ASTE 404, Computational Programming and Numerical Methods

Units: 3
Term: Fall
Day & Time: Tuesday: 12-2:50 PM
Location: TBD

Instructors: Dr. Lubos Brieda & Dr. Joseph Wang

Office: TBD
Office Hours: Tuesdays: 10-11:30am + by appointment
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

This course introduces students interested in numerical analysis to the ecosystem of programming languages, numerical methods, analysis tools, and hardware technologies encountered in the world of scientific computing. We start by reviewing Matlab before moving onto other frequently encountered languages, such as Python, C/C++, Java, and Javascript. Along the way we cover numerical techniques for solving problems encountered in astronautics and aerospace engineering, and learn how take advantage of modern hardware capabilities including multithreading, graphics cards (GPUs), computer clusters, microcontrollers, and FPGAs. Additional topics include object oriented programming, data visualization, 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.

Course Goals and Objectives:

This course combines materials from engineering numerical analysis with that from computational programming to provide a comprehensive study on programming languages and computational methods used in astronautics and aerospace engineering. The material goes beyond the Matlab based numerical method classes typically taught at Viterbi and focuses on the use of high-performance programming languages. 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.

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.

Co-Requisite(s): None
Concurrent Enrollment: None

Course Notes:

Lecture notes provided by instructor based on Brieda, Wang, Martin, *Introduction to Modern Scientific Programming and Numerical Methods*, CRC Press, 2021 (in preparation).

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. The literature review is based on finding an interesting computational journal or conference paper, and presenting a short (10 minute) summary. Several multiple choice / fill in the blanks quizzes will be used to test understanding of previously covered topics. The quizzes will be held approximately every other week. The final project involves working in small groups to develop a simulation program relevant to research interests.

Grading Breakdown:

	Points	% of Grade
Assessment Tool (assignments)		
Homework		40%
Quizzes		20%
Literature Review		10%
Final Project		30%
TOTAL		100%

ASTE 404: Computational Programming -- Course Schedule

Week	Topics/Daily Activities
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*
2	**Discretization Schemes.** Finite Difference is introduced. We use this method to develop a solver for the diffusion (heat) equation in 2D. We continue to use Matlab. *Reading: Ch. 2*
3	**Introduction to Python.** We learn how to install and use Python. We learn about Jupyter Notebooks, and see how to use NumPy, SciPy, and MatplotLib packages to replicate Lesson 2 results. *Reading: Ch. 1*
4	**Compiled Languages.** Performance of Matlab and Python cannot compare to those of compiled (or semi-compiled) languages such as C. We cover important concepts such as variable types, arrays, functions, dynamic memory, compilation, and the use of development environments in Fortran, C, C++, and Java. Here we briefly introduce the Linux command-line environment. *Reading: Ch. 1*
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, Fortran, and C++. We also cover data output and visualization in Paraview. *Reading: Ch. 2*
6	**Object Oriented Programming**. Data encapsulation, inheritance, virtual functions, and operator overloading are introduced in C++. These concepts are used to simplify the syntax of our diffusion equation solver. We also cover the implicit Crank-Nicolson integration scheme. *Reading: Ch. 4*
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*
8	**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	**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	**Fluid Modeling.** The diffusion equation we have been solving is an example of a simple fluid model. The advection-diffusion equation and the SIMPLE method for solving incompressible flows are introduced. We also cover the Finite Volume method and stability analysis. *Reading: Ch. 7*
11	**Stochastic (Monte Carlo) Methods.** Next we introduce Lagrangian particle methods based on picking random numbers. This lesson briefly introduces the DSMC and PIC methods. *Reading: Ch. 8*
12	**Parallel Processing.** We learn how to reduce computational time by utilizing multiple CPU cores (multithreading) and clusters of interconnected computers (MPI). *Reading: Ch. 9*
13	**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*
14	**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*
15	**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*
Final	**Final Project Presentation**

ASTE 404 - Course Project

Course project: The purpose of the class project is to practice developing a simulation program for conducting academic research. 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 sample topics provided if needed. All members are expected to contribute equally, with tasks divided among code development, testing, and documentation. Deliverables to include the code, preliminary results, user's guide outlining the numerical model, input files, and a report similar to a 10-15 page conference paper.

Project Timeline:

- Week 3: Identify team members and project topics
- Week 6: Proposal due (team member, topics and milestone)
- Week 8: Mid-term report due (code layout, test cases, preliminary results, draft documentation)
- Week 14: Project presentation (open to all faculty and students)
- Final: Final report due (problem statement, algorithm description, validation, major discovery, lessons learned)

Sample project: *The ESA ExoMars Rover mission is planning to use an ion mass spectrometer to investigate the Martian soil composition. The spectrometer uses a laser beam to desorb ions off a soil sample held at Mars atmospheric pressure. Above the sample is a long "straw" leading to a vacuum cavity containing the mass spectrometer. A sliding valve opens during the laser operation. The resulting pressure gradient accelerates the ambient gas molecules into the vacuum cavity, with ions becoming entrained in the gas flow due to collisional coupling. The objective of the project is to use techniques such as PIC and DSMC to investigate this ion and neutral gas transport.*

Grading breakdown of the course project*:*

- Proposal: 5%
- Mid-term report: 5%
- Final report: 5%
- Presentation: 10%
- Code and user's guide: 5%

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.