EPFL

MATH-505 HPC for numerical methods and data analysis

Grigori Laura

Cursus	Sem.	Type
Computational science and Engineering	MA1, MA3	Opt.
Computational science and engineering minor	H	Opt.
Ing.-math	MA1, MA3	Opt.
Mathématicien	MA1, MA3	Opt.

Language of teaching	English
Credits	5
Session	Winter
Semester	Fall
Exam	Oral
Workload	150h
Weeks	14
Hours	**4 weekly**
Lecture	2 weekly
Exercises	2 weekly
Number of positions	

Summary

The objective of this course is to provide the necessary background for designing efficient parallel algorithms in scientific computing as well as in the analysis of large volumes of data. The operations considered are the most costly steps at the heart of many complex numerical simulations.

Content

- Introduction to high performance computing
- Overview of state-of-the-art parallel architectures and MPI programming technique
- Factorization methods and communication avoiding algorithms
- Randomization for solving large scale problems
- Low rank matrix approximation algorithms, deterministic and randomized approaches
- Krylov subspace iterative solvers, deterministic and randomized approaches
- Applications to data science

Learning Prerequisites

Required courses

- Analysis
- Linear algebra
- Exposure to numerical linear algebra and numerical methods
- Programming skills in a language as Python, Julia, C ...

There is no requirement to have knowledge of parallel machines or experience with high performance computing programming. You will get an overview of those during the lectures. The two projects that will be graded are expected to be performed in Python and a library as MPI that allows to exhibit parallelism. They require a considerable amount of coding and performing numerical experiments while using theoretical mathematical background from the lectures.

Learning Outcomes

By the end of the course, the student must be able to:

- Choose a particular linear algebra algorithm to solve a specific problem

- Interpret the numerical and parallel efficiency results
- Assess / Evaluate the numerical stability of different algorithms
- Formulate the parallelization of a given algorithm
- Investigate the potential bottlenecks of a parallel algorithm

Teaching methods

Lectures + exercise sessions

Expected student activities

Students are expected to attend lectures and participate actively in class and exercises. Exercises will include both theoretical work and programming assignments.

Assessment methods

Projects and quizz. The grade will be based on two substantial projects that focus on randomized algorithms, an oral presentation of the second project, and a quizz for the last lectures.

Resources

Bibliography

-

Ressources en bibliothèque

- Matrix Computations / Golub

Moodle Link

- https://go.epfl.ch/MATH-505