Spring 2001 Biol 293
Simulating Ecological and Evolutionary Systems

Instructor: Will Wilson
Meeting Details: BioSci Rm 0049, Tues and Thurs 12:40–1:55
Office: BioSci 033; 660-7346; wgw@duke.edu

This course covers C programming applied to problems in ecological and evolutionary theory, ranging from stochastic birth–death models to foraging models to resource allocation in plants. Although each simulation problem is accompanied by analytic modelling, including differential equations and the like, providing a stronger theoretical foundation, the primary emphasis of the course is to learn simulation and computational techniques.

Expectations

There are no prerequisites for taking the course, for example, students are not expected to know how to program or work within the Unix computer environment. All students enrolled at Duke have access to the *acpub* computer system and its resources (the UNIX operating system and an ANSI-standard C compiler), however, students are encouraged to use their own computers with whatever operating system (Linux, Windows or Macintosh), editors, and C compilers (for example, Borland C).

At the beginning of the term, I cover the basic UNIX commands and file editors. Then, during the first part of the term, I'll cover basic C programming by examining "simple" birth-death population models from both individual-based simulation and analytic perspectives. I'll assign a series of computing projects to familiarize students with essential programming concepts and the language C. These projects begin with very straightforward programs designed to initiate students with editing files, and compiling and running a computer program. After a few programming projects, I'll introduce more advanced aspects of C, including pointers, structures, and compiler directives, and important numerical algorithms including random number generation, numerical integration, and visualization techniques.

An important part of the course is working on a self-motivated programming project. It could be either a computational technique (genetic algorithm, neural networks, numerical analysis, random number generation, etc.) or a biological application of a computational method (a published paper or general area). Two aspects should come together in your project: 1) the project should relate to your special interests (ideally for graduate students a publishable piece of work); and 2) I want you to demonstrate a mastery of programming skills. Four or five weeks into the term each student turns in a written outline and gives a general talk, to the class, on their proposed project. At the end of the term each student will give a final talk and write up a formal paper on the project. I leave open the possibility of a midterm and final.

Spring 2001 Biology 293 Lecture Schedule (January 16, 2001)

Date	Topic
1/11	Intro / Chap. 1
1/16	Chap. 2: Details of bd.c; Chap. 3: logbd.c; functions and arrays
1/23	Questions; Outline student projects
1/30	NO MEETING
2/6	Chap. 4: pointers and files
2/13	Chap. 5: using PS files; Chap. 6: Mini Projects
2/20	NO MEETING
2/27	NO MEETING (Thursday?)
3/6	Chap. 7: structures
3/13	NO MEETING
3/20	Chap. 8; numerical integration
3/27	
4/3	
4/10	Project Presentations (Meet Th, too?)
4/17	Project Presentations (Meet Th, too?)