GS500 Introduction to computer programming

Winter 2011 (2 credits)

Computational methods have become increasingly important for quantitative modeling and data analysis in the Natural Sciences. Many students are familiar with programs such as Excel, Mathematica, or Matlab, that provide easy-to-use graphical interfaces, robust numerical methods, and visualization tools. Yet, for work on extensive data sets, advanced modeling of physics and chemistry, parallel computing, and batch processing of many data sets it is often necessary to use more efficient and inherently more powerful computational techniques that are based on Linux computing and computer programming.

If you have ever felt the use of advanced programs was slow or tedious for your work this course will be well suited to develop your techniques in a more efficient environment. This course also allows for learning programming as a basis for numerical methods courses offered by Geology, Physics, Astronomy, and Engineering.

This course provides a basic introduction to Linux, shell programming, and Fortran90/C++ programming. The course is introductory and hands-on, with the grade determined by a number of bi-weekly problem sets and a final project. Part of the philosophy of the course is contrast & compare with existing Windows & Mac applications – a well suited choice for final project is one in which students optimize their own numerical tools for use in their graduate studies.

Instructors: Peter van Keken, 4502 CC Little Building (keken@umich.edu)
GSI: Zhangyi Hu, 4518 CC Little Building (zhangyhu@umich.edu).

Location: 5501 CC Little Building (Geological Sciences Computer Lab)
Time: Monday 2:10-5 (Full winter term; includes lab session)

Recommended reading:
Sobell, A practical guide to Linux, Addison-Wesley (any edition)
Prata, C++ Primer Plus, SAMS (later editions may be better)
Nyhoff and Leestma, Fortran 90 for engineers and scientists, Prentice-Hall

Course topics:
Introduction to Unix
 Comparison to MacOS/Windows; file management; editors
Computer programming
 History and philosophy; best practices; error checking and benchmarking
 Shell programming (csh and bash compared)
 Introduction to classical and modern languages (Fortran90/C++)
 Comparison with high-end computational applications (matlab/mathematica)
Basic numerical methods
 Forward modeling & data analysis
 Introduction to parallel programming
GS500 Introduction to computer programming. Tentative schedule

1/10  Introduction to linux/unix. Use of the shell.
1/17  No class (MLK)
1/24  History of computing. Editing files on linux: vi (and emacs).
1/31  Computer hardware. Writing simple programs in the bash shell
2/7   Computer languages. Introduction to Fortran90 and C++.
2/14  Structure programming. Programming structures: if-then-else, do-loops
2/21  Data storage. Arrays, file I/O, debugging.
2/28  No class (spring break)
3/7   Object oriented programming.
3/14  Introduction to numerical methods. integration and differentiation
3/21  Data analysis. Large data sets, statistical analysis

Last 3 weeks  Student project time (to be discussed) with presentations

We will meet Mo 2:10-5 in 5501 CC Little, starting with a lecture/demo hour, followed by work on the lab assignments and student project. Documentation is provided through ctools. You are encouraged to think about & discuss with us the topic of your final project at an early stage.