifically we want to perform addition, multiplication and QR-decomposition for finding inverse matrices. For each algorithm, we will first implement a correct solution on the CPU to get a baseline for performance. Then, we will improve the performance by utilizing parallelization on the GPU. We will measure the performance gains or losses for various input sizes and plot the results on a graph. We will also benchmark various implementations of each algorithm to see what has an impact on performance. We will implement a simple tool automate the process of benchmarking.

## Hardware

We have two systems with different GPUs and CPUs. System 1 has an NVIDIA GeForce RTX 3060 Laptop GPU a little under 4000 cores. System 2 has an NVIDIA GeForce RTX 4070 Ti GPU with a little over 7500 cores. We will run our implementations on both to measure the performance impact of different hardware.

## Limitations of scope

We will only consider rectangular, dense matrices in our implementations.

## Intended Learning Outcomes

After this project, we will be able to:

- Use the CUDA C programming language to perform calculations on the GPU.
- Compare performance metrics of matrix algorithms using the CPU and GPU.
- Compare performance metrics of different implementations of matrix algorithms on the GPU
- Compare performance metrics of running the implementations on different GPUs.
- Automate and structure tests and benchmarks for the performance of our implementations and visualize the results.
- Reflect critically on the results of performance benchmarks.
- Explain the advantages and disadvantages of writing and running algorithms on the CPU and GPU.

## Literature

Chapter 2, Numerical Recipes in C, William H. Press et al.

Afternotes on Numerical Analysis, G. W. Stewart

CUDA C PROGRAMMING GUIDE, documentation 2018

More articles and material on parallelization of matrices in C will be found and studied.

More articles and material on computer architecture of the GPU will be found and studied.