Individual Project

1 The Problem

Note that this is an **individual task**. In this final mandatory task in the course, you will work on an application or algorithm of your choice. The assignment must involve implementation of an algorithm, optimization of the code and parallelization either with Pthreads or OpenMP.

In Section 4, you will find a list of suggestions for projects, but you are encouraged to use your imagination to come up with something of your own, or add a variation to one of the suggestions.

2 The Report

The report should be written as a scientific paper of academic quality, written using either LaTeX (preferred) or a word processor (e.g. Word or LibreOffice), in either English (preferred) or Swedish. Plots can be created using e.g. Matlab or Python Matplotlib. The report should include the following:

1. *Introduction*, providing a background and motivation.
2. *Problem description*, presenting the task.
3. *Solution method*, description the algorithm, optimization and parallelization.
4. *Experiments*, presenting how you evaluate the performance of your solution along with your results and some observations and comments.
5. *Conclusions*, with explanations of the results and with ideas for possible optimizations or improvements.
6. *References*, listing relevant literature that was consulted in the project.

The code should be submitted as a compressed archive along with the report. Please submit the report itself as a PDF file.

3 Grading

The individual project will account for 60% of your final grade and the group assignments 1-3 (and 4) for 40%. When grading your work, we will take the following aspects into account:

- *Solution:* Choice of algorithm, serial and parallel efficiency.
- *Methodology:* Demonstration of correctness, and performance evaluation.
- *Code:* Design, robustness, documentation, and general quality.

- *Report:* Disposition, presentation of results, and language quality.

We will also take the complexity and difficulty of the problem into account when evaluating your work. It is also important that you meet the deadline for submission, late admission after the deadline will affect negatively on the grade. The different grade requirements are as follows:

- **Grade 3:** You must demonstrate an understanding of the fundamentals of the course. Your code works correctly and you have made a reasonable attempt at optimizing and parallelizing important aspects of the code. The report is of minimal requirements where you only describe the problem, solution algorithm and show performance results.

- **Grade 4:** You have tried to optimize everything, and you're able to reason about the performance of your code at a high level and know why certain optimizations worked/didn't work. The report is well structured with appropriate references to theory and related work. In addition to grade 3 you also reflect on and discuss your results.

- **Grade 5:** In addition to very good optimizations and understanding of performance issues, you must also show a higher level of understanding of the parallelized code. You must have optimized the code with respect to synchronization, data dependencies, load balance and parallel work. In your report, in addition to grade 4, you will also argue why you have chosen the specific method/technique and why your solution is optimal for the problem.

4 Suggestions

- Quicksort, Mergesort, Bucketsort, or some other sorting algorithm
- MxM with Strassens algorithm
- Sparse Matrix-Vector multiplication
- LU-factorization (with or without pivoting)
- Solution of triangular systems (Lx=b or Ux=b)
- QR-factorization
- Power method for eigenvalues
- Conjugate gradient
- Game of life
- Numerical PDE solver