Chapter 2. Software: (preview draft)
Getting Started with Stella and Vensim

Stella and Vensim are icon-based programs to support the construction and testing of system dynamics models. I use these programs in my own teaching and research, and they are used throughout the book. You will see mostly Stella models in this chapter. The Vensim examples will reveal the close similarity in the programs. I recommend you get started by downloading the trial version of Stella or the learning version of Vensim. They both provide good introductions, many examples and extensive documentation. Some of my students use both programs; others pick one or the other based on their personal situation and their research interests. You’ll see a mix of Stella and Vensim models throughout the book. The book’s website (BWeb) provides additional models using both programs, and it explains my thinking behind the selection of Stella or Vensim for the cases in this book. If you are new to system dynamics, don’t worry about which to choose. They are both excellent programs to support dynamic modeling.

Getting Started: Read and Verify

This chapter is written in work-book style, as if you are working along on your own computer. A good way to practice is to read about each model and immediately verify that you can reproduce the results on your own computer. Each model is described in a step-by-step manner as if you are following along, executing each step with your copy of the software. (As you do so, remember that these results are from ver 9.0.3 of Stella and ver. 5.3 of Vensim’s PLE, the Personal Learning Edition.) This chapter is written for readers who wish to learn how to build and test models. If you are reading this book for general ideas and concepts, skip over the step-by-step instructions. (The figures will give a general impression of what can be accomplished with the software.)

Exponential Growth in Population

Let’s start with an example of exponential growth. Imagine that you have drawn a reference mode similar to the exponential growth sketch in the previous chapter. Population is on the vertical axis, and time is on the horizontal axis. The current population is 100 million, and the population has been growing at 7%/year in recent years. Suppose that problems are anticipated when the population reaches 800 million, and we want the model to help us anticipate when that will occur.

Fig. 2.1 shows a map of a Stella model to simulate the population growth. The population is a stock, shown by the rectangle in the diagram. The births is a flow, shown by the double arrow with a valve. This model has one stock and one flow. Stocks and flows are the building blocks of system dynamics, so the software puts them in the first two positions on the panel of icons.
Now, imagine you are facing a blank screen and wondering how to start. The best place to start is with a stock. For this example, we will use a stock to represent the population. To create Fig. 2.1, select a stock from the panel; place it on the screen and give it the name population. Then select the flow icon from the panel; click below the stock to establish the cloud; and drag the icon toward the stock until Stella recognizes the connection. Name the flow births, and you have a map of your first model. If you wish to save your work, click on the file command and drag down to save. Stella will ask for a name and assign the STM suffix. This model has been saved as Exp Growth #1.STM.

![Fig. 2.1 Map of the first model.](image)

![Fig. 2.2. Click on the model tab.](image)

![Fig. 2.3. Need one more equation.](image)

![Fig. 2.4. The model is ready to simulate.](image)

The map shows the variables and how they interconnect. To write equations, click on the model tab, and Stella will respond with Fig. 2.2. The question marks alert us to the variables that need an equation. Let’s start with the population. All we need to do is type in the initial value, so we enter 1000000000 since the initial population is 100 million.

But take a closer look at the numbers. Did you see that I typed too many zeroes? Such mistakes are common, but they can be avoided if we rethink the units. Let’s measure the population in millions of persons and enter the initial value at 100. Close the population equation box, and Stella will show Fig. 2.3 with one question mark remaining. We could click on births and set the value to 7 if we expect births to be 7 million persons/year. Fig. 2.4 shows that the question marks are gone. The model is ready to simulate. To specify the length of the simulation, go to the run command and drag down to run specs. Time should be in years, starting with 0 and ending with 40. Click on run, and Stella will simulate the growth in the population for 40 years. Create a graph of the population, and you will see that it grows in linear fashion, increasing by 7 million per year or 70 million per decade. After 40 years, there will be 380 million persons, well below the 800 million that are said to pose problems.

If you get these results, congratulations on your first simulation. Unfortunately, the model did not deliver the intended results. It gives linear growth, but we are looking for exponential growth. To get exponential growth, the population has to grow faster and faster over time. This can happen if the number of births is proportional to the total population. Let’s assume that this is true for our population, and the birth rate is 0.07 per year. But how do we introduce the birth rate into the model?
Chapter 2. Software

This is where Stella’s converter can help. It is the circle located next to the flow on the panel of icons. The converters’ main function is to help explain the flows. Click on the converter; place it next to the flow; and give it the name birth rate. Then select the red arrow (the action connector) and click on the birth rate. Stella will establish a starting location with a small circle. Drag the arrow to the births and click again when Stella recognizes the connection. Stella will respond with Fig 2.5 which indicates that it needs an equation for the births and an equation for the birth rate. The birth rate can be set to 0.07, but what should we write for the births? They depend on the birth rate and on the population. This means we need another connector, as shown in Fig. 2.6. We now open the flow to write the equation for births.

Some students type in 7 since 7%/yr of 100 million is 7 million per year. If you give this a try, Stella will complain about unused inputs. It sees the action connectors in Fig. 2.6, and it wants your equation for births to use the connected variables. Stella doesn’t care how you use the two variables, but you know that the births is the product of the population and the birth rate. Enter this equation, and Stella will respond with Fig. 2.7. The model is now ready to simulate.

Before you click the run command, take the time to write down your guess as to the population at the end of the 40 year simulation. (If you don’t know how to guess, turn to appendix B.) If you do recall the doubling time rule, you’ll guess that the population will double to 200 million in the first decade, to 400 million in the second decade and to 800 million in the third decade. With no limits on the births, the population continues to grow, reaching 1,600 million by the end of the simulation.

To see these results, click on the graph pad icon and place it on the diagram. You’ll then select variables to be graphed and set the scales for best display of results. In Fig. 2.8, I selected the population and asked for the vertical axis to be scaled from 0 to 1600.

Figs. 2.9 and 2.10 show the corresponding images of the same model in Vensim. The first figure is a map or diagram of the model. The second figure highlights the
variables whose equations need to be written. The file name is Exp Growth #1.mdl, shown in the top panel next to the Vensim icon. Vensim’s box variable is used for the population. Vensim calls this a level, which is synonymous with a stock. The births add to the size of the population. The icon is a double arrow whose value is depicted by a butterfly valve. Vensim calls this a rate, a term which is synonymous with flow. The births depend on the population and the birth rate, as you can tell from the arrows.

Fig. 2.9. Vensim diagram of the population model.

Fig. 2.10. Equations are needed before we simulate.

The birth rate is entered by clicking on the VAB icon with a pencil. Vensim calls this a variable-auxiliary/constant. To write the equations, click on the Y=X^2 icon, and Vensim will respond with Fig. 2.10 which shows us the highlighted variables that need equations. Set the initial value of the population to 100 and the births to be the product of the population and the birth rate. Set the birth rate to 0.07, and Vensim will respond with a diagram free of highlighted variables. The model is ready to simulate. Go to the model command and drag down to settings, and ask for time bounds. Set time to be in years with an initial time of 0 and a final value of 40. Simulate the model with the model-simulate command. Vensim will ask if you want to assign a name to the data set or if you want to stick with the name current. If you agree, the results will be stored in a file named current.vdf. Give this a try and your simulation should show the population reaching 800 million by the 30th year and around 1,600 million by the end of the simulation.

Population Growth with Births and Deaths

The first model simulated the cumulative effect of births, but it ignored deaths. Suppose we are told that the death rate is 2% per year. If the population has been growing at 7% yr, we would suspect the birth rate is 9% per year. Fig. 2.11 shows a new version of the model with deaths, a flow which drains the stock of population. To add the flow, select the flow icon, click on the stock to establish the origin of the flow; drag to the right and release the flow. The new flow will be depicted as removing population from the stock and headed to a cloud. This cloud is sometimes called a sink. (The cloud on left side of the births flow is called a source.)
Fig. 2.11 shows the new model with the simulation results in view. The graph is identical to the graph shown previously. To create this image, place the graph below the diagram and click the push pin in the upper left corner. This pins the graph to the diagram, and it will be printed when you print the diagram. The graph confirms the results from before: the population reaches 800 million in 30 years and 1,600 million in 40 years.

Using Converters to Help Understand the Flows

The best way to build a model is to start with the stocks, add the flows and then use converters to explain the flows. In the previous model, we relied on a birth rate to explain the births and a death rate to explain the deaths. But we do not have to limit ourselves to a single converter to explain each flow. If we add more converters, a clearer picture of the flows may emerge. Fig. 2.12 illustrates with additional converters. First, notice that the deaths no longer depend on the death rate which was 0.02/year in the previous model. This could correspond to an average life time of 50 years, and it might be easier for people to understand if the model used the average life time. The equation for deaths will be the population divided by the average life time.

The main changes are the extra converters to help us explain the births. Let’s assume that the fraction female is 0.5 and 36% of the females are biologically mature. And finally, let’s assume that the mature females give birth every other year. This means that the births per mature female per year is 0.5. If you work through the numbers, you’ll
see that this combination of assumptions corresponds to a birth rate of 9%/year, the same value as in the previous model. The new model gives the correct results, and it provides a fuller explanation of the reasons for the exponential growth in population.

**Similarity in System Structure**

A human population can grow exponentially over time because a larger population leads to more births and more births lead to a larger population. Other systems exhibit exponential growth for similar reasons. The balance in your bank account will grow exponentially when the interest added increases the balance and that higher balance leads to still more interest added in the future. Fig. 2.13 shows a model to see if the balance will grow in exponential fashion. Except for the names, the model is identical to the population model in Fig. 2.7. Indeed, if you constructed the previous model, all you need to do is change the names and you will be ready to simulate the bank balance model. It starts with the balance at 100 and the interest added is the balance multiplied by the interest rate. I’ve selected the table pad so we can see numerical results from this simulation. The table is designed to report results every five years with the balance rounded to the nearest dollar. The tabular results confirm that the balance would double every decade and reach just over $1,600 by the end of the simulation.

**Exponential Decay**

Exponential decay is the second of the fundamental dynamics discussed in the previous chapter. It can appear when the rate of decay exceeds the rate of growth. The population model will exhibit exponential decay if the death rate exceeds the birth rate. Let’s simulate the model in Fig. 2.11 with an unusually high death rate of 16%/year. If we leave the birth rate at 9%/year, we would expect the population to decline at 7%/year. You know from appendix B that the half-life for this decay is ten years. So we expect to see the population fall by 50% every decade. Fig. 2.14 shows a time graph of the population with the vertical axis scaled from 0 to 100. It confirms the expected pattern of decay. For example, there are 50 million people after the first decade and 25 million after the second decade.
Sensitivity Analysis

To learn more about the importance of the death rate, we can conduct several simulations with different values. Such simulations are called sensitivity analysis because they teach us if the overall results are sensitive to changes in this parameter. To illustrate, let’s create simulations with the death rate set to 9%/year, 12.5%/year, and 16%/year. Go to Stella’s run command and drag down to sensi specs. Select the death rate, and ask for three simulations with incremental variations from a low of 0.09 to a high of 0.16. Then go to the run command and drag down to S-Run. Fig. 2.15 shows the results in a “comparative” graph. The first simulation assumes annual deaths of 9%. This is the same as the birth rate, so the population remains constant at 100 million. The second simulation uses 12.5% for the annual deaths, so we expect the population to decline at 3.5%/yr. You know from appendix B that the half-life will be 20 years. Fig. 2.15 confirms that the population falls to 50 million in the first two decades and then to 25 million in the next two decades. The third simulation uses a death rate of 16%/yr, the value simulated previously.

Fig. 2.15. Comparative graph of population.

Fig. 2.16 shows the corresponding results of three Vensim simulations of the population model. Vensim allows us to name the results. Rather than sticking with the default name (current), I have assigned the names 1st run, 2nd run and 3rd run. The three values of the death rate are the same as before. A new graph has been created to show the three results with the vertical axis scaled from 0 to 100. I asked for thick lines (lineW = 2) and for 4 divisions on both axes.

Fig. 2.16 Vensim analysis of the population decay.

Sensitivity analysis is a standard part of almost all modeling analyses. I recommend sensitivity analysis as the 7th of 8 steps in Table 1.2. Models are constructed and tested in an iterative fashion, so sensitivity analyses will be conducted many times.
during the modeling process. You’ll learn more about sensitivity analysis throughout the book, especially in chapter 21 and in appendix D.

**Nonlinear Relationships**

Environmental systems are highly nonlinear, and Stella and Vensim make it easy to include nonlinear relationships. To illustrate, let’s assume that the bank pays a higher interest rate if we have a higher balance in our account. The minimum rate is 4%/yr. But if we can build our balance, the interest rate will increase following the pattern in Fig. 2.17. Readers with a good command of algebra might go to work on a formula that will calculate the new interest rate as a function of the bank balance. But there is no need to devise a complicated formula. The better approach is to take advantage of Stella’s graphical function, as depicted in Fig. 2.18. It shows a model with the graphical function for the interest rate in view.

To bring this image onto your computer, click on the model tab and open the converter for interest rate. Stella wants you to use the bank balance, so click on the bank balance to make it appear in the equation window. Then click on the become graphical function button to see the window shown here. Ask for six data points and set the scale on the horizontal axis from 0 to 5000. Set the vertical scale from 0 to 0.08. Then enter six values of the interest rate to match the bank policy. The graphical function provides an opportunity for a visual check on our work by drawing the relationship as the values are entered.

Fig. 2.18 also shows two graph pads to the right of the model. The graphs are not open, so you can not see the results, but you can see that they have long names. Long names are useful as a reminder of the contents. The first graph shows the balance and the interest rate over time. The second shows a scatter plot of the interest rate versus the bank balance. These are left for you to create in the exercises for this chapter.
Conclusion

These examples will help you get started. We’ll return to Stella and Vensim in chapter 14. You’ll see that both programs provide many advanced features that have not been described here.

System dynamics models can also be constructed with Powersim. Like Stella and Vensim, Powersim was designed to provide a user-friendly, icon-based approach to modeling based on the principles first published by Forrester (1961). I am less familiar with Powersim, so I do not use it in the book. But my colleagues who use Powersim assure me that it is an equally powerful way to build and test system dynamics models of environmental systems. A variety of other programs are available to provide icon-based support of dynamic modeling (i.e., Simile, Simulink and GoldSim). These programs were not designed with the central focus on the system dynamics ideas (Forrester 1961). Nevertheless, they have many useful features in common with Stella and Vensim, and it is possible to build models with similar structure and similar results. These programs are described in appendix C.

Exercises

Exercise 2-1. Verify
Build the model in Fig. 2.18 with the new interest rate policy. Set the initial bank balance to 500 and run the model for 80 years.

Document your results with a “time series” graph to match the results in Fig. 2.19.

Fig. 2.19. Results with the new policy on interest rate.

Exercise 2-2. Scatter Graph. Fig 2.20 shows a “scatter graph” to provide a check on the interest rate policy. Select the bank balance to be on the X axis and the interest rate to be on the Y axis. Ask for thick lines and set the X-Y scales to match Fig. 2.20. Run the model for 80 years, and the results should trace out a pattern of dots in the X-Y space. The dots will confirm that the interest rate is consistent with the bank’s policy shown in Fig. 2.17.

Fig. 2.20. Scatter graph.
Exercise 2-3. The Graph Pad: Fig. 2.18 shows two graph pads, each with their own name to remind us of their purpose. But Stella allows you to store many graphs on the same graph pad. (Think of a pad of paper, with a new graph on each sheet in the pad.) To learn about this feature, open the graph pad from exercise 2-1 and ask for a new page in the pad. Use the new page for the scatter graph in Fig. 2.20.

Exercise 2-4. Discontinuous Graph Function: The interest rate policy in Fig. 2.17 changes gradually with changes in the bank balance. But what if the interest rate were to change abruptly when the balance reaches certain thresholds? Click on the graph icon in the lower left corner of Fig. 2.18. (When you do, the interest rate changes abruptly at the $2,000 threshold and the $4,000 threshold.) Run the new model for 80 years and document your results with graphs that correspond to Figs. 2.19 and 2.20.

Exercise 2-5. Vensim Graphical Lookup: Fig. 2.21 shows how the nonlinear relationship for the interest rate policy would be implemented in Vensim. Vensim uses a separate variable to hold the values of the nonlinear relationship. These are assigned the type “lookup,” and I normally put lookup in their name to remind me of their purpose. The lookup is shown in Fig. 2.21 as it would appear if you click the “as graph” button. This window resembles the graphical function window in Fig. 2.18. You specify the scales for display on the two axes, and Vensim draws the graph as you make the entries. These entries are evenly spaced in this example, but you can define the lookup with uneven spacing on the horizontal axis. When the lookup is completed, open the interest rate to write the equation. Click on the lookup (in variables) and Vensim will place the name of the lookup at the start of the equation window. Then enter a left parenthesis, click on bank balance and enter a right parenthesis.

Fig. 2.21 Vensim’s graph lookup in view.
Exercise 2-6. Verify
Simulate the model from the previous exercise over 80 years and document your results with a custom graph shown in Fig. 2.22. Select the bank balance to be scaled from 0 to 16,000 and the interest rate to be scaled from 0 to 0.08. And to match the graph exactly, ask Vensim to use 4 divisions on both axes and to assign LineW = 2 to both curves.

![Graph showing bank balance and interest rate over 80 years.](image)

Fig. 2.22. Vensim results with a variable interest rate.

### When do I get to write code?

Joe raises his hand to ask about programming. His roommate told him that the top programmers earn big money if they can produce pages of code each day. Joe has done all the exercises so far, but he worries that he has not produced a single page of code.

This question comes up frequently when students have heard about writing code with programs such as Fortran. But system dynamics modeling is not about writing code. Indeed, when you talk to practitioners, they will explain that they spend most of their time away from the computer (i.e. talking with people about the nature of the dynamic problem). When they do go to the computer, they rely on Stella or Vensim to build and test their models. The equations provide the closest correspondence to the code written in programming languages. But Joe has done all the exercises so far, and he hasn’t seen a list of equations. That’s quite revealing, for it shows that we don’t need to see the equations to build and test models. But the equations do exist, as you can verify in the concluding exercises.

### Exercise 2-7. View the Stella Equations:

To see the equations in Stella, click on the equations tab (located just below the model tab). You’ll recognize the variable names, and you can guess that $t$ stands for time. (The $dt$ stands for the small step in time as we proceed through a numerical simulation, as explained in chapter 4.) Print the equations from exercise 2-1 to verify that they match Table 2.1.

```plaintext
bank_balance(t) = bank_balance(t - dt) + (interest_added) * dt
INIT bank_balance = 500
INFLOWS:
interest_added = bank_balance*interest_rate
interest_rate = GRAPH(bank_balance)
(0.00, 0.04), (1000, 0.04), (2000, 0.05), (3000, 0.05), (4000, 0.06), (5000, 0.06)
```

Table 2.1. Stella equations from exercise 2-1.

### Exercise 2-8. Experiment with a poorly formulated model:

Fig 2.23 shows a model with an initial population of 800 million people. The births are constant at 100 million/year. The deaths are constant at 200 million/year. Build this model and verify the results shown here. The population declines in a linear manner reaching zero by the end of the 8th year.
At this point, the population should become negative because the deaths is fixed at 200 million/year no matter what the size of the population. But the graph shows an abrupt change in deaths at the end of the 8th year. For some reason, the software decides to override our equation and changes the value to 100 million/year. The population then remains at zero for the rest of the simulation.

**Exercise 2-9. Turn off the non-negative option**

Click on the population stock and pay attention toggle switch for making “population” a non-negative stock. If you see a check mark, the software will not allow the stock to become negative. It does this by ignoring our equation for the outflow that is about to drive the stock below zero. At first glance, this might seem like a useful thing to do. After all, the population can not become negative, so isn’t it nice of the software developers to have written a rule to replace our equation. In my view, this is the last thing we need. Covering up bad results will simply make the modeling process more difficult. We need to see the bad results as soon as possible so we can reformulate the equation that is driving the stock negative. Click the non-negative toggle to turn off this option. Then rerun the model from Fig. 2.23. Are the bad results visible? What is the population at the end of the simulation?

**Exercise 2-10. Add a death rate.** Suppose the deaths are based on a death rate of 25%/year. This rate would create 100 million deaths per year at the start of the simulation, but deaths would decrease over time as there are fewer and fewer people remaining. Add a converter for the death rate to the model in Fig. 2.23 and set the deaths to the product of the death rate and the population. Simulate the model for 20 years. What is the population at the end of the simulation? Will you ever see a negative population in this model?