COMPUTER SCIENCE 501
Computing for Science and Engineering
FALL 2008
COURSE INFORMATION
August 29

CS 590C: See page 4 for modifications pertaining to CS 590C.

Course description: Computational concepts, tools, and skills for computational science and engineering: scripting for numerical computing, scripting for file processing, high performance computing, and software development. Project may be required. Credit in this course may not be used toward a graduate degree in Computer Science. Prerequisites: C S 159, MA 262.

Role of course: This is a core course in the CS&E Program. The purpose of the course is to expose students to computational concepts, tools, and skills useful for research in computational science and engineering, beyond what is learned in a first programming course (and basic mathematics courses). A related aim is to prepare students for other courses in CSE core by teaching CS material mostly unavailable in existing graduate level courses.

Prerequisites:

- mathematical knowledge and maturity of an MS student in the physical sciences. (It is not assumed that the student is in the physical sciences, only that he(/she) has this level of mathematics.) In particular, some familiarity with matrix algebra and elementary probability is expected.
- programming experience in C, C++, Java, or Fortran or extensive scripting experience; also, commensurate computer skills.

Staff:

Instructor	**Teaching Assistant**
Robert Skeel	Mehdi Azarmi
Lawson 1205	Haas 145
(765)494-9025	(765)494-6702
fax (765)494-0739	
`skeel at cs dot`	`mazarmi at`
Office hours:	
Tuesdays 1:30–2:00	
Wednesdays 2:30–3:00	11:00–11:50, 5:30–6:20
Fridays	2:30–4:20

Feel free to call instructor any time he is in my office; best times are 1:30–3:30 daily. *All times are Eastern.*

Web page: The class web page is `http://www.cs.purdue.edu/homes/skeel/cs501.html` . This will be neglected in favor of a discussion board at `http://cs501.proboards104.com` . This will also have links to files for downloading. There will be an "Announcement" Topic for new announcements. Everyone should register by himself/herself. Within a day you should be approved by the administrator. After registration of all student, registration will be closed. We will use Blackboard for grade records and turning in homework.

Textbook: There is no suitably comprehensive textbook. Notes will be distributed having links to web resources. Some useful books:

- Python in a Nutshell
  by Alex Martelli, O'Reilly, March 2003, ISBN: 0-596-00188-6,
- Learning Python, 2nd Edition
  by Mark Lutz, and David Ascher, O'Reilly, December 2003, ISBN: 0-596-00281-5,
- Python Scripting for Computational Science, 2nd Edition
  by Hans Petter Langtangen, Springer, 2006, `http://folk.uio.no/hpl/scripting/ebook/index.html`
- The C Programming Language, 2nd Edition
  by Brian W. Kernighan and Dennis M. Ritchie, Prentice-Hall, 1988, ISBN 0-13-110362-8,
- An Introduction to Parallel Computing: Design and Analysis of Algorithms, 2nd Edition
  by Ananth Grama, George Karypis, Vipin Kumar, and Anshul Gupta Addison-Wesley, 2003, ISBN: 0-201-64865-2.

These books are on reserve in the Mathematical Sciences Library.

Examination schedule

Wednesday, **Oct 8** 1st exam, 1:30-2:20 pm,

Wednesday, **Nov 12** 2nd exam, 1:30-2:20 pm,

Dec 15–19 final exam, 2 hours

Anticipated course outline

Aug 25 introduction to CS&E

Theme I: scripting to get things done with numerical data of various types and text data

Aug 27, 29; Sep 3, 5 Unix & Python scripting

Sep 8, 10 files and text processing in Python

Sep 12, 15, 17, 19, 22 basic numerical computing in Python

Sep 24, 26, 29; Oct 1, 3 graphics and polynomials

Oct 6, 10, 15, 17, 20, 22, 24, 27 matrix computing in Python

Theme II: programming for performance

Oct 29, 31; Nov 3, 5, 7, 10 C programming

Nov 14, 17, 19, 21, 24; Dec 1, 3 parallel programming

Theme III: developing software

Dec 5, 8, 10, 12 developing software

Policies

a. *Copying* is the *evident* use of outside material in your solution (anything not present in lectures and handouts). In the case of collaboration or use of outside written material, the score for an assignment may be discounted. For example, 3 students turning in identical correct answers each may receive as little as one third credit. The course staff reserve the right to determine the discount, if any. Use of your work by another will be assumed to be done with your consent unless there is evidence to the contrary.

b. *Cheating* is the evident use of outside material in your solution *without including a written acknowledgment of the source*. (Citing sources is simply good scholarship.) This does not preclude some discussion when starting an assignment. However, the onus is on you to either acknowledge the collaboration or ensure that it is sufficiently limited so as not to be evident in your work. In practice this is enforced when the course staff believe that the collaboration goes beyond the sharing of ideas. but again the course staff reserve the right to make this judgment. *Cheating is penalized in accordance with university rules and the instructor's policy is to impose substantial penalties.*

c. Excuses. Failure to attend an exam or failure to submit an assignment on time is recorded as a zero except when it is unavoidable because of some legitimate emergency (a medical excuse from a physician, or a death in the immediate family). In such a case it is desirable that the instructor be contacted before the exam or before the due date of the assignment.

   The teaching assistant may grant you an extension for an assignment, but he has the prerogative to refuse you.

d. Requests for assignment regrades should be submitted to the TA within one week of it being returned to the class. If you remain dissatisfied after the regrade, you may appeal to the instructor.

e. No extra work will be provided for students wishing to improve their grade. Also, final grades will not be changed after they are posted (except if they were incorrectly computed).

f. In the event of a major campus emergency, course requirements, deadlines and grading percentages are subject to changes that may be necessitated by a revised semester calendar or other circumstances. Information about changes in this course will be announced on Blackboard Vista and possibly the instructor's web page.

Assignments There are about six assignment sets, much of it requiring the writing of programs. Tentative due dates are Sep 12, Sep 26, Oct 17, Oct 31, Nov 21, and Dec 5. The plan is to drop the lowest score. There is no project this semester. We will use Python and C. There will be considerable computer use. Ideally, the student already has access to a multicore computer for his/her research and will use this for the assignments. The student will be expected to install the necessary (free) software on his/her desktop. As an alternative, accounts on an ITaP/RCAC multiprocessor will be requested. We will attempt to accommodate the following platforms:

- Linux,
- Mac OS 10.4 and 10.5,
- Cygwin on Windows XP or Vista.

For Windows users, a better choice than Cygwin is to install Ubuntu. (A disadvantage is the need to restart when switching between Ubuntu and Windows.) Installation is very easy and straightforward. The link for downloading windows installer of ubuntu is `http://wubi-installer.org/` . The default installation will create a dual boot menu at startup time. In any case, we will ask you what platform (hardware and system software) you are using.

Grading

assignments	40%	A	80–100%	D	40–50%
in-class exams	30%	B	65–80%	F	0–40%
final exam	30%	C	50–65%		

Plus and minus grades will also be given. The grade boundaries may be lowered by as much as 10% in order to achieve a reasonable grade distribution. Also, there may be minor upward or downward adjustments of the grade boundaries.

COMPUTER SCIENCE 590C
Computing for Life Sciences
FALL 2008
COURSE INFORMATION
August 25

Course description: Computational concepts, tools, and skills for computational life sciences: scripting for numerical computing and file processing. Credit in this course may not be used toward a graduate degree in Computer Science.

Prerequisites: Math/CS/Stats required of undergraduates in biology at Purdue.

To be omitted: Any books or software dealing with C or parallel programming, e.g., SWIG and PyMPI.

Examination schedule

Wednesday, **Oct 8** final exam, 1:30-2:20 pm,

Anticipated course outline topics from **Aug 25** through **Sep 26**

Assignments There are two assignment sets, much of it requiring the writing of programs. Tentative due dates are Sep 12 and Sep 26. We will use Python. There will be considerable computer use. The student will be expected to install the necessary (free) software on his/her desktop.

Grading

assignments	52%
in-class exam	48%

ASSIGNMENT # 0 The following actions are required:

1. Send an email to the TA stating precisely what hardware and what operating system you are using. In the case of Windows, state whether or not you will be using cygwin.

2. Install items 1–6 below.

Installation of Software

Reference: Langtangen(2006), Chapter 1, Appendix A, and Appendix B.1.3.

Installation of software is done in two ways:

1. from a binary. This is typically performed by point-and-click and generally seems to require that you have superuser/administrator privileges.

2. from source. This done at the command line. If you have superuser privileges, you can preface the install command with `sudo`. If not, you can install in `$HOME/local` using an option like `--prefix=$HOME/local` or `--home=$HOME/local`. In such case, make sure that `$HOME/local/bin` is in your path.

The unpacked download can be discarded after you are satisfied with the installation. You may wish to save a few text files, the documentation, and examples.

For installation and use of some software, you need the location of the python binary. Following is an example:

```
$ which python
/usr/bin/python
$ ls -l /usr/bin/python
lrwxr-xr-x  1 root  wheel  72 Nov 16  2007 /usr/bin/python@ -> ../..
/System/Library/Frameworks/Python.framework/Versions/2.5/bin/python
$ PYDIR=/System/Library/Frameworks/Python.framework/Versions/2.5
```

The following are required for this course: Some of this software may already be installed.

1. a C compiler that supports OpenMP, e.g., version 4.2 or late from `gnu.org`. Test with the program `hello_omp.c`

```
#include <omp.h>
#include <stdio.h>
int main (int argc, char *argv[]) {
    int id, nthreads;
    #pragma omp parallel private(id)
    {
        id = omp_get_thread_num();
        printf("Hello World from thread %d\n", id);
        #pragma omp barrier
        if ( id == 0 ) {
            nthreads = omp_get_num_threads();
            printf("There are %d threads\n",nthreads);
        }
```

```
    }
    return 0;
}
```

as follows:

```
$ export OMP_NUM_THREADS=4
$ gcc -fopenmp -lgomp hello_omp.c
$ ./a.out
```

2. Python 2.5, from `http://www.python.org/download/`, by clicking on one of the 4 operating systems listed on the left. Version 2.5.2 has numerous bug fixes.

3. NumPy, from `http://www.scipy.org/Download`. The latest version is 1.1.1. You may wish also to install SciPy. Afterwards, execute "`import numpy; numpy.test()`".

4. SWIG (Simplified Wrapper and Interface Generator) from here. Select either `swig-1.3.36.tar.gz` or `swigwin-1.3.36.zip`. To install on Unix, change to the directory `swig-1.3.36` and enter
```
$ ./configure
$ make
$ sudo make install
$ make clean
```

5. `pyMPI-2.5b0.tar.gz` from `http://pympi.sourceforge.net/` . It is suggested that you use `./configure` or `./configure --prefix=\$HOME/local` if you do not have superuser/administrator privileges. Then do the `make` . Quite possibly you will get the error message
```
pyMPI_util.c:22:31: error: numpy/arrayobject.h: No such file or dire
ctory
```
If so, look in a directory near the python binary for one of the missing files, e.g.,

```
$ find $PYDIR | grep arrayobject.h
/System/Library/Frameworks/Python.framework/Versions/2.5/Extras/lib/python/
numpy/core/include/numpy/arrayobject.h
/System/Library/Frameworks/Python.framework/Versions/2.5/Extras/lib/python/
numpy/core/include/numpy/ndarrayobject.h
```

Then redo
```
./configure --prefix=$HOME/local \
    --with-includes='-I$PYDIR/Extras/lib/python/numpy/core/include/'
```
Redo the `make` followed by `make check`. There may be an error message but you may be able to continue, so try `make install` . The installed program may not work interactively but should work for
```
mpirun -np 8 pyMPI examples/fractal.py
```
which creates an image file for viewing.

6. a plotting program, e.g., Matplotlib from `http://matplotlib.sourceforge.net`. Installation can be tested with

```
import matplotlib
from pylab import *
ion()
plot([1, 2, 3])
```

The following is useful:

7. IPython from `http://ipython.scipy.org/moin/`. If you do not have administrator permissions for your machine, use `python setup.py install --home \$HOME/local` .

The following may be useful, though it may not work with the other tools:

8. VPython from `http://www.vpython.org/download.html`.

Proceed carefully and search the web for advice. Chapter 1 and Appendix A of Langtangen(2006) may be helpful. The book is dated and does not mention the existence of the Numpy package.

Python programs can be run noninteractively from the operating system command line. They can be run interactively in various ways:

- by typing "python" from the command line.
- by launching IDLE, which is an integrated development environment GUI. Either double-click its icon or enter "`idle -n &`" (or equivalent) from the command line. The "`-n`" flag is needed if using Matplotlib.
- by typing "ipython" from the command line.

Platform specific notes follow:

Linux

Although Python is pre-installed, you may have to build a binary to get a current version.

Mac OS 10.4

Useful information is found at `http://www.mtheory.co.uk/support/index.php?title=Mac_Specific_Help` where the following steps are given:

1. Install "Xcode Tools" either from the install DVD or from `http://developer.apple.com/tools/download/`.
2. Install Xwindows from the "Optional Installs" on the install DVD and select X11 from among the Applications and run Software Update. (Reference: `http://developer.apple.com/opensource/tools/runningx11.html`)
3. Go to `http://www.pythonmac.org/packages/` where there is a link to binaries. Download and install python, matplotlib, and numpy. Open a new Terminal window before testing python.
4. SciPy, if you want it, has to be installed from source, and this may not be easy.

5. To install IPython, if you want it, go to `http://ipython.scipy.org/moin/Download` and click on "downloads." Control-click on `ipython-0.8.4.tar.gz`

   ```
   tar -xvf ipython-0.8.4.tar    or    tar -xvzf ipython-0.8.4.tar.gz
   cd ipython-0.8.4
   python -c "import sys;print sys.prefix"
   ```

   It should return "`/Library/Frameworks/Python.framework/Versions/2.5`".

   ```
   python setup.py build
   sudo python setup.py install
   ```

   After installation open a new Terminal window and type

   ```
   ipython
   ```

   Alternatively, you might try `ipython-0.8.4-py2.5.egg`, but see the comments under MacOS 10.5.

MacOS 10.5

The "SciPy Superpack for Python 2.5" also includes IPython, Numpy, and Matplotlib. If you install it, you may wish to learn a little about Python Eggs and EasyInstall. Unfortunately, the automatic installion is into the directory

`/Library/Python/2.5/site-packages/` ;

whereas, for everything to work correctly, it should be in

`/Library/Frameworks/Python.framework/Versions/Current/lib/python2.5/site-packages/`

.

Windows XP and Windows Vista

Binaries are plentiful except for swig and pyMPI. Installing from source may be difficult, especially for pyMPI.