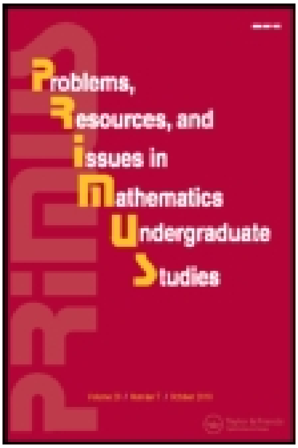

PRIMUS: Problems, Resources, and Issues in Mathematics Undergraduate Studies

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/upri20

Experimental Mathematics for the First Year Student

David Brown

Accepted author version posted online: 16 Dec 2013.Published online: 28 Mar 2014.

To cite this article: David Brown (2014) Experimental Mathematics for the First Year Student, PRIMUS: Problems, Resources, and Issues in Mathematics Undergraduate Studies, 24:4, 281-293, DOI: 10.1080/10511970.2013.870264

To link to this article: http://dx.doi.org/10.1080/10511970.2013.870264

PRIMUS, 24(4): 281–293, 2014

ISSN: 1051-1970 print / 1935-4053 online
DOI: 10.1080/10511970.2013.870264

Taylor & Francis
Taylor & Francis Group

Experimental Mathematics for the First Year Student

David Brown

Abstract: We describe a course that highlights mathematical experimentation as an introductory course for first year mathematics majors. We discuss the benefits of an experimental approach as an alternate pathway into the mathematics major. We also explain how this course serves as a gentle lead-in to later research experiences.

Keywords: Experimental mathematics, *Mathematica*, Pythagorean triples, technology.

1. INTRODUCTION

When students walk into the Mathematical Experimentation course offered by the Department of Mathematics at Ithaca College, they enter a world of mathematical learning that very few of them have ever experienced. There are no lectures, mathematical process takes precedent over structured content, and there is an expectation that students will follow their own paths to the discovery of mathematical concepts using ideas generated via technology. Students are taking their first steps into the world of experimental mathematics.

Borwein and Bailey [4, p. 2] describe experimental mathematics as a mathematical methodology that includes the use of computations for:

1. gaining insight and intuition;
2. discovering new patterns and relationships;
3. using graphical displays to suggest underlying mathematical principles;
4. testing and falsifying conjectures;
5. exploring a possible result to see if it is worthy of formal proof;
6. suggesting approaches for formal proof;
7. replacing lengthy hand derivations with computer-based derivations;
8. confirming analytically derived results.

Address correspondence to David Brown, Department of Mathematics, Ithaca College, Ithaca, NY 14850, USA. E-mail: dabrown@ithaca.edu

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/upri

Examples of experimental mathematics in action are easy to illustrate, even if the mathematics behind the examples becomes deep and challenging. One such example, which often becomes a research project for students in the Mathematical Experimentation course, involves exploring the bifurcation structure for the logistic difference equation

$$x_{n+1} = rx_n(1 - x_n), \quad 0 \le r \le 4, \quad 0 \le x_n \le 1.$$

Simple analysis of fixed points and two-cycles in this family leads to algebraic equations that allow us to determine that attracting fixed points exist when $1 < r < 3$ and attracting two-cycles exist when $3 < r < 1 + \sqrt{6}$ [8]. Understanding these bifurcation values, $\mathrm{Bif}_1 = 3$ and $\mathrm{Bif}_2 = 1 + \sqrt{6}$, leads to the image shown in Figure 1, created using repeated iteration.

The interesting work occurs when trying to determine higher bifurcation values. Bif_3 is the value of r for which stable four-cycles bifurcate into eight-cycles. This bifurcation value can be determined by solving the following degree-12 polynomial with integer coefficients:

$$\begin{aligned} r^{12} - 12r^{11} + 48r^{10} - \ & 40r^9 - 193r^8 + 392r^7 + 44r^6 \\ & + 8r^5 - 977r^4 - 604r^3 + 2108r^2 + 4913 = 0. \end{aligned}$$

Details about the derivation of this polynomial are found in [3]. Numerical solvers in Mathematica are more than capable of giving the solution (closest to 3.5) to whatever accuracy we wish. For instance, the Mathematica code

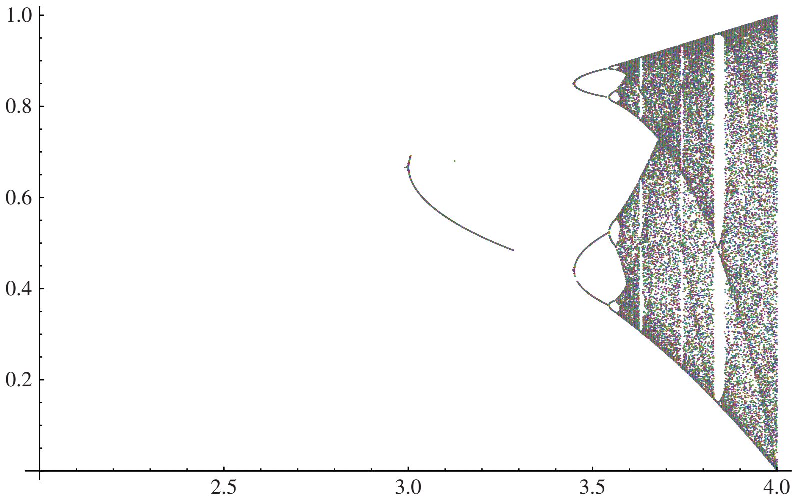

Figure 1. Bifurcation diagram for the logistic difference equation.

```
FindRoot [r^{12} - 12 r^{11}+48 r^{10} - 40 r^9 -
    193 r^8+392 r^7+44 r^6+8 r^5 - 977 r^4 -
    604 r^3+2108 r^2+4913 == 0, {r, 3.5},
    WorkingPrecision -> 40]
```

yields the desired result that

$$\mathrm{Bif}_3 \approx 3.544090359551922853615965986604804540583.$$

This sort of computation was challenging prior to easy access to computational software. Now, students can explore mathematically rich problems and make progress with these computational tools. This sort of simple example, which is fairly easy to motivate and understand, drives the philosophy of the Mathematical Experimentation course. All students, when given the proper introduction to experimental mathematics, can be taught how to discover mathematical ideas. And, they can learn to support those ideas even in the early stages of their mathematical training.

Of course, in an introductory course, it would be difficult to accomplish all of the aspects of experimental mathematics as defined above. Mathematical Experimentation was designed to focus on a specific set of goals:

1. Explore mathematical phenomena experimentally.
2. Detect patterns and provide mathematical explanations.
3. Explore mathematical thinking and process of conjecture.
4. Design and implement mathematical algorithms with computer algebra systems.

These particular goals were chosen to lay the foundation for the remaining aspects of experimental mathematics and to provide an introduction to the mathematics major that emphasizes deep critical thinking and exploration. We also have in mind that almost all of our students eventually complete a research project in mathematics (usually in the junior year) and this course provides many of the basic research skills that we expect students to employ. Thus, in many ways the course was modeled on our department's successful research course sequence [6].

Mathematical Experimentation requires only a basic understanding of high school mathematics, including algebra and geometry. Students majoring in mathematics enroll in this course concurrent with a calculus course, and thereby experience a content-driven mathematics course along with the process emphasis in the experimentation course. The course also attracts non-majors who have a general interest in mathematics, but are looking for an alternative to calculus.

2. COURSE STRUCTURE

The majority of the Mathematical Experimentation course centers on students working in groups on weekly laboratories in order to explore and discover mathematical ideas. The design of the course around laboratories was influenced by the approach taken by math faculty at Mt Holyoke College [7]. It is in the laboratory environment that students really dig in and engage with mathematics from the experimental point of view.

The class meets 3 hours per week, in a computer lab, during a 15 week semester. Students are first exposed to the software that is integral to their work throughout the course: *Mathematica* and LaTeX. *Mathematica* is the workhorse of investigations and LaTeX is the tool for communicating ideas. LaTeX was chosen for writing reports in order to expose students to standard typesetting program in mathematics, highlighting the professional aspects of mathematics.

The typical semester schedule is shown in Table 1.

3. WEEKLY LABS

Students spend 11 of the 15 weeks investigating labs designed to engage in the experimentation process. A fairly extensive library of labs has been created over the past 6 years, including labs on linear iteration (on the real line and in the plane), Pythagorean triples, fractals, and distribution of prime numbers.

Labs provide some direction of exploration for the students, but leave it to the students to decide what ideas they want to investigate in greater depth. In this way, the labs become somewhat open-ended. This means that the instructor spends a fair amount of time talking with individual

Table 1. Typical semester schedule for Mathematical Experimentation

Week	Topic
1	Guided introduction to *Mathematica* and LaTeX
2	Work on Lab 1
3	Edit Lab 1
	Continue deeper work with Lab 1
	Complete quiz 1
4–11	Repeat the process of weeks 2–3
	Work on Labs 2–6
	Complete biweekly quizzes
12	Develop final project outline
13, 14	Work on final projects
15	Present final project
	Submit final project report

groups about their ideas, interacting with the students rather than prescribing mathematical content. Class time becomes a time for student groups to explore, discuss ideas with each other, and make sense of their work to the instructor.

Much of the class time is devoted to helping students design and implement programming to model the mathematics they're exploring. Instructors show students the parallels between debugging and coding and careful reasoning in mathematics, and students come to realize that computers are far less forgiving of fuzzy thinking than even a faculty member. As the semester progresses, students become much better at the logic of programming and effective modeling of mathematics.

The following is an example of an introductory lab that has been given the past few years. It draws on a familiar result, the Pythagorean theorem, pushing students to see how deeper mathematics is related to this simple idea. Since it is a familiar topic, students are easily convinced that they can "jump in" and explore this topic.

3.1. Example - Pythagorean Triples Lab

The instructor displays the following image in Figure 2 and asks the class to discuss anything they know about this triangle.

After a few moments of peer-to-peer discussion, the instructor solicits input and the Pythagorean Theorem is quickly offered up. Of course, we rarely get a precise statement of the theorem, and so, we spend a minute or two finding counterexamples to imprecise statements such as "$a^2 + b^2 = c^2$." We discuss why precision in statements is important mathematically and give a correct statement of the Pythagorean Theorem. After a very brief discussion of Pythagorean triples (without explicitly calling them "Pythagorean triples" to avoid an immediate Google search) students are set loose in groups to investigate much more deeply. The opening section of the lab reads:

"In this lab, we explore a topic that should be very familiar to you. The Pythagorean Theorem is probably the most widely recognized theorem (what

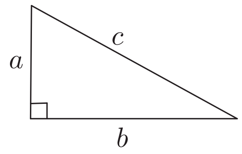

Figure 2. Picture to motivate discussion.

is a *theorem*?) Recall that the Pythagorean Theorem tells us that the legs (a and b) and hypotenuse (c) of a right triangle are related by the equation

$$a^2 + b^2 = c^2.$$

Given a and b, this equation always has a solution when working with positive *real* numbers, since every positive real number has a square root that is a real number. The story is different if the side lengths are required to be *whole* numbers; that is, positive integers.

You are probably aware of the famous triples: $(3, 4, 5)$ and $(5, 12, 13)$ since

$$3^2 + 4^2 = 5^2 \quad \text{and} \quad 5^2 + 12^2 = 13^2.$$

An obvious question is, *are there more such triples? Are there infinitely many?* These and many questions can be asked and explored. In this lab, we provide several such questions, but we look forward to your asking some questions to explore yourself."[5]

The lab continues with a brief example of using *Mathematica* to investigate triples.

"Enter the following into *Mathematica*, press ENTER, and see what happens.

```
a=3; b=4;
If [Element[Sqrt[a^2+b^2], Integers],
 Print[a, " ", b, " ", Sqrt[a^2+b^2]]]
```

Now, change the a to 2 and change b to 6 and re-run the code. What happens this time? Why? In your lab write-up, explain what this code is doing.

If you want to generate large lists of triples, the following program does so. You could write a different program, and you are encouraged to think about how you might do so. But, for now, play around with this code that you should enter into *Mathematica*.

```
Do[If[Element[Sqrt[a^2+b^2], Integers],
  Print[a, " ", b, " ", Sqrt[a^2+b^2]]],
  {a, 1, 100}, {b, 1, 100}]
```

In your lab write-up, explain what this code is doing. Be sure to understand what each command is doing and how they fit together to produce what we are interested in studying.

To further play, figure out what the following code does.

```
Do[c = Sqrt[a^2 + b^2];
 If[c \[Element] Integers && GCD[a, b, c] == 1 &&
   Ordering[{a, b, c}] == {1, 2, 3} && OddQ[a],
  Print[a, " ", b, " ", c]], {a, 1, 100}, {b, 1,
  100}]"
```

This code checks which values of a, b, and c satisfy the Pythagorean Theorem. In particular, for each choice of a and b between 1 and 100, the code checks if $c = \sqrt{a^2 + b^2}$ is an integer. If so, the code displays the triple (a, b, c); otherwise, the code moves on to the next values of a and b. The GCD command guarantees that a, b, and c share no common factors, eliminating multiples of primitive Pythagorean triples. The OddQ command forces the code to only consider those triples in which a is odd, as this leads to patterns that students can discover.

During the early weeks of the course, more detailed code is provided to the students as they learn the software. As the semester continues, students are expected to create more and more of the code on their own.

As this is the first lab of the semester, several leading questions are posed in order to give students directions to explore with *Mathematica*, including:

- If you write down a basic triple in increasing order, such as (3, 4, 5) or (8, 15, 17), then the lowest number can be odd or even. Write down all of the basic triples (in increasing order) for which an odd number is the lowest number. Do you notice a pattern that will allow you to write a formula for all such triples? Explore and see what find. Think about relationships between a, b, and c. Anything that you discover is a valid avenue to explore.
- Write down all basic triples that have multiples of 4 in the triple, such as (4, 3, 5), (8, 15, 17) and (12, 35, 37). If you continue this list, can you find a way to express all such basic triples? [5]

Students spend substantial time on these questions, generating output in *Mathematica*. We want students to wrestle with their data and try to understand what is and is not valuable for their work, such as what patterns they see and which of the output supports that pattern. We emphasize that, while all of the output may not support a conclusion, there may be families (even infinite ones) of output that do support their claims. For example, in the final bullet above, students eventually realize that triples with shortest side a multiple of four yield triples of the form

$$(a, b, c) = (4k, (2k)^2 - 1, (2k)^2 + 1). \tag{1}$$

They may not express this relationship so succinctly at first, but we help them to refine their work (through the editing and rewriting process) to reach the point where most see that they can justify their pattern (1) by showing it satisfies the Pythagorean Theorem:

$$(4k)^2 + ((2k)^2 - 1)^2 = 16k^4 + 8k^2 + 1 = ((2k)^2 + 1)^2.$$

Of course, this is not the most general expression of Pythagorean triples, but it is one that students can easily unveil and justify. Similarly, many students discover the triples pattern

$$(a, b, c) = (2k + 1, 2k(k + 1), 2k(k + 1) + 1) \quad (2)$$

by observing that for many triples (a, b, c) with a odd that $c = b + 1$. The Pythagorean Theorem then quickly gives the relation (2). The fact that many more triples are not accounted for by these two families pushes students to look deeper, thereby fueling the tenacity we hope to foster as students continue their mathematical studies.

3.2. Writing and Editing

We reinforce the main goal of introducing students to experimental mathematics by requiring that results from labs are written in a format consistent with good writing in mathematics. Each lab report goes through the writing and editing process. Although time-consuming, writing and editing helps students understand the importance of careful reasoning and precision when explaining mathematics. Students receive a lot of feedback from the instructor on both the content and exposition. This feedback provides students with the necessary directions for improving their experimentation and writing.

A preliminary report is submitted after 1 week of exploration and a final draft is submitted after receiving feedback on the preliminary draft and a second week of exploration. Students learn to focus on clear exposition and carefully crafted mathematical ideas. Supporting evidence such as patterns revealed by experimentation and/or deductive reasoning must be provided. After about half a semester of writing lab reports, we note that the exposition, and more importantly, their ability to engage in experimental mathematics, improves.

3.3. Other Labs...and How to Obtain Them

Many other labs have been written and used since the first offering of this course in the Spring semester of 2007. Some of the labs used more frequently are the following:

- *Discrete Dynamical Systems*. This lab focuses on the recurrence relation $x_{n+1} = ax_n + b$, studying the effects of a and b on the convergence of the sequence of numbers created. This lab is heavily inspired by the work done by the Mount Holyoke group.
- *Prime Numbers*. This lab studies the infinitude of primes and twin primes, investigates the Prime Number Theorem, and develops formulas for producing prime numbers.
- *Wheels on Wheels*. This lab studies cycloids, trochoids, and other parametric plots as they relate to number theory. In addition, students apply their work to understanding engineering of mechanical engines.
- *Euclidean Algorithm*. This lab introduces students to the algorithm via the programming of loops. Students study the efficiency of the algorithm and its connection with the Fibonacci numbers.
- *Linear Iteration in the Plane*. This lab focuses on the iteration of 2×2-matrices. Students also compare their work on this lab with that of the Discrete Dynamical Systems lab completed earlier in the semester.
- *Iterated Function Systems*. This lab asks students to develop understanding of iterated families of contractive mappings and the resulting topology of attractors. Students develop the basic mathematics involved in the creation of fractals.

These labs, as well as all items mentioned in this article, are available from the author; please send an email to the author for the latest version of all materials.

4. WEEKLY QUIZZES

Throughout the course, we emphasize the use of programming in order to generate mathematical ideas. In early offerings of the course, we found that students need to be pushed to understand the basics of simple coding that are needed to succeed in experimentation. By the third offering of the course, a biweekly quiz was introduced in order to focus attention directly on the programming portion and as a way to foreshadow future programming needs. Thus, we struck a balance between technical skill in programming and the process of mathematical experimentation.

For example, Figure 3 ties previous work on sums of squares of integers to the idea of counting how often something (such as numbers expressible as the sum of two squares) occurs. Future labs will use the idea of a counter, so we introduce the concept in a quiz that focuses on counting and nothing else. Students adapt previous code in order to understand how this new structure can be integrated. The introduction of quizzes focusing on specific programming structures has improved student ability to generate effective codes in their exploration of mathematical ideas on labs.

- **Sometimes, we are interested in how many times something occurs; we use a counter to do so. For instance, we may want to know how many numbers between 1 and 100 are divisible by 3. Notice how we use the variable "count" to keep track of how many times divisibility by 3 occurs.**

```
count = 0;
Do[
  If[Divisible[k, 3], {Print[k], count++}],
  {k, 1, 100}];
Print["There are ", count, " integers below 100 that are divisible by 3."]
```

- **Question 1 : (10 points) Explain how the code above is constructed to achieve the counting of the numbers that are divisible by 3.**
- **Question 2: (10 points) Write *Mathematica* code that counts how many integers between 1 and 100 are the sum of two squares.**

Figure 3. A quiz that introduces the idea of a counter.

The quizzes provide enough of a challenge to push students to see the value of developing good programming habits, without becoming a time burden. The quizzes are short and very focused, introducing only one or two new programming tools. Inevitably, students make very complicated programming requests in the pursuit of experimenting mathematically. The quizzes allow the students to grow in technical skill to meet their own programming needs.

5. FINAL PROJECTS

The course culminates with students developing their own projects to define, explore, and analyze. This culminating experience serves to evaluate how successful the students are in asking meaningful questions to explore in relation to a mathematical topic. The labs are designed to help students learn the process of experimentation in order to understand mathematical phenomena; the final project asks students to develop (as provided in the labs) questions of interest which serve as the basis for experimentation.

Students spend 1 week developing their topics and formulating the questions they will explore. This process is done in close consultation with the instructor to ensure that they are asking good questions that can be reasonably addressed in the final 3 weeks. A draft of their work, including a report and their computer programs, is due a week before the final due date. The final report is due during the final exam period when students give a presentation (oral or poster, depending on the instructor) highlighting their work. Students have been very excited to show off their projects and the presentations are a great way to end the course.

Students have developed many different final projects, some building on labs from earlier in the courses and others completely new. Some project examples are *Kaprekar-type Constants*, *Bifurcation Diagrams for Non-linear Functions*, *Periods of Base-b Expansions of Reciprocals of Prime Numbers*, and *Alternate Pascal Triangles and Related Identities*. These mini-research projects have, at times, become the basis for extensive research projects in the student's junior year research course.

6. CHALLENGES AND SUCCESSES

The major challenge in teaching a course in experimentation is students' discomfort with the experimental mathematics approach. Many students, including those who claim to be "good at math," find it disconcerting to not have a specific problem to solve. The thought that, through exploration, they may or may not discover mathematical ideas and results causes frustration, especially during the initial weeks of the semester. Patient guidance from the instructor is key in mitigating this frustration and almost every student eventually gains comfort with the uncertainty in trying to discover mathematical ideas and support those ideas through experimentation and reasoning.

Another challenge is resistance to writing in mathematics, which is not unique to this course. Mathematics majors are accustomed to writing solutions to problems, but asking them to write about their experimentation process and reasoning is often new to them. A major complaint is the time needed to write lab reports; most students report that labs require about 3 to 4 hours of writing outside of class during the week. This is certainly not excessive for a three-credit course.

Students using the internet to look up results in labs can be an issue at times. This is usually easy to spot because the students will write something in their report that is not supported by experimentation. This provides an opportunity to discuss with the students the importance of the goal of learning to experiment rather than "getting the right answer to a specific question." In our experience, this conversation is sufficient to curtail internet usage.

A question that faculty often ask is whether labs are used year after year and, if so, does lead to student sharing reports. We do use some labs year after year, but we have not experienced students sharing information from one year to the next. In fact, most student discussion about the course centers on the process and the shared experience of hours in the lab experimenting and writing. The author has occasionally dealt with students sharing lab reports within the same class, leading to discussions about plagiarism. These conversations (and resulting grade penalties) stops this sharing.

However, these challenges pay off as students progress to upper-level courses and move on to work on research projects. Students continue to use technology as they work on problems in other courses and many continue to write using LaTeX. The greatest success is observed when students enter the required research courses in the junior year of the major. The research projects given often require computer experimentation to develop understanding of the project and to create results. Students come into the course having been exposed to this approach and move quickly to the research process without the need for instruction. The instructors view the Mathematical Experimentation course as part of the continuum of engaging with research in mathematics, helping students see that mathematics is a vibrant subject, one in which they can make contributions.

7. CONCLUSION

When David Bailey and Jonathan Borwein [1] published about experimental mathematics in the *Notices of the AMS* in 2005, they sparked debate about the role of computers in mathematical research. The ideas presented in that paper were much less controversial by the time they published another article in the *Notices* 6 years later [2]. By 2011, the use of computation in the discovery and verification of mathematical results was more fully embraced by the research community.

Bailey and Borwein [2] make the case for understanding “when a computation is or can - in principle or practice- be made into a rigorous proof and when it is compelling evidence or entirely misleading.” They conclude that developing this understanding requires “curriculum that carefully teaches experimental computer-assisted mathematics.” The course described in this paper is an example of just such an effort.

Finally, approaches to teaching experimental mathematics have been tried by others. We have mentioned the work of those at Mount Holyoke, and Bailey and Borwein [2] mention several people engaging in teaching experimental mathematics. These courses teach our students how to use computers in doing mathematics and allow them to engage in the discovery process in a way that previous generations could not. Mathematical Experimentation provides the carefully structured introduction to the methodology of experimental mathematics called for by Bailey and Borwein.

ACKNOWLEDGEMENTS

The author thanks the editors and anonymous referees for the thoughtful suggestions that helped improve the content and exposition of this paper.

REFERENCES

1. Bailey, D. and J. Borwein. 2005. Experimental mathematics: examples, methods, and implications. *Notices of the AMS*. 52(5): 502–514.
2. Bailey, D. and J. Borwein. 2011. Exploratory experimentation and computation. *Notices of the AMS*. 54(10): 1410–1419.

3. Bailey, D., J. Borwein, V. Kapoor, and E. Weisstein. 2006. Ten problems in experimental mathematics. *American Mathematical Monthly*. 113(6): 481–509.
4. Borwein, J. and D. Bailey. 2008. *Mathematics by Experiment: Plausible Reasoning in the 21st Century*, *second edition*, Natick, MA: A.K. Peters.
5. Brown, D. 2013. Mathematical experimentation labs. http://faculty.ithaca.edu/dabrown/mathexp. Last accessed 31 January 2014.
6. Brown, D. and O. Yürekli. 2007. Undergraduate research in mathematics as a curricular option. *International Journal of Mathematics Education in Science and Technology*. 38(5): 571–580.
7. Cobb, G., G. Davidoff, A. Durfee, J. Gifford, D. O'Shea, M. Peterson, H. Pollatsek, M. Robinson, L. Senechal, R. Weaver, and J. W. Bruce. 1997. *Laboratories in Mathematical Experimentation: A Bridge to Higher Mathematics*, California: Key College Publishing.
8. Strogatz, S. 1994. *Nonlinear Dynamics and Chaos*, California: Perseus Books Publishing.

BIOGRAPHICAL SKETCH

David Brown is an Associate Professor of Mathematics at Ithaca College in Central New York. He received his Ph.D. from Cornell University and BA from Ithaca College. He is interested in research on fractals and iterated function systems and has published several articles with undergraduates on these topics. He is also interested in cryptology and teaches a humanities course in the history of Cold War espionage. He is heavily involved in K-12 mathematics education, giving professional development workshops for teachers and preparing pre-service teachers to teach in high-needs school districts.