Electrical & Computer ENGINEERING

Carnegie Mellon

Course Syllabus

18647: *Computational Problem Solving for Engineers*
Spring 2024

Instructor: Franz Franchetti
Office Location: HH A312 & Scott 6143, https://cmu.zoom.us/my/franzf
Email Address: franzf@andrew.cmu.edu
Office Hours: Monday 2pm—3pm

Teaching Assistant: Eric Tang
Office Location: HH 1306, https://cmu.zoom.us/my/erictang
Email Address: erictang@andrew.cmu.edu
Office Hours: Wednesday 2pm—3pm

Course Support: Academic Services Center
Office Location: HH 1113
Website: https://www.ece.cmu.edu/academics/academic-services-center.html

.

Course Description: Computing platforms used in engineering span an incredible dynamic range from embedded and wearable processors through handhelds/laptops to high performance computing servers and the cloud. Similar engineering and AI/ML problems need to be solved across the entire dynamic range. When developing algorithms or solving R&D problems, one usually starts with MATLAB and Python using frameworks like Torch, Spark, and TensorFlow, and only resorts to C/C++ when needed. This course covers how to solve AI/ML and engineering research and development problems across the entire range of machines in a productive and performant way. It discusses how to scale problems from the initial concept stage usually executed on a laptop to more powerful computing systems like enterprise or HPC servers, GPU accelerated systems, and cloud computing platforms.

This course addresses a wide range of computational and informatics problem families from traditional numerical simulation and symbolic data processing to AI and machine learning problems. It covers the most important scalable parallel algorithms used in engineering computing and discusses frameworks providing problem-specific and general implementation templates. It covers algorithm analysis from the numerical and complexity perspective, parallelization approaches and scalability, algorithm optimization, evaluation and analysis of results.

Students in this course learn to productively solve AI/ML and engineering research and development problems on advanced computer systems across the dynamic range of computing systems. Further, they learn to carry algorithms from the concept stage to efficient-enough scaled-up implementations necessary to solve large scale problem instances, or squeeze them into the small footprint of embedded and wearable devices. Students will solve assigned homeworks and do a final blog post about lessons learned.

Number of Units: 12

Recommended pre-requisites: 18-213/613, 18-645/646, 18-202 or equivalent

Pre-requisite for:

Undergraduate Course Designation:

Undergraduate Course Area:

Class Lecture:

- *Tuesday and Thursday* *9:30am – 10:50am*

Recitation:

- *Friday* *10:00am – 10:50am*

Required Textbook:
none

Suggested Reading:
Advanced Engineering Mathematics, 10th Edition
Edward J. Norminton, Herbert Kreyszig, Erwin Kreyszig

Computer Systems: A Programmer's Perspective, Third Edition
Randal E. Bryant and David R. O'Hallaron, Carnegie Mellon University

Numerical Recipes: The Art of Scientific Computing
William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery

Parallel Programming In C with MPI and OpenMP
Michael Quinn

Other Supplemental Materials:

Brief List of Topics Covered:

- Parallel algorithms, scalability, and numerical analysis of these algorithms
- Software stack and hardware for computational engineering and AI/ML problems
- Hardware and software available to CMU students (on-site, cloud, NSF XSEDE)
- Frameworks and execution environments for quick problem solving
- How to scale from initial concepts to large-scale problems
- Next steps: How to transition to C/C++ with CUDA/OpenCL, OpenMP, MPI etc.

Course Canvas:
To access the course canvas from an Andrew Machine, go to the login page at: https://cmu.instructure.com/. You should check the course canvas daily for announcements and handouts.

Course Wiki:
Students are encouraged to use the ECE wiki to provide feedback about the course at: http://wiki.ece.cmu.edu/index.php.

Grading Algorithm:

20%	In-class participation
20%	Blog post
60%	5 Homeworks

Lectures:
Lectures will be presented in person and may include slides and handwriting. Lecture notes and slide presentations will be posted after lecture.

Recitations:
Recitations will be presented in person and include blackboard and in-class discussion. Specifically, homework questions are effectively addressed as a group during recitations.

Homeworks:
Expect 5 homeworks, roughly, one due every other week. Homework assignments will be posted and submitted via Canvas. Completed homeworks are due at 11:59:59pm ET at the date indicated when posted. Students are encouraged to discuss the problems together but must submit their own independent work.

Machines
In this course we will use a large variety of machines in the homeworks and projects:

- Large Linux servers with direct ssh access. The machines we will access are the ECE number cluster and the data science cloud.
- Large number of small Linux machines in high throughput mode via ECE's HTCondor.

- HPC multicore nodes and multi-node configurations with and without GPUs via a batch system at PSC and CMU.
- Hadoop/Spark/TensorFlow with and without GPUs at PSC.
- Special hardware at PSC: Bridges AI multi-GPU resources (HPE Apollo and DGX-2) and large memory machines.
- AWS instances with or without multicore and GPU support.

Software Platforms
In the course we will try out a number of high-level languages and libraries in these languages:
- Python with NumPy, SciPy, and matplotlib
- R with CRAN packages
- MATLAB
- Mathematica
- Haskell
- Frameworks: Torch, Theano, OpenCV
- High-end engineering software
- C/C++ with HPC math/ML libraries

Algorithms
We will use algorithms covered in the lecture as simple examples to experiment across the machines and software platforms. Students either implement the algorithms themselves or find implementations online, and then run them on the target machines to perform scalability studies. Algorithm groups targeted in the homeworks are as following:
- Dense numerical linear algebra, CNNs/DNNs/FFTs
- Graph algorithms and sparse numerical linear algebra
- ODE and PDE solvers, stencils, filters, discretization
- Discrete and continuous optimization, ML training
- Informatics, symbolic computing, higher level AI algorithms
- Statistics: Monte Carlo, MCMC, statistical machine learning
- Deep Learning, Big Data Analytics

Homework 1—5 Deliverables
Homework 1 to 5 will have as deliverables examples that run the specified/chosen algorithms on the specified/chosen machines for a range of problem sizes. The homework submission consists of the following:
- The source code for the algorithm, with source attribution as applicable
- The necessary data files
- All necessary scripts and make files
- The captured output of the submission runs
- Measurements in CSV files and performance plot

Blog post—Deliverables
As final project students submit a blog post addressing the following aspects of one of the homework problems:

- *Problem specification:* explain what problem you are addressing and where it is used.
- *Algorithm:* describe the algorithm you used.
- *Hardware and software:* describe the hardware platform and software infrastructure you used.
- *Outcome:* describe the result you obtained and how larger computing resources were necessary.
- *Scalability and complexity:* describe the algorithm's complexity, how the algorithm scales with problem size, and how it scales number of processors/cores in strong and weak scaling sense.
- *Quality of the result:* assess the numerical or approximation quality of your result.
- *Performance analysis:* measure the runtime behavior, model fit it to the theoretical growth behavior, and assess the efficiency of the implementation.
- *Suggested performance optimization:* Assume a limited engineering budget but the need for a faster solution. Suggest which parts of the algorithm one would optimize and what optimization approach and target hardware should be used.

Tentative Course Calendar:

Date	Day	Class Activity
January		
16	Tues.	**State of computing:** What is the current state of the art from embedded devices through desktops, servers, and consumer system all the way to cloud, HPC, and supercomputing
18	Thurs.	**Computer architecture:** Relevant computer architecture concepts
23	Tues.	**Software stack:** ISA, operating system, virtualization, messaging, containers
25	Thurs.	**Mathematics for Engineers:** The central role of numerical linear algebra ***release HW 1***
30	Tues.	**The ECE Computing Environment:** number cluster, Andrew systems, capability machines, GPU access, cloud access, Pittsburgh Supercomputing Center and ACCESS
February		
1	Thurs.	**Algorithm analysis, scalability, complexity:** Getting answers in time
6	Tues.	**Guest Lecture:** Quantum Computing and Quantum Algorithms ***HW1 due, release HW2***
8	Thurs.	**Parallelization:** Sequential vs. parallel algorithms, scalability vs. performance
13	Tues.	**Need for speed:** Principles of code optimization, when and how to optimize code
15	Thurs.	**Cloud computing and HPC:** Amazon EC2/Windows Azure/Google Cloud, Computational Grids, Scientific Workflows, Computing Centers, ECE ITS
20	Tues.	**Numerical Analysis:** How good are your answers? How to make them better?
22	Thurs.	**Scalable algorithms:** Dense numerical linear algebra, CNNs/DNNs, FFTs
27	Tues.	**Scalable algorithms:** Graph algorithms and sparse numerical linear algebra
29	Thurs.	**Scalable algorithms:** ODE and PDE solvers, stencils, filters, discretization ***HW 2 due, release HW3***
March		
12	Tues.	**Scalable algorithms:** Discrete and continuous optimization, ML training
14	Thurs.	**Scalable algorithms:** Informatics, symbolic computing, higher level AI algorithms
19	Tues.	**Scalable algorithms:** Statistics: Monte Carlo, MCMC, statistical machine learning
21	Thurs.	**Data:** Obtaining real data sets, data visualization, data bases, data stores, file systems ***HW 3 due, release HW4***
26	Tues.	**Modern CPUs:** Super-Scalar Out-of-order, multicore
28	Thurs.	**Modern CPUs:** Vector Extensions
April		

2	Tues.	**Hardware Acceleration:** GPUs, FPGAs, TPU, Tensor Cores ,...
4	Thurs.	**From productivity to performance:** C++, OpenMP, MPI, CUDA, Autotuning ***HW 4 due, release HW 5***
9	Tues.	**Guest Lecture:** Software Engineering vs. Performance Engineering
16	Tues.	**Guest Lecture:** Pittsburgh Supercomputing Center and Usable HPC and High Performance AI/ML
18	Thurs.	***HW 5 due*** **Research Talk:** SPIRAL: Formal Software Synthesis of Computational Kernels
23	Tues.	**Student Blog Presentations and Homework Lessons Learned**
25	Thurs.	**Summary and Beyond** ***Blog post due***

Education Objectives (Relationship of Course to Program Outcomes):

1. **an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics**
   Formulate a question that can be solved by large scale computing.
2. **an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors**
   Implement scalable parallel algorithms, analyze scaling behavior and result accuracy.
3. **an ability to communicate effectively with a range of audiences**
   Multiple presentations of computing-related topics and the final project need to be understandable for non-experts.
4. **an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts**
   How powerful computing has become to be the third leg of science.
5. **an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives**
   N/A
6. **an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions**
   Measure performance and analyze it
7. **an ability to acquire and apply new knowledge as needed, using appropriate learning strategies.**
   Implement software that solves a large scale problem without the need of extensive programming.

ECE Academic Integrity Policy (http://www.ece.cmu.edu/programs-admissions/masters/academic-integrity.html**):**

The Department of Electrical and Computer Engineering adheres to the academic integrity policies set forth by Carnegie Mellon University and by the College of Engineering. ECE students should review fully and carefully Carnegie Mellon University's policies regarding Cheating and Plagiarism; Undergraduate Academic Discipline; and Graduate Academic Discipline. ECE graduate student should further review the Penalties for Graduate Student Academic Integrity Violations in CIT outlined in the CIT Policy on Graduate Student Academic Integrity Violations. In addition to the above university and college-level policies, it is ECE's policy that an ECE graduate student may not drop a course in which a disciplinary action is assessed or pending without the course instructor's explicit approval. Further, an ECE course instructor may set his/her own course-specific academic integrity policies that do not conflict with university and college-level policies; course-specific policies should be made available to the students in writing in the first week of class.

This policy applies, in all respects, to this course.

CMU Academic Integrity Policy (http://www.cmu.edu/academic-integrity/index.html):
In the midst of self exploration, the high demands of a challenging academic environment can create situations where some students have difficulty exercising good judgment. Academic challenges can provide many opportunities for high standards to evolve if students actively reflect on these challenges and if the community supports discussions to aid in this process. It is the responsibility of the entire community to establish and maintain the integrity of our university.

This site is offered as a comprehensive and accessible resource compiling and organizing the multitude of information pertaining to academic integrity that is available from across the university. These pages include practical information concerning policies, protocols and best practices as well as articulations of the institutional values from which the policies and protocols grew. The Carnegie Mellon Code, while not formally an honor code, serves as the foundation of these values and frames the expectations of our community with regard to personal integrity.

This policy applies, in all respects, to this course.

The Carnegie Mellon Code

Students at Carnegie Mellon, because they are members of an academic community dedicated to the achievement of excellence, are expected to meet the highest standards of personal, ethical and moral conduct possible.

These standards require personal integrity, a commitment to honesty without compromise, as well as truth without equivocation and a willingness to place the good of the community above the good of the self. Obligations once undertaken must be met, commitments kept.

As members of the Carnegie Mellon community, individuals are expected to uphold the standards of the community in addition to holding others accountable for said standards. It is rare that the life of a student in an academic community can be so private that it will not affect the community as a whole or that the above standards do not apply.

The discovery, advancement and communication of knowledge are not possible without a commitment to these standards. Creativity cannot exist without acknowledgment of the creativity of others. New knowledge cannot be developed without credit for prior knowledge. Without the ability to trust that these principles will be observed, an academic community cannot exist.

The commitment of its faculty, staff and students to these standards contributes to the high respect in which the Carnegie Mellon degree is held. Students must not destroy that respect by their failure to meet these standards. Students who cannot meet them should voluntarily withdraw from the university.

This policy applies, in all respects, to this course.

Carnegie Mellon University's Policy on Cheating
(http://www.cmu.edu/academic-integrity/cheating/index.html) states the following:
According to the University Policy on Academic Integrity, cheating "occurs when a student avails her/himself of an unfair or disallowed advantage which includes but is not limited to:

- Theft of or unauthorized access to an exam, answer key or other graded work from previous course offerings.
- Use of an alternate, stand-in or proxy during an examination.
- Copying from the examination or work of another person or source.
- Submission or use of falsified data.
- Using false statements to obtain additional time or other accommodation.
- Falsification of academic credentials."

This policy applies, in all respects, to this course.

Carnegie Mellon University's Policy on Plagiarism
(http://www.cmu.edu/academic-integrity/plagiarism/index.html) states the following: According to the University Policy on Academic Integrity, plagiarism "is defined as the use of work or concepts contributed by other individuals without proper attribution or citation. Unique ideas or materials taken from another source for either written or oral use must be fully acknowledged in academic work to be graded. Examples of sources expected to be referenced include but are not limited to:

- Text, either written or spoken, quoted directly or paraphrased.
- Graphic elements.
- Passages of music, existing either as sound or as notation.
- Mathematical proofs.
- Scientific data.
- Concepts or material derived from the work, published or unpublished, of another person."

This policy applies, in all respects, to this course.

Carnegie Mellon University's Policy on Unauthorized Assistance
(http://www.cmu.edu/academic-integrity/collaboration/index.html) states the following: According to the University Policy on Academic Integrity, unauthorized assistance "refers to the use of sources of support that have not been specifically authorized in this policy statement or by the course instructor(s) in the completion of academic work to be graded. Such sources of support may include but are not limited to advice or help provided by another individual, published or unpublished written sources, and electronic sources. Examples of unauthorized assistance include but are not limited to:

- Collaboration on any assignment beyond the standards authorized by this policy statement and the course instructor(s).
- Submission of work completed or edited in whole or in part by another person.
- Supplying or communicating unauthorized information or materials, including graded work and answer keys from previous course offerings, in any way to another student.
- Use of unauthorized information or materials, including graded work and answer keys from previous course offerings.
- Use of unauthorized devices.
- Submission for credit of previously completed graded work in a second course without first obtaining permission from the instructor(s) of the second course. In the case of concurrent courses, permission to submit the same work for credit in two courses must be obtained from the instructors of both courses."

This policy applies, in all respects, to this course.

Carnegie Mellon University's Policy on Research Misconduct (http://www.cmu.edu/academic-integrity/research/index.html) states the following:

According to the University Policy For Handling Alleged Misconduct In Research, "Carnegie Mellon University is responsible for the integrity of research conducted at the university. As a community of scholars, in which truth and integrity are fundamental, the university must establish procedures for the investigation of allegations of misconduct of research with due care to protect the rights of those accused, those making the allegations, and the university. Furthermore, federal regulations require the university to have explicit procedures for addressing incidents in which there are allegations of misconduct in research."

The policy goes on to note that "misconduct means:

- fabrication, falsification, plagiarism, or other serious deviation from accepted practices in proposing, carrying out, or reporting results from research;

- material failure to comply with Federal requirements for the protection of researchers, human subjects, or the public or for ensuring the welfare of laboratory animals; or

- failure to meet other material legal requirements governing research."

"To be deemed misconduct for the purposes of this policy, a 'material failure to comply with Federal requirements' or a 'failure to meet other material legal requirements' must be intentional or grossly negligent."

To become familiar with the expectations around the responsible conduct of research, please review the guidelines for Research Ethics published by the Office of Research Integrity and Compliance.

This policy applies, in all respects, to this course.

Take care of yourself. Do your best to maintain a healthy lifestyle this semester by eating well, exercising, avoiding drugs and alcohol, getting enough sleep and taking some time to relax. This will help you achieve your goals and cope with stress.

All of us benefit from support during times of struggle. You are not alone. There are many helpful resources available on campus and an important part of the college experience is learning how to ask for help. Asking for support sooner rather than later is often helpful.

If you or anyone you know experiences any academic stress, difficult life events, or feelings like anxiety or depression, we strongly encourage you to seek support. Counseling and Psychological Services (CaPS) is here to help: call 412-268-2922 and visit their website at http://www.cmu.edu/counseling/. Consider reaching out to a friend, faculty or family member you trust for help getting connected to the support that can help.

Every individual must be treated with respect. The ways we are diverse are many and are critical to excellence and an inclusive community. They include but are not limited to: race, color, national origin, sex, disability, age, sexual orientation, gender identity, religion, creed, ancestry, belief, veteran status, or genetic information. We at CMU, will work to promote diversity, equity and inclusion because it is just and necessary for innovation. Therefore, while we are imperfect, we will work inside and outside of our classrooms, to increase our commitment to build and sustain a community that embraces these values.

It is the responsibility of each of us to create a safer and more inclusive environment. Bias incidents, whether intentional or unintentional in their occurrence, contribute to creating an unwelcoming environment for individuals and groups at the university. If you experience or observe unfair or hostile treatment on the basis of identity, we encourage you to speak out for justice and support in the moment and and/or share your experience anonymously using the following resources:

Center for Student Diversity and Inclusion:
csdi@andrew.cmu.edu, (412) 268-2150, www.cmu.edu/student-diversity

Report-It online anonymous reporting platform:
www.reportit.net username: *tartans* password: *plaid*

All reports will be acknowledged, documented and a determination will be made regarding a course of action." All experiences shared will be used to transform the campus climate.

If you have questions about this or your coursework, please let me know.