**CPS644** **Fall, 2008** MW 5:30 – 7:00 PM
**Supercomputing and Applications** 114 Holmes

**Instructor:**
Dr. Robert E. Tuzun
128 Smith Hall
Office/FAX: 395-5368/5020
email: rtuzun@brockport.edu

**Office hours:**
Wed 1:00 – 3:00 PM
Thurs 1:00 – 3:00 PM
and by appointment

My Web page: http://www.cps.brockport.edu/~tuzun
Course Web page: http://www.cps.brockport.edu/~tuzun/courses/cps644/cps644.html

**Course materials**

MPI 1.1 standard from the Web (an excellent resource).

Ananth Grama *et. al.*, *Introduction to Parallel Computing, 2nd ed.* (Benjamin/Cummings, 2003).

**Course description**

An introduction to parallel computing, with an emphasis on the how-to rather than the theory. Parallel computing is the use of two or more processors to perform some task or computation. This may be necessary because the entire problem is too big to fit on a single machine, or it would take an unacceptably long time on a single machine. One of the best illustrations of parallel computing is weather prediction. Essentially, the map is divided into several geographical regions, and each region assigned its own processor. Each processor is responsible for all computations covering its region, and for sending/receiving any required information from other processors to allow, for example, weather fronts to move between regions. The processors working concurrently can perform the calculation in a short enough time to be useful, whereas a single machine would take too long to perform the same calculation.

Although some theoretical underpinnings will be given, the focus of this course will be practical. Much of the course will concentrate on the application of the MPI (Message Passing Interface) protocol for interprocessor communication that has been implemented on a wide variety of parallel architectures. The course will also focus on practical considerations that are necessary for performing the computations most efficiently. The course will cover parts of the main text and most of the MPI standard. All programming will be in Fortran 90. Topics include:

(1) Definition of, historical context for, and general capabilities of parallel computing.
(2) Models of parallel computing: architectures, interconnection networks, embedding, communications costs, and so on.
(3) Design of simple parallel algorithms and basic MPI calls: point-to-point communication, blocking versus non-blocking communication, reduction and broadcast operations.
(4) Collective communication operations, mostly gather and scatter.
(5) Data packing and user-defined data types.
(6) Communicators and topologies.
(7) A variety of algorithms with varying degrees of complexity (integration, matrix multiplication, sorting, graph algorithms, etc., depending on time).
(8) Performance issues.

You will work on the department's PC cluster, which has a Linux computing environment. If you are not already proficient in the use of Unix or Linux, you should become so immediately.

**Homework**

Homework assignments will be given every one to two weeks, including one assignment due during finals week. These will include

- *Paper and pencil problems.*
- *Programming*—modification of code provided to you, or writing some simpler codes from scratch. Concepts to be developed include generality, robustness, accuracy, and efficiency, as well as bookkeeping issues.

Homeworks will consist of short written presentations that you hand in, as well as any accompanying files on your computer account. Projects will require longer presentations and more elaborate calculations; specific guidelines will handed out with each project. Please follow the following guidelines.

- For any question that calls for interpretation of numerical results, please provide succinct written presentation. It is not enough to just provide numerical results; you must demonstrate that you understand the results.
- You may sometimes want to show raw data or diagnostic output in order to make some point. In your written presentations, please show snippets as short as possible, since the full output may consist of thousands of lines.
- On each graph you generate, be sure to include axis labels with units, as well as a title.
- Maintain a separate directory for this course. Organize your directory space so that each homework assignment or project has its own directory. You may sometimes find it advantageous to make subdirectories.
- In order for me to be able to provide the best feedback possible, I will want to be able to follow and to reproduce your work (including all numerical results). Please be sure to save all the source code, shell scripts, etc., required to generate your data, as well as output files. If you have a large number of files, please provide a table of input and output files (with their most relevant run parameters) and names of source codes, shell scripts, and any other relevant files.
- Please adhere to coding style guidelines shown later in this syllabus.

**The Web**

The course Web page will be updated frequently. In addition to being passed out in the classroom, assignments and handouts will be posted on the course Web page. Answer keys, class announcements, and so on will also be posted. No grades will be accessible from the Web.

**Grading**

Two projects: 40%, weighted equally.
Homework: 60%, weighted equally.
There will be no tests or final exam.

Late homework/projects will penalized 25% per day.

Writing/presentation skills and coding style and readability will count toward the grades. Your coding style does not have to be the same as mine; however, your code does need to be readable and to have a consistent style.

I am a great believer in partial credit. I can't give credit for a one-line wrong answer, but if you show your work, you will receive partial credit for what you did correctly. (You'll also receive higher quality, more targeted feedback).

Final grades will be assigned according to the following cutoffs. The cutoffs may be lowered under some circumstances, but they will never be raised.

| Grade | Average, % | Grade | Average, % |
|---|---|---|---|
| A | 93.0 & above | B- | 80.0-82.9 |
| A- | 90.0-92.9 | C+ | 77.0-79.9 |
| B+ | 87.0-89.9 | C | 73.0-76.9 |
| B | 83.0-86.9 | E | Below 73.0 |

**Coding style guidelines**

In order to make it as easy as possible to understand and grade your work, I am gradually introducing a set of coding guidelines to be used in all of my upper level courses. At a minimum, please adhere to the following guidelines:

1. Begin your program with a comment block that includes your name, the homework problem or project number, and any other relevant identifying information, as well as any other comments you wish to include.
2. Separate your code into paragraphs interspersed with comment blocks. Paragraphs and/or comment blocks should be separated from one another by one empty line.
3. Indent in a way that enhances readability:
    a. Indent each for, do, while, or do-while statement block by some number of spaces (at least two).
    b. Indent each if block by some number of spaces (at least two).
    c. Do not use TAB characters to indent.
    d. In C/C++ programs, it does not matter to me whether braces off-setting statement blocks are on their own separate lines or not. However you choose, be consistent throughout the program.
4. In Fortran programs, use implicit none in all main programs, functions, and subroutines. Never rely on the Fortran naming convention.
5. Name variables, and other identifiers such as function names, in a manner that makes it as easy as possible to determine their scope and purpose. Here is a sampling of common practices, meant to inspire you to develop your own style:
    a. Local loop variables are named i, j, k, …
    b. Variables appearing within only a small section of code begin with lower case letters (iStart, iEnd).
    c. Variables that are larger in scope, or other identifiers such as subroutine names, consist of short capitalized words strung together (e.g., SumX for sum of X, SumXX for sum of squares of X, NumIterMax for maximum number of iterations). An alternative is to use lower case and to separate words with underscores (e.g., sum_x, sum_xx).
    d. Parameters (in Fortran) or defined constants (in C/C++) have all upper case letters, with words separated by underscores (e.g., MAX_SIZE).
    e. Variable names are no longer than 15 characters.

An excellent book on all aspects of software construction is Mike McConnell's *Code Complete* (Microsoft Press). Although the book is aimed at business programmers, it is full of advice useful to all programmers.

**General policies**

Course policies are in accordance with "*SUNY Brockport 2008-2009: Your Right to Know & Academic Policies Handbook*".

Students are expected to attend all classes. Absences will be excused for (a) documented illness, (b) official representation of the college, (c) death of a close relative, (d) religious holidays, and (e) other circumstances beyond the control of the student. The student is responsible for obtaining class notes and handouts for classes missed. Missing classes does not absolve the student of responsibility for any course work. In accordance with suggested College policy, any student whose unexcused absences exceed 15% of classes may receive a lowered grade or failure at the instructor's discretion.

No incompletes or withdrawals will be given for this course except in cases of illness, personal tragedy, or extraordinary circumstances beyond the student's control, and then only if documented to the instructor's satisfaction.

While you are encouraged to work together, any assignments handed in must be your own work. Submitting material as your own work that is not entirely your authorship or knowingly providing an answer to another person constitutes cheating. If I am convinced beyond a reasonable doubt that cheating has occurred, the person(s) submitting the illicitly obtained answer, and any person(s) knowingly providing the same, will receive a grade of zero for that assignment. Further disciplinary procedures may also be considered.

SUNY Brockport is committed to maintaining a workplace and learning environment free of sexual harassment and intimidation. Sexual harassment is unacceptable behavior, unlawful, and intolerable.

I do not have the authority to cancel classes, no matter how severe the weather. Only the President of the College makes that decision. On days of severe weather, please listen to WBSU (89.9 FM) or WHAM (1180 AM) for class cancellations. You may also call 395-COLD. If I have any information, I will do my best to send email or post a message on my Web page.

**Disability statement**

Students with documented disabilities may be entitled to specific accommodations. SUNY Brockport's Office for Students with Disabilities makes this determination. Please contact the Office for Students with Disabilities at 395-5409 or osdoffic@brockport.edu to inquire about obtaining an official letter to the course instructor detailing approved accommodations. The student is responsible for providing the course instructor with the official letter. Faculty and staff work as a team with the Office for Students with Disabilities to meet the needs of students with disabilities.

**Disclaimer**

Announcements given in class or on the Web are considered official addenda to course policies. All policies described are subject to change as the situation warrants. Should changes be necessary, all students will be notified in a timely fashion.