Syllabus

High Performance Scientific Computing

University of Colorado Boulder

Course No. CSCI 4576/5576

Fall 2020 (August 24, 2020 through December 7, 2020)

Lectures via Zoom on Monday-Wednesday-Friday from 9:10am-10:00am
Labs held via Zoom on Fridays from 1:50pm-3:30pm

Instructor: Scott R. Runnels, Ph.D. ECOT 411
scott.runnels@colorado.edu 505-695-9241

Contents

Course Description

This course introduces high-performance computing ("HPC") systems, software, and methods used to solve large-scale problems in science and engineering. It will focus on the intersection of two separate aspects of scientific HPC: Codes that implement methods for performing complex simulations and HPC systems on which those codes run. Four classes of methods will be covered: Spatial discretization using meshes, particle methods, ray-tracing, and linear solvers. Each of these types of methods have different requirements and lead to different perspectives regarding data structures. They will be intersected with three elements of HPC: Shared-memory parallelism (or "threading" via OpenMP), distributed-memory parallelism (via MPI), and accelerator-based computing (e.g., on GPUs). The focus will be on how the methods are adapted to these lines of HPC to increase performance. Choice of data structures and strategies of coding will be key design decisions. Students will write and run software on a supercomputer, and will learn how to interact with the supercomputer environment, including managing batch jobs, queues, and allocations.

Recommended Prerequisites

APPM 4650 or CSCI 3656 or MATH 4650 or MCEN 3030.

Class Structure

Grading

This class will consist of lectures and laboratories. There are no exams in this class; all work for grading is accomplished through the weekly laboratory, pre-lab reports, and follow-up post-lab reports. The laboratories will occur on Fridays and will be conducted via Zoom. In each laboratory, students will work in groups of 2 or 3 people, who will share their screens with each other in the Zoom session. Each laboratory assignment will require some type of programming or execution of software on either a student's local computer or a CU Boulder supercomputer. While only one student will be able to operate the computer at a given time, it is expected that the team will work together on each laboratory to accomplish its goals. Students are required to share with each other all files they generate as a group during the Friday laboratory. Lab teams will not remain constant throughout the semester, but will instead be shuffled. It is expected that each individual has the opportunity to be the one writing code, while the others serve as checks; this is a powerful method of code writing called "pair programming" (although, here, there can be three on a team).

Weekly Pre- and Post-Lab Reports

Each week, students will submit a pre-lab report, that is typically 1 or 2 pages in length. These pre-lab reports are individual reports and are due Thursday night. The Friday lab is then collaborative in nature, as discussed above. The post-lab report is a group report, and is due the Wednesday of the following week. It is these lab reports that will, together, constitute each student's grade in the course. Individuals or groups may be invited, on occasion, to share their pre- or post-lab reports and ideas with the entire class.

Teamwork and Design Decisions

Throughout the course, lab teams will be required to make decisions regarding how to go about improving performance of some code, and there may be differing designs provided by the members, and differing opinions. Because time will be limited, students will be asked to use what we will call here the "Standard Decision Making Tool", where each team member ranks the other 2 team members' designs, but not their own. The design/idea with the best score wins and is pursued by the group. Ties are broken with a coin toss. Students are highly encouraged to not become attached to a design but, instead, to stay goal oriented and emphasize the value of teamwork. When on real code teams, it is sometimes the case that a design inferior to yours is ultimately chosen. This step of working on teams is a valuable opportunity for becoming acclimated to that possibility.

Office Hours

Office hours are subject to change based on how things go. The are currently scheduled to be held via Zoom on Mondays from 11:00am-noon, and Wednesdays from 1:00pm-2:00pm.

Recommended Resouraces

- *Parallel and High Performance Computing* by Robert Robey and Yuliana Zamora, available from Manning publishers.
- CU Boulder Research Computing documentation:
  `https://curc.readthedocs.io/en/latest/index.html`

Estimated Schedule and Course Content

Note that the schedule and content are subject to change.

Week #1 Introduction and Basics (8/24)

In this first week, we acquire some basic skills to introduce the world of high-performance scientific computing.

How to use a supercomputer, compile nodes vs. compute nodes, queues, file systems, batch jobs, slurm.

In-class development of a basic MPI code, demonstrating basic commands.

Running a parallel code on a supercomputer.

Laboratory (5%): Students will log into a supercomputer, build a code on it, and run a parallel job on it.

Week #2 Application – MPI on Spatial Discretization (8/31)

This week, we apply MPI commands and the ability to run parallel problems to a numerical method that solves a partial differential equation.

In-class development of a finite difference code

In-class development of an iterative linear system solver

Parallelizing the finite difference code with MPI

Laboratory (5%): Parallelize the finite difference code, including the linear solver, with MPI.

Week #3 Application – Particles in a Spatial Mesh (9/7)

This week, we apply our knowledge to a different kind of scientific computing application, one involving particles moving through a spatial mesh.

In-class development of a rudimentary particle transport code, where the particles have a prescribed velocity

Analyzing data structure and computational requirements of the particle transport code

Brainstorming on ways to parallelize it using MPI

Laboratory (5%): Develop key part required to parallelize the prescribed velocity particle transport code using MPI.

Week #4 Combining two MPI Performance Requirements (9/14)

This week, we investigate what happens when we combine two different computing applications into one code and try to optimize performance.

In-class development of combined finite difference and particle-in-cell code, i.e., a ”PIC” code

Brainstorming on ways to parallelize it using MPI

Laboratory (5%): Parallelize the PIC code using MPI.

Week #5 Details of Processor and Memory Architecture (9/21)

Distributed memory computing with MPI may be thought of as separate PCs operating on their own part of the problem, communicating when needed. But task parallelism, vectorization, and GPU programming require more intimate knowledge of CPUs and memory. This week, to prepare us for those aspects of HPC, we explore those details.

Structure of a multi-core CPU

Memory access through cache levels, lines

Cache misses, distribution of memory per core

Commands for exploring a computer’s architecture

Laboratory (5%): Students will use commands and concepts learned in class to learn about and report on the computational and memory architecture of Summit.

Week #6 Measuring Performance (9/28)

This week, we learn how to measure and analyze the performance of our MPI parallel electrostatic PIC code (esPIC), including thelinear solver.

Performance metrics

Analyzing a code for performance bottlenecks

Demonstrating how to remove bottlenecks

Laboratory (10%): Write a report on the performance of the esPIC code using all of the metrics and tools learned in class. Identify and attempt to fix a bottleneck.

Week #7 Introduction to Threading and OpenMP (10/5)

This week, we learn about the difference between MPI programming, which is for distributed memory, and task parallel programming, which relies on shared memory.

Tools for ”threading”

Introduction to OpenMP

In-class development of a code using OpenMP directives

Laboratory (5%): Students will complete a code that solves Laplace’s equation using high-level threading, will measure its speed-up, and will explore methods for making it faster.

Week #8 Linear Solvers (10/12)

This week, we explore linear solvers in more depth, adding and parallelizing a new solver to the code base.

Cell-based matrix formation

Parallelization of Jacobi iteration without ghost nodes

The Conjugate Gradient algorithm

Non-linear systems

Successive Approximation and Newton-Raphson iteration

Laboratory (10%): Students will learn the conjugate gradient algorithm in coded form and will parallelize it using MPI.

Week #9 Combining Distributed and Shared Memory Parallelization (10/19)

This week, we combine shared- and distributed-memory parallelism in a nonlinear spatial discretization solver .

Newton-Raphson Iteration

Relaxation

Analyzing code for thread parallelism performance; Intel Advisor

Laboratory (15%): *This is a two-week lab.* Students will analyze a code designed to solve a nonlinear spatial PDE under MPI to determine how to combine MPI and threading for performance enhancement.

Week #10 Multi-Material Cells (10/26)

This week, we expand our finite difference PIC code to handle multi-material cells, which introduce new complexities into the data structure requirements.

In-class: Expand the PIC code to allow for multi-material cells with volumetric averaging

Analyzing code for its computation, memory, and data structures needs

Laboratory (0%): *This week will continue the lab from last week.*

Week #11 Parallel I/O (11/2)

This week, we learn about the design principles and methods for parallel input and output.

Parallel input

n-to-1, n-to-n, and n-to-m parallel output methods

MPIIO

Guest Lecture: Summit Parallel File Systems

Laboratory (10%): Students will write an n-to-1 parallel output writer for a uniform spatial grid, i.e., the finite difference code from earlier, using MPIIO.

Week #12 Resource Management (11/9)

This week, we learn about how to manage large computations on an HPC time-sharing environment.

Restart dumps

Automatic restart

Guest Lecture: Summit Queues and Partitions

Laboratory (10%): Students will add a restart capability to a 2D transient diffusion solver and complete an automatic restart script to go with it.

Week #13 Application – Ray Tracing (11/16)

This week, we learn about ray tracing, its application in scientific computing, and study how to optimize ray tracing performance.

Ray tracing theory and application

In-class development of a 3D ray-tracing code

Laboratory (10%): This is an individual (not a team) lab, but collaboration is encouraged. In this lab, students either complete the ray-tracing code developed in class or develop their own, to meet specified requirements.

Week #14 GPU Programming, Part 1 (11/23)

This week, we will introduce the GPU and develop an understanding of the various design constraints associated with it.

Guest Lecture: GPU architecture

Tools for GPU programming

Laboratory (0%): The lab for this week is combined with that for next week. Since this is Thanksgiving Week, the lab will not meet as usual. See next week for the actual lab assignment summary statement. Students will be provided with the lab assignment this week, but will have more than one week to complete it. This final lab will be an individual lab; students will not work on teams for this one, but will be encouraged to collaborate freely.

Week #15 GPU Programming, Part 2 (11/30)

This week, we will apply tools we learned last week to speed up one of the applications we have developed as part of this class.

Special Guest Lecture: Code Portability

Tools for GPU programming

In-class application of GPU directives

Laboratory (5%): Students will work individually on this lab. Students will be using pragma directives to speed up a code using a GPU. They will use Summit if they do not have an appropriate GPU on their local computer. Collaboration is encouraged.

Week #16 Recapiluation (12/7)

This week has only one class day in it, which is Monday. During this lecture various elements of the course will be placed into broader contexts.

Course recapitulation

Laboratory (0%): None

Policies

Classroom Behavior

Both students and faculty are responsible for maintaining an appropriate learning environment in all instructional settings, whether in person, remote or online. Those who fail to adhere to such behavioral standards may be subject to discipline. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, political affiliation or political philosophy. For more information, see the policies on classroom behavior and the Student Code of Conduct.

Requirements for COVID-19

As a matter of public health and safety due to the pandemic, all members of the CU Boulder community and all visitors to campus must follow university, department and building requirements, and public health orders in place to reduce the risk of spreading infectious disease. Required safety measures at CU Boulder relevant to the classroom setting include:

- maintain 6-foot distancing when possible,
- wear a face covering in public indoor spaces and outdoors while on campus consistent with state and county health orders,
- clean local work area,
- practice hand hygiene,
- follow public health orders, and
- if sick and you live off campus, do not come onto campus (unless instructed by a CU Healthcare professional), or ifyou live on-campus, please alert CU Boulder Medical Services.

Students who fail to adhere to these requirements will be asked to leave class, and students who do not leave class when asked or who refuse to comply with these requirements will be referred to Student Conduct and Conflict Resolution. For more information, see the policies on COVID-19 Health and Safety and classroom behavior and the Student Code of Conduct. If you require accommodation because a disability prevents you from fulfilling these safety measures, please see the "Accommodation for Disabilities" statement on this syllabus.

Before returning to campus, all students must complete the COVID-19 Student Health and Expectations Course. Before coming on to campus each day, all students are required to complete a Daily Health Form. Students who have tested positive for COVID-19, have symptoms of COVID-19, or have had close contact with someone who has tested positive for or had symptoms of COVID-19 must stay home and complete the Health Questionnaire and Illness Reporting Form remotely. In this class, if you are sick or quarantined, you can still participate in lectures and the laboratory since they are all remote, but only if you are feeling well enough to do so. If you are too sick to participate in laboratory, please do your best to notify the instructor. However, do not feel any obligation to reveal the details of your illness (whether COVID-19 or otherwise). The main point will be to convey that you will not be able to participate.

Accommodation for Disabilities

If you qualify for accommodations because of a disability, please submit your accommodation letter from Disability Services to your faculty member in a timely manner so that your needs can be addressed. Disability Services determines accommodations based on documented disabilities in the academic environment. Information on requesting accommodations is located on the Disability Services website. Contact Disability Services at 303-492-8671 or dsinfo@colorado.edu for further assistance. If you have a temporary medical condition, see Temporary Medical Conditions on the Disability Services website.

Preferred Student Names and Pronouns

CU Boulder recognizes that students' legal information doesn't always align with how they identify. Students may update their preferred names and pronouns via the student portal; those preferred names and pronouns are listed on instructors' class rosters. In the absence of such updates, the name that appears on the class roster is the student's legal name.

Honor Code

All students enrolled in a University of Colorado Boulder course are responsible for knowing and adhering to the Honor Code. Violations of the policy may include: plagiarism, cheating, fabrication, lying, bribery, threat, unauthorized access to academic materials, clicker fraud, submitting the same or similar work in more than one course without permission from all course instructors involved, and aiding academic dishonesty. All incidents of academic misconduct will be reported to the Honor Code (honor@colorado.edu; 303-492-555). Students found responsible for violating the academic integrity policy will be subject to nonacademic sanctions from the Honor Code as well as academic sanctions from the faculty member. Additional information regarding the Honor Code academic integrity policy can be found at theHonor Code Office website.

Sexual Misconduct, Discrimination, Harassment and/or Related Retaliation

The University of Colorado Boulder (CU Boulder) is committed to fostering an inclusive and welcoming learning, working, and living environment. CU Boulder will not tolerate acts of sexual misconduct (harassment, exploitation, and assault), intimate partner violence (dating or domestic violence), stalking, or protected-class discrimination or harassment by members of our community. Individuals who believe they have been subject to misconduct or retaliatory actions for reporting a concern should contact the Office of Institutional Equity and Compliance (OIEC) at 303-492-2127 or cureport@colorado.edu. Information about the OIEC, university policies, anonymous reporting, and the campus resources can be found on the OIEC website.

Please know that faculty and instructors have a responsibility to inform OIEC when made aware of incidents of sexual misconduct, dating and domestic violence, stalking, discrimination, harassment and/or related retaliation, to ensure that individuals impacted receive information about options for reporting and support resources.

Religious Holidays

Campus policy regarding religious observances requires that faculty make every effort to deal reasonably and fairly with all students who, because of religious obligations, have conflicts with scheduled exams, assignments or required attendance. In this class, students are required to notify the instructor at the beginning of the semester so that proper accommodations can be made for the laboratory assignments. See the campus policy regarding religious observances for full details.