Math 447 - Introduction to Parallel Computing

Summer 2014

Matthias K. Gobbert, Kofi Adragni, Bradford E. Peercy, and Nagaraj K. Neerchal

---

This page can be reached via my homepage at http://www.math.umbc.edu/~gobbert.

This senior-level three-credit course is integrated with the NSF-funded REU Site: Interdisciplinary Program in High Performance Computing. The participants of the REU Site will take this course as part of their program. Several additional seats are available to the general public as part of UMBC's Summer Program. See the bullet on Schedule below for an explanation of the 2014 Schedule of class meetings for this course. This course is set up as consent required course. **Please contact the instructor for questions and for permission at gobbert@umbc.edu as soon as possible.** *Notice that the class meetings start on June 16, 2014, but since the class is set up as a 12-week summer session, you must register much earlier, namely by approximately May 20, 2014, to avoid issues with a late fee.*

Basic Information

- Instructors:
  - Matthias K. Gobbert, Math/Psyc 416, (410) 455-2404, gobbert@umbc.edu
  - Kofi Adragni, Math/Psyc 411, (410) 455-2406, kofi@umbc.edu
  - Bradford E. Peercy, Math/Psyc 436, (410) 455-2436, bpeercy@umbc.edu
  - Nagaraj K. Neerchal, Math/Psyc 437, (410) 455-2437, nagaraj@umbc.edu

  See also the 2014 People tab at the webpage of the REU Site: Interdisciplinary Program in High Performance Computing.
- Schedule: Please see the tab 2014 Schedule at the webpage of the REU Site: Interdisciplinary Program in High Performance Computing. This course consists of the 29 numbered, boldfaced classes plus associated labs listed in the 2014 Schedule of the REU Site.
- Prerequisites: Math 341 or 441, Stat 355, CMSC 201, proficiency with a high-level programming language, and familiarity with the Unix/Linux operating system, or permission of instructor; please contact the instructor with information about your background to ensure that you are ready to take this course.
- Copies of the following books are on reserve in the library.
  - Required textbook on parallel computing: Peter S. Pacheco, *Parallel Programming with MPI*, Morgan Kaufmann, 1997. Associated webpage: http://www.cs.usfca.edu/~peter/ppmpi.

    We will have explicit reading assignments for several chapters from this book.
  - Recommended book on the programming language C: Brian W. Kernighan and Dennis M. Ritchie, *The C Programming Language*, second edition, Prentice-Hall, 1988. Associated webpage: http://cm.bell-labs.com/cm/cs/cbook/.
    This is the classic book on C written by its creators.
  - Recommended book on Matlab: Desmond J. Higham and Nicholas J. Higham, *Matlab Guide*, second edition, SIAM, 2005. The associated webpage http://www.ma.man.ac.uk/~higham/mg includes updates, code, and a list of errors.
  - Recommended book on professional issues in mathematics and the sciences: Nicholas J.

Higham, *Handbook of Writing for the Mathematical Sciences*, second edition, SIAM, 1998.
  - Recommended book on LaTeX: Leslie Lamport, *LaTeX: A Document Preparation System*, second edition, Addison-Wesley, 1994.
    Introduction to LaTeX by the author himself.
- Grading policy:

Homework and Quizzes	Participation
90%	10%

  - The *homework* includes the computer assignments that are vital to understanding the course material. A late assignment accrues a deduction of up to 10% of the possible score for each day late until my receiving it; I reserve the right to exclude any problem from scoring on late homework, for instance, if we discuss it in class. The *quizzes* will generally be unannounced and brief and will include the use of learning groups formed by the instructor. For instance, they may be designed to initiate class discussion or to give me feedback on your learning. They may be technical or non-technical in nature.
  - The graded *participation* component rewards your professional behavior and active involvement in all aspects of the course. Examples of expected professional behavior include attending class regularly, reading assigned material when requested, cooperating with formal issues such as submitting requested material on time, and participating constructively in class, specifically in group work. In this course, professional behavior also includes adhering to good user behaviors on the shared computing facilities that you will work on.

  All assignments will be conducted by student teams assigned by the instructor for the duration of the course. It is vital that you are willing to participate actively by giving and receiving feedback with team members. Additional details or changes will be announced as necessary. See also general policies and procedures for more information.
- Blackboard is a course management system that allows for posting and communicating among registered participants of a course. We will actively only use the "Course Documents" area of our course in Blackboard. I will post class summaries and PDF files of the lecture notes as well as other material including the homework assignments in this area. I will also use Blackboard to send e-mail to the class, which goes to your UMBC account by default. Therefore, you must either check your UMBC e-mail regularly or have the mail forwarded to an account that you check frequently.
- **Homework submission:** Each homework is to be submitted as one single PDF file attached to an e-mail the appropriate instructor, with the attachment name to include both the number of the homework and your unique team name. For example, if you are Team 1, the submission of Homework 2 would use the filename "HW2_Team1.pdf". Please include this information also in the Subject field of the e-mail, such as "HW 2 from Team 1". If in doubt, contact me about these issues; the goal is to have clear and unique filenames when I download the files from the e-mail system. Please also make sure that the From field of your e-mail shows your full name clearly, not just your username or some other internet handle; in order to guard against confusion about the sender of the mail, I require the information in the Subject field, as stated above. The contents of e-mail for homework submission should be essentially empty, since I have no way to integrate this with your PDF file; make sure that all contents for grading is in the PDF attachment! But a sensible, short message, signed by your full name is of course useful to guard against misunderstandings. Please submit each homework only once; if you send several copies, I cannot guarantee which one I end up grading. Cc all team members on your submission.

Course Description

Parallel computing has become an ubiquitous way to perform computer simulations involving large amounts of data or intensive calculations. The basic purpose of using several processors is to speed up

computations of large problems by distributing the work. But large problems typically involve vast quantities of data as well; by pooling the memory from several processors, problems of previously unsolvable size can now be tackled in reasonable time.

This course will introduce the basic aspects of parallel programming and the algorithmic considerations involved in designed scalable parallel numerical methods. The programming will use MPI (Message Passing Interface), the most common library of parallel communication commands for distributed-memory clusters. Several application examples will demonstrate how parallel computing can be used to solve large problems in practice. We will also consider the options for multi-threading on multi- and many-core CPUs and for using graphics processing units (GPUs).

Registered students in this course will gain access to the state-of-the-art resources in the UMBC High Performance Computing Facility (HPCF; www.umbc.edu/hpcf). This class is intended to familiarize students with this cluster, if you expect to use it for your research in the future. One of the side benefits of this class is to help in the creation of a user community on campus.

The class will include an efficient introduction to the Linux operating system as installed on the HPCF cluster, and it will include a review of serial programming in the source code language C that is integrated into the initial presentation of sample codes. This review assumes some experience with compiling and debugging in a high-level source code programming language. It will only include a restricted set of features of C, but these are selected to be sufficient for work on the homework assignments in the class.

Learning Goals

By the end of this course, you should:

- understand and remember the key ideas, concepts, definitions, and theorems of the subject. Examples include understanding the purpose of parallel computing and why it can work, being aware of potential limitations, and knowing the major types of hardware available. This information will be communicated in class and in the textbook, but also in additional reading.
  --> This information will be discussed in the lecture as well as in the textbook and other assigned reading.
- have experience writing code for a Linux cluster using MPI in C, C++, and/or Fortran that correctly solves problems in scientific computing. The sample problems are taken from mathematics and your code has to compile without error or warning, run without error, and give mathematically correct results first of all. In addition, it needs to run on a Linux cluster without error and you need to be able to explain its scalability, i.e., why or why not it executes faster on several processors than in serial. We will have problems stated in different ways and from various sources to provide you with exposure to as many issues as possible.
  --> This is the main purpose of the homework and most learning will take place here.
- have gained proficiency in delivering code written by you to others for compilation and use. This includes the concept of providing a README file that gives instructions how to compile and run the code as well as of providing a sample output file to allow the user to check the results. We will work together in class to discuss best practices to transfer code for homework problems of increasing complexity.
  --> You will submit your homework code by e-mail to the instructor and it needs to compile and run in parallel for credit; this is complemented by a report that shows and explains your results.
- have some experience how to learn information from a research paper and to discuss it with peers.

Group work requiring communication for effective collaboration with peers and supervisors is a vital professional skill, and the development of professional skills is a declared learning goal of this course.
--> I will supply some research papers carefully selected for their readability and relevance to the course. Learning from research papers is a crucial skill to develop.

- have experience with independent work and presenting it both in a written report and in an oral presentation. It is vital to gain experience with setting your own goals, estimating a realistic time line, working with peers and supervisors on regular updates and giving and receiving suggestions, submitting and editing a written report to standards of a research paper, and presenting your results in an oral presentation as part of a conference session.
--> The homework will cover all of these components, and we will discuss the various steps necessary in class.

Other Information

- Please see the tab 2014 Schedule at the webpage of the REU Site: Interdisciplinary Program in High Performance Computing. This course consists of the 29 numbered, boldfaced classes listed in the 2014 Schedule of the REU Site.
- General policies and procedures including grading guidelines
- Guidelines on how to report on computer results

- The recommended literature on my homepage
- A brief introduction to Unix/Linux at UMBC
- Introduction to Matlab at UMBC (maintained by CIRC)
- An introduction to LaTeX including a sample file and a template file for project reports.

- UMBC High Performance Computing Facility (HPCF) including general information, list of projects, publications page, and resources for users.
- Center for Interdisciplinary Research and Consulting (CIRC)

UMBC Academic Integrity Policy

By enrolling in this course, each student assumes the responsibilities of an active participant in UMBC's scholarly community in which everyone's academic work and behavior are held to the highest standards of honesty. Cheating, fabrication, plagiarism, and helping others to commit these acts are all forms of academic dishonesty, and they are wrong. Academic misconduct could result in disciplinary action that may include, but is not limited to, suspension or dismissal. To read the full Student Academic Conduct Policy, consult the UMBC Student Handbook, the Faculty Handbook, the UMBC Integrity webpage www.umbc.edu/integrity, the UMBC Undergraduate Student Academic Conduct Policy (PDF) for undergraduate students, or the University of Maryland Graduate School, Baltimore (UMGSB) Policy and Procedures for Student Academic Misconduct (PDF) for graduate students.

This page version 1.2, June 2014.