AMSC663/CMSC663: Advanced Scientific Computing I - Fall 2020

Course description

AMSC/CMSC 663-664 is a two-semester project course in which each student will identify and carry out a scientific computing project with a focus on:

- Understanding of scientific computing algorithms related to the project.
- Code development, which could include
    - Modularity, portability, memory management
    - Post-processing, restarting, and writing to databases
    - Interactivity and effective scientific visualization
    - Proper documentation, version management tools, and accessibility
    - Debugging and profiling tools
    - Validation, verification, and unit testing.

Each project must be focused on computing and involve the development of high-quality computer code implementing contemporary or original numerical techniques.

Each project is for academic use only. Data used in each project must be publicly available.

Logistics

Classes: Tue-Th 12:30--1:45, online via Zoom

Maria Cameron, email: mariakc@umd.edu

Radu Balan, email: rvbalan@cscamm.umd.edu

Course web site: umd.instructure.com mirrored at

https://www.math.umd.edu/~mariakc/advanced-scientific-computi.html (Links to an external site.)

Prerequisites

AMSC/CMSC 660 and 661 or similar scientific computing/numerical analysis graduate classes

Course requirements

- **Each student must have a faculty advisor (who is not an instructor of this course)**

- Deliverables:
    - Written documents and oral presentations (see Section Presentations and proposal documents below)
        - Project proposal (5-7 page document, due Oct. 8)
        - Proposal presentation (20 min oral presentation)
        - Mid-year report (an edition of project proposal plus current results, due Dec. 10)
        - Mid-year presentation (20 min oral presentation)
    - Code posted on GitHub (Links to an external site.) (if appropriate) and accompanied with documentation (see Section Code requirements) (due Dec. 16)
- Weakly update email should be sent by each student to the instructors.

Grading

- Projects will be graded on the quality of
    - code and documentation for it;
    - written documents: project proposal and mid-year report;
    - oral presentations: proposal presentation and mid-year presentation.
- Oral presentations and written documents should reflect:
        - critical thinking,
        - ability to formulate and achieve research goals,
        - ability to identify and overcome difficulties,
        - newly acquired technical skills.
    - Note that a solid effort without significant technical advances will **not** result at a good grade.

Code requirements

- Code should be well-organized, clean, loosely coupled, and extensible
- Code should be tested and validated.
- Code should be documented, well-commented, and accompanied with a user-friendly guide.
- Code should be distributable via GitHub (Links to an external site.) if appropriate

Oral and written presentation requirements

- Project proposal and presentation should include the following components.
    - Background on the problem being addressed.
    - Why is this problem important?
    - What are state-of-art methods for solving it? Provide refs.
    - Project goals: what are you hoping to achieve?
    - Approach. How will you achieve these goals? What components will need to be implemented to get there?
        - Describe specific algorithms and how they will be implemented.
        - Describe hardware/software platform you target. What programming languages will be used.
    - Validation methods: how you plan to test your code.
    - Deliverables: specific components of the code you plan to develop.
    - Milestones and a rough timeline.
- Mid-year presentation and report should include the following items.
    - Shorter versions of the first five bullet points above.
    - Detailed description of what has been accomplished.
    - Description of what has not been accomplished and why.
    - A link to the completed code and documentation for it.
    - Itemized list of deliverables such as code, data, code documentation.
    - A research plan for AMSC664

Course calendar

Sept. 1: an introductory class, setting up course goals and expectations; self-intro presentations by the instructors.

Sept. 3 -- Sept. 10: students' self-intro presentations.

Sept. 3 — Sept. 28: project kick-off meetings: the student, his/her project advisor, and the instructors.

Sept. 15: ***deadline for picking a project and an advisor.***

Sept. 22 — Oct. 8: project presentations.

Oct. 8: ***project proposal is due.***

Oct. 26 — Nov. 6: individual meetings with students for code review.

Nov. 16 — Dec. 3: individual meetings with students for code review.

Dec. 1 — Dec. 10: mid-year oral presentations.

Dec. 16: ***mid-year report is due.***

Academic integrity

- Sources such as text and figures must be properly cited.
- You can import and use third-party libraries in your code if it is rational. In this case, provide a full and due credit to the third party.
- Link to the UMD Code of Academic Integrity: https://www.studentconduct.umd.edu/academic-dishonesty