USING STARLOGO TO INTRODUCE DIFFERENTIAL EQUATIONS

PHILIP ANDERSON AND CARL R. SEAQUIST

Abstract. Massively parallel programming languages, like StarLogo, provide a rich environment for introducing differential equations to students with an unsophisticated mathematical background. In this paper we describe the basic software for simulating and monitoring various population dynamics. Simple differential equations that describe the observed dynamics are introduced. The same environment is then used to explore solutions to the differential equations using Euler's method. Many solutions can be displayed simultaneously and viewed as a flow, which is easier to understand than the more traditional slope fields. All software is written in StarLogo, a massively parallel version of Logo, and can be easily modified, thus permitting students to embark on their own explorations.

1. Introduction

Many college students take their first differential equations course as second-semester sophomores or first-semester juniors. These courses typically involve learning techniques for solving differential equations in closed form; theoretical results involving existence and uniqueness of solutions; and modeling situations with differential equations. Although every effort is made to motivate particular equations as models of important real world problems, often the students become confused because of the multiplicity of views presented. In an introductory course they are unsure whether they should concentrate on memorizing a collection of baroque techniques for solving the few differential equations that can be solved in closed form, whether they should concentrate on the theory, or whether accurate modeling is the issue. Our goal here is to provide a particular set of situations that can be modeled by differential equations and then allow the students to experiment with these situations. We provide an environment for a series of activities that allow the student to use an agent-based simulator for modeling various situations that can also be modeled by differential equations. The students are encouraged to compare the actual results of their simulations with those predicted by the appropriate differential equations. Our goal is to introduce differential equations to mathematically unsophisticated students in preparation for the material presented in the standard introductory differential equations course. In particular, we concentrate on population dynamics and emphasize the fact that differential equations are only one of many possible mathematical models that can be used to understand population dynamics. By applying differential equations to model actual population dynamics of artificial communities that they create, students discover many modeling issues, they become aware of the existence and uniqueness of solutions, and they are motivated to learn both numerical and analytic techniques for solving differential equations. Our approach is constructionist [P1991] in that students build knowledge by constructing both the situation to be modeled and then the differential equations to model the situation. We believe that

in this way they obtain a better understanding of the area and are then ready to obtain maximum benefit from a standard course.

There are many ways to simulate population dynamics on a computer. Here we use agent-based simulations. In such an approach each agent in a population is simulated. The actions of each object that is under the control of a procedure are determined by its environment and its interaction with other agents. One of the most popular simulation systems that allow such agent-based simulations is a platform developed at the Santa Fe Institute called Swarm [B1994, M1996]. It is currently finding many applications in the area of artificial life and in the social sciences.

In our work we use a similar language, StarLogo, developed by Mitchel Resnick of the Epistemology and Learning Group at the MIT Media Laboratory [R1994]. We chose this language because of its similarity to the Logo language [P1979], which was originally designed for use by young children. We hoped that because of the relative success [C1992] of the Logo language in elementary schools, we might eventually be able to expose high school and junior high school students to differential equations in a natural way, especially if they had been exposed along the way either to Logo and/or StarLogo.

In the simplest of cases, StarLogo is like Logo in that under program control a turtle crawls around the screen, sometimes drawing as it crawls, sometimes not. In StarLogo, however, instead of one turtle, there can be hundreds of turtles, even thousands, that crawl around and interact with each other and their environment. In StarLogo there are actually three programming objects: turtles that crawl around, patches that represent pieces of the world on which the turtles crawl, and observers that monitor the activity of the turtles and patches. All the turtles execute the same procedure over and over again in lock step. Thus the kind of parallelism principally supported by StarLogo is in Flynn's classification that of SPMD, or single program multiple data. StarLogo was originally implemented on the Connection Machine, a massively parallel Lisp machine that runs a parallel version of Lisp called *Lisp. Now versions of StarLogo exist on Unix workstations, Macintosh computers (now called MacStarLogo) and recently a version was implemented in JAVA that will run on many platforms, including a PC [B1999].

2. Modeling Artificial Communities with Differential Equations

Our approach is to help students create an "artificial" community of turtles. These turtles give birth to new turtles, compete for limited resources, and die. The number of turtles is monitored by an observer that plots the results. These results can then be compared with the solutions to various differential equations that model the situations. Discrepancies are explained. Because we want to be able to eventually apply this approach to students with little mathematical sophistication, we first introduce an artificial community and encourage students to experiment with it and to discuss why the plot of the population varies as it does. A mathematical model in the form of a differential equation is proposed. Solution curves to the differential equation are computed for various initial conditions simultaneously. This is done in StarLogo by taking a collection of turtles and placing them at places which represent various initial conditions. Each turtle then begins to crawl in a direction determined by evaluating the differential equation at the turtle's particular location. By coloring the turtles appropriately and displaying them all simultaneously, they give the impression of creating

a flow along the solution curves. Once the students are comfortable enough with the StarLogo program to create this flow, they are told about Euler's method and its relation to the program. If time permits, ways of improving on Euler's method can be discussed. It can also be pointed out that since the direction of a turtle at any point is completely determined by its position, the paths of the turtles do not cross; i.e., the solutions are unique. If a computer-based algebra system, like Maple or Mathematica, is available, then students can also plot solutions to the differential equation(s), which they can compare with the actual data they observed in the community of turtles. Finally, many of these differential equations have closed-form solutions. For students with little mathematical sophistication, these functions can be introduced as possible solution candidates because they have same general shape as the numerical solutions they have found. For more sophisticated students, the functions can be shown to be solutions to the particular differential equation, either directly or by using a computer-based algebra system. Finally, the student can be shown how to find the family of solutions analytically by hand or by using a Computer based Algebra System.

3. Specific Examples

We address three specific examples: exponential growth, logarithmic growth, and the Volterra-Lotka predator/prey system. In each instance we give a running template of a StarLogo program that creates a community of turtles which behaves in the desired manner. Various buttons on the user interface of each program allow the student to vary initial conditions and parameters.

As our first example, see Figure 1. It shows the user interface for the exponential growth program. Here the student has two sliders: one to control the initial population and the other to control the birth rate. In addition, Figure 1 shows the program, the plot of population versus turtle time, and the field on which the turtles live. The students are encouraged to experiment with various input conditions and to modify the program to make the community more realistic.

Figure 2 shows the portion of the StarLogo program that controls the turtles in the logistic growth situation. The global variable "switch" is set when the simulation is supposed to end. Each turtle checks to see if the simulation should end. If not, then the turtle makes a random turn (less than 90 degrees) to the left and then to the right and moves forward one unit. These actions simulate a random walk. Then, based on settings controlling the birth and death rates, the turtle randomly decides whether or not to give birth and/or to die. Other parts of the program not included above take care of initialization and the user interface. There is also a short procedure that the observer executes in order to plot the number of turtles versus time.

This is a barebones approach to simulating a turtle community. A more interesting approach could include a limited resource like space. Up to a certain point increased interaction of turtles caused by more turtles in the confined space could increase the birth rate, while eventually more interactions might increase the death rate as well. Students are encouraged to experiment with these possibilities.

Figure 3 shows the phase plane map for the system of differential equations modeling the Volterra-Lotka predator/prey system. Here two competing breeds of turtles, red and green, are introduced. The red breed, the predator, feeds on the green breed, the prey. As the population of the predator

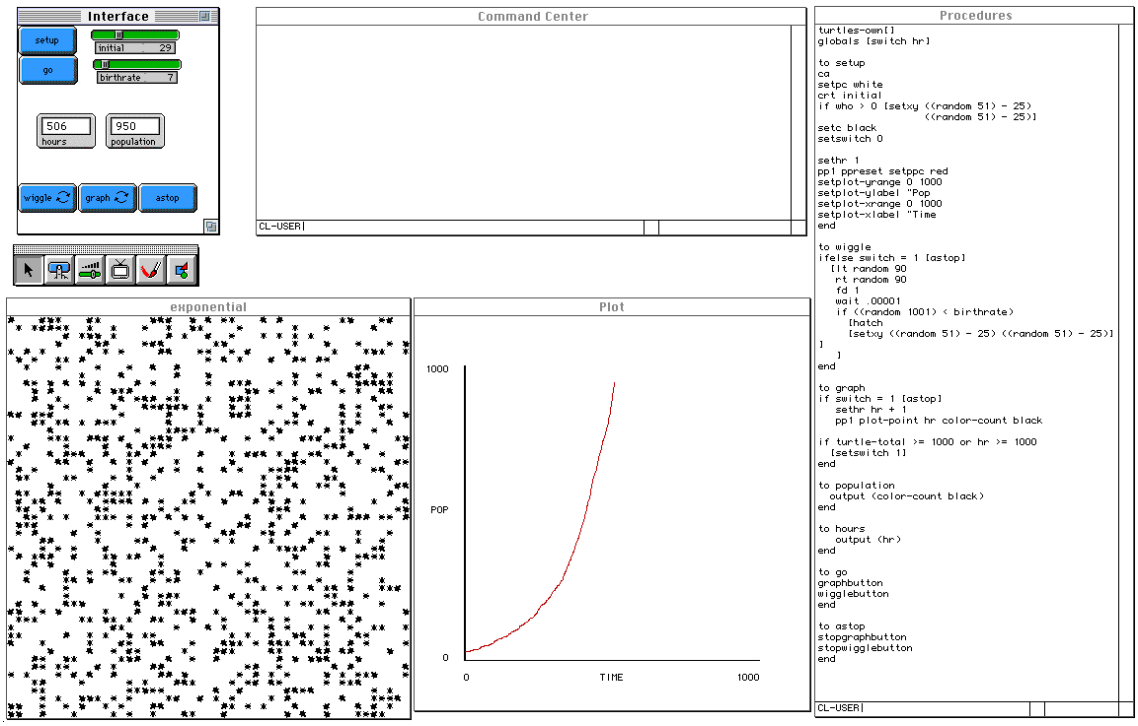

FIGURE 1. The StarLogo display for the exponential growth community.

```
to wiggle
  ifelse (switch = 1) [astop]
   [lt random 90
    rt random 90
    fd 1
    wait .00001
    if ((random 1001) < a)
      [hatch [setxy ((random 51) - 25) ((random 51) - 25)]]
    if ((random 10001) < (bb * turtle-total)) [die]
   ]
end
```

FIGURE 2. The turtle procedure for the logarithmic growth community.

increases, the population of the prey decreases, but as the food supply for the predator decreases, predators began to die off, allowing the prey population to rebound. In the phase plane map the population of the predator is plotted on the x-axis against the population of the prey on the y-axis. Thus the cyclical behavior as shown in the figure is observed.

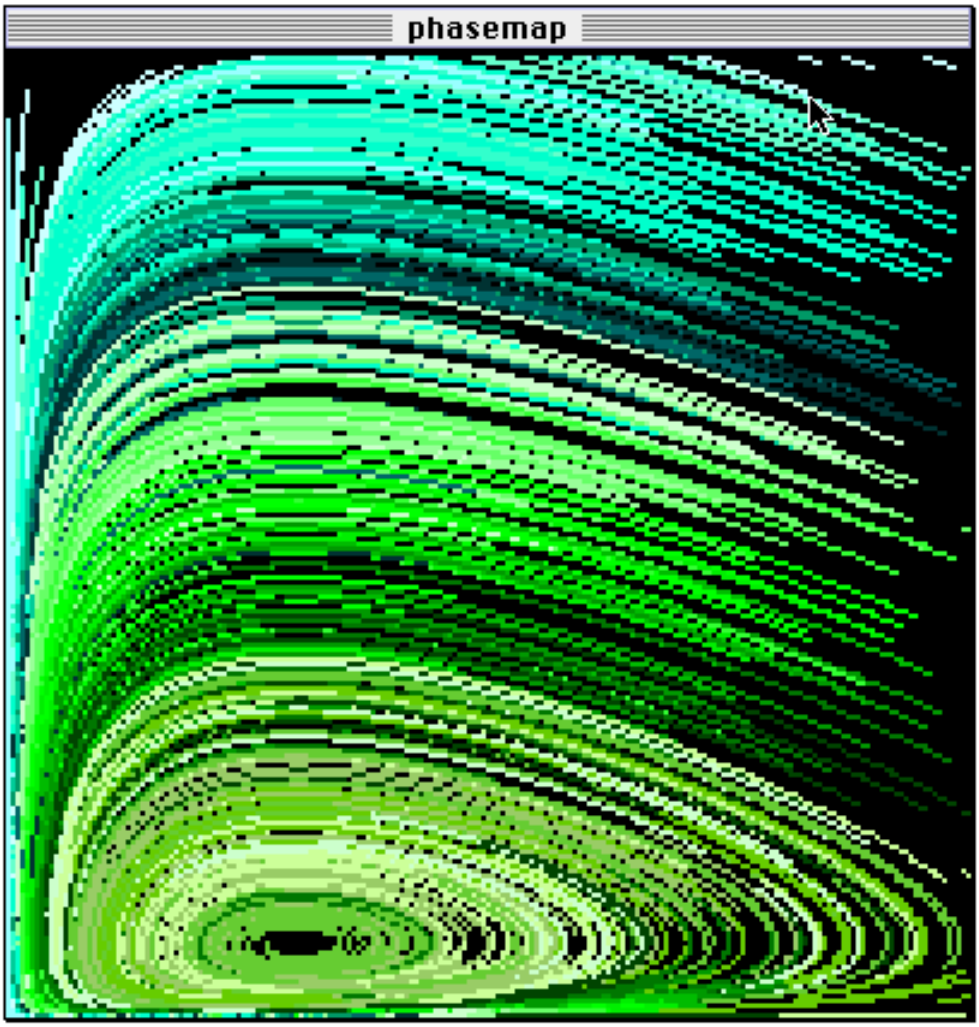

FIGURE 3. The phase plane map for the predator/prey community.

4. Conclusions and Future Work

Our work is very preliminary but both authors are optimistic about our approach. Even though both authors have taken and/or taught an introductory course in differential equations, we both believe that by developing this approach we increased our understanding of population dynamics and differential equations. Perhaps most surprising was how difficult it is to get the population of an actual community of artificial turtles to behave as predicted by the differential equations. We knew the differential equations we wanted to introduce but it was difficult to get a community of turtles to behave as predicted. Figure 4 shows the population of the original community we designed, which was to follow a logistic growth pattern. It does, in general, follow such a pattern, but we were very surprised to find the cyclic behavior once the population began to approach the carrying capacity

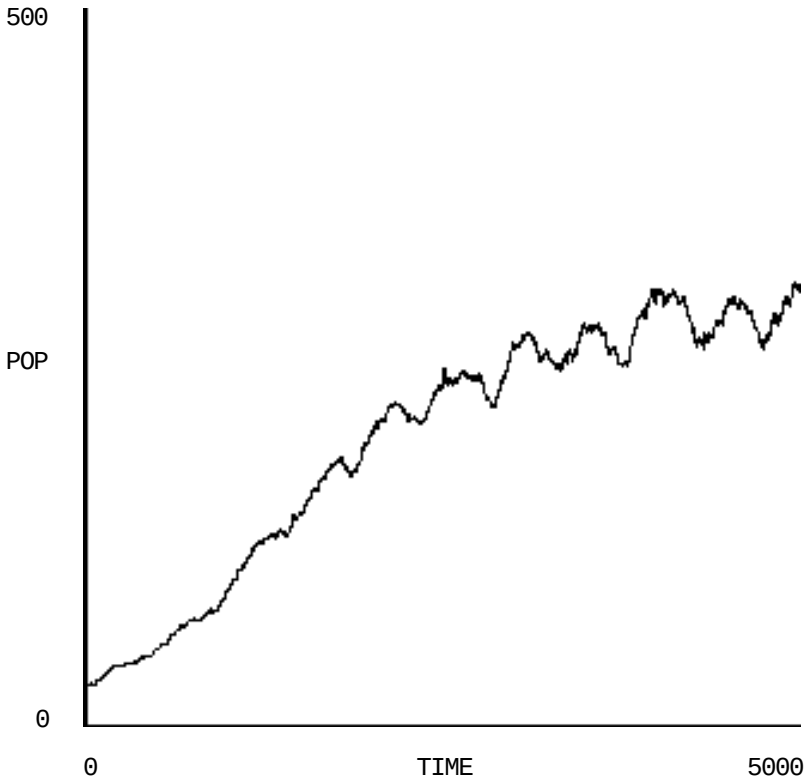

FIGURE 4. The phase plane map for the predator/prey community.

of the community. This did not appear to be a noise phenomenon but rather was caused by a feedback delay that was unintentionally introduced. This phenomenon is typical of delay differential equations.

We still do not have experience with trying out our approach on undergraduate students, although we have used the StarLogo programs for demonstration purposes. There are many reasons we have not yet tried our approach. One is that StarLogo has only recently become available on platforms that are widely available at our respective schools. The fact that this new release is still in a beta version also makes it difficult to use in a classroom situation. Another difficulty we have encountered is that although we teach Logo programming to our prospective mathematics teachers, the K-12 teachers in our community are not teaching Logo to their students. Thus we initially expected that students in high school or early years of college would have some expertise in Logo, but instead found that they have never even heard of the language or its turtle paradigm. When talking to elementary school teachers about this situation, we were told that kids did not need to learn programming any more because software packages were now widely available and programming was no longer a germaine subject. Unfortunately, independent of the fact that the act of learning to

program might be valuable, the teachers we talked to did not seem to understand that in order to control a sophisticated software package some sort of programming is necessary. Our final difficulty is that most syllabi are so crowded with important information that there is very little time for an undergraduate to explore and experiment with a field. Our hope is that as StarLogo becomes more stable, we will be able to introduce it to some of the math clubs in the local junior high schools. We hope then to introduce our approach to learning about differential equations to these students. These students then will be in a position to appreciate the areas discussed in an introductory course on differential equations: the finding of closed formed solutions, the theory of existence and uniqueness of solutions, and the art of modeling.

References

[A1998] P. Anderson, *Teaching Differential Equations with StarLogo*, working paper, 1998.

[B1999] A. Begel, Email from Andrew Begel, abegel@cs.berkeley.edu, 1999.

[B1994] R. Burkhart, *The Swarm Multi-Agent Simulation System*, Technical Report, Object-Oriented Programming Systems, Languages, and Applications (OOPSLA) '94 Workshop on "The Object Engine", 7 September 1994, http://www.santafe.edu/ rmb/oopsla94.html.

[C1992] D. H. Clements and J. S. Meredith, *Effects and Efficacy*, Logo Foundation, NY, 1992, http://el.www.media.mit.edu/groups/logo-foundation/Publications.

[F1966] M. J. Flynn, *Very High Speed Computing Systems*, Proc. IEEE, Vol. 12, 1966, 1901–1909.

[M1996] M. Minar, R. Burkhart, C. Langton, M. Askenazy, *The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations*, Santa Fe Institute, 1996, http://www.santafe.edu/projects/swarm.

[P1991] S. Papert, *Situating Constructionism*, In *Constructionism*, edited by I. Harel and S. Papert, Ablex Publishing, Norwood, NJ, 1991.

[P1979] S. Papert, et al. *Final Report of the Brookline Logo Project*, Part II: Project summary and data analysis (Logo Memo No. 53), Cambridge, MA: MIT, AI Laboratory, 1979.

[R1994] M. Resnick, *Turtles, Termites, and Traffic Jams, Explorations in Massively Parallel Microworlds*, MIT Press, Cambridge, MA, 1994.

[R1997] M. Resnick, *StarLogo Reference Manual*, Epistemology and Learning Group, MIT Media Laboratory, Cambridge, MA, 1997, http://starlogo.www.media.mit.edu/people/starlogo/documentation.

Department of Mathematics, Box 134, South Plains College, Levelland, TX 79336
E-mail address: panderso@spc.cc.tx.us

Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX 79409-1042
E-mail address: seaqucr@math.ttu.edu