ASTE 404, Computational Programming and Numerical Methods

Units: 3
Term: Fall 2023
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: 10-11:30am + online by appt
Contact Info: brieda@usc.edu

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

IT Help: N/A
Hours of Service: N/A
Contact Info: N/A

Course Description
My goal with this course is to introduce you to the ecosystem of computational tools and techniques relevant to aerospace and astronautical engineering that I picked up over the past 15 years of my career as a developer of simulation codes for the space environment and plasma physics. 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. During the course, each student will develop and implement a simulation code for solving a numerical problem of interest. Prerequisite: basic programming background.

Catalogue 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 simulation codes for spacecraft contamination transport, plasma propulsion, and fusion applications through my company Particle in Cell Consulting LLC. Some examples of my past projects include: analyzing the amount of ice buildup during the deployment of JWST for NASA, developing a Hall effect thruster simulation module for a plasma simulation framework used by AFRL, helping the Aerospace corporation select pumping configuration for a new vacuum chamber, performing thruster plume impingement studies for numerous satellites, aiding in due diligence studies for fusion start ups, and (currently) supporting the JHU/APL Dragonfly mission to fly a rover on Saturn's moon Titan. I also co-advise the undergraduate spacecraft propulsion club ASPEN.

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: Basic programming experience with Matlab, C/C++, Python, or other programming languages, and/or similar exposure on the level of ITP115, or ITP165.

Course Notes:
Lecture notes provided by instructor based on Brieda, Wang, Martin, *Introduction to Modern Scientific Programming and Numerical Methods*, CRC Press, 2023 (in preparation).
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 assigned at the start of the following class. It generally consists of "follow along" programming assignments. You need to complete only 6 assignments for a full credit. The literature review is based on finding an interesting computational journal or conference paper, and preparing a short (10 minutes / 4 slides) presentation. Several multiple choice / fill in the blanks quizzes will be used to test understanding of previously covered topics. The quizzes will be assigned approximately every other week and will be primarily take-home. 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
8/24	1	**Introduction to Scientific Computing.** We start by reviewing numerical simulations relevant to astronautics, and mechanical/aerospace engineering. Basic concepts such as variables, loops, conditions, and functions are also introduced. Sample code is implemented in Matlab to demonstrate the graphical user interface for running the code and visualizing results. *Reading: Ch. 1*
8/31	2	**Discretization Schemes.** Finite Difference is introduced. We use this method to develop a solver for the diffusion (heat) equation in 2D in Matlab and Python. We learn about Google Colab, and NumPy, SciPy, and MatplotLib packages *Reading: Ch. 2*
9/7	3	**Web technologies.** We learn how to develop an interactive Javascript version of the diffusion equation solver that runs in a web browser. *Reading: Ch. 5*
9/14	4	**Compiled Languages.** We start introducing C. We cover important concepts such as variable types, arrays, functions, dynamic memory, compilation, and the use of development environments. We also introduce the Linux command-line environment. Reading: Ch. 1
9/21	5	**Linear Solvers and Visualization**. We cover popular matrix solvers such as Jacobi and Gauss-Seidel Iteration, and the Thomas Algorithm. We see how to implement these methods in Python and C++. We also cover data output and visualization in Paraview. Reading: Ch. 2
9/28	6	**Object Oriented Programming**. Data encapsulation, inheritance, virtual functions, and operator overloading are introduced in C++ and are used to simplify syntax of our diffusion equation solver. We also cover the implicit Crank-Nicolson integration scheme. *Reading: Ch. 4*
10/5	7	**Code Testing and Documentation.** We cover uncertainty analysis, convergence studies, unit testing, version control, linear algebra libraries, documentation systems, and LaTeX. Reading: Ch.6
10/12	-	**Fall Recess, No Class**
10/19	8	**Numerical Analysis.** This lesson covers few additional important topics related to numerical analysis including interpolation, smoothing, signal processing, and Newton-Raphson method. We compare implementation in Python and C++. *Reading: Ch. 3*
10/26	9	**Gas Simulation Methods.** Fluid and kinetic methods for gas modeling are compared. We introduce the advection-diffusion equation, the Finite Volume Method, Crank Nicolson, stability analysis, DSMC, PIC. *Reading: Ch. 7 and 8*
11/02	10	**Parallel Processing.** We learn how to reduce computational time by utilizing multiple CPU cores (multithreading) and clusters of interconnected computers (MPI). *Reading: Ch. 9*
11/09	11	**Graphics Cards.** The use of graphics cards to perform computations with the CUDA C++ language extension is discussed. We learn about main bottlenecks, and compare performance. *Reading: Ch. 9*
11/16	12	**Embedded Systems.** We introduce the Arduino platform, and demonstrate how to interface with an external sensor. We then build a solver running on the microcontroller and interfacing over the serial port. Verilog and the use of FPGAs/ASICs on flight hardware is discussed. Reading: Ch. 11
11/23	-	**Thanksgiving, No Class**
11/30	13	**Optimization and Machine Learning.** Genetic algorithms and neural networks are reviewed. We see how to implement a simple neural network in Python (with TensorFlow) and C++. *Ch. 10*
12/12 (Tue)	-	**Final Project Presentation, 11am – 1pm (most likely online)**

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. There may be some deviation from this schedule due to my expected conference travel, mostly in September. During those weeks, I will run the lesson remotely over Zoom, and possibly reschedule to a different day.

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, and you will receive full credit as long as you attempt all parts, even if some parts don't work for whatever reason. This year, as an experiment, you will need to complete only 6 of the assignments to receive full credit. This should help free some of your time to focus ton the project.

Quizzes

We will have about 6 "test your understanding" open book quizzes. Most of these will be take home. The quizzes usually consist of about 10 multiple choice, true/false, matching, and short answer type of questions. They are due by the following class. They need to be individual work.

Paper Review

Use websites such as Google Scholar to find a relatively recent academic article on an astronautical topic in which the authors performed numerical analysis. Put together a short 3-4 slide presentation describing the paper, the implemented numerical model, the obtained results, and (important!) a short summary of things you found to be wrong or missing in the paper. Historically, I would ask you to present the paper to class live, but depending on the enrollment, you may be asked to upload a video file recording of your presentation.

Course Project

The project is the main focus of the class. Its purpose is to practice programming skills learned in the class, but also to work with your group on meeting the deadline and deliverables. This project is envisioned to be completed in groups of 2-4 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:

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

Project Timeline:

- Week 3: Identify team members and decide on the topic
- Week 6: Proposal due
- Week 13: End of term update due
- Final: Presentation, summary website and code due

Grading breakdown of the course project:

- **Proposal (20%):** Deliver a short document outlining project title, team members, planned project objectives, work distribution, and work load distribution. 2-3 page long Word/PDF document.
- **End of term report (20%):** Update of the prior report describing work done to date and the remaining work. 2-3 page long Word/PDF document.
- **Final Presentation (10%):** 10-15 minute long presentation along with slides delivered summarizing your project, along with lessons learned / future work

- **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 the day of our “final”.
- **Code Delivery (10%):** Code needs to contain comments, ideally with Doxygen, and also contain a "readme" outlining compilation / use instructions. Due at the end of the semester, along with the final report.
- **V&V (20%):** Description of code validation (as much as possible) and/or unit testing. For a simulation code, I would like to see some write up on using the code to solve a case with an analytical solution. Where not possible, perform and describe some convergence testing. In other words, you need to convince me that the code is capable of producing results that can be trusted. This description is to be included as part of the final report .html writeup.

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. You can also describe the testing you anticipated to complete. Your report / website needs to 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

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.