Parallel and High Performance Computing

Dr. Pablo Antolín

Series 8 - Assignment 2

April 17 2025

CUDA - Conjugate gradient

1 CUDA

This exercise will cover a CUDA implementation of the (already well-known) conjugate gradient solver studied in Series 05, but this time you will be using dense matrices instead of sparse ones.

You are requested to implement a parallel version of the solver using CUDA. Once implemented and tested, you should vary the sizes of the blocks to see how the performance is affected.

As a starting point, pull from the git repository `math454-phpc/exercises-2025` to get access to the conjugate gradient solver code for dense matrices in the subfolder `lecture_08`. This solver accepts the same (sparse) matrix files as the ones in Series 05 (the parser will transform them into dense matrices).

You have to submit a report (1 page max!) that includes a graph with the total number of threads per block on the x-axis (logarithmic scale) and the time per iteration on the y-axis (linear scale). Comment on the obtained results and explain the observed behavior. You should also submit your code in the form of an archive (tarball).

For launching the solver, you can use the following bash script in `izar`

```bash
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --time=00:05:00
#SBATCH --partition=gpu
#SBATCH --gres=gpu:1
#SBATCH --qos=math-454
#SBATCH --account=math-454

module purge
module load gcc cuda

srun nvprof <executable> <input_arguments>
```

where `<executable>` is the name of the executable to launch, and `<input_arguments>` is the list of input arguments (beyond the matrix file, you can modify your main function to accept as input the number of threads per block).

Important note: Do not use any CUDA functions (e.g., `cublasDaxpy`, `cublasDdot`, `cublasDgemv`, etc.) in your implementation: any GPU acceleration of the original code has to be implemented in CUDA by yourself.

Note: We recall that the `GridSize` and `BlockSize` are subject to limits: Features and Technical specifications (Compute Capability 7.5).