ASTE 404, Computational Programming and Numerical Methods

Units: 3
Term: Fall 2024
Day & Time: Thursdays 12:30-3:20 PM
Location: DRB 146
Website: https://sites.google.com/usc.edu/aste-404

Instructors: Dr. Lubos Brieda

Office: OHE 530J or RRB 115
Office Hours: Thursdays: 10:30-11:30am + online by appt
Contact Info: brieda@usc.edu

Teaching Assistant: TBD
Office: TBD
Office Hours: TBD
Contact Info: TBD

Course Description
The goal of this course is to introduce you to the ecosystem of computational tools and techniques relevant to aerospace and astronautical engineering. We start by introducing the basics of numerical integration in Matlab before introducing the same concepts in Python and C++. Along the way, we cover mesh and particle-based techniques for solving gas dynamics problems, and learn how take advantage of modern hardware capabilities including multithreading, graphics cards (GPUs), computer clusters, microcontrollers, and FPGAs. We learn how to visualize results in Paraview and how to develop interactive applications that run in web browsers. Additional topics include object oriented programming, validation and verification, debugging, machine learning, and optimization. You will practice these concepts by working on a group project focusing on a topic of your choice.

Course Goals and Objectives:
At the completion of the subject, students will master at least one high-performance programming language and apply it to develop and implement their own computer simulation models to solve complex problems in astronautics and aerospace engineering. The student will also have basic understanding of numerical integration, Eulerian and Lagrangian mechanics, object oriented programming, code debugging and verification, parallelization, web-based technologies, use of microcontrollers and FPGA, and machine learning.

Recommended Preparation:
You are expected to have some introductory experience with programming and numerical analysis in languages on the level of ITP115 or ITP165.

Catalog Data:
Programming of serial and parallel simulation codes with high-performance languages such as C++ and Fortran. Covers numerical techniques for continuum and rarefied gas flows, data visualization, embedded systems, graphics cards, and machine learning.

Instructor Bio:
I am a part time lecturer at USC Department of Astronautics, where I have been teaching ASTE-404 and ASTE-546 since 2021. My "real job" involves developing plasma and rarefied gas simulation codes applicable to electric (plasma) propulsion, plasma-wall interface in fusion reactors, and spacecraft contamination transport through my company Particle in Cell Consulting LLC. As an example, I worked on simulating Hall effect thrusters and their plumes, hollow cathodes, and ice build up during deployment of the James Webb Space Telescope. I am currently supporting the JHU/APL Dragonfly mission that will fly a rover on Saturn's moon Titan. I also co-advise the undergraduate spacecraft propulsion club ASPEN that is building a novel solid-fuel electric propulsion thruster for small satellite deorbiting. I hold a Ph.D. in Mechanical and Aerospace Engineering from the George Washington University, and my other work experience includes the roles of a research engineer at the Air Force Research Laboratory at Edwards AFB, and Contamination Control Engineer at NASA Goddard SFC.

Course Notes:
Textbook (not required):
- Brieda, Wang, Martin, *Introduction to Modern Scientific Programming and Numerical Methods*, CRC Press, 2024

Additional recommended books:
- Brieda, L. *Plasma Simulations by Example*, CRC Press, 2019
- Stroustrup, B., *Tour of C++*, Addison-Wesley, 2022

Grading:
Course grading will be based on homework, quizzes, a literature review presentation, and a final project. Homework is assigned weekly and is due at the start of the following class. It generally consists of "follow along" programming assignments. You need to complete only 10 assignments for a full credit. The literature review is based on finding an interesting computational journal or conference paper, and recording a short (10 minutes / 4 slides) presentation. About 8 multiple choice / fill in the blanks quizzes will be used to test understanding of previously covered topics. The quizzes will be open books, and some will be take-home, while others will be in class. The final project involves working in small groups to develop a program or an application relevant to your interests.

Grading Breakdown:

Homework	30%
Quizzes	20%
Literature Review	10%
Final Project	40%

TOTAL 100%

ASTE 404: Computational Programming -- Course Schedule

Day	#	Topics/Daily Activities
8/29	1	**Introduction to Scientific Computing.** Basic concepts such as variables, loops, conditions, functions. Numerical integration introduced by developing algorithm to integrate a tennis ball trajectory.
9/5	2	**Discretization Schemes.** Finite difference, 2D heat equation in Python and Matlab. Google Colab, Python NumPy and SciPy packages.
9/12	3	**Compiled Languages.** C and Fortran are introduced. Variable types, arrays, functions, dynamic memory, compilation, use of development environments. Linux command-line environment.
9/19	4	**Linear Solvers and Visualization**. Jacobi and Gauss-Seidel iteration, Thomas algorithm. Data visualization using Paraview. C++ version of the 2D heat equation solver.
9/26	5	**Object Oriented Programming**. Data encapsulation, inheritance, virtual functions, and operator overloading. Revised version of the C++ heat equation solver. Crank-Nicolson method.
10/3	6	**Web technologies.** Use of HTML and Javascript to develop interactive applications running in a web browser. Rendering using Canvas and WebGL.
10/8 (T) **(ZOOM)**	7	**(Fall Recess) Code Testing and Documentation.** Uncertainty analysis, convergence studies, unit testing, version control, linear algebra libraries, documentation systems, and LaTeX.
10/17	8	**Numerical Analysis.** Additional important topics related to numerical analysis such as interpolation, smoothing, signal processing, Newton-Raphson method, multigrid, distribution functions.
10/24	9	**Fluid Methods**. Introduction to fluid methods for gas modeling. Advection-diffusion equation, vorticity-stream function and SIMPLE methods, Finite Volume Method, Runge-Kutta and multistep methods, stability analysis.
10/31	10	**Particle Methods.** Introduction of simulation methods based on test particles, Direct Simulation Monte Carlo (DSMC), Particle in Cell (PIC), Molecular Dynamics (MD).
11/7	11	**Multithreading and Distributed Computing.** Code parallelization using multiple CPU cores (multithread) and clusters of interconnected computers (MPI), race condition, deadlock.
11/14	12	**Graphics Cards.** Use of CUDA to develop parallel algorithms running on graphics cards (GPUs), OpenGL rendering, performance bottlenecks.
11/21	13	**Embedded Systems.** Arduino platform, interfacing with external sensors, serial port communication. Field Programmable Gate Arrays (FPGAs), Verilog, interfacing with Arduino.
11/26 (T) **(ZOOM)**	14	**(Thanksgiving) Emerging Technologies**. Quantum computing, game engines, virtual reality, apps, spacecraft data processing, calibration, data compression.
12/5	15	**Optimization and Machine Learning.** Adaptive search, genetic algorithms, neural networks, physics-informed learning, back propagation, Tensorflow, simple C++ neural network.
12/17	-	**Final Project Presentation, 11am – 1pm**

Lessons will be given in person, but will also be streamed (and recorded) over Zoom. I however expect you all to attend in person as much as possible. Please note that we will hold two online Tuesday (subject to change, time TBD) classes during the week of Fall Recess and Thanksgiving to provide the standard 15 weeks of lectures.

Grading Details

Homework

Homework will be assigned every week. Many of the assignment are follow-along "labs", where you are given the full code as a screenshot, and you need to transcribe it an run it. The homework is mainly graded on effort, this however requires you attempting to complete (and sufficiently debug) all parts.

Quizzes

We will have about 8 "test your understanding" open book quizzes. The quizzes usually consist of around 10 multiple choice, true/false, matching, and short answer type of questions. Some quizzes will be given in class, while others will be take home and will be due before the following class. The take home quizzes may involve some simple programming.

Paper Review

I would like you to become more familiar with astronautical numerical analysis field by finding and reading some relatively recent journal or a conference paper (using sites such as Google Scholar, or AIAA, IEEE, APS society websites). You are then to prepare a short presentation (no more than 5 slides!) describing a) the objective of the paper, b) a brief overview of numerical method, c) summary of the most important results, and d) summary of shortcomings or simplifications you observed in the work. Video recording of your presentation then needs to be uploaded.

Course Project

The main focus of the class is a group project. Its purpose is to practice programming skills learned in the class, while also learning to work as part of a project team. This project is envisioned to be completed in groups of 3-4 persons. The group is expected to suggest a topic relevant to individual interests, with all members contributing equally. The topics are up to you but should be at least marginally related to astronautics. Below are suggestions to kick off brainstorming. Also see https://sites.google.com/usc.edu/aste-404/projects for examples of past projects.

Examples:

- HTML+Javascript site parsing data from a NASA mission repository
- Arduino-based robot that uses a light sensor for navigation
- Script using machine learning to identify coefficients that improve fit to some input data
- Particle-based simulation of galaxy formation
- Timing study of GPU or MPI acceleration to determine optimal decomposition parameters
- Code processing USB microscope images to identify dust particle sizes
- Gas flow solver running on a cell phone
- Molecular dynamics simulation of molecular contamination buildup

Project Timeline:

- Week 8: Project proposal, identifying team members, project objectives, and completion timeline
- Week 13: Project update summarizing accomplishments to date and remaining work
- Final: in-person presentation, delivery of HTML report and slides

Grading breakdown of the course project:

- **Proposal (20%):** One to two page document outlining project title, team members, planned project objectives, work distribution, and completion time line.
- **End of term report (20%):** Word/PDF report describing work done to date along with preliminary results. Also identify the remaining work and the completion strategy.
- **Final Presentation (20%):** 10-15 minute long in class presentation along summarizing your project, along with lessons learned / future work.
- **V&V (15%):** Description of code validation (as much as possible) and/or unit testing discussed in both your presentation and final report. For a simulation code, I would like you to compare the code to some simplified analytical case and to also

perform mesh and/or time step sensitivity studies. For an embedded / hardware design, I would like you to characterize the domain over which the design can be expected to operate normally.

- **Final Report (20%):** Final report, describing your project and team members, objectives, summary of outcomes / results, verification and validation (separate grading rubric, see below), and future work. This report is to be written as a standalone HTML (and possibly Javascript) website, and delivered as a .zip file containing all the required media and style sheets. It will be posted online on the class website. Due two days after the final presentation.
- **Code Delivery (5%):** Delivery of the developed source code. Code needs to contain comments, ideally with Doxygen, and also contain a "readme" file with compilation and usage instructions. Due two days after the final presentation.

Note: your code or device does not necessarily need to work. Many tasks take longer than expected. However, you still need to complete the above items. For instance, you may be able to run V&V on some reduced case. Your report / website can describe the difficulties you encountered and summarize any proposed solutions that could be investigated as part of future work.

Statement on Academic Conduct and Support Systems

Instructor note on use of AI technologies:

You are allowed to utilize technologies such as ChatGPT as an "enhanced search engine". In other words, you can use it to find information you could otherwise find by Googling or vising websites such as StackExchange, that is subsequently used *by you* to complete the assignment. You may not use these these technologies to complete the work for you.

Academic Conduct:

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in SCampus in Part B, Section 11, "Behavior Violating University Standards" policy.usc.edu/scampus-part-b. Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct, policy.usc.edu/scientific-misconduct.

Support Systems:

Counseling and Mental Health - (213) 740-9355 – 24/7 on call
studenthealth.usc.edu/counseling
Free and confidential mental health treatment for students, including short-term psychotherapy, group counseling, stress fitness workshops, and crisis intervention.

National Suicide Prevention Lifeline - 1 (800) 273-8255 – 24/7 on call
suicidepreventionlifeline.org
Free and confidential emotional support to people in suicidal crisis or emotional distress 24 hours a day, 7 days a week.

Relationship and Sexual Violence Prevention and Services (RSVP) - (213) 740-9355(WELL), press "0" after hours – 24/7 on call
studenthealth.usc.edu/sexual-assault
Free and confidential therapy services, workshops, and training for situations related to gender-based harm.

Office of Equity and Diversity (OED)- (213) 740-5086 | Title IX – (213) 821-8298
equity.usc.edu, titleix.usc.edu
Information about how to get help or help someone affected by harassment or discrimination, rights of protected classes, reporting options, and additional resources for students, faculty, staff, visitors, and applicants. The university prohibits discrimination or harassment based on the following *protected characteristics*: race, color, national origin, ancestry, religion, sex, gender, gender identity, gender expression, sexual orientation, age, physical disability, medical condition, mental disability, marital status, pregnancy, veteran status, genetic information, and any other characteristic which may be specified in applicable laws and governmental regulations. The university also prohibits sexual assault, non-consensual sexual contact, sexual misconduct, intimate partner violence, stalking, malicious dissuasion, retaliation, and violation of interim measures.

Reporting Incidents of Bias or Harassment - (213) 740-5086 or (213) 821-8298
usc-advocate.symplicity.com/care_report
Avenue to report incidents of bias, hate crimes, and microaggressions to the Office of Equity and Diversity |Title IX for appropriate investigation, supportive measures, and response.

The Office of Disability Services and Programs - (213) 740-0776
dsp.usc.edu
Support and accommodations for students with disabilities. Services include assistance in providing readers/notetakers/interpreters, special accommodations for test taking needs, assistance with architectural barriers, assistive technology, and support for individual needs.

USC Support and Advocacy - (213) 821-4710
uscsa.usc.edu
Assists students and families in resolving complex personal, financial, and academic issues adversely affecting their success as a student.

Diversity at USC - (213) 740-2101
diversity.usc.edu

Information on events, programs and training, the Provost's Diversity and Inclusion Council, Diversity Liaisons for each academic school, chronology, participation, and various resources for students.

USC Emergency - UPC: (213) 740-4321, HSC: (323) 442-1000 – 24/7 on call
dps.usc.edu, emergency.usc.edu
Emergency assistance and avenue to report a crime. Latest updates regarding safety, including ways in which instruction will be continued if an officially declared emergency makes travel to campus infeasible.

USC Department of Public Safety - UPC: (213) 740-6000, HSC: (323) 442-120 – 24/7 on call
dps.usc.edu
Non-emergency assistance or information.