Introduction to Dragoon

# Training problem with one accumulator node

Dragoon is a computer program for creating models. A Dragoon model consists of a set of nodes (shaped like diamonds, circles and rectangles) connected by links. Each node represents both (a) a numeric quantity whose value may change over time, and (b) a mathematical function for calculating the value of that quantity. The links represent inputs. That is, when node *A* has a function that takes node B as an input, then there is a link from B to A. You can think of numbers flowing along the links. When a model is complete, all of the calculations are performed by Dragoon. You can see how each node’s value changes over time by clicking on the Show Graph button.

This kind of model is called a “system dynamics model” because it *represents how a system changes over time*, where the system is modeled (represented) by a set of quantities. System dynamics models are widely used in science, engineering, management, and government.

Dragoon has three types of nodes:

* If the value of quantity is constant and its numeric value is specified in the problem, then you used a *parameter* node for it.
* If you need to know the previous value of a quantity in order to calculate its current value, then you use an *accumulator* node to represent it.
  + In proper mathematical language, an accumulator is called an “integral function.” That is, it is a function and its calculation involves integration, which is an operation taught in calculus courses.
* If you can calculate the value of a quantity directly from the values of other quantities, and you don’t need to know the previous value of the quantity, then you use a *function* node to represent it.
  + In proper mathematical language, there is no short name for this kind of function, so it might be described as a “function without integration.”

Although it may sound simple to tell when to use each type of node, it is surprisingly subtle and difficult. Over the next few exercises, you will learn how to decide what quantities to include in your model and how to define nodes that represent the quantities and their interrelationships.

# Exercise: Jared’s weight

The purpose of this problem is to introduce you to the accumulator node and to the user interface for creating models in Dragoon.

In many advertisements for Subway restaurants, Jared is concerned about his weight. Suppose he starts at 200 pounds and loses 2.5 pounds a week. This is a very simple system, so let’s build a model of it.

Jared’s weight is a quantity that changes over time. It should be represented by a node. But what kind of node: Accumulator? Function? Parameter? In order to know how much Jared weighs next week, we need to know how much he weighed this week. Thus, we must use an *accumulator* node to represent Jared’s weight. An accumulator node is used whenever the calculation of the node’s next value depends on the node’s current value.

But what exactly is that calculation? Each week, the accumulator node’s value should have 2.5 subtracted from it, because Jared loses 2.5 pounds a week. That is, every week the calculation is:

Next value (Jared weight) = Jared weight – 2.5

In general, if you know how much a value *changes* each week, then you use an accumulator node. For instance, if you know that Jared’s weight *increases* each week by 1 pound, then you set up the accumulator to be:

Next value (Jared weight) = Jared weight + 1

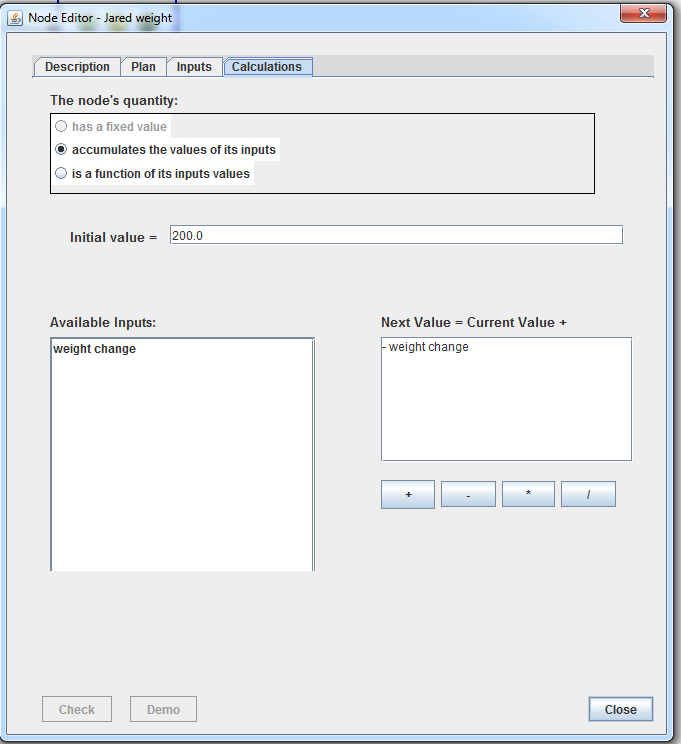
Thus, the increase or decrease of a quantity is determined by the sign of the associated constant.

Although we could write the equation for Jared’ weight using the number 2.5 as shown above, it is better to give such a constant a name, because that helps us remember what it represents. Named constants are called “parameters.” For instance, let us use weight change as a parameter name and the equations:

Weight change = 2.5

Next value (Jared weight) = Jared weight + weight change

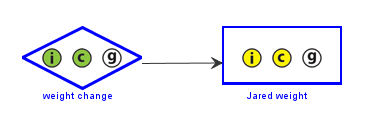
These equations let us calculate Jared’s weight, but only if we know this initial weight, which is 200 pounds. Notice that 200 does not appear in the equations above. Dragoon provides a special place for you to input the initial value of an accumulator. Here is the relevant tab:



You will type the initial value for Jared’s weight, 200, in one box. You will type an algebraic expression for the change in Jared’s weight, “- weight change”, into another box. Instead of typing, you can also click “- weight change” into the box using the mouse.

It might seem that this document should now describe every step for how to create a model of Jared’s weight using Dragoon. Such descriptions are tedious. Most students prefer to discover the steps themselves. So:

1. Go to dragoon.asu.edu and click on Introduction
2. Type in your name
3. Click on the button “Jared’s weight”
4. Create a model. (Hints: Start by clicking on the “Create Node” button; Double clicking on a node opens the node editor)
5. When your model is complete, it should look like this:

  
except that the colors inside the little circle may be different. Green means that you did it yourself, and yellow means that you asked Dragoon to demonstrate it for you. (That’s what the Demo button does). Also, you can move nodes by dragging them.

1. When your model is complete, click on the “Show Graph” node. This will execute the model (i.e., perform all the calculations) and display the resulting graphs. When you hover over the points on the graph you will get coordinates (values).
2. When you have studied the graphs, write answers to the questions below then close Dragoon.

Questions

1. What is the difference between the graphs of “weight change” and “Jared’s weight”?  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. What is the slope of this graph? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. According to this graph, what is will Jared’s weight be in 7 weeks? \_\_\_\_\_\_\_\_\_\_\_\_  
   In 10 weeks?\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. What might cause Jared’s weight to change? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. In your model of the Jared’s weight, why does the arrow (→) point towards “Jared’s weight”?  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. Why is “weight change” a diamond and “Jared’s weight “a square?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
7. What is the difference between an “accumulator” and a “parameter” node?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
8. The function for calculating Jared weight is linear. Circle either True or False.

# Training problem with one function node (addition)

In the preceding section, you learned about accumulator nodes and parameter nodes. In this section, you will learn about function nodes.

Let’s start with a system that we will model. Suppose Jared is a diet tester for Oprah Winfrey. He eats exactly what she eats, and does the same exercises too. He and Oprah compare weights each week, and it turns out that Jared is always 50 pounds heavier than Oprah. Thus, when Oprah weighs 120 pounds, then Jared weighs 170 pounds (170 is 120+50). When Oprah weighs 100 pounds, then Jared’s weight is 150 pounds.

Now in this case, we use a function node to represent Jared’s weight, and the calculation is (in units of pounds):

Jared’s weight = 50 + Oprah’s weight

In general, we use a function node whenever we can calculate a quantity *directly* in terms of other quantities.

In the preceding exercise, you were told how much Jared’s weight *changed* each week e.g., he lost 2.5 pounds a week. Here you are *not* told anything about how Jared’s weight *changes* each week; you are told what his weight *actually is* each week, namely, 50 + Oprah’s weight. So the difference between using an accumulator node and using a function node depends on whether you are told about the *change* in a quantity’s value or about the *actual value itself.*

So go ahead and make a model assuming Jared always weighs 50 pounds more than Oprah. Please return to the Introduction page of dragoon.asu.edu and click on the button “Jared & Oprah.” When you have completed your model and viewed the graphs, answer the questions below then close Dragoon:

Questions:

1. What is Oprah’s initial weight? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Why is Oprah’s weight represented by a parameter node?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. Why is Jared’s weight is represented by a function node? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Why do the arrows point toward Jared’s weight? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. What do you notice about the graphs? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. Why do these graphs not tell you if Jared loses or gains weight? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Training problem with one function and one accumulator.

In the preceding two exercises, you learned when to use an accumulator node and when to use a function node. That is, you should use an accumulator node when you’re told about the *change* in value, and you should use a function node when you’re told that *the value itself is a function of other values*. You should use a parameter node (diamond shape) when you are told a specific number for a quantity. These three kinds of nodes (accumulator, function and parameter value) are the only kinds. So you now you have experience with all three types of nodes.

In general, a model can have several accumulator nodes, several function nodes and several parameter value nodes. The more nodes a model has, the more confusing it can be to construct it. In the remaining exercises, you’ll learn how to create gradually more complex models.

In this exercise, you’ll create a model with four nodes:

* *Jared weight*: Jared always weighs a certain amount more than Oprah.
* *Jared extra weight*: This is 50 pounds, which is how much more he weighs than Oprah.
* *Oprah weight*: She starts at 140 pounds and decreases by a certain amount a week.
* *Oprah weight loss*: She loses 1.5 pounds a week.

In general, you should finish one node completely before starting to work on another node. For now, the software will require you to do this. Later you will have the freedom to edit nodes in any order you like. But it is still a good idea to complete one node before working on others.

Please return to the Introduction page and click on the button “Jared & Oprah +.” When you have completed your model and viewed the graphs, please answer these questions then close Dragoon:

**Questions**

1. Why is Oprah’s weight an accumulator and not a function? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Why is Jared’s weight a function and not an accumulator? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. What would happen to the graph of Jared’s weight is Oprah began to gain weight? \_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Training problem with accumulator & one function (multiplication)

The previous Jared and Oprah exercises used addition and subtraction, but not multiplication and division. This exercise introduces the use of multiplication in calculations.

Suppose that Jared always weighs 1.3 times as much as Oprah. That is, when she weighs 100 pounds, then Jared weighs 130 pounds. When Oprah weighs 120 pounds, then Jared weight 156 pounds (156 is 1.3\*120). Note: Dragoon uses \* to denote multiplication.

Next, you will build a model of this system. To save you work, the nodes for Oprah’s weight and weekly weight loss have been defined already. Thus you will only have to define two nodes:

* *Jared weight:* This is how much Jared weighs at the end of each week.
* *Jared extra weight:* This is a factor that is used to convert Oprah’s weight to Jared’s weight

Please return to the Introduction page and click on the button “Jared & Oprah \*.” When you have completed your model and viewed the graphs, please answer the questions below then close Dragoon.

Questions

1. The graph of Jared’s weight and Oprah’s weight is linear with a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ slope because they are both \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ weight.
2. Why was it necessary to use the \* operation in the calculation of Jared’s weight? \_\_\_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. How many inputs are there for Jared’s weight? List them here. Explain each one. \_\_\_\_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
     
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Exponential decay

This exercise introduces a new kind of complexity: a model with a loop in the links. The behavior displayed by this model is so common that it has a name: *exponential decay*.

This exercise has you create a model containing a loop. The model represents the extinction of the Dodo (pronounced “dough dough”), a bird that became extinct in the middle of the 17th century (<http://en.wikipedia.org/wiki/Dodo>). The model assumes that the Dodo became extinct because invasive species ate its eggs and thus prevented births. All the numbers are, however, made up just for this exercise. Here is a description of the system:

Suppose that 1000 Dodos were living on the island of Mauritius when sailors landed. The sailors accidentally introduced a new species, rats. The rats ate essentially all the Dodo eggs as soon as they were laid, so the Dodo birth rate dropped to zero. Meanwhile, the annual Dodo death rate continued to be 20%. That is, 20% of the Dodos died each year and no Dodos were born. Build a model and observe how the Dodo population changed over 20 years.

The key idea for this model is that the Dodo population changes each year due to the Dodo deaths, but the number of Dodo deaths also depends on the population. That’s a loop! More specifically, we need:

* A node, *population*, to represent the number of Dodos still alive on Mauritius. This decreases each year due to deaths.
* A node, *deaths*, to represent the number of Dodos that die each year. To calculate it, multiple death rate by population.
* A node, *death rate*, to represent the percentage of the Dodo population that die each year (Dragoon doesn’t understand percentages, so always use numbers instead e.g., use 0.20 instead of 20%).

Note that deaths is an input to population and population is an input to deaths. This is indicated by a double-headed arrow.

Please return to the Introduction page and click on the button “Dodo extinction.” When you have completed the model and studied the graphs, answer the questions below.

**True/False Questions (circle T or F):**

1. The graph for the Dodo’s population is a straight line, sloping downward. T F
2. The graph for the Dodo’s population is a curved line, sloping downward. T F
3. Around year 6, the Dodo’s population is 250. T F
4. The graph for the Dodo death rate is a curved line, sloping downward. T F
5. The best node type representing Dodo population is an accumulator. T F
6. One should use a function node to represent the Dodo deaths per year. T F
7. The death rate depends on the population, so it should be represented by a function node. T F

# Exponential growth

This section introduces another common behavior, called *exponential growth*. Like the preceding model (for Dodo population), the model that you will create in this section has a loop. By now, you have learned enough about Dragoon and modeling that you can construct the model without further help. Please return to the Introduction page and select the problem “Rabbit population.” When you have completed the model and studied the graphs, answer the questions below.

True/False Questions (circle T or F)

1. The rabbit population increases. T F
2. The graph of the rabbit population is a straight line sloping upwards. T F
3. The graph of rabbit births is a straight line sloping downwards. T F
4. The best node for representing rabbit deaths is an accumulator. T F
5. The best node for representing the rabbit population is an accumulator. T F