Course summary

This course will focus on the parallel implementation of computational mathematics problems using modern accelerated C++. The aim of this course is to learn how to quickly write useful efficient C++ programs. The students will not learn low-level C/C++ instead they will learn how to use high-level data structures, iterators, generic strings, and streams (including interactive and file I/O) of the C++ ISO Standard library. In addition, highly-optimized linear algebra libraries are introduced since the course teaches to solve problems, instead of explaining low-level C++ and computer science algorithms, like sorting algorithms, which are provided in the C++ standard library.

The first part, provides a brief overview of the containers, strings, streams, input/output, and the numeric library of the C++ standard library. For linear algebra, we will look into Blaze which is an open-source, high-performance C++ math library for dense and sparse arithmetic.

The second part will solve computational mathematics problems based-on the the previous introduced features of the C++ standard library.

The third part will focus on the parallel features provided by the C++ standard library. Here, the implemented computational problems in the second part of the course will be parallized using the C++ standard library for parallelism and concurrency.

Since programming skills can only be improved by doing, there will be weekly programming exercises and a small project. After this course students have a basic overview of the C++ standard library to solve efficiently computational mathematics problems without using low-level C/C++.

Prerequisites

None, but some basic knowledge about C++ is beneficial.

Lectures

Tuesday and Thursday, 09:00 to 10:20, 130 LCKT

Important dates

- Midterm exam: 15.10 during lecture
- Final exams: 12.12 from 12:30 to 2:30
- Fall holidays: 17.10-20.10
- Thanksgiving holidays: 27.11-1.12

Reading

We will use material from the following two books in the course:

- Andrew, Koenig. Accelerated C++: practical programming by example. Pearson Education India, 2000.
- Stroustrup, Bjarne. Programming: principles and practice using C++. Pearson Education, 2014.

For advanced reading, following books are recommended:

- Chacon, Scott, and Ben Straub. Pro git. Apress, 2014. Ebook
- Stroustrup, Bjarne. A Tour of C++. Addison-Wesley Professional, 2018.
- Scott, Craig. Professional CMake: A Practical Guide, 2018.

Other resources

- Mailing list: `https://mail.cct.lsu.edu/mailman/listinfo/par4997`
- Web page: `https://www.cct.lsu.edu/~pdiehl/teaching/2019/4977/`

Please don't hesitate to ask questions related to the course by sending me emails: Patrick Diehl.

Projects, Homework, and Quizzes

Grading

- Quizzes and homework 30%
- Project 20%
- Midterm exam 20%
- Final exam 30%

Overall, in the end of the semester 90% of all points or more will give you an 'A', 80% or more a 'B', 70% or more a 'C', and 60% or more results in a 'D'. Below that you'll fail the course, but I'm sure that will not happen to anyone

Topics

Table 1 indicates roughly how much time we will spend on each topic. There is a full lecture calendar available on the course web page outlining the topics in more detail. There is also some flexibility in shortening some of these topics and adding other advanced topics.

Office hours

After the lectures, I will be around for discussion. During the first lecture, we will arrange a office hour which will be announced on the course web page.

Portion	Topic
1/3	Accelerated/Modern C++
1/3	Modern C++ implementation of computational mathematics problems
1/3	Parallel computation using the C++ standard library for parallelism and concurrency

Table 1: Brief outline of the topics of the course

Course Policy

Grading

It is course policy that whoever graded something will be responsible for handling grading disputes. I will grade the midterm exam and the final exam. The grader will grade the homework and the project. Grades become final one week after homework or exam is handed back. This should leave ample time to resolve grading disputes.

Homework Standards

All homework and the project will be submitted using Gihtub Classroom. There will be instructions on the first exercise sheet. All work submitted must carry the student's name and must have sufficient comments and be well organized. A work that can not be open or read easily will get less credits. A reasonable standard of English expression and grammar is also required. The same requirements apply to exams. Additional requirements may apply for any of the separate assignments and will be outlined in the corresponding descriptions.

Cheating

Cheating is a very serious offense and will not be tolerated. Supplying others with homework solutions or materials is forbidden. The policy is that the supplier and the receiver of information will both be dealt with in accordance with and as outlined by the LSU Code of Student Conduct[1].

Programming standards

For this course the C++17 standard is used and obviously your submitted code should compile and run using this standard. Due to the complexity of the program, no credit can be given for a program that does not compile. If a program only partly runs, only partial credit will be given.

Laptops

My experience has shown that taking notes on electronic devices is quite challenging and might distract others. Therefore, I like to restrict the usage of electronic devices during the lectures, if I feel that the lecture will be disturbed. During live coding session, I encourage you to use your device and play with the code.

[1] `https://www.lsu.edu/saa/students/codeofconduct.php`