Complexity Theory and the Mathematics Lab-Classroom

MARGARET SINCLAIR
York University

The main thesis of a recent article by Davis and Simmt (2003) is that "mathematics classes are adaptive and self-organizing complex systems". This thesis is intriguing. It helps explain a phenomenon teachers often witness—the spontaneous emergence of a special learning community in a particular class. And it raises the question—if this thesis applies to all mathematics classes how do the additional elements of a computer lab environment affect the evolution of the "learning system"? In this article three technology experiences have been analyzed in light of the thesis, to determine how software, organization, and task impact the blossoming of a complex learning system in the lab-classroom.

Complicity: An International Journal of Complexity and Education
Volume 1, Number 1 • pp. 57–71 • www.complexityandeducation.ca

Introduction

Complexity theory arose from studies of mathematical processes and biological systems, but it has been applied in a variety of contexts. Recently, researchers have begun to discuss its potential in the social sciences (Cronbach, 1988), in education (Doll, 1989) and more specifically in mathematics education (cf. Davis, 2003; Davis & Simmt, 2003; Lesh, 2000).

Unlike mechanistic theories, which assume a centrally controlled governing structure, complexity theory rests on the idea that order emerges through the interactions of organisms or agents. Analysis has shown that systems as diverse as ant colonies, cities, and the stock market provide examples of such "bottom up" development (Johnson, 2001; Kelly, 1994). Although the interactions between agents follow simple rules, complex systems are capable of innovation if they satisfy five necessary but insufficient conditions: internal diversity, redundancy, decentralized control, organized randomness, and neighbour interactions (Davis & Simmt, 2003).

Internal diversity refers to the idea that parts or members of a system have different capabilities. When there is a high level of diversity, there are more opportunities for a system to develop new and creative responses to situations. In the classroom, internal diversity is linked to a range of possible innovations and thus to the range of experiences and skills of the agents (Davis & Simmt, 2003, p. 148).

Redundancy is a characteristic of biological systems. Nature produces many more organisms than necessary to ensure that enough survive. Davis and Simmt suggest that shared vocabularies and experiences are examples of redundancy in the classroom.

Control in complex systems is decentralized. There is no "king" who sends out commands to every working part; instead, the response of the system depends on the interactions of individual agents. The counterpart in the classroom is that learning emerges from shared mathematical insights (Davis & Simmt, 2003, p. 152). In other words, the focus of learning is neither the teacher, nor the individual.

Complex systems grow and develop within boundaries, but by random processes—that is, they display organized randomness. In education settings one might think about organized randomness by focusing on the idea of constraints. Within the boundaries of particular tasks students can respond and react with varying degrees of freedom.

Neighbour interactions, in a biological sense, concern the impact of one organism on another and the effect of such interactions on development and behaviour. In the classroom, neighbour interactions could be interpreted as peer-peer interactions, but another option is to consider ideas that interact or "bump against one another"(Davis & Simmt, 2003, p. 156).

This paper aims to offer insight into how the use of lab-based technology affects or is affected by these five conditions and to consider the question: Can we—or how can we—nurture the development of a "learning system" in a technology-supported environment? The three situations that provide the data for the reflections will be referred to as: Linear Transformations, JavaSketchpad Research, and Independent Study Project. A brief overview of each is provided.

Linear Transformations

Over a period of several years, students in my OAC Algebra and Geometry course were assigned an in-class project on Linear Transformations.[1] The observations in this paper relate to work with the initial group. The assignment required students to investigate the effects of a variety of linear transformations with the spreadsheet As-Easy-As, 5.5.[2] The cell addresses were used to define the product of a transformation matrix with a matrix of values that produced a unit square. Students recorded their observations in a table and conjectured relationships between characteristics of the transformation matrix, and the shape, orientation and area of the image. (e.g., The orientation of the original unit square was counter-clockwise. If the determinant of the transformation matrix was negative, the orientation of the image square was clockwise.) The reader may access the entire assignment at http://www.yorku.ca/sinclair/. The online file includes set-up instructions, assignment questions, matrices, and tabulation sheet.

In the computer lab students worked with a partner of their choice while I circulated through the class, assisting with technical and mathematical problems. A discussion of results was held in the regular classroom on a subsequent day.

JavaSketchpad Research

The second technology application was my thesis research, which looked at the benefits and limitations of using *JavaSketches*[3] in senior mathematics. Students in three grade 12 classes at two different schools participated in the study. Although the students had done introductory work on deductive geometry, none had worked with dynamic geometry software.

JavaSketchpad (Jackiw, 1998), was used to prepare four web-based, dynamic geometry sketches for student pairs to explore during the three (or four) lab sessions, two extra sketches for those who finished early, and one sketch for a group discussion. The labsheet that accompanied each sketch provided directions for opening and manipulating the sketch, a statement of the problem, and questions related to the task. Problems chosen as the basis for the web-based sketches were similar in difficulty to those in the student text, *Mathematics: Principles and Process, Book 2* (Ebos, Tuck, & Schofield, 1986) and related to congruence and parallelism. The reader may view the sketches at: http://www.yorku.ca/sinclair/JavaSketchpad/index.htm. Analysis of some results is available in the following articles: *The provision of accurate images with dynamic geometry*, (Sinclair, 2003a) and *Some implications of the results of a case study for the design of pre-constructed, dynamic geometry sketches and accompanying material*s (Sinclair, 2003b).

Independent Study Project

The Independent Study Project was carried out with four successive OAC classes.[4] Students were required to investigate a topic using *The Geometer's Sketchpad* (Jackiw, 1991), the As-Easy-As spreadsheet ("As-Easy-As, version 5.5," 1993), or MapleV, ("MapleV, release 2," 1994). The work was completed after school and at lunch in the school's computer lab over a period of several weeks. I supervised and provided a number of resources. The requirements for the project were as follows:

> During 4 timeslots in the lab, carry out one task from the following list, on the computer, with a partner. On the last two days of class be prepared to present a three to five minute demonstration of your findings using the computer and the overhead display. In the demonstration explain briefly what your project is about and show how the computer program helped in investigating. Hand in an explanation of your work (no more than 3 pages) and a disk with the animation/graph/sketch.
>
> *TASK LIST*
>
> - Use *The Geometer's Sketchpad* to create tessellations.
> - Use the Scripting tool in *The Geometer's Sketchpad* to create the Koch snowflake and other fractals.
> - Create ellipses and hyperbolas through animation using *The Geometer's Sketchpad*.
> - Create and investigate the Sierpinski triangle using *The Geometer's Sketchpad* or a graphics calculator.
> - Construct sine and cosine wave tracers using *The Geometer's Sketchpad.*
> - Examine three proofs of the Pythagorean theorem using *The Geometer's Sketchpad.*
> - Investigate recursive relationships using a graphics calculator.
> - Investigate the GLaD construction[5] using *The Geometer's Sketchpad.*
> - Create and investigate polar graphs with MapleV.
> - Investigate graphs of planes, cylinders, and/or spheres using MapleV.
> - Investigate linear and affine transformations using As-Easy-As.
> - Create and investigate polar graphs on a spreadsheet.
> - Use the CBL [Calculator-Based Laboratory] to collect data from a light source and determine the equation of the resulting curve.
> - Use one of the programs to explore famous curves: for example, the cycloid or the witch of Agnesi.
> - Investigate the relationship between Euler's line and the 9-point circle using *The Geometer's Sketchpad.*

Observations

The following observations, clustered according to the five necessary but insufficient conditions for the emergence of complexity, represent a re-examination of data from a new perspective. Since the original projects were not designed to provide information about characteristics of complex systems, the reflections are somewhat fragmented; nevertheless, I believe they offer some important insights into the technology-supported classroom experience.

Internal Diversity

As noted earlier, a high level of diversity helps a system develop new and creative responses to situations; but what does this mean in a technology-supported classroom?

The students involved in the Linear Transformations project had completed the same mathematics courses, although mathematics achievement levels ranged from poor to excellent. Thus, there was some level of diversity with regard to mathematical skills. On the other hand, the level of diversity was high with respect to technological skills. Some students had never touched a computer, while others had a moderate to high level of experience for the time. No students in the initial group had used a computer for mathematics; however, a few had used a spreadsheet to enter numerical data.

The project involved the spreadsheet program As-Easy-As. Students were expected to follow the instructions for entering information line by line, and then to calculate various results and record these on a labsheet. At the beginning most dutifully complied. Soon, however, a few students who had used spreadsheets before began to experiment. Although these students had limited experience, they were aware that spreadsheets could be made to carry out arithmetic procedures. They soon constructed spreadsheet formulas to do the required calculations, and shared their technique with others in the class. Students had seized on the capabilities of the program and taught themselves new procedures. In doing so, they had altered the assignment itself—for the better. It no longer provided extra practice at calculating dot products and determinants, but focused more effectively on developing an understanding of linear transformations.

The circumstances surrounding the JavaSketchpad research were similar in some ways. Student achievement levels in mathematics ranged from poor to excellent; there was a wide range of technological expertise; only one program was used (a browser to view the JavaSketchpad applets); and students were expected to work in pairs. However, unlike As-Easy-As, which could support a range of approaches, JavaSketchpad itself constrained student options. The sketches could be investigated in various ways—but only via routes pre-planned by the sketch designer. Students could not devise new methods as they had with the spreadsheet, nor could they spontaneously move in a new direction as they might have with a pre-constructed *Geometer's Sketchpad* sketch.

The design of the Independent Study Project was substantially different. Each of the programs—Sketchpad, As-Easy-As, and MapleV, could be used for mathematical applications at a range of levels. In addition, the atmosphere was casual. Pairs used the lab after school or at lunch, worked on their own topic, borrowed resources as needed from my collection (the Internet was not available in the lab at the time), and wandered around to see what other pairs were doing. Students shared their expertise—both mathematical and technical, and discovered so many new possibilities that I was quite taken aback. Students taught themselves how to generate fractals using iteration, and to produce intricate animated tessellations; they examined visual representations of complicated algebraic and trigonometric expressions, and experimented with creating new images by tweaking those expressions. The final presentations were quite marvelous, and the excitement drew other teachers and students into the lab to watch and listen!

These examples suggest that the diversity of options within a software program plays a part in the lab experience. It can either dampen or intensify the impact of the technical and mathematical diversity of the students.

Redundancy

Redundancy supports continuity by ensuring that vital skills are distributed among many agents. More than enough organisms are produced. More than enough agents possess the same knowledge. In the mathematics classroom, shared vocabularies and experiences are examples of redundancy (Davis & Simmt, 2003).

In the Linear Transformations project, the shared algebra knowledge of the students provided a solid basis for discussion, but the program language caused some difficulty. As-Easy-As was a DOS program. Basic tasks were not difficult to learn but routines were less intuitive than those of the Windows-based programs students were beginning to use. Although some students became quite proficient, many students wasted valuable class time troubleshooting minor errors in formulas.

Many students in the JavaSketchpad study had an inadequate geometry background and consequently, a poor grasp of geometry language. The resultant lack of shared terminology caused communication difficulties for some students, e.g., a student said AB, but was referring to an angle. On the other hand, the technical "language" (i.e., the icons, menus, commands and routines through which software and user communicate) was not a problem because the browser interface was familiar to, or easily learned by the students.

The students in the Independent Study Project shared a mathematics background but had little or no knowledge of the three programs. Those who chose *Sketchpad* found it intuitively easy (perhaps because the program uses the same conventions as Macintosh and Windows software), and rapidly moved ahead to animate sketches and create iterative designs. Although students had used As-Easy-As before, they had difficulty learning the commands to investigate topics such as polar graphs. In contrast, despite the complicated and unfamiliar structure and syntax of MapleV students became enthusiastic users. They used the Help files to teach themselves the syntax of the commands they needed (e.g., to solve a trig equation, or to plot a polar graph).

Many programs provide Help files that offer only definitions and cryptic instructions. MapleV, on the other hand, included dozens of examples that could be copied back to the main window and activated. Students quickly discovered that the Help files, in addition to syntactical examples, contained a treasure trove of incredible 2D and 3D graphs, some fully animated. Students altered the expressions to investigate what would happen. They competed to find, or create the most beautiful, most convoluted, most interesting plot. Some were so enthralled that they bought a copy of the program and printed colour copies of their discoveries to adorn their assignment covers.

These examples suggest that shared program terminology in addition to shared mathematical language plays a role in the development of a complex learning system. Software that adopts the syntax and routines of familiar programs exploits redundancy to facilitate interactions; software that presents an unusual interface can compensate for this deficiency by providing generous examples rather than detailed instructions.

Decentralized Control

At the time of the Linear Transformations project I was beginning to recognize several conditions that affected the teacher's control in a computer lab. The lab layout made it difficult to move around to help students and to ensure that they were on topic. Other important influences were: the relative independence of student pairs, and the level of complexity of the software. Two observations are particularly relevant. First, lab-classes fragmented quickly. After a brief time, not only was it physically difficult to pull the students together to summarize and share, it was *conceptually* difficult because students were at different stages in the activity, and it was *emotionally* difficult because many students didn't want to abandon their investigation. Second, students were able to set up a math investigation using unfamiliar software by following lines of instructions, but in some cases the meaninglessness of the instruction sequence interfered with the development of mathematical understanding. For example, many students who typed entries such as A1*$D1+A2*$D2, lost track of the fact that this was a step in finding the product of two matrices. Despite these difficulties, however, there is evidence that learning emerged from shared understanding; the student-initiated modifications mentioned earlier helped many in the class to move past the procedures and to focus on the visual representations.

By the time I was involved with the JavaSketchpad research, I was aware of the need to facilitate, to provide simple directions, to ask a small number of important questions, and to draw ideas together. I did not attempt to micro-manage the pace of student investigation and only interrupted the process for the class discussion. Thus, some control was effectively decentralized, but how did this play out in the interchange between student partners?

Listening to tapes of the student conversations I developed a new awareness of the role of peers in helping one another, and in directing the course of an investigation. Peer interventions, when properly monitored, helped propel students towards new understanding by providing opportunities for sharing and vetting ideas; however, in some cases the relative isolation of student pairs in a lab set the stage for peer interventions that interfered with the development of mathematical understanding. This finding is echoed in an article by Sfard and Kieran (2001) that analyses the communication between two students. The authors call into question whether our emphasis on student talking as a means of developing understanding is justified. Even students paired with a knowledgeable peer can form erroneous conclusions or fuzzy notions that go unnoticed and unchecked if students are isolated.

In the Independent Study Project, the out of class timeslots gave "permission" for distributed control, and the individual topics provided the matter over which to exercise that control. New ideas emerged from student interactions with one another and with the various programs. Pairs developed expertise and taught one another both math concepts and computer skills. Since students were not isolated, and because the student to teacher ratio on most days was low, any difficulties were noticed quickly and addressed. Although the class did not focus on a single goal, shared understanding emerged with regard to the use of technology, and with regard to the nature of mathematics; for the first time many students saw mathematics as a creative endeavour.

Organized Randomness

Davis and Simmt use the idea of task constraints to exemplify organized randomness in the classroom. They note that tasks that are proscriptive (i.e., that tell you what you can't do), allow more options for responding than tasks that are prescriptive (i.e., that tell you what you must do).

Recent research into the role of "serious play" is relevant to a discussion of how organized randomness operates in a technology-supported classroom. Serious play is described by Reiber, Smith, and Noah (1998) as a "special kind of intense learning experience in which both adults and children voluntarily devote enormous amounts of time, energy and commitment and at the same time derive great enjoyment from experience"(p. 1).

Research has shown that play motivates and engages students. With regard to science, Stone and Glascott (1997) report that free play promotes "curiosity and willingness to consider varying options" (p. 2). And Reiber et al. (1998) note that games are a way of telling stories, which are fundamental to both understanding and learning. Nevertheless, technological applications for learning often fail to build in opportunities for play. De Castell and Jenson (2003) state:

> Technologically re-mediated curriculum so far has largely rendered education "witless," by eroding and finally eliminating that playfulness which makes formal schooling an engine of intelligence rather than obedience. (p. 49)

At the heart of 'playfulness' is the idea of self-direction (Reiber & Matsko, 2001). Tasks that allow students to set their own goals within broader constraints (i.e., proscriptive) incorporate this central idea, but the complex environment in the lab-classroom can sometimes reshape even prescriptive tasks.

As defined, the Linear Transformations task was quite prescriptive. Although it was a novel activity in terms of the OAC Algebra and Geometry course, students needed to set up their spreadsheet in a specific way and answers were either right or wrong. However, in practice the task was more flexible. The software permitted students to devise new approaches, which broadened the nature of the task. In addition, interest in animations, and curiosity about which matrix controlled which aspect of the shape led some students to experiment with additional features of the program.

In the Independent Study Project there were minimal constraints. A number of programs with very different affordances could be used to investigate a wide range of topics. For example, to create polar plots, students could use a numerical approach on a spreadsheet or experiment with equations in MapleV. Students played with the programs—they were absorbed, set their own pace, showed enthusiasm and spent time investigating peripheral topics that caught their interest. In many cases the students explored topics that were well beyond the curriculum.

They developed:

- Tessellations that waved!
- Intricate polar graphs
- Fractal trees and designs

Along with the visuals the students gave competent explanations of the mathematics.

As designed, the JavaSketchpad tasks fell somewhere between proscriptive and prescriptive. All tasks were designed to allow students to think about the mathematics using both traditional Euclidean methods and symmetry concepts, but these options were set by the sketch designer and students could not freely decide on a course of action. However, in practice, since students had not used computers to explore mathematics, the tasks did engage students and provided opportunities for them to set and modify goals—in a limited setting.

The following excerpts illustrate two different types of play that were observed in the JavaSketchpad study: play to explore new possibilities, and play to investigate a question. In the first example Sue and Paul, two above-average students, jabbed at the screen as they referred to items; however, they didn't leave the sketch static. They continued to play with the sketch and explore new possibilities.

Paul: GBH is congruent to GCH because angle, angle, given, common, common
Sue: So wait a second. Side, side—no side, angle, side
Paul: Angle B and angle C are equal and then H is the midpoint…. It's the right bisector too

Other students played to investigate a particular question—and actually called it "play" as seen in this quote:

Sarah: I feel obligated that there must be 3 pieces of information given. Let's play with it and maybe we'll see.

These examples suggest that there is a strong incentive to 'play' in a technological environment. If so, organized randomness will be a characteristic of a lab-class as long as the task is engaging and the software allows students to set and modify their own goals.

Neighbour Interactions

Davis and Simmt propose that we consider the interactions of ideas when we examine complex learning systems in education; this section will focus on the conditions in which those interactions take place within the lab environment.

Peer-peer

When students work in pairs on computer activities, exchanges between partners are central. Teasley and Roschelle (1993) maintain that the partners construct a Joint Problem Space. They note:

> We propose that the fundamental activity in collaborative problem solving [in a computer environment] occurs via the students' participation in the creation and maintenance of a JPS. (p. 229)

Although students collaborating without computers may construct such a space, I contend that the relative isolation of students in a lab, caused to some degree by the configuration of furniture, supports the development of a strong link between student partners—for better or for worse.

The Linear Transformations project was an early lab experience for me. I did not permit students to move around and work with other pairs. Additionally, the layout of the lab, the paucity and inaccessibility of board space, and the poor quality of the overhead monitor (an old television) made it difficult to share results in the lab, and limited students' opportunities to take advantage of the range of skills within the class. Nevertheless, some interactions did occur between pairs clustered in the same area, and student-invented modifications spread on a small scale. By the second day, students in other parts of the class had learned the new methods and were busy passing on their expertise.

For the JavaSketchpad research, students were strongly linked to their partners. Interactions with other pairs were less frequent. As noted earlier in the discussion of decentralized control, partner interactions were sometimes beneficial and sometimes detrimental to the growth of understanding.

In the Independent Study Project students were strongly linked to their partner, but also interacted frequently with other student pairs. This secondary relationship acted as an error check and allowed students to try out their ideas on a local audience before the presentation.

These observations suggest that physical conditions can affect the exchange of ideas and in turn support or prevent the emergence of shared mathematical insight.

Task interactions

Equally important in lab classes are the interactions between a student pair and the task, and between task questions and the affordances of the computer program.

The JavaSketchpad research showed that the relationship between student pairs and a computer-based task is affected not only by the subject matter (i.e., topic, difficulty level), but also by the way in which the problem and accompanying questions are presented. The task itself steps into the learning space through its prompts and questions. To support the growth of mathematical understanding the questions must invite students to explore, and the instructions must model the use of terminology and encourage students to explain their reasoning.

A task also intervenes through the pre-constructed sketch or the computer object that students create. Results of the JavaSketchpad research suggest that this image is most effective if it generates surprise. I believe the success of the Linear Transformations project, despite the heavy setup demands, was related to students' surprise at seeing the result of each transformation. Their delight fueled their curiosity, which led them to search for explanations. Each Independent Study Project was different, however, again, surprise at the visual rendition of a mathematical idea was a crucial factor; in many cases it led students to embark on extensive research, and to acquire new technological skills.

These results provide evidence that the visual and kinesthetic features of computer applications can stimulate the "bumping" of ideas. At the same time, analysis of the JavaSketchpad research data showed that failure to address the relationship between task question and software affordance can hamper the development of new ideas by leaving students confused and frustrated (Sinclair, 2003b). At the most basic level, if students are prompted to carry out a particular action (e.g., drag vertex A and observe the change in the perimeter of ABCD) the sketch must provide several affordances—a dragging tool, a movable quadrilateral, and an updatable measure. If a question introduces uncertainty or surprise the sketch must support student experimentation (i.e., it must provide students with the means to satisfy their curiosity about the result). Four simple but important criteria for linking task question and sketch affordance are listed in Table 1.

Purpose of Question	*JavaSketch* must
Focus attention	Draw attention
Prompt action	Provide affordances
Invite exploration	Provide alternate paths
Introduce uncertainty	Support experimentation

TABLE 1. Question/Sketch interplay

Conclusions

Complex systems are adaptable, evolvable, resilient, boundless, and novel (Kelly, 1994). However, Kelly points out that such systems are also nonoptimal, noncontrollable, nonpredictable, nonunderstandable and nonimmediate. If our mathematics classes are complex systems we must develop methodologies that allow us to take advantage of the good qualities. The first step is to focus our attention on evidence from practice.

This paper examined three examples of mathematics classes with respect to five conditions that Davis and Simmt (2003) identify as necessary but insufficient for the development of a complex system that is able to learn. The analysis suggests that diversity, redundancy, distributed control, organized randomness, and neighbour interactions were present to varying degrees but were affected in the lab-based environment by:

- Lab configuration;
- Program ease of use and/or depth of options;
- Task design;
- Student opportunities to share knowledge with peers, and to communicate with the teacher

In the Linear Transformation project and in the JavaSketchpad study there were some indications that tentative understandings were beginning to emerge. For example, the affordances of As-Easy-As enticed some students to move beyond the constraints of the assignment, and the visual impact of the spreadsheet images and the geometry applets encouraged students to play and explore. However, neither of these situations epitomized an emergent learning system.

The Independent Study Project satisfied all the conditions. The diversity of math skills, technical skills, and computer affordances supported exploration and analysis. The familiar conventions of Sketchpad, prior experience with As-Easy-As, and the MapleV help files, coupled with shared mathematics basics supplied enough redundancy in the system to support communication. The tasks were proscriptive. Each encouraged investigation and open-ended question posing. The free interactions between pairs allowed the sharing of mathematical and technical expertise, and the development of innovative ideas. In addition, the out of class timing dampened the negative effects of a cramped lab space. Learning flourished as the classes 'gelled'.

Observations of the Independent Study Project groups suggest that a structure allowing for sharing, play, and individual choice, that involves activities based on broadly applicable and adaptable software can result in the emergence of a beneficial complex learning system in a technological environment.

Endnotes

1. The assignment was originally designed by mathematics teacher and researcher, Ysbrand de Bruyn.
2. In later years students used Microsoft Excel. ("Microsoft Excel, version 5," 1993).
3. *JavaSketches* are html versions of sketches created in *The Geometer's Sketchpad* and then saved using *JavaSketchpad.* A *JavaSketch* behaves differently than its *Geometer's Sketchpad* counterpart. In the non-web sketch, independent points can be moved anywhere, and other points, lines and line segments can be translated by dragging; however, in the web version, only independent points can be dragged. Although action buttons can be included to hide/show or animate, the applets cannot be edited. At present some geometric details (e.g., rays) are not supported in the web version.
4. OAC refers to Ontario Academic Credit. OAC courses were University preparation courses taken in the 4th or 5th year of high school. These courses were discontinued in 2002.
5. Proof using Sketchpad of a method to divide any line segment into a regular partition of any number of parts. See - Dietrich, C. H., Litchfield, D. C., & Goldenheim, D. A. (1997). Euclid, Fibonacci, Sketchpad. *Mathematics Teacher, 90*(1), 8–12.

References

As-Easy-As, version 5.5. North Andover, MA: Trius, Inc.

Cronbach, L. J. "Playing with chaos." *Educational Researcher* 17, no. 6 (1988): 46–49.

Davis, B. "Toward a pragmatics of complex transformation." *Journal of the Canadian Association for Curriculum Studies* 1, no. 1 (2003): 39–45.

Davis, B., and E. Simmt. "Understanding learning systems: Mathematics education and complexity science." *Journal for Research in Mathematics Education* 34, no. 2 (2003): 137–167.

De Castell, S., and J. Jenson. "Serious play: Curriculum for a post-talk era." *Journal of the Canadian Association for Curriculum Studies* 1, no. 1 (2003): 47–52.

Doll, W. "Complexity in the classroom." *Educational Leadership* 7, no. 1 (1989): 65ç70.

Ebos, F., R. Tuck, and W. Schofield. *Mathematics: Principles & Process, Book 2*. Scarborough, Ontario: Nelson Canada, 1986.

The Geometer's Sketchpad Version 3. Key Curriculum Press, Berkeley, CA

JavaSketchpad DR3. Key Curriculum Press, Berkeley, CA

Johnson, S. *Emergence: The connected lives of ants, brains, cities, and software*. New York: Simon and Schuster, 2001.

Kelly, K. *Out of control: The new biology of machines, social systems and the economic world*. Cambridge, MA: Perseus Books, 1994

Lesh, R. "Dealing with complexity: New paradigms for research in mathematics education." In *Proceedings of the Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education,* edited by M. L. Fernandez, 2-20. 2000. ERIC Document Reproduction Service, ED 466728.

MapleV release 2. Waterloo Maple Software, Waterloo, Ontario.

Microsoft Excel version 5. Microsoft Corporation, Cambridge, MA

Reiber, L. P., and M. J. Matsko. "Serious design for serious play in physics." *Educational Technology* 1, (2001):14–24.

Reiber, L. P., L. Smith, and D. Noah. "The value of serious play." *Educational Technology* 38, no. 6 (1998): 29–37.

Sfard, A., and C. Kieran. "Cognition as communication: Rethinking learning-by-Talking Through Multi-Faceted Analysis of Students' Mathematical Interactions." *Mind, Culture, and Activity* 8, no. 1 (2001): 42–76.

Sinclair, M. P. "The provision of accurate images with dynamic geometry." In *Proceedings of the 27th Conference of the International Group for the Psychology of Mathematics Education held jointly with the 25th Conference of PME-NA*, edited by N. A. Pateman, B. J. Dougherty and J. T. Zilliox, 191–198. Honolulu, HI: Center for Research and Development Group, University of Hawai'i, 2003.

———. "Some implications of the results of a case study for the design of pre-constructed, dynamic geometry sketches and accompanying materials." *Educational Studies in Mathematics* 52, no. 3 (2003): 289–317.

Stone, S. J., and K. Glascott. "The affective side of science instruction." *Childhood Education* 74, (1997): 102–104.

Teasley, S. D., and J. Roschelle. "Constructing a joint problem space: The computer as a tool for sharing knowledge." In *Computers as Cognitive Tools,* edited by S. P. Lajoie and S. J. Derry, 229-258. Hillsdale, NJ: Lawrence Erlbaum Associates, 1993.

About the Author

Margaret Sinclair is an Assistant Professor in the Faculty of Education at York University in Toronto, where she teaches preservice and graduate courses in mathematics education. She brings to this work almost twenty years of experience as an elementary and secondary classroom teacher and a secondary administrator. Sinclair has written course profiles for the Ontario Ministry of Education and Training, has co-authored several textbooks published by Pearson Education for the new Ontario secondary curriculum and has been involved in the development of Internet mathematics materials. Her ongoing research focuses on the use and design of pre-constructed dynamic geometry sketches. (Email address: msinclair@edu.yorku.ca)