Syllabus
MATH664/AMSC664 Spring 2013
Advanced Scientific Computing II (ASC II)

Instructors: Radu Balan, Kayo Ide
Classes: Tuesday, Thursday, 5:15pm – 6:30pm (or as needed) in CSIC 4122
Office Hours: Radu Balan: Wednesday, 1-2pm in CSCAMM; Kayo Ide: Tuesday, 4:00-5:00pm in CSCAMM; or by appointment otherwise

Contact Information:
Kayo Ide: ide@umd.edu, (AOSC) CSS Building, Room 3403, x50491 / (CSCAMM) CSIC Building, Room 4127, x50162
Radu Balan: rvbalan@math.umd.edu, Math Building, Room 2308, x55492 / (CSCAMM) CSIC Building, Room 4131, x51217

Prerequisites: AMSC 663 (ASC I)

This course is a continuation of AMSC 663 (ASC I). This course focuses on issues of code development and validation relevant to large scientific computing projects on high performance computing systems.

Students’ responsibilities in AMSC664:

- Students will provide weekly statements of their project status. The statements are due before the beginning of the scheduled weekly class period.
- Students will prepare and give a 15-20 minute oral presentation between March 1st and March 15th detailing the status of the project. At the time of this presentation the implementation must be in its final stages or complete.
- Students will prepare and give a comprehensive 30-minute final project presentation and submit the final project report at the end of the course. This project report should be comprehensive. Your advisor’s presence is required. Students are advised to start preparing this report well ahead of the last class to allow for feedback and corrections to occur before final submission.

Students are responsible for achieving the project goals that were listed in the proposal. Grading takes into account student’s understanding of the project, its implementation, timing of the weekly reports, presentation quality and the final report quality.

The class will meet once a week in CSIC 4122 as needed, except during those times when presentations are scheduled and meetings may be more often.

Tentative lecture topics:

- Large scale optimizations, particularly quadratic programs and implementation issues.
- Examples of Parallel Algorithms and Code Development
Several examples of algorithm development and implementation from instructors’ areas of research (e.g. OpenMP)

- Computational Science Seminars
  - Local and visiting computational scientists representing a broad selection of scientific disciplines will explain the computational aspects of their work with emphasis on code development and validation. Each student is expected to listen to these seminars critically.

Students are encouraged to meet with the instructors by appointment to discuss their progress.