# TEACHING STATEMENT

W. GARRETT MITCHENER

## 1. My teaching philosophy

Mathematical understanding comes roughly in three levels. *Comprehension* is when a student understands all the steps and the overall picture of a proof or the solution to a problem. Significantly more difficult is *production,* the ability to use mathematical tools to discover and state such a proof or solution without guidance. At the highest level, students are able to *create* new mathematical tools and solve new kinds of problems. With that framework in mind, I use three main principles to design assignments and lectures:

- *Production is deeper than comprehension. Creation is deeper still.* At the end of a typical course, students should be able to produce at the level of the main material, and to comprehend at the level of a future course. Majors should strive for creation.
- *Students must learn to realize when they are producing nonsense.* Students who merely comprehend are overly trusting of problem solving recipes and will happily follow them to logically inconsistent conclusions. Clear writing and presentations are vital to discovering and correcting nonsense and advancing to production.
- *I refuse to protect my students from hard problems.* I aim just above their comfort zone. I challenge them to go beyond stereotypical problems to prepare them for more advanced mathematics.

In the following sections, I will illustrate how I apply these principles in developing courses, projects, and student research.

## 2. Planning courses and lectures

For any individual course, students need to reach at least the level of production. I bring aspects of production into lecture by taking on the role of a co-pilot: When I lecture students comprehend, but when they participate they produce. I write a problem and wait until someone proposes a step toward the solution, then write what they suggest and so on. During review sessions, I have students take the chalk and present problems, with the understanding that they are welcome to present a partial solution, to make mistakes, and to ask me and the rest of the class for suggestions. I intervene only when we get stuck. I create a low-pressure atmosphere where everyone is allowed to make mistakes, and we respond by learning to check for mistakes, looking out for them, and fixing them.

Each course must prepare students for the next. In 131, Duke's differential equations course for majors, I show proofs of error bounds for numerical methods and the existence and uniqueness of solutions to initial value problems. I also introduce the Hilbert space $L^2$ when we solve the heat equation with Fourier series. This material is beyond the mastery of differential equations required for 131. However, mathematics majors will eventually need

*Date*: October 28, 2005.

to learn it. My goal for 131 is for them to comprehend this material so that they will be better prepared to produce proofs when they take analysis.

## 3. How I design projects

Duke has been using a book by Edwards and Penney for 131. The book contains a project about plotting solutions to $y' = \sin(y - x)$. Deriving the explicit symbolic solution requires an inverse tangent but the principal arctangent only gives parts of some solutions. To get complete solutions for all initial values we must combine several branches of arctangent by adding appropriate multiples of $\pi$. The book avoids these complications, resulting in a bland project. I re-wrote the lab and spiced it up by asking harder questions: Does the existence and uniqueness theorem for solutions to initial value problems apply? Why does the graph of the symbolic solution look so different from the numerical solution? I ask them to find the symbolic solutions for two initial value problems and graph them, and they discover that the unknown constant is the same for both. They are unable to graph them until they figure out that something is going wrong with the arctangent. Unsurprisingly, many students ask for help at this point. My goal is not so much for them to figure everything out on their own as for them to identify the complication and avoid producing nonsense such as "There is no solution even though the theorem applies." I engineered this combination of questions to illustrate how recipes can lead to nonsense and that they must thoroughly understand and synthesize multiple problem solving techniques to produce a complete correct answer.

A second example is a project I designed for 196S, the seminar on mathematical modeling in Spring 2005. It is a guided computer project based on a paper by linguist Anthony Kroch: We are given several time series of usage counts of alternative ways of constructing certain sentences in Middle English. Over decades, an old form gives way to the modern construction. In the first part of the project, I ask students to fit a sigmoid to the time series, using the traditional method of performing a least squares fit on linearized data. Linearization fails on some points where they would need to take $\log 0$ and they must find a way to work around the difficulty. In the second part, I ask them to fit a sigmoid directly using non-linear least squares minimization. I ask what happens if the data were to change slightly, as might happen if new manuscripts were discovered. This question introduces sensitivity analysis: Students must evaluate their answers and determine how trustworthy they are. This is part of how I train them to catch when they produce nonsense. Finally, I ask them to reproduce Kroch's maximum likelihood fit, which gives the most robust results and illustrates how an approach tailored to a problem gives better results than generic techniques.

Later projects require them to create and evaluate their own models. The final projects were impressive in their creativity and quality of writing and presentation. Two groups applied learning algorithms (neural networks and support vector machines) to phonetic data. One group used statistical methods to identify languages from written text. One group looked for patterns in the Voynich manuscript, and two modeled dialect formation. To help them write, I presented guidelines for what a good modeling paper looks like, including an anonymous packet of the best and worst writing from their mid-term projects. Groups were required to exchange and proofread each other's drafts. I made extensive comments on mid-term projects, pointing out inconsistency, vagueness, and points that needed clarification.

## 4. Student research

Mathematical creativity is best built through projects and research. I have supervised two undergraduate research projects. The first was a summer project with Adam Chandler, who was interested in language models. I suggested that we build mathematical models based on a linguistics paper about the spread of a sound change in the speech of Pennsylvanians. We developed a simulation in Mathematica and studied how such learning processes naturally lead to traveling wave solutions of reaction-diffusion equations.

The second project grew out of the mathematical contest in modeling (MCM) run by COMAP. The contest requires teams of three students to spend a weekend writing a paper about an open-ended real-world problem. In February 2005, I coordinated Duke's teams, and the team of Adam Chandler, Pradeep Baliga, and Matthew Mian earned an Outstanding rating, the highest possible, for their paper on the optimum size for a toll plaza. They gave an excellent presentation on their solution at Mathfest, August 2005. Under my supervision, Chandler and Baliga are continuing their MCM project on models of toll roads using queuing theory and cellular automata. They will write papers on their work as part of the requirements for graduation with distinction in mathematics, and we hope to present their results to the North Carolina Department of Transportation. I help them search the literature, decide which potential lines of thought are most likely to be fruitful, and write. Chandler and Baliga are quite independent and work well as a team. It is a pleasure to see their creativity at work.

## 5. Conclusion

Here are a few facts that I use to gage my success as a teacher. My 131 students are generally able to answer even the hardest questions (proofs and dynamical systems) on projects, tests, and exams. They willingly go beyond the scope of the course: One student came to my office repeatedly to hash out a theorem about Wronskians, and during one lecture someone raised a subtle point about an improper integral. These students deserve the credit for their performance, but I like to think that my classes provide them opportunities to flourish. Students enjoy me as a teacher: When I taught 196S, which is an elective, eight of the twelve students taking it for credit had taken 131 with me in a previous semester. One student said it was the perfect class for his interests and was very disappointed when a scheduling conflict forced him to drop. Many times hopeful students ask what I am teaching next term and when I will run 196S again. Even the shyest students are comfortable asking questions and talking to me.

In conclusion, I design courses to encourage students to synthesize problem solving techniques. I teach them to produce and express clear, complete, and correct results, and to exercise creativity. I introduce them to more difficult concepts to prepare them for future courses. Students like my teaching and seem to thrive under it.