

PopDyn—an ecological simulation program

Models, whether conceptual, mathematical, or experimental, have become an increasingly important area of ecological and evolutionary research. Models vary in size, nature, and complexity, but typically share a common goal: to enhance understanding of a system. Students of ecology and evolution are introduced to models early in their academic development, but rarely do they have the opportunity to explore these models in any detail. In most cases, dynamic processes are presented in lecture as static displays and are not dealt with further. Consequently, many students fail to develop a sophisticated understanding of these systems and do not appreciate the advantages of formulating models.

Much of contemporary ecology rests on three fundamental models: some form of density-dependent (e.g., logistic) growth, competition, and predation. The computer program PopDyn was written to enable instructors to present these fundamental models dynamically and to allow students to explore how changes in the parameters affect the dynamic behavior of the populations.

PopDyn¹ is an interactive computer program written for the Macintosh that allows users to graphically explore several classical ecological models: logistic growth; competition; predator-prey interactions with linear, type II, or type III functional response; food chains; and food webs. The parameters used in each model are set by adjusting visual metaphors on the screen and may be altered before and during a simulation. The ability to alter parameter settings during a simulation allows one to explore how ecological and/or evolutionary changes (e.g., improved capture efficiency of one trophic level

File Edit Run Models Model Options View Options

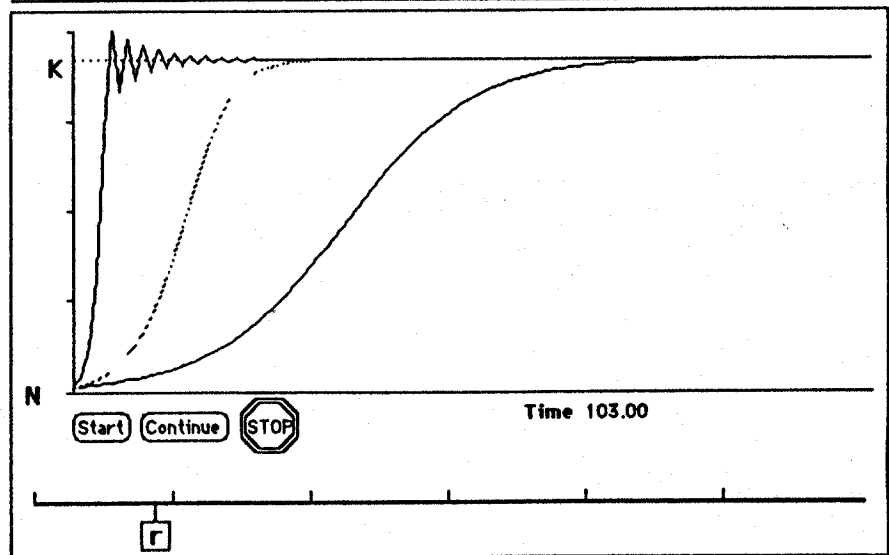


Figure 1. The logistic window in PopDyn. The icons N , r , and K can be moved with the mouse to set the parameters in the logistic equation. The icons for N and K move vertically along the y-axis, increasing or decreasing the value of the parameter used in the simulation. The icon for r moves horizontally along the scale bar. The effect of increasing the parameter r during three separate runs is shown. Sequential simulations alternate between a solid and dotted trace on a black-and-white monitor and among six colors on a color monitor.

on another) affect the dynamics of each population in a community. Parameters can be deterministic or stochastic with the magnitude and type of variability (such as normal or uniform) set by the user. The results are plotted in real time as density versus time and simultaneously in phase space when appropriate. There is context-sensitive help in all models, and particular parameter settings can be saved to a file and reused. A manual presents a series of exercises to lead students through each model and describes the models and various options in PopDyn.

Logistic model

The window for the logistic model is shown in Figure 1. The parameter values of the logistic equation

$$\frac{dN}{dt} = r \left(1 - \frac{N}{K} \right) N$$

can be set by moving the appropriate screen icons (N , r , and K) with the mouse. Once the parameters are set, clicking on the start button begins the simulation. The population trajectory is plotted during the simulation. Simulations can be stopped and continued at any point using the three control buttons (Start, Continue, and Stop) located below the plot. There

by Ron J. Etter

¹The program can be obtained free of charge from R. J. Etter.

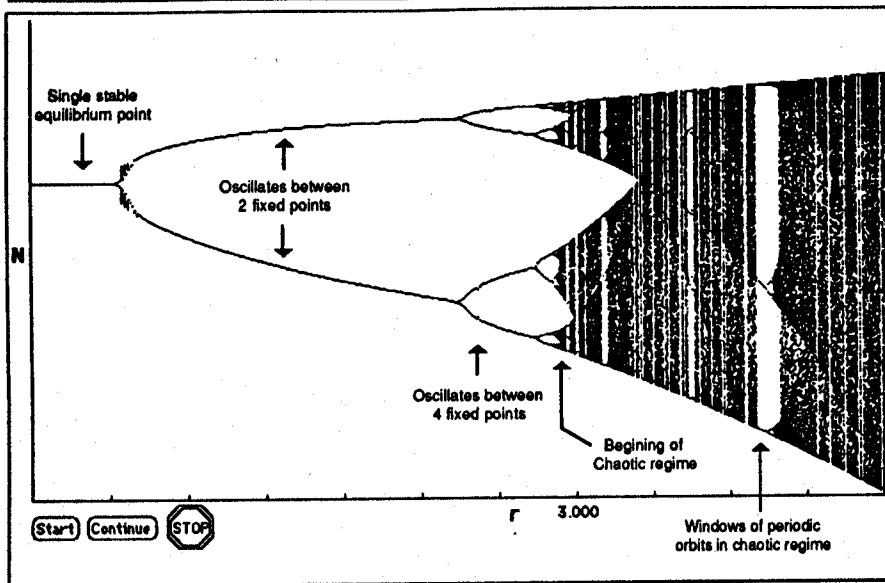


Figure 2. This plot depicts the cascade of bifurcations in the dynamic behavior of the logistic equation leading to chaos as a single parameter (r) is changed. The dynamics change from a single equilibrium point to oscillations with periods of 2, 4, 8, 16, and on into chaos (aperiodic behavior). On a color monitor, each bifurcation is drawn in a different color.

are no limits on how long a simulation runs.

An example of how one might use this program to explore the effects of increasing the population-growth rate on logistic growth is shown in Figure 1. These results were obtained by running the simulation three times with increasing values of r . Note that on one run (that with the largest r value) the trajectory overshoot K and exhibited a damped oscillation. If I had continued increasing r , the dynamics of the system would have moved through a series of bifurcations (periodic orbits) and on into chaos (aperiodic behavior).

To see a more synoptic view of the effect of increasing r on the dynamic behavior of logistic population growth, one can select the Bifurcation Map menu item under the models option menu. Click on the start button and the values of N that the system equilibrates to, or oscillates between, will be plotted as a function of the value of r (Figure 2). The period of the orbit is indicated by the number of points above each value of r .

Any parameter can also be stochastic such that the actual value used during each time step is drawn from a random sample with the mean and variance set by the user. The magnitude and type of variance is set from

within a dialog box (see Figure 9). Two types of variability for each parameter can be chosen: uniform (white noise) or normal. An example of several runs of the logistic model with both r and K stochastic is shown in Figure 3. Notice that the trajectories do not repeat but instead show

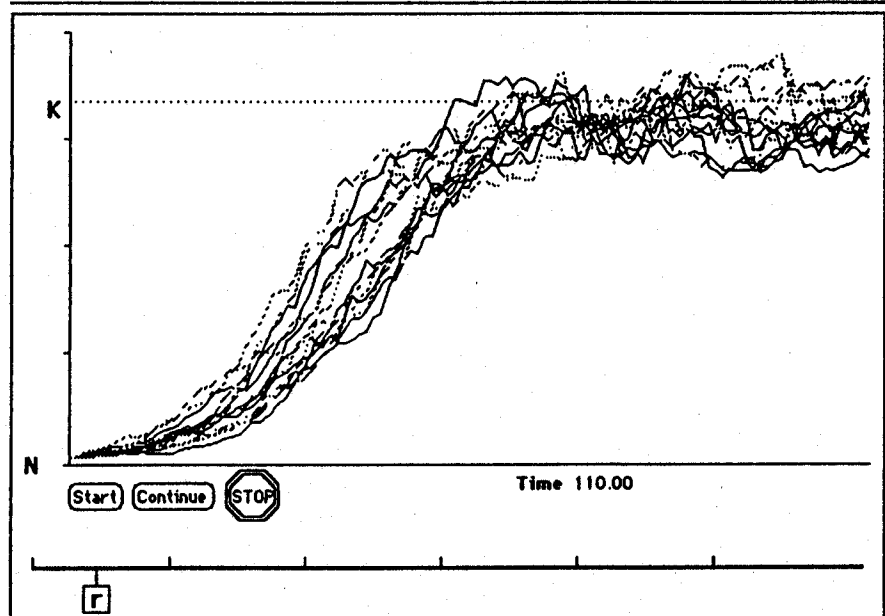


Figure 3. Several runs of the logistic model with both r and K stochastic. Changing parameters from deterministic to stochastic can be done easily through dialog boxes similar to that shown in Figure 9.

some variance about the mean (the trajectory if r and K were not stochastic). With this option, one can explore a variety of questions about how fluctuations in the parameters of the logistic model affect the dynamic behavior of the population.

Competition model

The competition model uses the Lotka-Volterra equations for competition between two species with the competing species represented by the light-colored (species 1) and dark-colored (species 2) hare icons. A sample competition screen is shown in Figure 4. The left panel is N (population density) plotted against time. The initial population size and carrying capacity for each population of hares is set by using the mouse to move vertically the appropriate icons located to the left of the y -axis. The reproductive rate for each species is represented by the width of the arrows at the bottom of the left panel and is set by increasing or decreasing the width of the arrow.

For example, to change the reproductive rate of the dark-colored hare, place the cursor over its arrow and click and hold the mouse button down while dragging the mouse. The number above each arrow is the

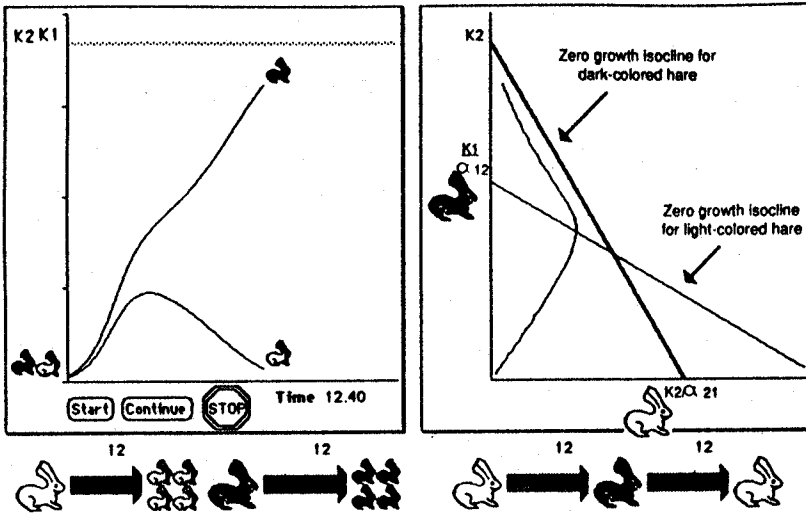


Figure 4 (top). The competition window in PopDyn. The competition coefficients (arrows in right panel), reproductive rates (arrows in left panel), carrying capacities, and initial densities for the competing hares can all be modified by moving the appropriate icon or adjusting the width of the appropriate arrow with the mouse before and during a simulation. The width of each arrow is adjustable and sets the magnitude of a parameter. The number above each arrow indicates its width, allowing users to return to the same parameter settings. The results from a simulation that has run for 12.40 time steps are shown.

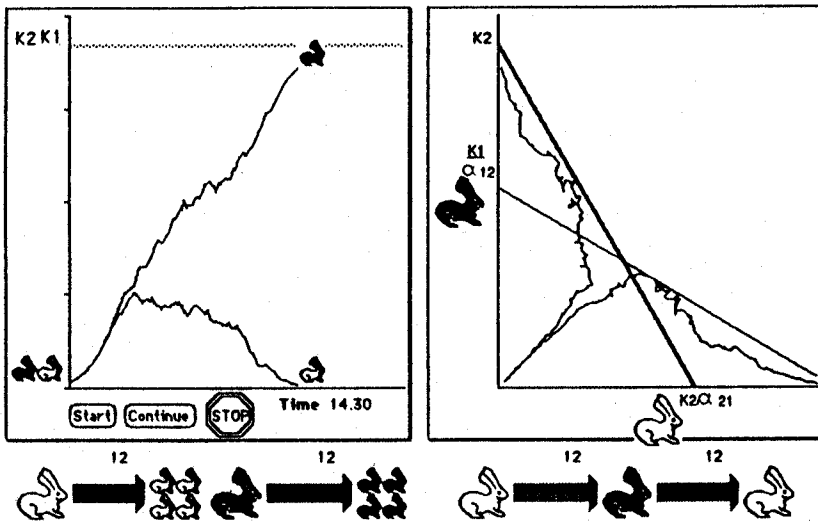


Figure 5 (middle). Two runs of the competition model with identical parameter settings (only one run is shown in the left panel because the previous run is erased before each simulation) but with stochastic competition coefficients. The contrasting results (which species wins) in phase space reflects the variance in the competition coefficients. The values of the competition coefficients for each time step in the simulation were drawn stochastically from a normal distribution with a mean set by the arrows below the phase space and by the variance set in a separate dialog box.

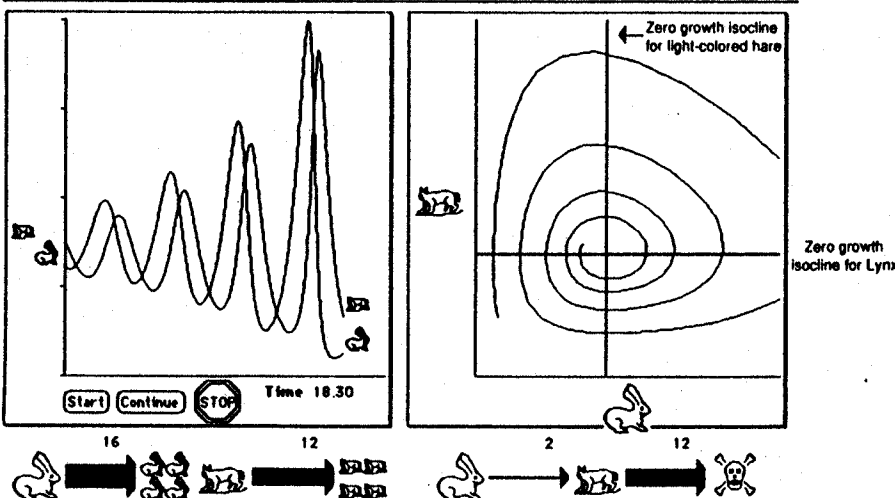


Figure 6 (bottom). One of the predator-prey windows in PopDyn using the typical Lotka-Volterra equations. The window functions in a manner similar to that described for the competition window. The arrows under the phase portrait set the capture efficiency and mortality rate of the lynx.

width of the arrow and is a relative estimate of the parameter (in this case, the reproductive rate for the dark-colored hare). The ability to rapidly change parameter values on screen without having to type in a series of numbers allows one to quickly determine the effects of different parameter states on population dynamics. All models use the same metaphor for setting parameter values.

The right-hand panel is the phase space of the two species, where the number of dark hares is plotted against the number of light hares. The thick and thin diagonal lines in the phase space are the zero-growth isoclines for the dark hares ($dN_2/dt = 0$) and light hares ($dN_1/dt = 0$), respectively, and each line adjusts dynamically as the parameters of the equations are altered. The arrows below the right panel set the competition coefficients. When a simulation is started, the trajectories of both popu-

lations are plotted simultaneously in N versus time and phase space.

An example is shown where the reproductive rates and competition coefficients are the same for both species, but the initial density of the dark hare is slightly greater, and its carrying capacity slightly less, than that of the light hare. In this case, the system has an equilibrium point where both species could coexist (where the two isoclines intersect), but it is an unstable equilibrium; the dark hare wins because it has a slight (one individual) advantage initially.

If the competition coefficients (or any of the other parameters) are stochastic, the outcome (which species wins) of a simulation with the same parameter settings as in Figure 4 is unpredictable (Figure 5). The species that wins is determined by which species gains an advantage during the simulation. The phase space in Figure 5 shows the outcome of two simulations with identical parameter settings.

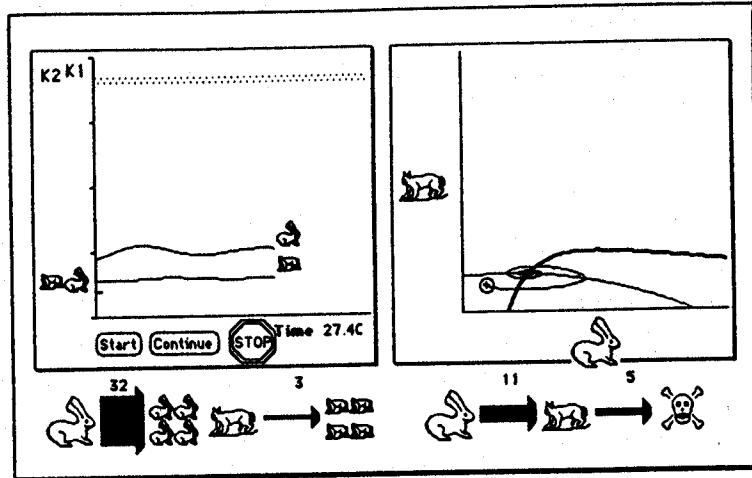
Predator-prey model

The predator-prey model uses the Lotka-Volterra equations with the prey (species 1) represented by the hare icon and the predator (species 2) by the lynx icon (Figure 6). The parameters in the equations are set in a manner similar to that for the competition model, and the isoclines in the phase space adjust dynamically to reflect any changes.

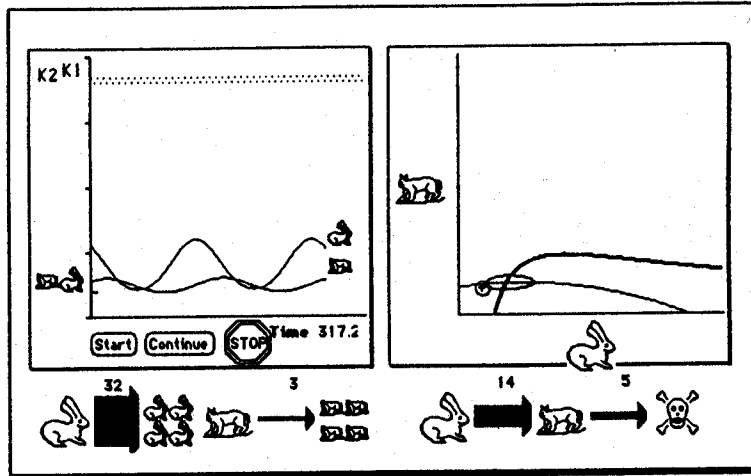
A sample run of the simulation is shown by the trajectories in each panel. Note that although the isoclines are perpendicular to one another, the oscillations are unstable, leading eventually to the extinction of both species. These results contrast with the examples typically presented in textbooks, where a stable limit-cycle results from this model. The explanation for the disparity is that there is an inherent time lag in all computer models, which in this case leads to unstable dynamic behavior. The time lag is essentially the integration time step, which can be set through a menu item (Integration Time Step) found under the Model Options menu. By setting the integration time step to a value less than 0.01, the stable behavior can be observed.

It is unlikely that the predator can

a



b



c

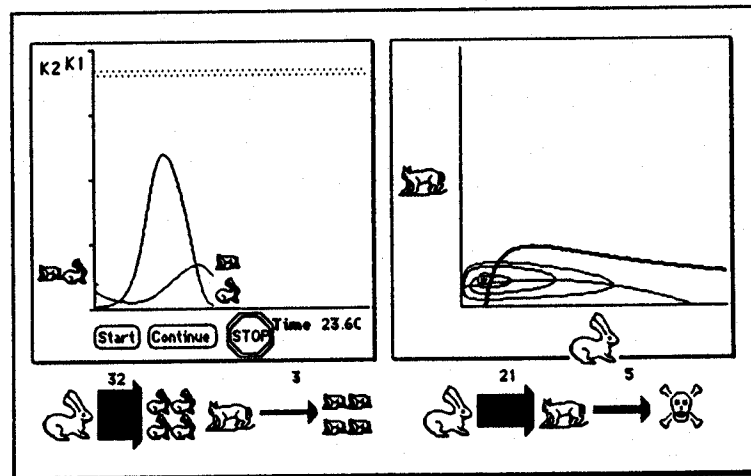


Figure 7. A predator-prey system with a type II functional response. Note the shape of the isoclines in the phase portrait. The successive plots show the change in the dynamic behavior of the populations as the capture efficiency of the lynx is increased. a. Poor capture efficiency: there is a damped oscillation to an equilibrium defined by the intersection of the two isoclines. b. Intermediate capture efficiency: the system reaches a stable-limit cycle. c. High capture efficiency: the predator drives the prey to extinction and then also goes extinct.

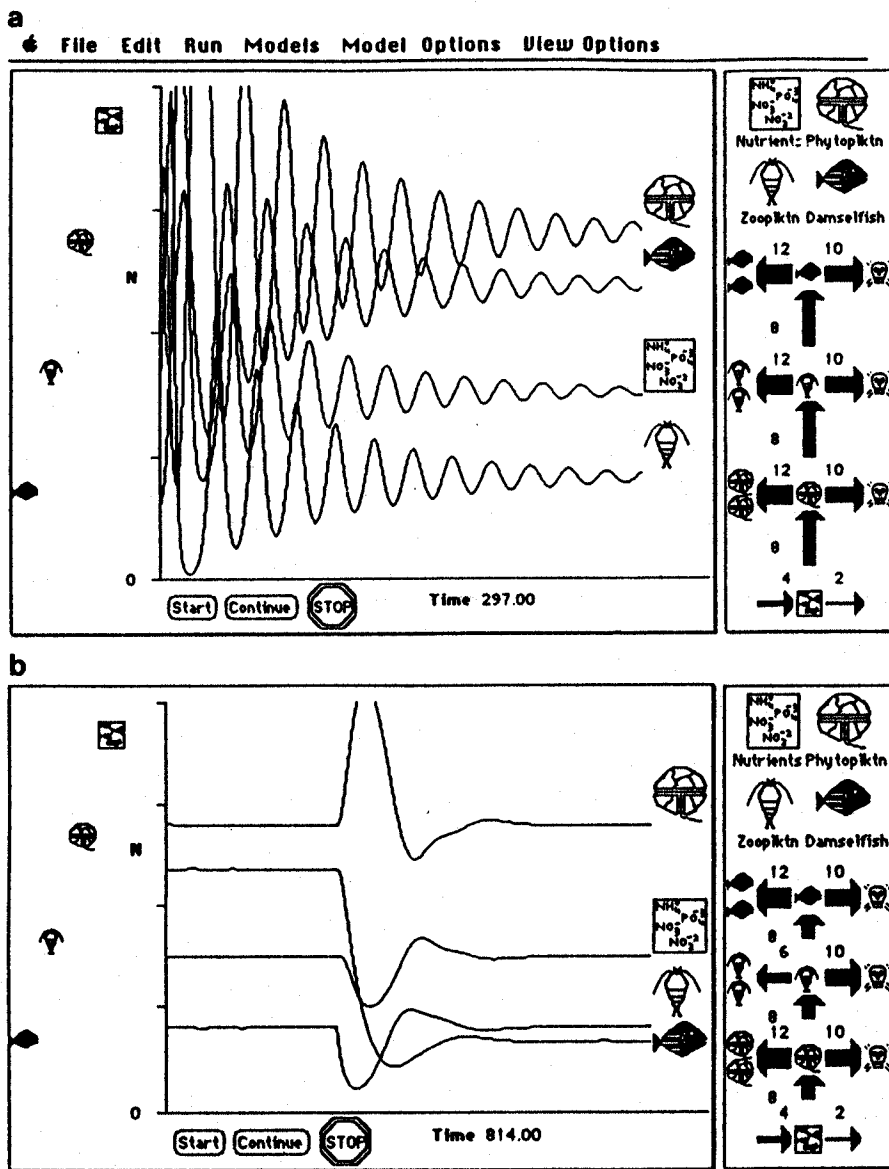


Figure 8. The food-chain model window. The controls for the window are described in the text. a. Sample simulation run for 297 time steps with the parameters set at the default conditions. Note that all population trajectories are damping. b. Continuation of the simulation in a, but once the populations had reached a stable equilibrium (first third of plot), the reproductive rate for the zooplankton was halved. This occurs where the population trajectories suddenly diverge.

increase its consumption of prey linearly with an increase in prey density, as assumed in the Lotka-Volterra models. At some point, the predator should become satiated such that the number of prey consumed per unit time remains constant, even though prey density increases. In addition, it is unlikely that if there were no predators the prey would increase without bound. Both the density dependence of the prey and the functional response of the predator can be incorporated into the predator-prey model

by choosing the type II or type III functional-response options. A sample predator-prey screen with a type II functional response is shown in Figure 7. The separate panels illustrate the change in the dynamic behavior of this interaction when the capture efficiency of the predator on the prey is increased.

Food-chain model

The predator-prey models are extremely simplified representations of

nature; two species would rarely interact in isolation from other organisms. The next level of complexity is added to these models in the food-chain model. The food chain model extends the simple Lotka-Volterra predator-prey model to several trophic levels, which increases the biological realism but also complicates intuitive understanding of the system. The effects of parameter changes or perturbations in the simple predator-prey models are fairly easy to predict, yet how those effects are translated through a food chain is often not quite so apparent. In addition, simultaneous changes may occur at several trophic levels, which makes the task of formulating predictions even more difficult. The food-chain model allows one to experiment and test intuition on how various perturbations might affect the dynamic behavior and equilibrium values of the component species in the ecosystem.

An example of the food-chain window is shown in Figure 8. In this model, the user controls the complexity (number of trophic levels) of the ecosystem by turning icons on or off. Nutrients represent the base of the ecosystem (trophic level 1) and are always on. The higher trophic levels rely on the trophic level immediately below them for nutrition. The trophic level icons are located in the upper right and indicate which species are included in the simulation. When the icon is on, it appears black and white (or colored on color monitors); when it is off, it will appear gray. To toggle an icon on or off, simply move the pointer over the icon (a lightning bolt should appear) and click the mouse button once.

Below the four icons are smaller icons with several arrows representing the interactions within and between trophic levels. The vertical arrows represent the capture efficiency or rate of consumption of one trophic level by another. The arrows pointing left represent the reproductive rate (e.g., the number of fish produced per unit of zooplankton consumed). The arrows pointing right represent the density-independent mortality rates. All rates can be increased or decreased as described previously. The starting density of each population can be adjusted by moving the density icons to the left of the y-axis. Densi-

ties can also be changed during a simulation by clicking the stop button, moving the plotting density icons (the four icons inside the plot region in Figure 8a), and clicking the continue button. This feature enables the user to model disturbance, emigration, and/or immigration.

One may also isolate various components of the ecosystem and analyze the effects of parameter variation on these isolates while all other components are held constant. This is accomplished by locking resources at their current levels such that they never increase or decrease. This option may be helpful when analyzing how two higher trophic levels affect each other independent of lower trophic levels.

The food-chain model allows one to explore how ecological and/or evolutionary changes may affect the dynamics of a simple ecosystem. For example, Figure 8a shows the population trajectories that occur for the four trophic levels set at the default conditions. Notice that the oscillations are damping. If the simulation is allowed to run longer, each population attains an equilibrium, as shown in the left third of Figure 8b.

After the system attained an equilibrium, I stopped the simulation, halved the reproductive rate of the zooplankton, and continued the simulation (in Figure 8b this occurred where the trajectories suddenly diverge). Although one might have expected that halving the reproductive rate of the zooplankton would reduce its equilibrium population size, notice that after an initial decline it attains a population size similar to that before the parameter change. However, the reduction of the zooplankton reproductive rate had a much greater long-term effect on the fish and nutrients. The fish dropped to approximately a quarter of their previous numbers, and the nutrients declined by approximately a third. The value of this simulation is not the actual outcome; rather, it demonstrates the genre of unexpected results from altering parameters within even a simple ecosystem.

Another level of complexity and biological realism can be added by making some or all of the parameters stochastic. Figure 9 shows the sort of dialog box that appears when Stochastic Parameters is selected from

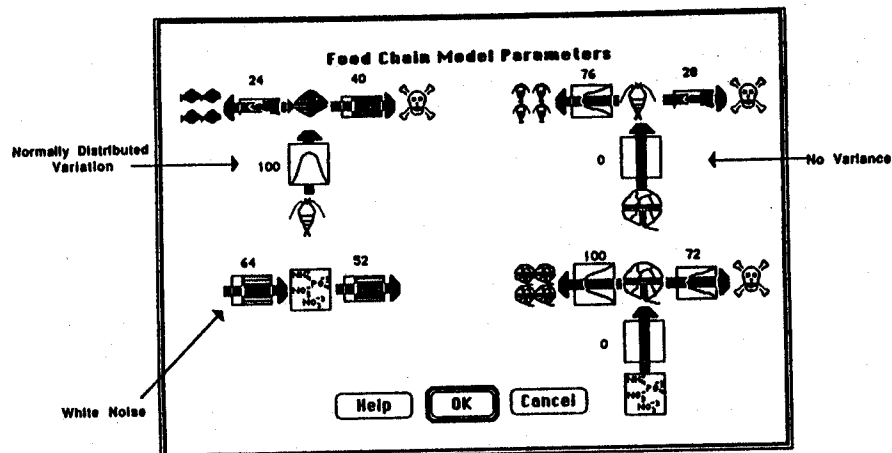


Figure 9. A dialog box is available in all models. This dialog box allows the user to set the magnitude and type of variance (normally distributed, white noise, or no variance) associated with each parameter of a model. To toggle between the different modes of variability (shown by the icons placed over the arrow representing the parameter), hold down the option key and click on the icon. The magnitude of the variance is set by changing the width of the icons with the mouse. The numbers adjacent to the variance icons indicate the magnitude of the variance as a percentage of the mean. If a parameter is defined to be variable, the actual parameter value used in each time step is drawn stochastically from an appropriate distribution with a mean set by the arrows in the model and a variance set by the width of the variance icons shown above. The example above is for the food chain model.

the options menu. The dialog box varies with the model and would appear initially with all variance icons as empty boxes, indicating all parameters are constant.

Figure 10 shows the population trajectories of the food-chain model with a small amount of variance in the mortality rate of the fish. The stochastic effect was implemented after the populations had attained an

equilibrium. The oscillations apparent in all trophic levels are driven totally by the small variance in the mortality rate of the fish.

Food-web model

The food-web model allows simultaneous interactions within and between trophic levels incorporating another level of complexity. In this

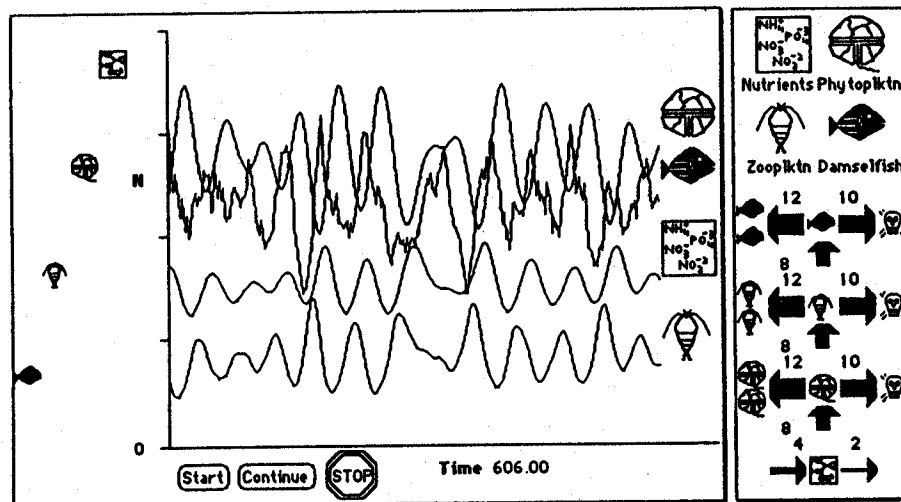


Figure 10. A simulation of the food chain model with parameters set to the default conditions. After the populations had reached an equilibrium (as shown in Figure 8a), the mortality rate of the damselfish was set to be stochastic with a variance 16% of the mean. The oscillations are totally driven by the small variability in the mortality rate of the damselfish.

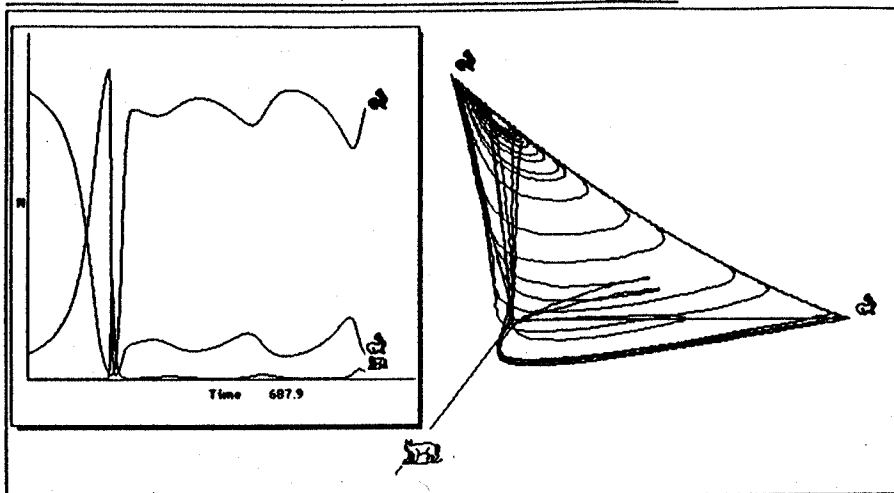


Figure 11. The N -versus-time (left) and phase (right) plots of a food-web model where a single predator (lynx) is preying on two prey (the light and dark hares). The prey are competing and the lynx has a preference for the light hares. As one changes the capture efficiency of the lynx on the preferred prey, the dynamics of these three species change from a stable equilibrium, to a damped oscillation, to a stable-limit cycle, to chaos. The system shown above is in the chaotic regime. On a color monitor, the attractor of the phase portrait is spectrally mapped such that the colors indicate the dynamic velocity of the system. The view of the 3D-phase portrait can be rotated around any of the three axes to better visualize the shape of the attractor.

model, there is a predator (the lynx) feeding on two prey (the light and dark hares) that compete for resources. The model is not completely implemented yet, but when finished it will enable users to alter all parameters of the three interacting species.

Figure 11 depicts a sample run of the food-web model. Both the N -versus-time plot (left) and the 3D-phase plot (right) are shown. Note that the 3D-phase portrait can be rotated around any of the three axes to better visualize the shape of the attractor. In this particular simulation, the light hare is the superior competitor but is also the preferred prey of the lynx. All other parameters are equal. As one increases the capture efficiency of the lynx on the preferred prey, the dynamic behavior of the system changes from a stable equilibrium, to a damped oscillation, to a stable limit cycle, and on to chaos. The simulation shown in Figure 11 is in the chaotic regime.

If a color monitor is used, the trajectory in phase space changes color according to the dynamic velocity of the system (the rate at which population size of the three species change:

$\sum dN_i/dt$, where i varies over the three species). The thermal properties of the colors indicate the dynamic velocity: red is high velocity, cyan is low velocity.

Spectrally mapping the dynamics in this manner provides some fascinating insights. The dynamic velocity of the system has important ramifications for interpreting the outcome of experiments designed to elucidate putative interactions among species. For example, to experimentally test for competition between the two hares, one would typically exclude one of the species from an area. The outcome of removing one of the hare species from the system will depend on what part of the attractor the system is traversing during the experiment. In the blue region, removing one of the hares would have little effect on the dynamics, whereas in the red regions the effects would be quite dramatic. Thus, if nature behaves in this manner (I do not claim that it does), it is important for ecologists to know where their system is on the attractor to properly interpret the outcome of their experiments.

Also note that the system spends

the majority of time in the blue and cyan region of the attractor. This condition simply reflects that the populations are changing slowly in these regions. If one needed to guess where on the attractor the system was, the best guess would be in these regions. Keep in mind that one loop around the attractor may take 2500 generations (i.e., iterations).

Conclusions

PopDyn is useful for dynamic demonstrations of important points in lecture, but a better use of the program is to allow students to work with the models in a laboratory setting. I have written an exercise manual to complement the PopDyn program that forces students to think about particular ecological problems. The simulations, along with the exercises, enable students to explore how ecological systems respond to various perturbations and allows them to improve their ecological intuition. It is the hands-on experience that allows students to develop a better grasp of ecological concepts. Students can ask "what if" questions and immediately run a simulation to obtain an answer.

At Harvard University, we rented computers for a two-week period for the laboratories that dealt with PopDyn. We rented enough computers so that each group of three students had a Macintosh. Students were asked to solve a number of problems using PopDyn and their knowledge of ecology and evolution. In addition, Harvard has a computer lab filled with Macintoshes that students could use independently. The labs were very successful.

Several universities have used a prototype version of this program. The responses I have received indicate that students understood the concepts of competition, predation, and logistic growth better than any other concepts covered. In addition, many students stated that they enjoyed working with the models. □

Ron J. Etter is an assistant professor in the Biology Department, University of Massachusetts, Boston, MA 02125. © 1991 American Institute of Biological Sciences.