

## Article

# From Biology to Mathematical Models and Back: Teaching Modeling to Biology Students, and Biology to Math and Engineering Students

Hillel J. Chiel,<sup>\*,†,‡</sup> Jeffrey M. McManus,<sup>\*</sup> and Kendrick M. Shaw<sup>\*</sup>

Departments of <sup>\*</sup>Biology, <sup>†</sup>Neurosciences, and <sup>‡</sup>Biomedical Engineering, Case Western Reserve University, Cleveland, OH 44106-7080

Submitted March 15, 2010; Revised June 3, 2010; Accepted June 4, 2010  
Monitoring Editor: John Jungck

We describe the development of a course to teach modeling and mathematical analysis skills to students of biology and to teach biology to students with strong backgrounds in mathematics, physics, or engineering. The two groups of students have different ways of learning material and often have strong negative feelings toward the area of knowledge that they find difficult. To give students a sense of mastery in each area, several complementary approaches are used in the course: 1) a “live” textbook that allows students to explore models and mathematical processes interactively; 2) benchmark problems providing key skills on which students make continuous progress; 3) assignment of students to teams of two throughout the semester; 4) regular one-on-one interactions with instructors throughout the semester; and 5) a term project in which students reconstruct, analyze, extend, and then write in detail about a recently published biological model. Based on student evaluations and comments, an attitude survey, and the quality of the students’ term papers, the course has significantly increased the ability and willingness of biology students to use mathematical concepts and modeling tools to understand biological systems, and it has significantly enhanced engineering students’ appreciation of biology.

## INTRODUCTION

Nearly 50 yr ago, an outstanding teacher was attempting to convey key ideas in his field through lectures. After he was done, he described some of his feelings of frustration at the difficulties he encountered:

“I think, however, that there isn’t any solution to this problem of education than to realize that the best teaching can be done only when there is a direct individual relationship between a student and a good teacher—a situation in which the student discusses the ideas, thinks about the things, and talks about the

things. It’s impossible to learn very much by simply sitting in a lecture, or even by simply doing problems that are assigned.” (Feynman *et al.*, 1963 p.5)

Richard P. Feynman, who made fundamental contributions to the theory of electrodynamics, had just concluded a brilliant series of lectures on physics for freshmen and sophomores at Caltech in the early 1960s. He was lecturing to a very select group of students who were clearly interested in science and mathematics; otherwise, they would not have chosen to go to Caltech. His lectures, in printed form, continue to inspire both students and teachers, but even under conditions that might be considered optimal for teaching through lecturing, he found the results disappointing.

The problem is compounded when students have little interest in the subject matter, or find it boring, or difficult, or even frightening. It is one thing to lecture on physics to aspiring physics students; it is quite another to lecture to students who have no intention of being physicists or mathematicians. Dreger and Aiken (1957) first named the feelings

DOI: 10.1187/cbe.10-03-0022

Address correspondence to: Hillel J. Chiel (hjc@case.edu).

© 2010 H. J. Chiel *et al.* CBE—Life Sciences Education © 2010 The American Society for Cell Biology under license from the author(s). It is available to the public under Attribution–Noncommercial–Share Alike 3.0 Unported Creative Commons License (<http://creativecommons.org/licenses/by-nc-sa/3.0>).

of dislike toward quantitative subjects “math anxiety.” Sheila Tobias popularized the importance of math anxiety for career choices, especially for young women, in her influential book, *Overcoming Math Anxiety* (Tobias, 1993). Psychometric studies have shown a consistent negative correlation between math anxiety and math achievement, and these strong feelings may dominate choices of college majors and subsequent careers; moreover, the negative affect associated with these subjects reduces performance by students (Ashcraft and Moore, 2009). These problems may be less pressing for students in math, physics, and engineering, but they have a major effect on many students who choose to major in biology (Gross *et al.*, 2004).

Opportunities to enhance biology using the concepts and tools of math, physics, and engineering have grown considerably in the past few years. Historically, biophysics (e.g., the analysis of ion channels and calcium regulation in excitable cells) has drawn heavily on math and physics (Jack *et al.*, 1975; Hille, 2001; Friel and Chiel, 2008). More recently, the detailed description of metabolic pathways and of genomes has led to the application of mathematical concepts from network theory to biology (Jeong *et al.*, 2000).

At the same time, opportunities to use concepts based on biology for developing new approaches in math, physics, and engineering also have grown considerably. Interdisciplinary efforts to integrate biological tools and inspiration into artificial devices have led to new areas, such as biologically inspired robotics (Beer, 2009), biomimetics (Bar-Cohen, 2006), evolutionary algorithms (Mitchell, 1996), and artificial neural networks (Kollias *et al.*, 2006).

Recognizing these opportunities, 7 yr ago, the National Research Council of the National Academies of Sciences issued a call for incorporating more quantitative concepts and skills into the biology curriculum. Some of the key points they made were that biology departments should

“... Consider the importance of building a strong foundation in mathematics and the physical and information sciences to prepare students for research that is increasingly interdisciplinary in character . . . Concepts, examples and techniques from mathematics, and the physical and information sciences should be included in biology courses . . . Successful interdisciplinary teaching will require new materials and new approaches . . . Laboratory courses should be as interdisciplinary as possible . . . All students should be encouraged to pursue independent research as early as is practical in their education . . .” (National Research Council, 2003, *BIO2010*, pp. 8 and 9)

The creation of programs in systems biology that use tools from computer science and engineering for bioinformatics and network analyses (Ideker, 2004) has been one indication that aspects of the vision of *BIO2010* are beginning to be realized. In addition, new programs in mathematical biology have begun to be created (e.g., see the concentration in Computational and Mathematical Biology at the University of Pennsylvania, [www.bio.upenn.edu/programs/undergraduate/concentrations/compbio.html](http://www.bio.upenn.edu/programs/undergraduate/concentrations/compbio.html); the Mathematical Biology program at the University of Utah, [www.math.utah.edu/research/mathbio/index.html](http://www.math.utah.edu/research/mathbio/index.html); or the program in Mathematical and Computational Biology at Harvard University, [www.oeb.harvard.edu/research/math\\_comp.html](http://www.oeb.harvard.edu/research/math_comp.html)).

Nevertheless, it is probably fair to say that the majority of biology students are not systematically exposed to math, physics, and engineering concepts and tools.

Similarly, recent attempts to enhance the training of engineers have led faculty to realize that closer integration of math, science, and engineering would be valuable (Froyd and Ohland, 2005) and that biology should be integrated into engineering training, not just for biomedical engineers, but for many other engineering disciplines that might profit from a closer connection to biology. Nevertheless, it is probably fair to say that the majority of students majoring in mathematics, physics, and engineering are not systematically exposed to biology.

What obstacles are preventing the integration of math, physics, and engineering into the biology curriculum and vice versa? Institutional obstacles can create barriers. For example, the number of credits that are needed to satisfy a major may restrict the number of courses that students take outside of their major. Professional course requirements may make it difficult for students to take courses in other areas: for example, requirements for premedical students, or requirements for engineering students in programs approved by the Accreditation Board for Engineering and Technology (ABET, Inc; see [www.abet.org](http://www.abet.org)) may restrict course choices.

More fundamentally, however, there is a problem similar to the “two cultures” identified >50 yr ago by Snow (1959), although his focus was on the vast gulf that had grown between the humanities and the sciences. A similar cultural gap exists between many biologists, both faculty and students, and the (historically) more quantitatively oriented sciences. There are exceptions, of course, but in general biology students and faculty have a “different way of knowing” than students and faculty in mathematics, physics, and engineering.

Biology students and faculty are trained in very specific ways to achieve mastery of complex biological systems. First, biological terminology is a way of communicating about biological systems shared by all biologists. Whether a researcher works on genes and their regulation, intermediary metabolism, neurophysiology, human anatomy and physiology, or evolutionary biology, the names of the various components are an important part of understanding what is currently known and what still needs to be understood. Second, it is often not clear a priori which details matter for the function of a biological system and which can safely be ignored, in part because this may change with the context in which a biological system is studied. This leads to a tendency to encourage students to know all the details. Third, because of the complexity of biological systems, students must develop a qualitative “feeling for the organism” (Keller, 1983), and this internal qualitative model of the system may be difficult to articulate quantitatively but may nevertheless be very useful for guiding experiments that successfully analyze a biological system.

In contrast, students in math, physics, and engineering are trained very differently. The focus is on finding the right simplifications and abstractions to describe a system and then analyzing the resulting simplified system as fully as possible, whether the area is abstract algebra, quantum electrodynamics, or systems and control engineering. Furthermore, memorizing formulas is discouraged, because it is not

an effective strategy for solving problems, especially in upper-level courses. Students are expected to turn qualitative intuitions into mathematical statements that can be formally and rigorously analyzed and that also can be used to make precise predictions to guide design and experiments.

The training process for biology tends to attract students who are good at memorization, who work effectively from concrete examples to general principles, who enjoy struggling to understand complicated systems even if they must sometimes reason qualitatively, and who revel in finding and characterizing new details of a system. The very same training process repels students who are most interested in abstract principles, who have difficulty memorizing facts, and who are impatient with what they regard as unnecessary complexity. Conversely, what attracts students to mathematics, physics, and engineering tends to repel students who are interested in biology. These divisions among the different groups of students are usually established by high school. This is the fundamental obstacle that must be overcome.

Over the past 10 yr, one of us (H.J.C.) has developed a course called Dynamics of Biological Systems that attempts to bridge the gap between biology and concepts and tools in math, physics, and engineering and that has been taught to both biology students and students in math, physics, and engineering. The rationale for the course is that unless a person has some mastery of these different areas, it will not be possible to fully integrate them. Furthermore, the best way to begin to master these skills is to have immediate and regular feedback from using them to build something of intrinsic interest to the student. This article describes the course in some detail.

## METHODS

One of us (H.J.C.) conceived of the course in 1999, and in fall 2000, was granted release time to develop the initial course units. The course was first offered at Case Western Reserve University (CWRU) in spring 2001. Initially, it was offered every semester, but starting in 2005, it was offered in the spring term only. It has been offered 13 times (through spring 2010). The description below is based on the current form that the course has taken, which developed based on how students did in the course, feedback from students, and our own ideas.

### *Encouraging Enrollment*

Although some biology departments have begun to encourage students to take courses that emphasize mathematics, physics, or engineering, it has generally been difficult to mount mathematically based biology courses within a biology department, unless the department has faculty who have those interests and abilities. Furthermore, in biology departments that are oriented more toward molecular biology or more descriptive forms of biology, a math-based course may not be seen as essential. Finally, many biology students may instinctively avoid a course that relies heavily on mathematics and computer programming. Thus, several steps were necessary to ensure enrollment in the course.

During the first two semesters that the course was offered, advertisements for the course were posted. Also, notices were sent to chairs of other departments whose students might have found the course of interest. One of us (H.J.C.) has a secondary appointment in biomedical engineering and was able to have the course approved for listing both as a Biology course (BIOL 300) and as a Biomedical Engineering course (EBME 300). H.J.C. also worked with colleagues in the Department of Biology (Drs. Robin Snyder and Joseph Koonce) to develop a Systems Biology major, for which BIOL 300 is a required course. This degree program was officially approved in spring 2007, so that students in the major began enrolling in the course in their sophomore year, beginning in spring 2008.

To increase enrollment, especially among biology students, we deliberately did not require any prerequisite courses. Biology students often delay taking their math and computer programming requirements until their senior year, because none of their regular biology courses require these prerequisites. Because they also tend to delay taking any quantitative courses until their senior year, they would automatically be shut out of the course if math and computer programming were prerequisites. The course has been designed to be sufficiently self-contained that the key mathematical and computer programming concepts and skills are incorporated into the problems that students must solve during the first half of the semester.

Three developments played a crucial role in increasing enrollment in the course. First, from the beginning, students were expected to work at their computers as the instructor introduced material; thus, the course qualified as a laboratory. Because all biology majors are required to take two labs, they were motivated to take the course. Second, all students who planned to graduate with a B.S. degree in Biology (rather than a B.A.) are required to take at least one quantitative lab course, and BIOL 300 satisfies this requirement. Third, a chemical engineering student took the course and strongly recommended it as a technical elective for chemical engineers, which has ensured a steady enrollment of chemical engineers over the years. Finally, it is also likely that positive word of mouth has contributed to enrollment.

### *Educational Goals*

It is crucial to articulate clear educational goals. In the first session, which is the only lecture of the entire semester, the educational goals of the course are clearly stated. The goals are posted on the course website, and all assessments are tied to these goals. The goals are to teach students the skills necessary to:

1. Construct and extend mathematical models of biological phenomena;
2. Analyze these models using the concepts and tools of nonlinear dynamical systems theory; and
3. Write clearly about the modeling process and the results obtained from the model.

A reproduction of the course Web page in PDF format is at <http://slugoffice8.biol.cwru.edu/~hjc/DynamicsCourseMaterials/index.html>.

### Choice of Modeling Tools

There are three ways to introduce students to tools for constructing models. At one extreme, one can introduce programming languages (e.g., C++, Java). Students have control over every aspect of the models they create and well-written code can run very quickly, making it possible to rapidly simulate complex models. The disadvantages are that most programming languages require students to spend a great deal of time to get anything working, they may need considerable practice to remove programming errors, and they may require students to handle details irrelevant to the goals of the course (e.g., graphical user interfaces). The faculty member must also have facility in the chosen language, or if this choice is not restricted, in a range of programming languages. Thus, significant amounts of class time must be spent on teaching programming, or a programming course must be a prerequisite. Creating a course that requires programming experience would almost certainly reduce enrollment by regular biology majors, unless programming is a requirement of the major.

A second extreme would be to use dedicated modeling packages. Molecular modeling packages such as AMBER (<http://ambermd.org>), CHARMM ([www.charmm.org](http://www.charmm.org)), or GROMOS ([www.igc.ethz.ch/GROMOS/index](http://www.igc.ethz.ch/GROMOS/index)) make it possible for users to set up sophisticated and complex models of biological molecules. Neural modeling packages such as NEURON ([www.neuron.yale.edu/neuron](http://www.neuron.yale.edu/neuron)), SNNAP (<http://snnap.uth.tmc.edu>), or GENESIS ([www.scholarpedia.org/article/GENESIS](http://www.scholarpedia.org/article/GENESIS)) make it possible for users to set up complex models of individual neurons, small neural circuits, or large biologically based neural networks. Dedicated packages have several advantages. One can rapidly put together sophisticated and complex simulations of biological structures. It is often possible to modify a previous simulation slightly to get a working model. The program takes care of the graphical user interface. Such packages, however, have several drawbacks. Students may need to learn a great deal to use the packages effectively, the packages may require powerful computing resources to run efficiently, and they may be very difficult to modify if a student wishes to model something that is not already "built in" to the program. Again, a significant part of the course may need to be dedicated to teaching the students how to use the packages.

An intermediate solution may be most effective. If students are taught to use a modeling platform that has powerful built-in constructs that allow them to build small working models quickly, that can work across multiple computing platforms, that can provide a graphical user interface, and that can be programmed to modify and extend models, then most of the focus of the course can be on modeling rather than on programming. Ideally, the platform should handle both symbolic mathematics and numerical simulation very well.

Examples of modeling platforms that can satisfy these criteria, to a greater or lesser extent, are MATLAB ([www.mathworks.com](http://www.mathworks.com)), Macsyma ([www.symbolics-dks.com/Macsyma-1.htm](http://www.symbolics-dks.com/Macsyma-1.htm)), Maple ([www.maplesoft.com](http://www.maplesoft.com)), and *Mathematica* ([www.wolfram.com](http://www.wolfram.com)). Many engineering schools have adopted MATLAB as their de facto standard programming language. We considered MATLAB seriously and may in the future adopt it as an alternative. In addition, there are open source modeling and analysis packages, many of them based in

Python ([www.scipy.org](http://www.scipy.org)), and open source packages that can do sophisticated data analysis, such as R ([www.r-project.org](http://www.r-project.org)), and these are also appealing, because advanced faculty and students can contribute directly to their development by creating new source code.

Several considerations finally determined our decision to adopt *Mathematica* as the basis for the course, despite the rather steep initial learning curve that it imposes on the students (and on faculty who have not used it previously!). First, it provides full word processing and mathematical typesetting capabilities, so course units could easily be developed and deployed in the classroom, and this has served as the basis for creating an interactive textbook. Second, it has a variety of built-in functions that allow students to import, transform, plot, fit, analyze, and export data. Third, it has powerful symbolic mathematical capabilities, so that students can be shown both mathematical and numerical solutions in a unified setting. Fourth, it has powerful numerical capabilities and can run complex models quickly. Fifth, it is programmable, and supports a range of programming styles (e.g., procedural, functional, recursive), providing students with a useful tool for creating or extending models of varying levels of sophistication. Finally, it provides a very useful function (Manipulate) that makes it easy to set up and interactively manipulate models.

Although the cost of *Mathematica* is high, our university has invested in a site license for *Mathematica*, which allows students to download and activate a copy that is good for 1 yr. Each subsequent year, they need to reactivate the program, but it remains functional as long as they are enrolled at the university.

### Classroom Architecture

Initially, the first half of the course was taught as a series of lectures while one of us (H.J.C.) developed the interactive textbook for the course. The classroom was thus used in a standard configuration, with the lecturer in front, and the students sitting in rows facing the front. Starting with the spring 2009 semester, however, all lectures except the first were eliminated and were replaced with students working in teams on benchmark problems. The rationale for this change was that it would encourage teamwork, and help the students focus on making continuous progress toward solving benchmark problems. In general, lectures lead students to regard the teacher as the sole source of information, and this inhibits peer interactions among them. By eliminating the lectures, and encouraging students to work in teams from the beginning of the semester, students much more quickly learn to rely on themselves and their peers.

In spring 2008, the classroom in which the course was regularly taught was renovated, so that students would be able to easily interact with one another. A false floor was installed so that table configurations and their connections to power and the network could be flexibly rerouted. Each table was provided with a power strip. There are six tables composed of two half hexagons, which seat six students, so that the total capacity of the classroom is 36 students (Figure 1).



**Figure 1.** Classroom design to enhance teamwork and a continuous progress method of instruction. (A) Layout of the classroom. Each hexagonal table can be separated into two independent tables if necessary. Six Ethernet connections to the CWRU fiber optic network are shown. A power strip runs along the front side of each half table. (B) Students working in the classroom.

### *Teamwork and Group Dynamics*

Although in earlier offerings of the course students worked alone on problem sets (in the first half of the semester) and then worked alone during the second half of the semester to reconstruct a model, it became clear that it was much better to require students to work together in teams of two. Teams larger than this did not tend to work well, because responsibility for the work was too diffused, and often one or two students out of three would end up doing most of the work, creating feelings of resentment and exclusion. Students working alone often got stuck and frustrated. In general, with a few notable exceptions, teams of two were able to work together very effectively to solve problems throughout the semester.

Because the class attracts students with heterogeneous backgrounds, and teams are assigned at random, teams may have two students with strong quantitative and/or programming skills, or with strong skills in biology, or one of each. Whatever the mix of skills, working together with another student raises difficult issues of group dynamics. Students must learn to respect each other's abilities and use them to maximum advantage. They may need to teach some of the material to the other member of the team or be comfortable asking questions and sharing ideas.

To facilitate this process, for the past 3 yr we have required students, on a weekly basis, to provide an evaluation of 1) how much they have contributed; 2) how much their

teammate has contributed; and 3) how well the team has functioned, on a scale of 0–10 (0, worst and 10, best). We analyzed the data after each class and used it to determine whether it was necessary to intervene. If any of the ratings were unusually low, we e-mailed students, arranged to meet with them individually, encouraged them to articulate what the team issues might be, and then met with them together if we felt that this would be helpful. By intervening relatively early, we are often able to head off more serious problems. We also interact regularly with students during class and thus can often observe and respond to potential problems in team interactions even if the ratings look fine.

### *Computing Resources*

Outfitting a classroom with state-of-the-art computers is expensive. It is also wasteful, because within 3 to 5 yr (or less) the hardware is outdated. Investment in computers also creates continuing expenses. Hardware may break and then needs to be repaired or replaced. A system administrator may need to be hired to keep the software virus free, updated, and working especially after it has been updated. These expenses must be added to the cost of the software itself. In addition, security may be necessary so that the computers are not stolen, and appropriate account security must be set up so that students can log into the computers to use them when they are not physically in class.

With the advent of wireless hot spots throughout campuses, students have largely shifted to purchasing laptop computers. Thus, we set up the classroom so that students could easily attach a laptop computer to the campus network through an Ethernet connector, or use a wireless connection, and could plug into a power source (Figure 1A). We encourage students to bring in their own computers. This immediately solves the problems listed above: the students are responsible for hardware and software maintenance and upgrades at no cost to the department; they have access to their machines all day and over the weekend, and they are more likely to take good care of a machine that they own. Each year, students bring newer computers to class, so the problem of obsolescence also is obviated.

To handle the minority of students who have not yet purchased a laptop, or whose computers are not working, we purchased eight MacBooks. These are powerful enough to run *Mathematica*, they provide Web access, and they allow students to use other programs available to them through the university's software computing center. They have required relatively little maintenance over several years.

### *Interactive Textbook and a Constructive Approach to Modeling*

To effectively convey mathematical and programming concepts, it is invaluable to give students immediate feedback, and the opportunity to experiment. This is crucial to allow students to build confidence in their ability to master the material. These considerations were the basis for creating an interactive textbook.

Most textbooks are static repositories of information. If a textbook is online, it can have animations and hyperlinks, which are helpful. But a more radical change for a modeling

textbook would be a mixture of text exposition and code to generate results and figures that could be reevaluated by the student after putting in slightly different values to see how the results change. Furthermore, students should be able to easily copy the code into their own workspaces, and further modify and manipulate it, so that it can serve as the basis for models that they create. If an error develops in their copy of the chapter, they can always download it again from the course website, so there is no penalty for experimentation.

These considerations have led the first author to create an interactive textbook. Because *Mathematica* makes it easy to integrate descriptive text, illustrations, numerical simulations, and symbolic mathematical manipulations seamlessly, it is a natural platform for constructing such a textbook (the chapters are available as *Mathematica* notebooks and as PDFs at <http://slugoffice8.biol.cwru.edu/~hjc/DynamicsCourseMaterials/index.html>; the *Mathematica* notebooks can be viewed using the free *Mathematica* player, which can be downloaded from [www.wolfram.com/products/player](http://www.wolfram.com/products/player)). Code for analyzing mathematical expressions, for running numerical simulations, and for generating figures are all part of the textbook, so students can rerun it and see what happens when they make small changes; they can also copy it into their own *Mathematica* notebooks and use it as the basis for their own models.

Since version 6, *Mathematica* has also had a very powerful built-in function, Manipulate, that can easily create interactive models with very little programming. Using this function, it is easy to set up simple models and move sliders (or other intuitive interfaces) to see how the model changes. *Mathematica* has set up an entire website devoted to demonstrations of mathematical and scientific phenomena based on Manipulate (<http://demonstrations.wolfram.com>). This has further strengthened the ability of students in the course to determine whether their model is working, and to understand how changes in initial conditions or in parameter values can induce quantitative or qualitative changes (i.e., bifurcations) in their models.

### ***Student–Instructor Interactions throughout the Semester***

Many students do not realize how much (or how little) they understand of material until they attempt to solve problems. As they solve problems, the timing and nature of their interactions with an instructor are very important. If the instructor simply shows the student how to solve the problem immediately, little is gained. If the instructor waits until the student has become stuck and frustrated with the problem and has given up, little is gained. Ideally, the instructor should give the student time to grapple with the problem, check regularly to see whether the student is making progress, discuss ideas for solution, and then allow the student to solve the problem on his or her own. A student also may feel that he or she has solved the problem once a final formula or graph or plot of data has been produced, but the instructor needs to probe by asking questions to ensure that the student has a deep conceptual grasp of the material. For example, it may be possible for a student to produce a correct bifurcation diagram but not really under-

stand what the plot means. Similarly, a student in math or engineering may have an excellent understanding of analytical techniques, but find the complex details of biological systems bewildering; seeing how these details can be analyzed in the context of a model can help that student grasp the significance of the details and master complex biological information.

As a consequence, the three authors make sure to have regular, individual discussions with students during every class. Each instructor is assigned to two of the six tables (i.e., has primary responsibility for 12 students in six teams) and the assigned tables change each class. Each instructor walks around the classroom to see how students are doing, answers their questions, and checks them off on the benchmark questions. The instructors generally use a Socratic approach, asking the students leading questions, and guiding the student to grasp the correct answer, rather than stating the answer. The instructors also strongly encourage students to rely on information from their teammates, from other members at the table, from the Web, from other textbooks, or from students and faculty outside of the class, if this helps them understand the material better.

After each class, the instructors meet to review how each student and each team performed. This allows us to assign extra credit for students who have provided useful feedback (e.g., indicating ways in which the book could be clarified or spotting errors), assign penalties (e.g., for absences or late work), identify potential team or student problems, and discuss how to help the students better understand the material.

### ***Assessment and Connection to Educational Goals***

Unless assessment is tightly tied to educational goals, the goals are unlikely to be achieved. For example, if a teacher claims that he or she wants to encourage critical thinking but then does not encourage students to challenge his or her statements, does not assess student progress based on their ability to critically analyze data and ideas, and does not administer tests on which students can demonstrate their ability to critically analyze new data (Chiel, 1996), students will quickly realize what the teacher really wants, and act accordingly. If the teacher actually assesses the students based on multiple-choice questions, short answers, or using problems that have a single correct numerical answer, students will use their previous strategies to do well in the course: they will memorize answers, repeat the teacher's statements, and memorize and apply "problem templates."

Ideally, feedback should be intrinsically based on the student's realization that he or she has solved the problem correctly, and not come from the teacher at all. From 1995 to 2006, one of us (H.J.C.) taught a course with Drs. Randall Beer and Richard Drushel (*Autonomous Robotics*; Beer *et al.*, 1999). During the first half of the semester, students were taught the basic principles of mechanical, sensor, and control design. During the second half of the semester, they worked in teams to construct autonomous robots that participated in a public competition at the end of the semester. What made the course so exciting and effective for the students was that they could tell very quickly whether their device was working. Especially in the second half of the

semester, when they were working to create their own novel autonomous robot, they were the experts on how their robot worked. The instructors could provide helpful suggestions for fixing programming or mechanical problems, but ultimately the solutions were up to the students. The feedback that mattered was whether the robot worked or not. At the same time, the final grade in the course did not depend on whether the robot won or lost in the competition, but on how well the student documented and analyzed the process of designing and testing the robot.

The structure of the Dynamics of Biological Systems course was based on the same principles: 1) provide students with the intellectual tools they need in the first half of the semester; 2) allow them to construct, analyze, and extend a complex model in the second half of the semester; and 3) assess them based on their effort to master the material, and their ability to clearly describe the model and the results they obtained.

We combined informal and formal assessment. Informally, we spoke to each student during each class session, answering their questions and asking them to explain the results they had obtained as they solved problems. During the second half of the semester, we asked them to describe the progress they had made toward reconstructing and analyzing their model, and we regularly helped them if they got stuck during the process. At the same time, we asked them to submit drafts of components of their term paper, so that we had a continuous record of how well they understood the significance of their model, its components, and the results they had obtained.

Formally, at the end of the semester, we carefully read the original journal article on which their reconstructions were based, and then carefully analyzed each component of the term paper. We evaluated the paper based on 1) the extent to which the model was replicated (i.e., full, partial, limited replication); 2) the level of understanding demonstrated by the term paper (on a scale from 1 to 5, in which 5 is the best); 3) the quality of the writing (on a scale of 1–5); 4) the kind of model extension (none, parameter manipulation, changing the inputs to the model, changing the form of the model, or doing dynamical analysis); and 5) the quality of the extension (on a scale from 1 to 5). The term paper grade constitutes half of the final course grade; the other half of the grade is based on class participation, which includes completing the benchmark problems by specific times, regular class attendance, and direct involvement in the process of learning the material.

Starting in the spring 2010 semester, we also asked students to fill out a brief questionnaire on their attitudes toward and sense of competence in biology, mathematics, and computer programming, to assess their initial views and how the course might affect these views. At the end of the semester, we also asked two additional questions to assess the effects of the course: 1) Before this course, had you ever constructed a mathematical model of a biological system? 2) After taking this course, are you more likely to construct or use a mathematical model to understand a biological system?

All statistical tests were done in the open source statistical program R ([www.r-project.org](http://www.r-project.org)) or in *Mathematica*.

### ***First Half of Semester: Key Intellectual Tools, Benchmark Problems, and Continuous Progress***

During the first half of the semester, students are taught the key intellectual tools they will need for the second half of the semester: 1) how to use *Mathematica* for basic calculations and for symbolic math; 2) how to program in *Mathematica*; 3) how to set up and numerically integrate coupled nonlinear differential equations; 4) concepts and tools from nonlinear dynamical systems theory, including equilibrium points, limit cycles and chaotic attractors, stability, and bifurcations; and 5) how to set up models in Manipulate, and use it to explore the effects of changes in initial conditions and parameters on quantitative and qualitative properties of the model.

Students are assigned reading in the textbook throughout the first half of the semester (<http://slugoffice8.biol.cwru.edu/~hjc/DynamicsCourseMaterials/index.html>):

Chapter 1 provides a general introduction to modeling, an outline of the book, and the approach to teaching modeling.

Chapter 2 presents a hands-on introduction to *Mathematica*, providing students with the ability to import, manipulate, plot and save data; to define their own functions, solve equations, numerically integrate differential equations; and to program. Small problems throughout each section of the chapter provide students immediate feedback as to whether they have mastered each component of the material. Solutions are provided in the text for these problems as quick self-checks for the students. The problems at the end of the chapter are more challenging and require students to combine information from the entire chapter. No solutions are provided for these problems. A quick reference key to *Mathematica* commands also is provided.

Chapter 3 introduces students to the iterative process of experiment and modeling, using the example of bacterial growth. An initial recursive model of bacterial growth is compared with actual data and improved using an exponential growth model. The new model can more accurately capture the data but fails after time because it predicts unlimited bacterial growth. This leads to the derivation of a model that can incorporate limits on population growth (the logistic model). The problems at the end of the chapter allow students to derive different models of growth, apply the models to data, do some additional mathematical analysis of components of the models, and practice programming for data analysis.

Chapter 4 uses the exponential and logistic models developed in the previous chapter as the basis for introducing one-dimensional nonlinear dynamical systems theory: trajectories and flows, limit sets, and bifurcations. The problems at the end of the chapter allow students to explore a saddle-node bifurcation in a one-dimensional system.

Chapter 5 presents the basic molecular mechanisms of the cell cycle and then focuses on a simplified model of the cell cycle developed by Tyson and Novak (2001), introducing molecular kinetics (zeroth order, first order, and second order) and enzyme kinetics so that students can understand each term in their two-dimensional model. Once the model is constructed, the problems allow the students to use Manipulate to visualize how the cell moves between two molecular checkpoints (i.e., two stable equilibrium points).

Chapter 6 uses the model developed in Chapter 5 to extend the concepts of nonlinear dynamical systems theory to higher dimensions, introducing the concepts of phase plane analysis, equilibrium points in two dimensions, and saddle points, and then introduces the students to eigenvalues and eigenvectors, bifurcations in higher dimensions, and hysteresis, so that they can understand the dynamical mechanisms underlying a bistable switch in the cell cycle as the cell grows in size. The problems at the end of the chapter allow them to use Manipulate to visualize the bifurcation in the phase plane as well as to deepen their understanding of numerical integration.

Chapter 7 develops the Morris/Lecar model of neuronal excitability from first principles (Rinzel and Ermentrout, 1989) and uses it to understand rhythmic behaviors, complex eigenvalues, and Hopf bifurcations. The problems at the end of the chapter allow students to study another two-dimensional system that undergoes a Hopf bifurcation and to explore the three-dimensional Lorenz attractor and chaos.

Once all students have checked off on the benchmark problems, the answers are posted on the class website, so they can be used as references for the students during the second half of the semester.

Chapters 8–10 develop other models, but these are beyond the scope of the material that can be covered in one semester, if the second half of the semester is to be devoted to model reconstruction. Thus, the final assignment is Chapter 11, which develops an agent-based model of fish schooling (Inada and Kawachi, 2002) as an example of the process of analyzing a complex model, breaking its reconstruction down into subcomponents, and then implementing and testing these components, providing a concrete example of the process of model reconstruction just before the students begin to reconstruct their own models.

By each class session, students are supposed to have read specific sections of the book (sections that are due are posted online on the course website, hosted by H.J.C.) and must solve assigned problems by specified dates. Each problem is designed to sequentially present students with challenges based on the key intellectual tools that they need to master. Toward the latter part of the first half of the semester, as students are posed a series of problems for analyzing coupled nonlinear differential equations, the guidance for analyzing the equations becomes more general, and the aspects of the model they must analyze and understand become more challenging.

After the first session, in which H.J.C. introduces the structure of the course, and discusses the rationale for using the tools of mathematics and simulation in biology, and the value of understanding biology to students of math and engineering, there are no lectures. We check off students on the benchmark problems throughout class sessions, making sure that they understand the material. Late check offs lead to penalties in class participation points, providing students with a strong motivation to keep up with course material. By speaking to students after they have worked hard with their teammates to master the material and solve the problems, the instructors can provide the maximally helpful guidance and feedback. The goal is to allow the students to make continuous progress toward mastery of the material they will need in the second half of the semester, rather than to

have them solve discrete problem sets. The instructors have observed that problem sets and midterms that are noncumulative encourage students to master material until they have taken the exam, after which students assume that they can forget what they've just learned, and move on to the next material to learn. In contrast, in this course, every problem that a student solves provides him or her with skills needed in the second half of the course, and the second half of the semester helps consolidate what students have learned by applying the concepts and tools in a new setting.

### *Second Half of Semester: Guidance during Model Reconstruction and Term Paper Writing*

To allow students to immediately apply the knowledge they have learned in the first half of the semester, during the second half of the semester, they reconstruct a model that has previously been published in the technical literature. Papers are not preselected by the instructors. This ensures students spend some time exploring the primary literature themselves, and that the papers chosen are more likely to interest students. Students are shown how to find appropriate modeling papers using a variety of databases (e.g., Science Citation Index, PubMed, Google Scholar) and encouraged to find a paper that interests them. General guidelines are provided: the paper should describe a mathematical model, not a curve fit; the number of differential equations should probably not be more than approximately 10 to 15; and the paper should not primarily involve mathematical proofs but should focus on some simulation results and some dynamical analysis.

After the students have selected possible papers, the instructors read and screen them. Papers are approved based on their suitability for reconstruction and are also matched to the level of the team, based on the team's performance on the benchmark problems and their understanding of the material. Thus, more challenging papers are approved for teams that can readily handle more advanced material, and teams that have had more difficulty in the first half of the semester are encouraged to work on a paper better matched to their skills. This helps to level the playing field for all the teams and ensures that all the students are appropriately challenged during the second half of the semester. Before the midsemester break, students must submit a one- to two-page proposal describing the model, the process of reconstruction, and the ways in which they will analyze and extend the model.

During the second half of the semester, benchmarks are focused on several goals: 1) breaking the reconstruction of the model into a manageable sequence of subgoals; 2) recognizing and describing the different components of the model verbally; 3) reconstructing and testing model components and then the model as a whole; and 4) creating drafts of components of the term paper that must be handed in as the model is being reconstructed, which are evaluated using a rubric that is shared with the students. Thus, the continuous progress approach also is used during the second half of the semester. However, as the semester progresses, the instructors emphasize that the students are now the experts on their models and encourage them to explain the problems they are having and how the model is working so that the instructors can provide advice and guidance. On some oc-

casions, one of the instructors will create a code component to help a group that has encountered an obstacle, but ultimately the students build their own models.

### Student Presentations

One potential drawback of the structure of the course is that the class as a whole does not get to see the full range of models that have been reconstructed during the second half of the semester. In the last two semesters that the course has been offered, we have addressed this problem by devoting the last session of the class to student presentations. Teams are encouraged to volunteer to give a 15-min presentation, for which they may gain up to 5 points of extra credit out of the 100 possible points for their term paper. The two students in the team present the rationale for the model, the central hypothesis tested by it, the equations and parameters, and the results they obtained. They discuss any discrepancies they found between the model as described in the paper and what they actually needed to do to implement the model, the extension to the model they created, and proposed future work. Students from the rest of the class then have time to ask questions. Students are thus exposed to a much wider range of models than just the one that they themselves reconstructed.

## RESULTS

### Class Composition

In the 10 yr that the course has been offered, 242 students have enrolled. The enrollment per semester has grown considerably. The initial offering of the course had two students. In the past 3 yr that the course has been offered, the enrollment has been 35, 36, and 38 students, respectively. Overall, the percentage of men taking the course has been 55%, and the percentage of women has been 45% (Figure 2).

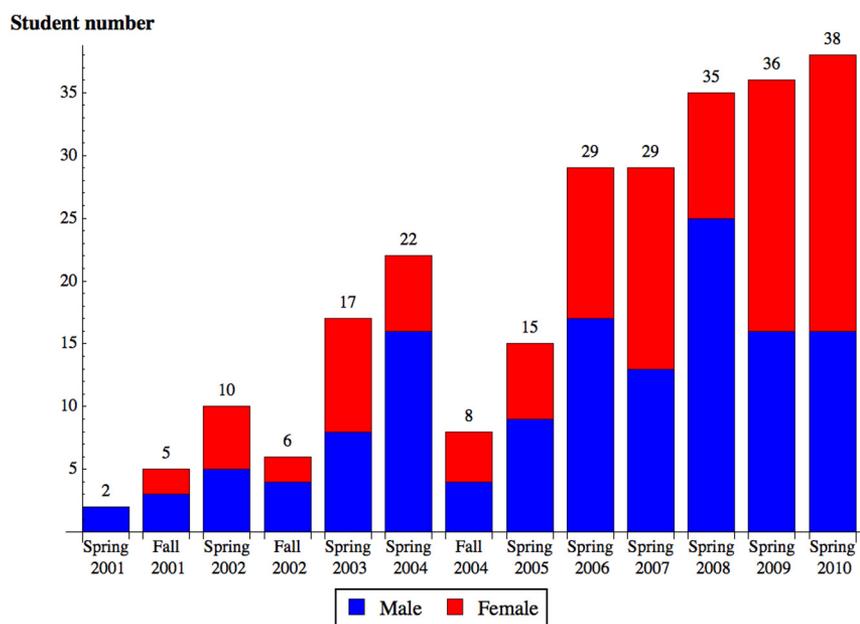
Students mainly take the course in their senior year or junior year. Graduate students and first-year students have enrolled in the course, and after the Systems Biology major was established in 2007, students began to take the course in their sophomore year. The percentage of seniors in the course (66.7%) has been, on average, about three times the number of juniors (21%) (Figure 3).

The class has attracted students from a wide variety of majors (Figure 4). Not surprisingly, biology majors generally constitute at least half the class. The majority of the rest of the students are biomedical engineers, chemical engineers, or majors in biochemistry or chemistry.

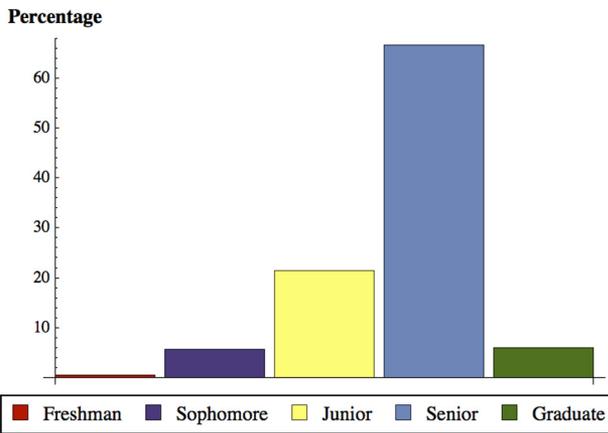
### Final Grade Distribution

In general, students who work hard in the course do very well. Slightly more than three-quarters of the students in the course (78%) earn a grade of "A" for the semester. Another 15% earn a grade of "B," and the remaining students receive lower grades (Figure 5). Distributions of grades did not differ significantly between males or females, or between students majoring in biology or in other subjects. Generally, the grades are not surprising either to the instructors or to the students, because students have received regular feedback throughout the semester, not only from the instructors but also from their teammates and from their models, which are working (or not working). For almost all of the students who take the course and are majoring in biology, this is the first time that they have actually created a working mathematical model, simulated it, analyzed it, and written about it (as we determined this semester, of the 17 biology majors, 14 [82%] had not previously constructed a mathematical model of a biological system).

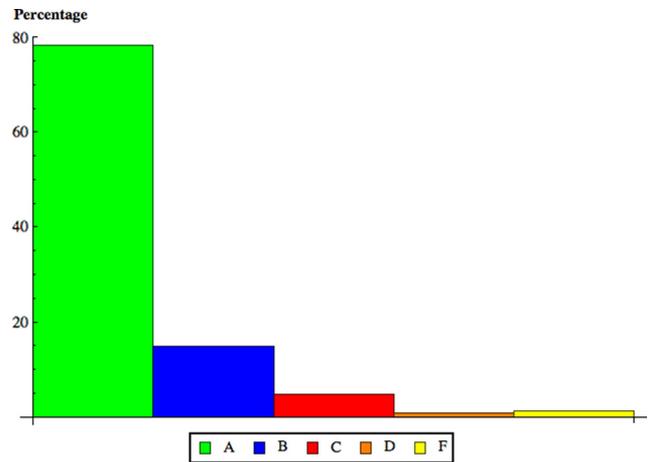
Students who do poorly in the course do so for fairly clear reasons that may have little to do with the course or its structure. Some students simply stop coming to class, or miss many classes without valid reasons to be absent. Other students



**Figure 2.** Total enrollment in the course from 2001 to 2010 by gender.



**Figure 3.** Distribution of students enrolled by year (freshman, sophomore, junior, senior, or graduate student) over all years that the course has been offered.



**Figure 5.** Grade distribution for all students that have taken the course from 2001 to 2010.

consistently fail to complete benchmarks on time, and accrue late penalties. Given that half of the grade is class participation, they have little chance of getting a good final grade. Some students fail to hand in a final term paper, or hand it in late. The late penalties for the term paper are strict and very steep: one letter grade per day that the paper is late. Because students have the entire second half of the semester to work on their models and write the paper, this penalty is reasonable, but some students fail to take it seriously, with devastating consequences for their final grade. Some of the weaker students write very poor term papers, demonstrating a fundamental lack of understanding of their model, reporting almost no results, no analysis and no extension. The instructors have ensured, for all students, that by the end of the semester, their model (or at least some of its components) works, and students who use that as a basis for a carefully

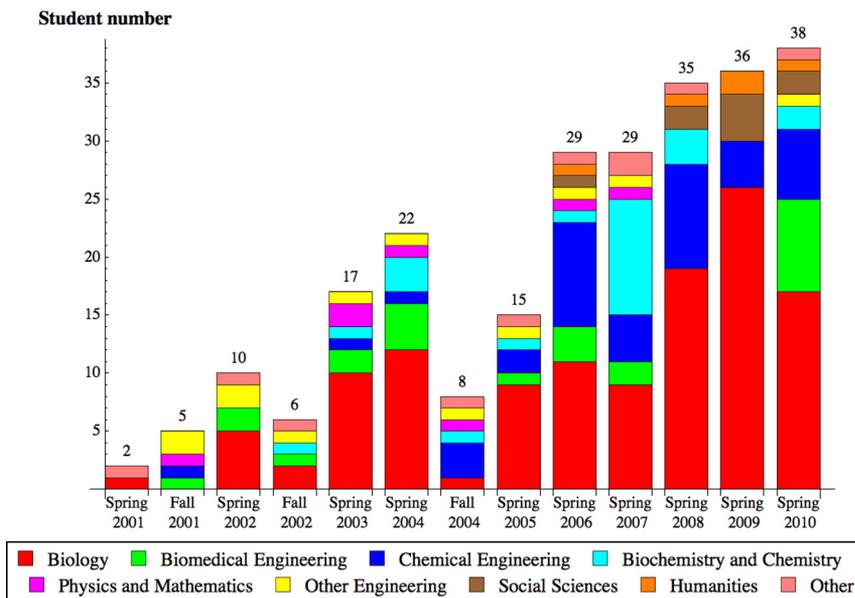
written and thoughtful term paper can still write a paper that can earn a grade of "A."

**Student Evaluations of Course and Student Comments**

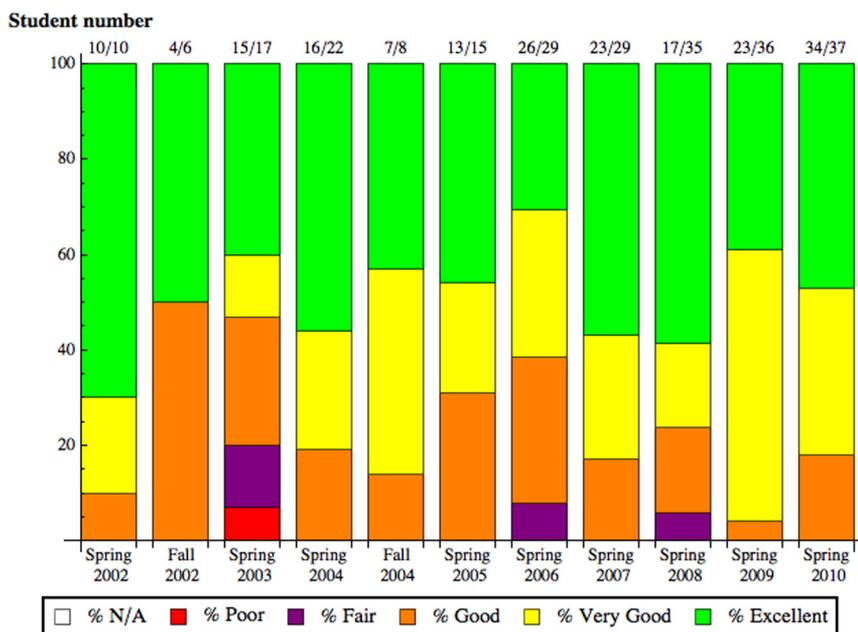
In general, student evaluations of the course have been strongly positive. In the past 4 yr, the sum of the percentage of responses that have been excellent or very good (green and yellow segments of bars in Figure 6) has ranged from 61% to 96%.

Student comments on the course are generally very positive. Comments from the course offerings in 2009 and 2010 are typical of those that have been provided throughout the years the course has been offered. Several students commented on the team structure:

"I really liked working in pairs. It was very helpful to have another person to work with."



**Figure 4.** Distribution of majors among students enrolled in the course from 2001 to 2010.



**Figure 6.** Student evaluations of the course. The fractions at the top of each bar indicate the number of students who provided evaluations over the total number of students in the course that semester. For example, in spring 2008, only 17 of the 35 students in the course provided evaluations (interestingly, this corresponded to the year in which CWRU switched from paper forms that were filled out in class to online evaluations that students could fill out at any time). Compliance was higher in 2010 (34/37) when the instructors repeatedly requested that students fill out the evaluations before the deadline.

“The partner system built in allows for those students who are more mathematically-based to supplement those that are more biologically-based, creating effective teams.”

“The partner system can work very well when supplemented by instructors, you are obliged to try especially hard when someone else is relying on you.”

Students acknowledged that the course was challenging but felt that they had the help they needed to succeed:

“I thought *Mathematica* was initially hard to understand, but the progress made during the course was exciting. I felt like I’ve learned a lot about how to use *Mathematica*.”

“I enjoyed doing the benchmark exercises. They were challenging, and gave good use of the material in the book.”

“I have found it rather incredible that I have basically learned a new program in such a short period of time. It is quite impressive and definitely something to be proud of.”

Students also appreciated the overall structure of the course:

“The course structure was excellent in the way it ushered in an understanding of the programming and worked up to a research project.”

“This was definitely a different laboratory class than any other offered. It was challenging and independent and well integrated. The course was well thought out and stayed particularly on schedule.”

“The first half = hands on, second half = project was a great setup.”

“The overall course structure and subject material was very well thought out and prepared us perfectly for

recreating other papers. It was obvious that the instructors devoted a large amount of time to the class, which was not spent preparing lectures, but discussing student progress etc. The textbook served as the lectures so instructors were free to focus on the individual.”

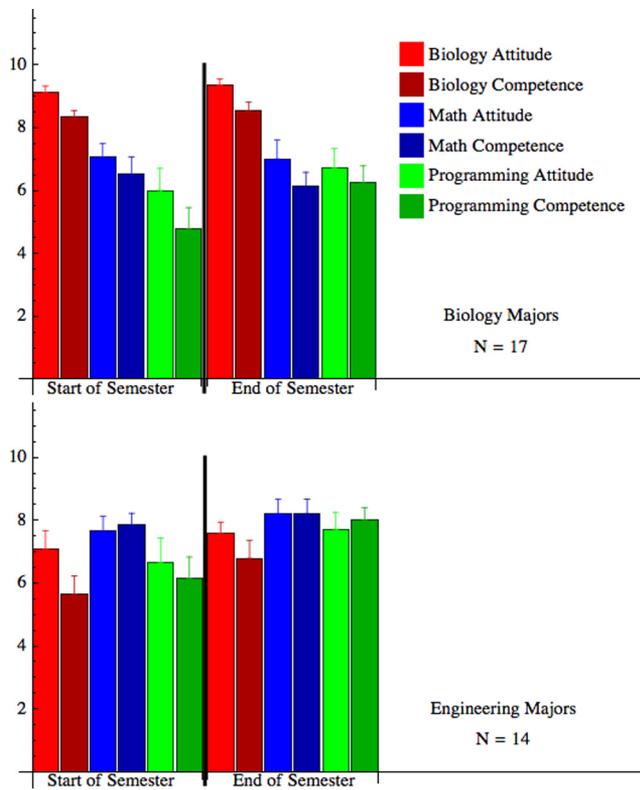
“It was a breath of fresh air compared with my other courses. A unique approach to teaching programming in a more relaxed and friendly environment.”

“GREAT CLASS. Push the biology department toward this style of teaching for genetics, dev. bio, and other classes. I think a teachers time is better spent explaining material a student has already read about instead of droning on in front of 100+ students.”

“This was one of the best classes I’ve taken here - I wish I would have taken it before some of my other computational modeling classes! I now understand some of the underlying principles better and feel *Mathematica*-adept and confident in my research writing abilities.”

### *Assessment of Student Attitudes and Competence*

In the 2010 semester, we implemented an additional form of student evaluation, a self-rating of attitudes and competence. At the beginning of the semester, just before spring break, and at the end of the semester, students were asked to fill out a survey in which they were asked six questions, on a scale from 0 to 10 (0, worst; 10, best): 1) How much do you like biology? 2) How competent are you in biology? 3) How much do you like mathematics? 4) How competent are you in mathematics? 5) How much do you like programming? 6) How competent are you in programming? The survey was administered on a secure course Web page, and although student names were linked to the survey results, one of us (Shaw) programmed the system so that instructors were not able to see this information until after grades were assigned; students were informed of this before being required to take the survey as part of their class participation grade.



**Figure 7.** Student attitudes toward and sense of competence in biology, math, and computer programming. The figure shows the average and s.e.m. for answers to six questions (e.g., How much do you like biology? How competent are you in biology?) that were rated on a scale from 0 to 10 at the beginning and at the end of the semester. The bar charts show all biology majors ( $N = 17$ ) separately from all engineering majors ( $N = 14$ ), i.e., students majoring in either biomedical engineering ( $N = 8$ ) or chemical engineering ( $N = 6$ ). It is clear that the initial attitudes and sense of competence of biology students significantly differs from that of engineering students. It is also clear that all students show a significant change in their responses after taking the course (for details, see *Results, Assessment of Student Attitudes and Competence*).

The data support the hypothesis that, before taking the course, biology and engineering majors have significantly different attitudes toward and senses of competence in the areas of biology, mathematics, and computer programming (Figure 7). A comparison of the biology majors to the engineering majors (i.e., students majoring in biomedical or chemical engineering) on all six questions demonstrated that the overall differences between the two groups were highly significant ( $N = 17$  biology majors,  $N = 14$  engineering majors; multivariate analysis of variance,  $p < 0.003$ ). The major differences were in the attitude and sense of competence in biology, which were significantly lower for engineering students ( $p < 0.002$  and  $p < 0.0001$ , respectively; post hoc Tukey's honestly significant difference tests). Although biology and engineering majors did not differ significantly in their attitudes toward or their sense of competence in computer programming, the engineers showed a clear trend toward a higher sense of competence in mathematics ( $p = 0.057$ ).

After the students took the course, the survey indicated that the course significantly affected the overall attitudes and sense of competence of students in both groups when they were compared with themselves. Biology majors showed significant changes over the course of the semester (the multivariate analogue of the paired  $t$  test, Hotelling's  $T^2$  statistic, yielded an overall  $p < 0.04$ ). Biology majors showed a highly significant increase in their sense of competence in programming ( $p < 0.006$  [post hoc  $t$  test using the Bonferroni correction]). No other changes were significant.

Engineering majors also showed significant changes over the course of the semester (paired Hotelling's  $T^2$  statistic,  $p < 0.002$ ). They showed a highly significant improvement in their sense of competence in programming ( $p < 0.0004$ ) and a strong trend toward an increased sense of competence in biology ( $p < 0.03$ ) and a more positive attitude toward programming ( $p < 0.023$ ; post hoc  $t$  tests using the Bonferroni correction).

Students also were asked two final questions at the end of the semester: 1) Before this semester, had you constructed a mathematical model of a biological system? 2) After this semester, are you more likely to construct or use a mathematical model to understand a biological system? We were particularly interested in the responses of those students who had not previously done mathematical modeling, because they would be most likely to be directly influenced by the experience of taking the course. Of the 17 biology majors, 14 had not previously constructed a mathematical model of a biological system, and 12 of these 14 students (86%) were more likely to construct or use a mathematical model to understand a biological system. Of the 14 engineering majors, seven had not previously constructed a mathematical model of a biological system, and six of these seven students (86%) were more likely to construct or use a mathematical model to understand a biological system. In the entire class, across all majors, 26 students had not previously constructed a model, and of these students, 22 (85%) said that they were more likely to construct or use a mathematical model to understand a biological system. Using a null hypothesis that half the students who had no prior modeling experience might have changed their views on modeling over the course of the semester even if they had not taken the course, the probability of seeing this many students adopting a more positive view was very low (binomial test,  $p < 0.0006$ ). This is an extremely conservative null hypothesis, because it is highly unlikely that half of the students would show spontaneous improvements in their views about mathematical modeling of biological systems; assuming a lower rate would increase the significance of the results.

### Assessment of Team Dynamics

Students provide weekly feedback on team dynamics (rating their contribution, their teammates' contribution, and the team function on a scale from 0 to 10). Two aspects of these ratings stand out. First, as teams encounter difficulties, either in solving the benchmark problems during the first half of the semester, or in reconstructing the model in the second half of the semester, ratings tend to drop. Second, there is no clear association between these changes in ratings and the composition of the teams. It was as likely that biology majors would lower their ratings as engineering majors, and it was

just as likely to observe changes in ratings in teams consisting of two biology majors, a biology major and an engineering major, or two engineering majors. Thus, the team dynamics was not appreciably better or worse as a function of the majors of the team members.

The overall ratings of teams varied over the course of the semester. In the most recent offering of the course, there were initially 19 teams for the 38 enrolled students. One student withdrew during the first half of the semester for health reasons, and the student who remained was added to a second team. Thus, there were 17 teams whose composition was unchanged during the course of the semester. Of these 17 teams, nine showed no significant change in ratings throughout the semester. Six teams showed steady improvements in ratings throughout the semester. Two teams showed worsening ratings. These two teams were both working on especially difficult models and struggled to complete them before the end of the semester. There was again no association between the majors of the team members and the tendency for the ratings to remain the same, to improve, or to worsen.

### *Formal Assessment of Student Performance*

Formal assessment of student progress was provided by the drafts of the sections of the term paper (Introduction, Model Description, Results, and Discussion) that were due on successive weeks during the second half of the semester (weeks 9, 10, 12, and 13 of a 13-wk semester). The three instructors met to read the relevant sections of the original papers from the technical literature and then to read and discuss the students' drafts of each section. We provided suggestions for improvement for all drafts of all sections. With permission from students who had submitted excellent work, we posted on the course website exemplary versions of student drafts of each of the sections to serve as a guide for all the students.

For the Introduction section, each submission was evaluated based on how well it addressed the following three rubric questions: 1) Have you explained the significance of the problem? 2) Have you stated the hypothesis that the model will test? 3) Have you provided citations? Drafts that did not meet the rubric were returned to the student for revision, and a late penalty was applied. We returned 11 drafts for revision (30%).

For the Model Description section, each submission was evaluated based on how well it addressed the following three rubric questions: 1) Have you provided a brief overview of the key state variables, parameters, and inputs to the model? 2) Have you described the assumptions underlying the model? 3) Have you provided a verbal description of the components of the equations of your model, and a description of how you simulated it? We returned nine drafts for revision (24%).

For the Results section, each submission was evaluated based on how well it addressed these rubric questions: 1) Have you described the results you obtained and compared them to the results obtained with the original model? 2) Have you provided figures with appropriate legends to show your results? 3) Have you described any discrepancies that you found relative to the original published model? We returned two drafts for revision (5%).

For the Discussion section, each submission was evaluated based on these rubric questions: 1) Have you drawn clear conclusions from the model simulations? 2) Have you discussed the limitations of your results? 3) Have you related your results to other work in the field, with appropriate citations? Because the Discussion section was submitted during the last week of the semester, we did not return any drafts, even if they fell below the rubric, but we provided feedback as to whether the section met the standard.

Student presentations were extra credit, because we could not accommodate presentations by all teams. We carefully assessed presentations based on the clarity of their introduction and background, the statement of the model hypothesis, the clarity with which they presented the model, how they extended the model, the results they obtained, and their discussion of results and future work. Students were given up to 5 points of extra credit for their presentations.

As described above, in addition to the rubric questions used for the drafts, term papers were evaluated based on the extent to which the model had been fully replicated, the difficulty of the model, the level of understanding students illustrated in their write up, the quality of the writing, the kind of extension to the model that students had created, and the quality of the extension. In the most recent semester, the mean grade was 90.7 of 100, and the median grade was 93. There was, however, no significant difference in grades of students majoring in biology relative to the grades of students majoring in engineering ( $92 \pm 6$ ,  $90.4 \pm 6$  [mean  $\pm$  std. dev.], two-sided *t* test,  $p = 0.26$ ), suggesting that the level of mastery in both groups of students was comparable.

### *Obstacles to Reconstructing Models*

Reconstructing models poses several challenges to students. There are both conceptual and technical obstacles that they may encounter. First, they may have difficulty understanding how the specific components of the actual biological system are mapped into the mathematical descriptions. Second, they may have difficulty understanding (and thus distinguishing) initial conditions, state variables, and parameters. Third, they may have difficulty in understanding the significance of the terms within the model equations. Fourth, the actual implementation of the model may pose difficulties, especially if the model requires that pulses be applied to the system (e.g., chemotherapeutic agents applied to cancer cells at particular times) or that state variables be reset to keep them within specific ranges of values. Fifth, they may have some conceptual difficulties with nonlinear dynamical systems theory, which is often used for analyzing biological models. These difficulties have served as the basis for the design of the benchmark problems in the first half of the semester and for the requirement that drafts of sections of the term paper be handed in during the second half of the semester. These benchmarks allow the instructors to regularly assess student understanding and to intervene when necessary to help students with either conceptual or technical problems.

In addition to these difficulties, some published peer-reviewed papers have errors that range from minor to very serious. Although the instructors encourage the students to assume that the papers are correct, after students have worked hard to reconstruct and check model components,

and continue to see serious discrepancies between their results and those shown in the original paper, it becomes more likely that there may be a problem in the paper, and not with the students' reconstruction. Sometimes key information is missing (e.g., initial conditions); sometimes the values of parameters are missing, or wrong; sometimes there are errors in the equations. The instructors encourage students to contact the authors of the papers via e-mail, and on occasion, students get back very helpful e-mails from the authors. In the most recent semester, of the 18 papers that served as the basis for model reconstructions, six had no errors; 10 had errors in parameter values so that students had to change the values to reproduce the published figures; and in two papers the model equations had typographical errors and were not sufficiently well specified (e.g., terms were not defined), so that students had great difficulty in reproducing the model results. These error frequencies are typical of those we have observed in other semesters.

### Assessing Student Creativity: Extending Models

Another way to assess students' comprehension and mastery as well as to provide them opportunities to express their own creativity, is to encourage them to extend the models that they have reconstructed. Extending a model allows students to apply the knowledge they have learned to a new but related problem. More generally, allowing students to explore areas that have not been previously studied gives them a much better sense for the excitement of research. Although we strongly encourage students to work together as they reconstruct the published models, we allow students to work together or separately on the extensions, depending on their personal preferences.

Extensions fall into four general categories: 1) Students carefully vary the values of key parameters of the model that were not varied in the original paper, and explore the effects on the model. Using the *Mathematica* function Manipulate, it is relatively easy to set up a model with an interface that can quickly vary initial conditions and parameters. 2) Students modify inputs to a model. For example, a model exploring the responses of tumors to chemotherapeutic agents can be varied by applying the treatments in different concentrations and in different temporal patterns. 3) Students modify the form of a model. For example, if a model has a term that allows one of the state variables to grow exponentially, replacing the term with a logistic term may more realistically capture the initial rapid growth of the state variable, followed by its reaching a long-term stable value. 4) Students apply the tools of nonlinear dynamical systems theory to analyze a model. For example, determining the stability of equilibrium points, and (for more advanced students) doing a bifurcation analysis of the model as key parameters are varied.

Extensions created by students in the current semester illustrate each of these alternatives. One group extended a model that explored the effects of preconditioning on the responses of Toll-like receptors to lipopolysaccharides (Riviere *et al.*, 2009) by using a different parameter value for the maximal expression of the Toll-like receptors, based on published research that described the effect of overexpression of Toll-like receptors (Bihl *et al.*, 2003). A second group modified the inputs to a model of wound healing that in-

corporated the responses of fibroblasts and the likelihood of infection (Menke *et al.*, 2009) to explore the effects of successive wounding, or of hyperbaric therapy (Tibbles and Edelsberg, 1996). A third group took a model of honeybee nest selection (Britton *et al.*, 2002) that explored how honeybees select between two nests and added new equations to explore how honeybee behavior changed if the honeybees needed to select among three nests. Finally, a fourth group took a model of drug efficacy and hepatitis B viral levels (Dahari *et al.*, 2009) and determined the stability of the equilibrium points of the model.

In the current semester, 11 students varied parameters of their models, five students altered the inputs to their models, 14 students changed the forms of their models, and four students did additional dynamical analyses. Two teams chose to reconstruct very complex models, and so four students did not have time to create extensions. The best work done by the students on model extensions could almost certainly serve as the basis for new publications.

## DISCUSSION

By creating an interactive textbook and providing students with regular feedback from both instructors and peers that ensures that they make continuous progress, we have created a course that successfully allows students to develop and analyze their own mathematical models of biological systems. At the same time, the course introduces students with strong backgrounds in math and engineering to some of the excitement of research at the frontiers of biology. Despite the challenging nature of the material, the majority of students do well in the course. In addition, it seems that the course induces significant improvements in a sense of competence in computer programming in all students.

### Assessing Continuous Progress

Regular quizzes, exams, and papers provide a great deal of "objective" information about student progress to both students and teachers. How useful are these assessments for predicting how well a student will do in the real world? One of us (H.J.C.) has had undergraduate students working on research projects in his laboratory since joining the CWRU faculty in 1987 and has repeatedly noted that many students with nearly perfect academic records do very poorly when confronted with real, open-ended problems, whether they are mathematical, computer, or experimental biology problems. In contrast, many students with much less stellar academic records do outstanding work when given the opportunity to solve meaningful problems that genuinely interest them.

Part of the problem is that the normal process of assessment is based on the assumption that a teacher conveys information to the student, and if a student can solve problems posed by the teacher using this information, the student has mastered the material. In real-world situations, however, problems do not have single, well-defined answers that are neatly solved by using one key piece of information. In fact, much of the solution may depend on first properly defining the problem itself, and the solution may draw on many disparate ideas and approaches.

The continuous progress approach provides students better training for real-world problems. Studies of continuous progress in elementary school suggest that it may be very effective (Ysseldyke and Bolt, 2007). Previous research also has shown that active inquiry, knowledge construction, interaction with peers and teachers, and reading and writing science greatly enhance K–12 science education (Pearson *et al.*, 2010), and our results suggest that this is equally true for undergraduate students. As students focus on problem solving with regular suggestions from the instructors, joint efforts with their teammates, and immediate feedback from the computer commands they have entered, they not only solve the specific problems but also gain confidence in their problem-solving ability, and their ability to use the tools at their disposal. Because of the nature of the continuous progress approach, the instructors have the time to get to know each student as an individual and therefore have a much deeper ongoing assessment of a student's level of mastery. If students can actually complete a difficult task like reconstructing a model, show creativity by extending the model, and write about their results clearly and with comprehension, it is clear both to them and to the instructors that they have made real progress toward mastering the material. This is far more satisfying to students than acing a tough exam, because it represents a meaningful piece of work. Indeed, one of us (H.J.C.) has successfully recruited several outstanding students for research work in his laboratory based on their success in this course, which is a far better predictor of a student's success in problem solving than his/her overall grade point average.

The additional assessments that we have reported in this paper—the levels of enrollment in an elective course; the positive student evaluations; and the self-evaluation questionnaire, which suggests that students perceive improvements in their attitudes and abilities after taking the course—are useful supplements to the most important form of assessment that we use, which is the actual ability of students to reproduce, analyze, and interpret models based on their term papers.

All three instructors taught the course in spring 2008, before introducing the continuous progress approach, and the same instructors taught the course in spring 2009 and 2010. Although the student/faculty ratio did not change, the ability of students to reconstruct models was strikingly enhanced after the continuous progress approach was introduced. Indeed, our ability to ask students to submit drafts of their term papers during the second half of the semester, and to devote one session to student presentations, was only made possible by the more rapid progress that students now make toward completing their models earlier in the semester. Students' depth of comprehension of their models and their ability to extend and analyze their models also has been reflected in the improved quality of their term papers.

In future offerings of the course, it might be useful for the students if we administer a conceptual pretest and posttest to help demonstrate to them more clearly the general level of conceptual mastery they have obtained. In addition, we intend to give students the opportunity to submit ratings of their own draft components according to the rubric we described, which we will compare with our own ratings, allowing us to determine the accuracy of student self-eval-

uations. We intend to try these other assessments when we next offer the course.

### *Self-Ratings of Changes in Attitudes and Competence*

Although the anonymous student ratings and comments have always been useful for the instructors in evaluating how to further improve the course, the self-rating questionnaire that we administered this semester allowed us to do a much finer analysis of changes, because we could distinguish students by majors, and use each student as his or her own control for determining changes over the semester. The results clearly support the general hypothesis stated in the *Introduction*: At the outset, biology students significantly differ from engineering students in their attitudes and sense of competence in key areas. Biology students have a significantly more positive attitude toward and sense of competence in biology than do engineering students (even those in biomedical engineering!); in contrast, engineers have a greater feeling of competence in mathematics. Given that new knowledge must build on pre-existing conceptions, and that motivation is crucial for learning, these data provide further support for the need to create undergraduate courses that directly address these pre-existing differences.

We were pleasantly surprised to see that, across the entire class, attitudes toward and the sense of competence in programming improved by the end of the semester. We hypothesize that this is due to two main effects. First, having students work to create a model that is intrinsically interesting to them is more motivating than solving general computer programming problems, even if these problems are well designed to illustrate important principles. Second, the Manipulate function in *Mathematica* makes it easy and even enjoyable to “play” with models, in ways that would ordinarily require much more intense programming effort.

We also were pleased to see that engineering students showed significant improvements in their attitude and sense of competence in biology. Building a quantitative model of a biological system, and applying mathematical tools to analyze that model, may constitute a much more effective way of introducing an engineering student to the complexity of biological systems. It is likely that the lack of significant changes in the attitudes of biology students is due to a ceiling effect: their attitudes toward and sense of competence in biology are near maximal before taking the course, so little further improvement is possible.

It is somewhat disappointing that biology students showed no significant improvement in their attitudes toward and their sense of competence in mathematics. This suggests that a single semester course, even one that explicitly focuses on the utility of mathematics for biology, is not enough to undo the years of negative experience that most biology students have had with mathematics. It is also likely that, after taking the course and interacting with engineering students, biology majors can now more accurately assess their competence in mathematics than they could before (Kruger and Dunning, 1999). In contrast, it is heartening that a very significant number of biology students with no prior modeling experience have indicated that, after the course, they are more willing to build or use mathematical models to understand biological systems.

There are limitations to the analysis that we have presented. Allowing students to serve as their own control group for determining attitude change is useful, but it is not feasible to track students who did or did not take the course and to determine whether biology students who have not taken the course continue to show negative attitudes toward mathematics and programming, whereas biology students who have taken the course show more positive attitudes. Unless the biology department adopted a policy of administering a required conceptual pretest to students shortly after they declare a biology major, and a required posttest in their senior year, it would be extremely difficult to obtain a random sample of students. A voluntary test would not attract a truly random sample of students. Individual instructors are unlikely to take limited class time to administer a conceptual test in areas that are irrelevant to those that they teach. Despite these caveats, our course does place engineering and biology students in the same classroom and encourages them to work together on the same material, providing an excellent measure of initial differences between them and their relative mastery of the material. Moreover, using students as their own controls provides a sensitive test of the effect of the course on their attitudes and sense of competence.

### *Team Dynamics*

Is it unfair to randomly assign students of differing abilities to teams? Stronger students with better backgrounds may feel burdened by working with weaker students. Friends may prefer to work together. We have tried a variety of approaches to setting up teams, including rotations, and letting students choose their own partners. In general, these approaches have been much less successful than random assignments. Students who are friendly out of class may not have worked together on assignments under deadline pressure, and this often creates great strains that affect teamwork. Students who do not know anyone else in a class may feel it is unfair that other students are allowed to work with friends. Two students who are equally strong sometimes fail to listen to one another, and end up accomplishing less than students who are less evenly matched. Rotations also make it impossible for a team to coalesce effectively, and generally lead to “parallel play” rather than a strong collaborative interaction.

When teammates are assigned randomly, we have often observed that students may initially think that their assigned teammate is weak, only to discover over time that both members of the team have complementary strengths. Other teams work well because the stronger student discovers that he or she has skills as a teacher and is able to clarify material by explaining it to the other student. More generally, studies of science teaching have emphasized the importance of argument and discussion for effective learning, which happens most readily between peers (Osborne, 2010). Because students write independent term papers, they are not assessed as a team and thus do not feel that the random team assignment penalizes them, even if they must work somewhat harder to accomplish the term goals. At the end of the semester, we allow students to work separately or as a team on their extensions, so that students who prefer to work alone are given this opportunity if they prefer it. These

observations are matched by the assessment of the team ratings that we reported in *Results, Assessment of Team Dynamics*.

### *Learning Styles*

The course attempts to address the variations in learning styles by allowing students to interact with each other and with the instructors as well as allowing them to create models in the computer and seeing how they work, which should be of help to students with visual, auditory, and kinesthetic learning styles. By working from concrete examples and allowing students to experiment, the course overcomes at least some of the obstacles that many biology students encounter to mastering mathematics and modeling. Similarly, by allowing engineering students to master large amounts of biological data in the context of creating a model, we give them a much more accessible introduction to biology that does not overwhelm them with detail. Students do comment that they would have liked some of the material presented through lectures, and this is likely to be especially helpful for auditory learners, but the gain in facility that all students show from working on the benchmark problems seems significantly greater than that which occurred when the first author lectured and assigned problem sets.

### *Consolidation of Learning*

A potential drawback of focusing on benchmark problems is that students will tend to read the minimum amount of the book that they need to solve the problems, and not try to learn the more general principles. This problem also occurs in more conventional courses, however, when problem sets are assigned and students are tested using exams. The advantage of the structure of the course is that students often need to revisit some of the techniques and concepts they learned in the first half of the semester during the second half of the semester, and this helps them to consolidate and generalize their understanding.

### *Grade Inflation?*

Although 78% of the students in the course have received a grade of “A,” students do not rate the course as easy. We are concerned about the possibility of grade inflation and carefully discuss and rate each term paper, and we also discuss our perception of each student’s conceptual grasp of the material. We see a grade of “A” as certifying that if the student were given a novel technical paper, he or she could extract its key equations, reconstruct the model, analyze it, and perhaps extend it. Given these criteria, we believe that students demonstrating this level of competence should be given a grade of “A” in the course.

### *Long-Term Impact*

What is the long-term impact of the course on students’ attitudes and career choices? This is a difficult question to answer, because one would need to do longitudinal studies of students matched by major and other variables, comparing the outcomes for students who did take the course with those who did not. A sampling of the career paths of the students who have taken the course suggests that many

have gone on to medical school, graduate school, and to professional careers in engineering; but no strong conclusions can be drawn from this, because these outcomes are typical for many graduates of CWRU. Because the majority of students who take the course are in the second semester of their senior year, they have made their career choices before taking the course, and many are in the midst of interviews for medical or graduate school. Anecdotally, several students have told the first author that the course had a significant impact on their choice of research, and their use of models.

### *Elective versus Required Courses*

How well can the approach outlined in this paper be applied to required courses, rather than elective courses? In general, the instructors have observed that when students are required to take a course, rather than choosing it freely, it is harder to motivate them. This is natural. Required courses imply that education is something done to the student, rather than something that the student chooses to do for himself or herself. If a required course has multiple sections, and an instructor wished to adopt some of the approaches outlined in this paper, it might be most effective to let students know that they have a choice of different approaches at the outset of the semester, and at least let them freely choose the different styles of instruction. Over time, word of mouth from students could increase enrollment in those sections that students found most effective at teaching them the material, and the approach could organically grow to dominate the course. Imposing it from the top down is unlikely to be as effective, especially if many instructors of the course are comfortable and effective with presenting material through lectures.

### *Effectiveness at Other Institutions?*

How successful would a course of this kind be at other institutions? One of us (H.J.C.) has had a longstanding interest in mathematics as well as biology, which made it much easier for him to create this course. At other institutions, it might require an ongoing collaboration between a mathematician and a biologist, who could coteach the course. Another strength of the student population at CWRU is that it includes both many students interested in the life sciences (often planning to go to medical school) and students who plan careers in engineering, and this has led to the very interesting mix of students in Dynamics of Biological Systems. Even if majors in biology were the only students in the course, the approach is likely to be successful, especially if there are enough instructors to help students past conceptual and technical problems.

### *Student/Instructor Ratio*

Would it be possible to scale up the size of this class? It is important to recognize that the course is a laboratory course, and these require relatively low student/faculty ratios to work well. Even in the large core courses at CWRU that enroll >300 students, the labs are taught in groups of 24–28 students, to which a graduate student and an experienced undergraduate are assigned, for a student/instructor ratio of 12–14, similar to the ratio in our course. Based on our experience,

if there are enough instructors and teaching assistants who know the material well to ensure that students can get help quickly during class, or via e-mail out of class, then it is likely that much larger numbers of students could be taught in this way. It may be possible to have undergraduates who previously took the course, and who were particularly effective in helping their teammates during the course, and who enjoyed the continuous progress approach, act as teaching assistants in subsequent semesters. Without this level of ongoing support, however, it is unlikely that the course would be as successful. From our experience, if the student/instructor ratio is much higher than 12 it becomes difficult to provide students sufficient individual attention and feedback to obtain the full benefits of the approach we have described.

### *Quality Control for Modeling Papers*

An unexpected consequence of teaching this course has been the discovery of the number of modeling papers that have errors, which we have found consistently in every semester. This strongly suggests that most reviewers of modeling papers cannot fully evaluate the models, because they do not have the time to replicate them. We do not generally ask experimentalists reviewing an experimental paper to replicate the same experiments before approving a paper, so this is understandable. These observations suggest, however, that it might be important to create a model repository, similar to the protein data bank, in which documented source code and executables for multiple computing platforms could be deposited, so that reviewers could at least spot check models and see that they perform as described in the papers.

## CONCLUSION

Starting a course like this can be daunting, and has risks. It also requires a great deal of work from faculty and from students. Given the success we have had, however, and the clear benefits to students, we encourage others to try it.

## ACKNOWLEDGMENTS

H.J.C. gratefully acknowledges support for development of the course from the two previous chairs of the Department of Biology, Drs. Norman Rushforth and Joseph Koonce, both of whom provided release time and regular moral support as the course was developed; and from the current Chair of Biology, Dr. Chris Cullis, who has continued to provide enthusiastic support for the course and the development of the interactive textbook. H.J.C. also gratefully acknowledges support from the former Chair of the Department of Biomedical Engineering, Dr. Patrick Crago, and from his colleagues Drs. Randall D. Beer, Michael Branicky, and Robin Snyder, who provided helpful comments and feedback on earlier versions of the textbook. The Howard Hughes Medical Institute (HHMI) has provided support to the Department of Biology through its Undergraduate Biological Sciences Initiative; HHMI supported the purchase of the laptops used in the course. H.J.C. also was supported through a Nord Grant from the University Center for Innovation in Teaching and Education (UCITE) during the initial development of units for the course. We are grateful for comments on an earlier draft of the manuscript from Drs. Robin Snyder, Peter Thomas, and Hilary Walton; Jocelyn Eckert; and two anonymous reviewers. Most of all, we gratefully acknowledge the helpful feed-

back and suggestions that we have received from students in the Dynamics of Biological Systems course; they have helped us to become better teachers.

## REFERENCES

- Ashcraft, M. H., and Moore, A. M. (2009). Mathematics anxiety and the affective drop in performance. *J. Psychoeduc. Assess.* 27, 197–205.
- Bar-Cohen, Y. (2006). *Biomimetics: Biologically Inspired Technologies*, Boca Raton, FL: Taylor & Francis Group.
- Beer, R. D. (2009). Biologically inspired robotics. *Scholarpedia* 4, 1531. [www.scholarpedia.org/article/Biologically\\_inspired\\_robotics](http://www.scholarpedia.org/article/Biologically_inspired_robotics) (accessed 17 June 2010).
- Beer, R. D., Chiel, H. J., and Drushel, R. F. (1999). Using autonomous robotics to teach science and engineering. *Commun. ACM* 42, 85–92.
- Bihl, F., *et al.* (2003). Overexpression of toll-like receptor amplifies the host response to lipopolysaccharide and provides a survival advantage in transgenic mice. *J. Immunol.* 170, 6141–6150.
- Britton, N. F., Pratt, S. C., Franks, N. R., and Seeley, T. D. (2002). Deciding on a new home: how do honey-bees agree? *Proc. R. Soc. Lond. B* 269, 1383–1388.
- Chiel, H. J. (1996). Critical thinking in a neurobiology course. *Bio-scene* 22, 3–14.
- Dahari, H., Shudo, E., Ribeiro, R. M., and Perelson, A. S. (2009). Modeling complex decay profiles of hepatitis B virus during antiviral therapy. *Hepatology* 49, 32–38.
- Dreger, R. M., and Aiken, L. R. (1957). The identification of number anxiety in a college population. *J. Educ. Psychol.* 48, 344–351.
- Feynman, R. P., Leighton, R. B., and Sands, M. (1963). *The Feynman Lectures on Physics*, Vol. I, Reading, MA: Addison-Wesley.
- Friel, D. D., and Chiel, H. J. (2008). Calcium dynamics: analyzing the  $Ca^{2+}$  regulatory network in intact cells. *Trends Neurosci.* 31, 8–19.
- Froyd, J. E., and Ohland, M. W. (2005). Integrated engineering curricula. *J. Eng. Educ.* 94, 147–164.
- Gross, L. J., Brent, R., and Hoy, R. (2004). Points of view: the interface of mathematics and biology. *Cell Biol. Educ.* 3, 85–92.
- Hille, B. (2001). *Ion Channels of Excitable Membranes*, 3rd ed., Sunderland, MA: Sinauer Associates.
- Ideker, T. (2004). Systems biology 101—what you need to know. *Nat. Biotechnol.* 22, 473–475.
- Inada, Y., and Kawachi, K. (2002). Order and flexibility in the motion of fish schools. *J. Theor. Biol.* 214, 371–387.
- Jack, J. J. B., Noble, D., and Tsien, R. W. (1975). *Electric Current Flow in Excitable Cells*. Oxford, United Kingdom: Clarendon Press.
- Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., and Barabasi, A.-L. (2000). The large-scale organization of metabolic networks. *Nature* 407, 651–654.
- Keller, E. F. (1983). *A Feeling for the Organism: The Life and Work of Barbara McClintock*, New York: Henry Holt, and W. H. Freeman.
- Kollias, S., Stafylopatis, A., Duch, W., and Oja, E. (2006). Artificial neural networks. In: 16th International Conference on Artificial Neural Networks, Proceedings, Part II, 10–14 September 2006, Athens, Greece. Berlin, Germany: Springer-Verlag.
- Kruger, J., and Dunning, D. (1999). Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments. *J. Pers. Soc. Psychol.* 77, 1121–1134.
- Menke, N. B., Cain, J. W., Reynolds, A., Chan, D. M., Segal, R. A., Witten, T. M., Bonchev, D. G., Diegelmann, R. F., and Ward, K. R. (2009). An in silico approach to the analysis of acute wound healing. *Wound Repair Regen.* 18, 105–113.
- Mitchell, M. (1996). *An Introduction to Genetic Algorithms*, Cambridge, MA: MIT Press.
- National Research Council (2003). *BIO2010: Transforming Undergraduate Education for Future Research Biologists*, Washington, DC: National Academies Press.
- Osborne, J. (2010). Arguing to learn in science: the role of collaborative, critical discourse. *Science* 328, 463–466.
- Pearson, P. D., Moje, E., and Greenleaf, C. (2010). Literacy and science: each in the service of the other. *Science* 328, 459–463.
- Rinzel, J., and Ermentrout, G. B. (1989). Analysis of neural excitability and oscillations. In: *Methods in Neuronal Modeling: From Synapses to Networks*, ed. C. Koch and I. Segev, Cambridge, MA: MIT Press, 135–169.
- Riviere, B., Epshteyn, Y., Swigon, D., and Vodovotz, Y. (2009). A simple mathematical model of signaling resulting from the binding of lipopolysaccharide with Toll-like receptor 4 demonstrates inherent preconditioning behavior. *Math. Biosci.* 217, 19–26.
- Snow, C. P. (1959). *The Two Cultures and the Scientific Revolution*, New York: Cambridge University Press.
- Tibbles, P. M., and Edelsberg, J. S. (1996). Hyperbaric-oxygen therapy. *N. Engl. J. Med.* 334, 1642–1648.
- Tobias, S. (1993). *Overcoming Math Anxiety*, New York: W. W. Norton.
- Tyson, J. J., and Novak, B. (2001). Regulation of the eukaryotic cell cycle: molecular antagonism, hysteresis, and irreversible transitions. *J. Theor. Biol.* 210, 249–263.
- Ysseldyke, J., and Bolt, D. M. (2007). Effect of technology-enhanced continuous monitoring on math achievement. *School Psychol. Rev.* 36, 453–467.