Introduction To Modeling

I. Introduction to the Modeling Process with STELLA

In order to illustrate some fundamental aspects of modeling with STELLA, we begin with a very simple system — a tub of water with a faucet and drain.

1. From Real World to Conceptual Model to Computer Model

The first step in modeling is to define and consider the system as it exists in the real world. This involves identifying the components of the system, the material or entity that is moving through the system, the major processes involved in moving this material or entity, and the other quantities that these processes depend on. This step necessarily involves simplifying the real world because it is obviously impossible to model the full complexity of nature. Drawing a cartoon of the system, as shown in Figure 1, is an important part of this process.

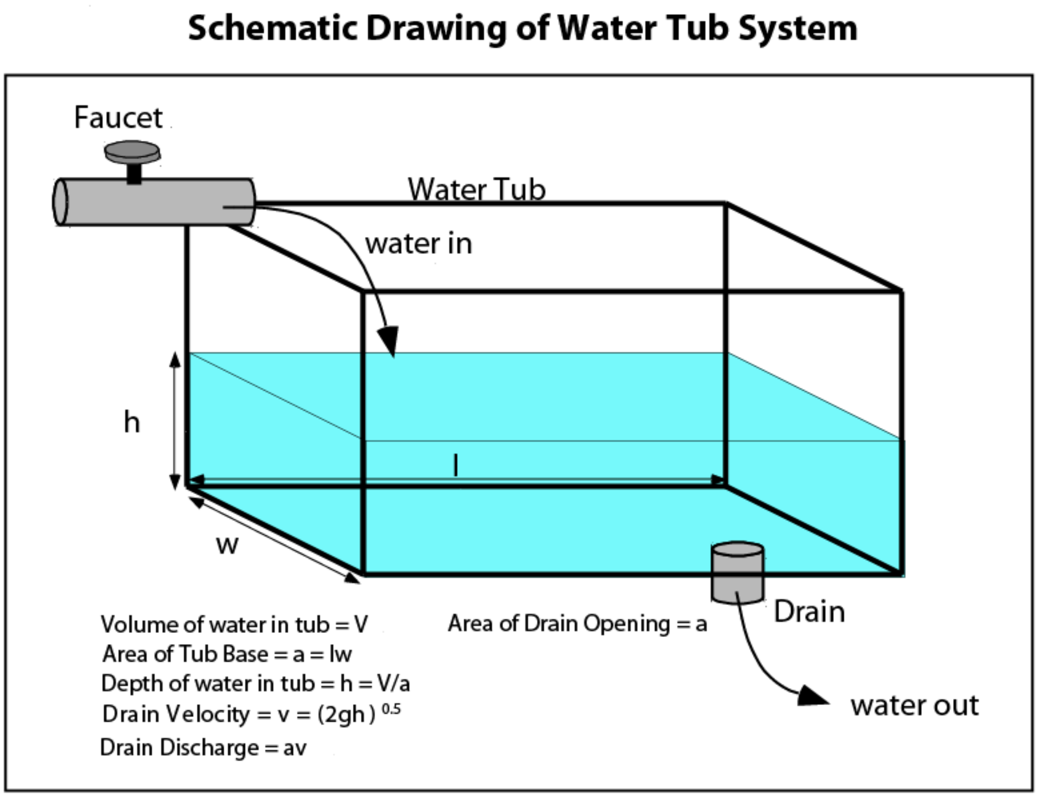


Figure 1. A simple sketch of the water tub system, consisting of a faucet, a drain, and tub that contains water. The faucet flow rate is independently controlled, but the rate of flow through the drain is a function of the water depth and the area of the drain opening.

The purpose of a schematic diagram like this is to clarify what we are modeling, the components of the system, and the relationships between these components. In this case, we are modeling the volume of water in the tub, which is a function of the amount added by the faucet and the amount removed through the drain. Here, we see that the faucet does not depend on anything else in the system — we can make it a constant, or we can make it something that changes over time in a specified manner. The flow of water out the drain, however, is dependent on the other parts of the system; namely the size of the drain (its area), and the depth of the water in the tub, which in turn depends on the volume of water divided by the area of the tub base.

A more precise description of this dependence of the drain flow on the water depth is provided by Torricelli's Law, which states that the velocity of water flowing out of a drain is given by:

$$v = \sqrt{2gh} \tag{1}$$

where v is velocity, g is gravity, and h is the depth. The velocity is then multiplied by the area of the drain opening to give a discharge in volume of water per unit of time.

Figure 2 shows how this system is represented in STELLA using the four building blocks of systems — reservoirs, flows, connectors, and converters.

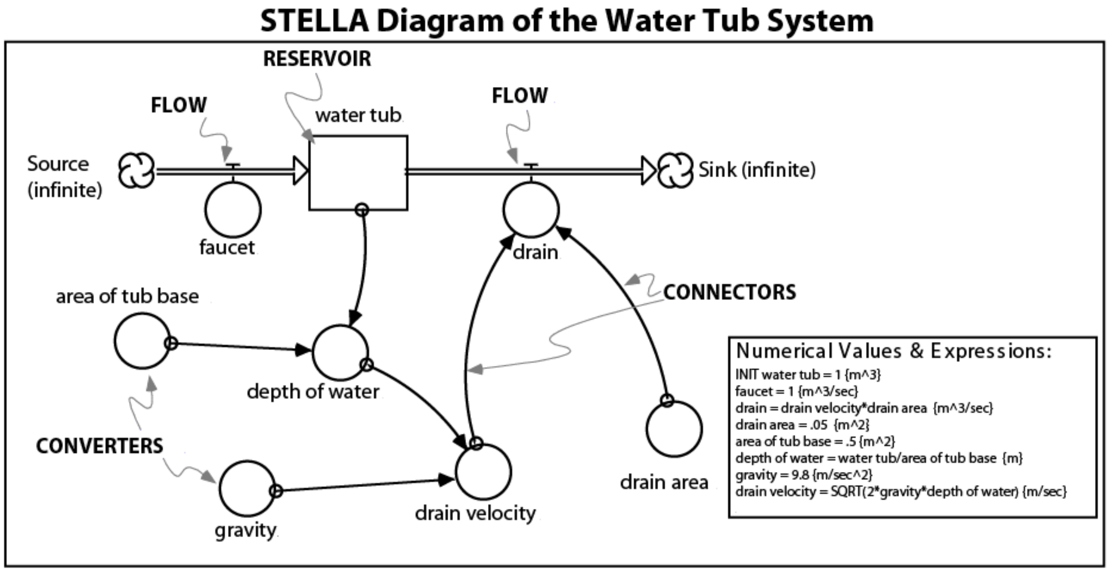


Figure 2. The water tub system as represented in STELLA. The reservoir, flows, and converters all have numerical values associated with them; hidden in this view, they can be seen and changed by double-clicking on each symbol. If no numerical value or expression is associated with a symbol, the program shows a question mark in that symbol. The numerical values and expressions used in this model are given in the box (INIT stands for the initial amount in the reservoir; comments enclosed in {} brackets are used to help keep track of units; note that the time units here are seconds.

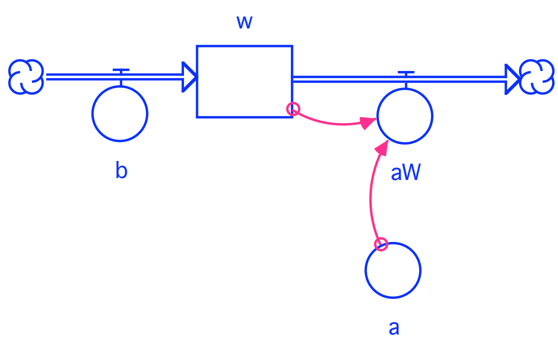
The connector arrows represent dependence; the depth of water is dependent on the amount of water in the tub and the area of the tub base, so connector arrows go from the water tub reservoir (the box) and the tub area converter (a circle). The converters represent either constants or variables defined by equations or graphs. Note that the two flows have cloud symbols at the ends away from the tub, indicating that this is an open system, drawing water from some unspecified source, and sending it to an unspecified sink at the other end.

2. Units

The next step is to figure out the units of the various system components. This is a very important step to take before we add numbers and make the model run — if we don't have the units figured out, we don't really know what is being calculated when the model is run. We start with the reservoir, which is a volume of water, which will have units of cubic meters [m^3]. This means that the flows both have to be expressed as cubic meters per unit of time, and we'll use seconds here [m^3/s] or [m^3s^{-1}]. Next, we have the converters to think about. The area of the tub base will have units of sqaure meters [m^2], and since the water depth is the volume [m^3] divided by the area [m^2], it will have units of meters [m]. Next, we have gravity, which is an acceleration with units of meters per second squared [m/s^2] or [ms^{-2}]. The velocity is calculated using the equation above, in which we take the square root of 2 times the gravity times the water depth, so the units here will be ([m/s^2] x [m])$^{-2}$, which is [m^2/s^2]$^{-2}$, which reduces to [m/s], the appropriate units for velocity. Next we have the area of the drain opening, which is in square meters [m^2], and when multiplied by the velocity [m/s], we have [m^3/s], the same units as the faucet. Thus we see that the units all work out, and we know we are dealing with cubic meters of water added to and removed from the tub on a timescale measured in seconds.

3. The Mathematics

To consider the mathematics, we will simplify the above model so that it looks like this:



The inflow, b, represents the faucet, while aW represents the drain. The faucet flow is a constant, while the drain flow is defined as the volume of water in the tub (W) times a rate constant. This system can be described by the following differential equation:

$$\frac{dW}{dt} = b - aW, \text{where } b = \text{faucet rate}, \text{and } aW = \text{drain rate}.$$

We also know the starting value, W_0, the amount of water in the tub at time = 0.

This is a classic type of first-order, linear differential equation and the general strategy for solving this is to separate the variables W and t and then integrate both sides — this will allow us to figure out W at any time t. If we do this separating of variables and integrate, the first equation we get is:

$$\int \frac{dW}{b - aW} = \int \mathrm{dt} \qquad (2)$$

The problem with this is that you can't solve these integrals, so we use a trick, which is to multiply both sides by $-a$, giving us:

$$\int \frac{-adW}{b - aW} = \int -adt \qquad (3).$$

This is a good trick because now the left-hand side of the equation has the form of:

$\int \frac{du}{u}$, where $u = b - aW$ and $du = -adW$

We then solve the integrals above to give:

$ln(\mathrm{b\text{-}aW}) = \mathrm{-at + C}.$

Applying the natural exponential function to both sides helps us get closer to our goal of isolating W on one side of the equation.

$$e^{\ln(b-aW)} = e^{-at+C} \text{ which can be simplified to: } b - aW = e^{-at+C}$$

From here, if we recall that $e^{x+y} = e^x e^y$, and since e^C is a constant, which we'll designate K, we can rewrite the right hand side of the last equation as Ke^{-at}. Next we need to find out what K is, and we get help if we look at the initial condition, at t=0, when $W = W_0$. Our basic equation then at t=0 would be:

$b - aW_0 = Ke^{-a0}$ and since $e^0 = 1$,

$b - aW_0 = K$ then if we substitute this into the general equation,

$b - aW = (b - aW_0)e^{-at}$ this can be rearranged to isolate W:

$$W(t) = \frac{b}{a} - \left(\frac{b}{a} - W_0\right)e^{-at}$$

This is our final equation for the water tub system — we sometimes cll this the ***analytical solution*** — it can be used to calculate the amount in the tub at any point in time. This analytical solution, done the old-fashioned way was not too hard to arrive at, but remember that this is a very simple system. As systems become more complex, the analytical solutions are increasingly difficult, and often impossible; this is when a computer is needed.

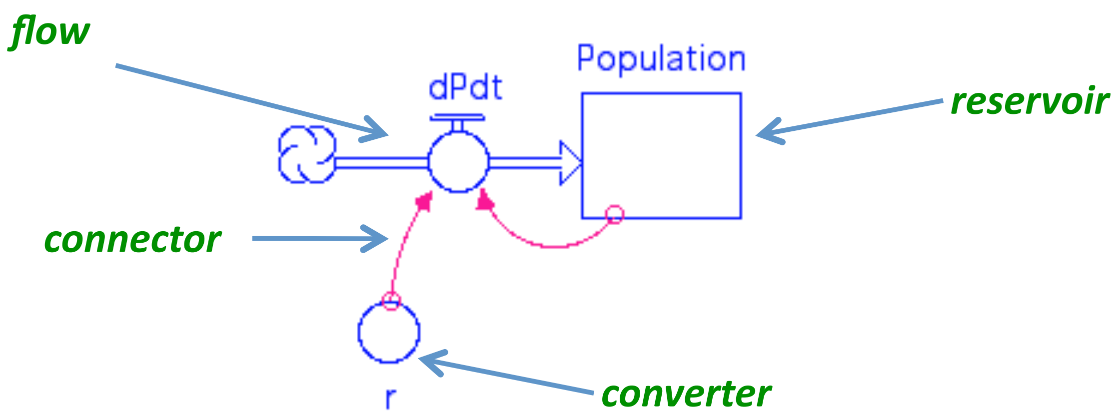
One more point about this system — we can just look at the equations that describe it and figure out what it will look like at steady state, when the inflow is equal to the outflow. In other words:

$$b = aW_{ss} \text{ or } W_{ss} = \frac{b}{a}$$

4. How STELLA Solves the Equations

Consider now an even simpler model, one of global population growth, which looks like this:

Global population model



In terms of an equation, this system is represented as:

flow reservoir

$$\frac{dP}{dt} = rP$$

converter

Now let's look at what STELLA does to solve this equation over time. It uses what is called a finite difference approach, in which it calculates the change over a short interval of time and then adds/subtracts the change from the previous value of the reservoir quantity. The reservoir quantity is thus a sum, or an integral over time. For the global population model, here is the basic information that STELLA has, and how it relates to the equations described above:

INIT Population = 6.7e9

Here we just specify the starting amount in the reservoir (units here are people) — we'll refer to this as P_0 in what follows.

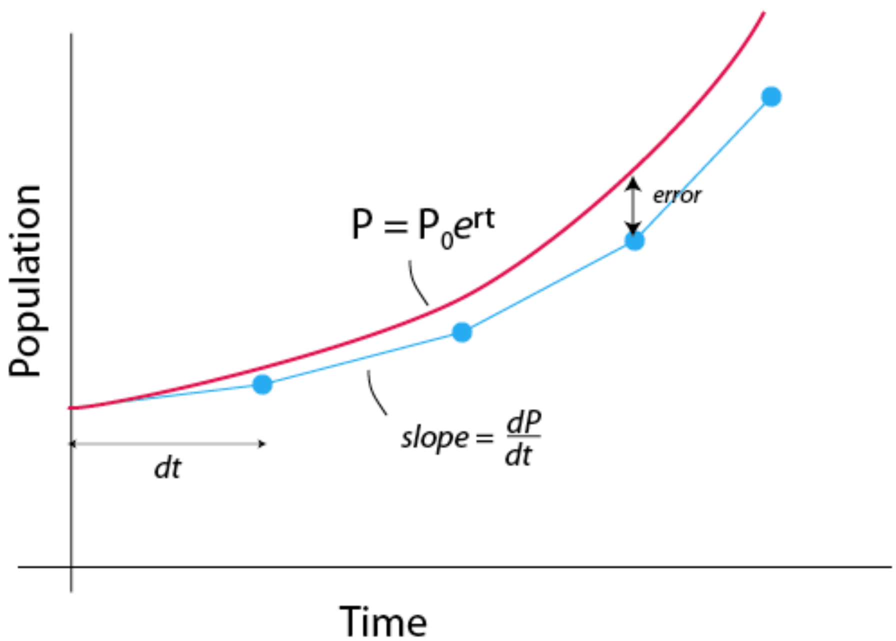
CONVERTER:
r = .0121 {constant annual growth rate = birth rate - death rate; units are per year or 1/t}

INFLOW:
dPdt = Population*r {units here are people/yr}

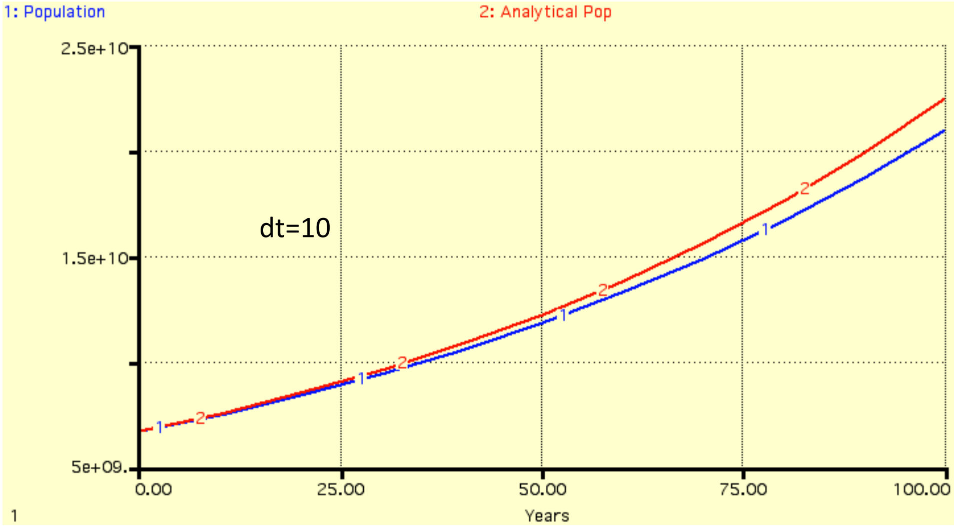
Population(t) = Population(t - dt) + (dP/dt) * dt

This says that the population at any time, *t*, is equal to the population in the previous time (***dt*** is the increment of time or *time step* between each calculation) plus the rate of change in the population (people/yr) times the increment of time (some fraction of a year). *This is the finite difference equation that gets solved again and again over the duration of the model.* In reality, this finite difference equation is just the first two terms in a Taylor series expansion. The other terms in the expansion are ignored, and this gives rise to errors in this mode of estimation, but those errors can be reduced by making the time step, ***dt***, very small.

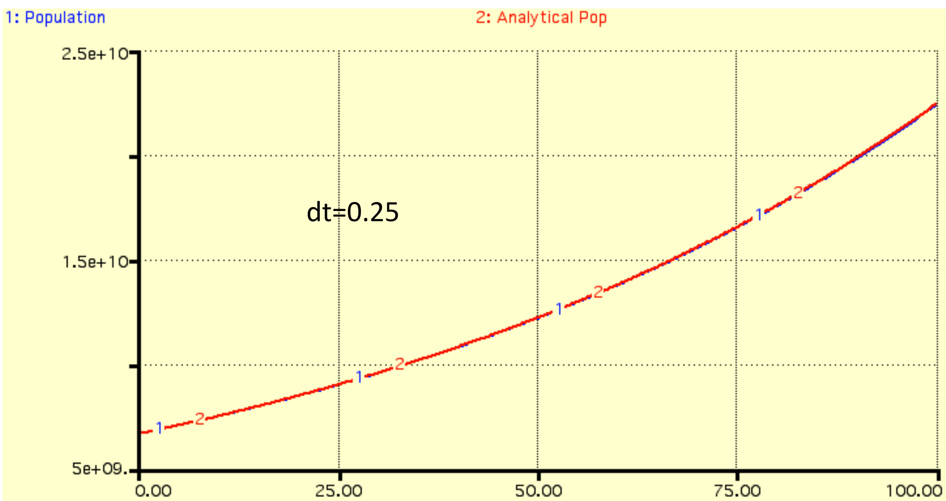
STELLA uses a variety of strategies, or algorithms, to solve this basic equation. The simplest one is called Euler's Method, and can be visualized in the figure below.



Here, the red line is the analytical solution to the equation, while the blue line is what gets calculated by Euler's Method, in which the computer approximates the exact solution by calculating the slope at the points in time indicated by the blue dots; these slopes are extrapolated for discrete units of time given by **dt**. As you can see, if **dt** is large, Euler's Method results in errors that tend to get bigger as time goes on. Below, we can see the actual calculation for the simple global population model described above, with a very large time step — **dt** = 10 years.



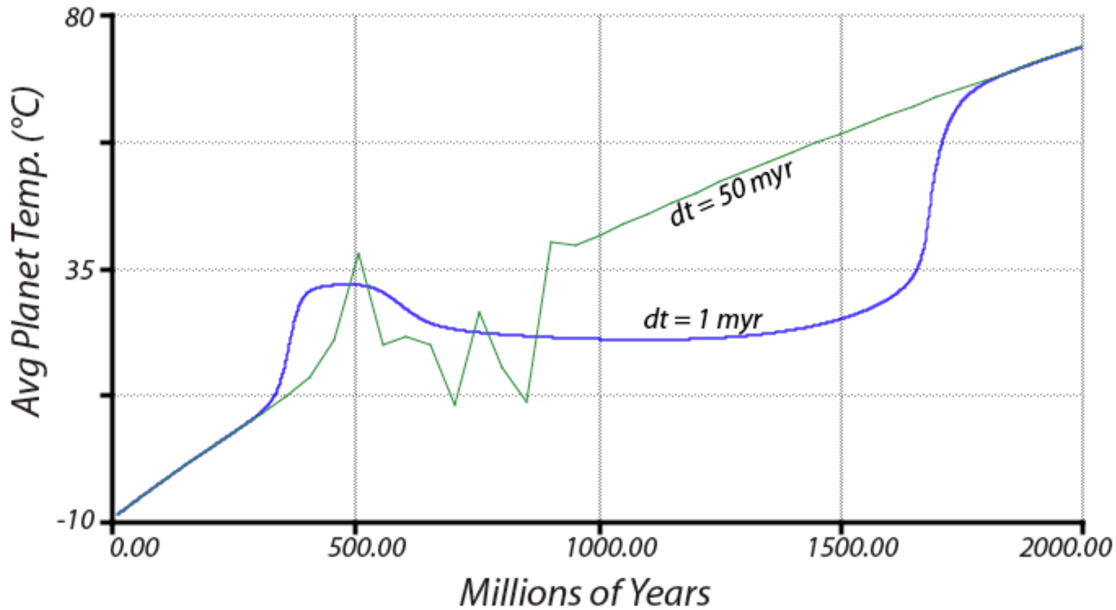
For comparison, see what happens as we make the time step much smaller — **dt** = 0.25 years. Now, the Euler's Method solution gives us a much better approximation to the exact solution.



In STELLA, you have the option of choosing two other methods of integration called Runge-Kutta 2 and Runge-Kutta 4; these methods both involve doing calculations at fractions of the time step (**dt**), as a way to avoid straying from the true solution. If you select one of the Runge-Kutta methods, the computation takes a bit longer, but you can get away with a longer time step.

5. Choosing the Right Time Step (dt)

You can clearly see the importance of picking the right time step in figures above, but it can get much worse. Take this example, which is from a climate model of an imaginary planet populated by daisies (Daisyworld), which we explore in more detail in a separate module.



The blue curve, with the time step of 1 million years is a good result, which we verify by reducing the time step to 0.5 and comparing the model output with that where the **dt** was 1; the model output in both cases is identical, and so we know that a time step of 1 is fine. But as we increase the time step, the model output begins to change, and as shown by the green curve, the model output begins to differ greatly from the shorter **dt** version. As a general rule, if the model output is full of abrupt, dramatic swings, then your time step is probably too large, so you need to reduce it by half until you stop seeing changes in the model output. But even if you don't see wild fluctuations in your model results, it is a good idea to experiment with shorter time steps just to be sure that you are not looking at some artifact of an inappropriately large time step.

II. Using the Model to Illustrate Systems Concepts

There are a number of important concepts connected to dynamic systems that can easily be illustrated with our model of the water tub.

1. Steady State

If we run the model for 100 seconds (see figure), we see some interesting changes take place as the system evolves towards a **steady state**, where the amount of water in the tub stays constant.

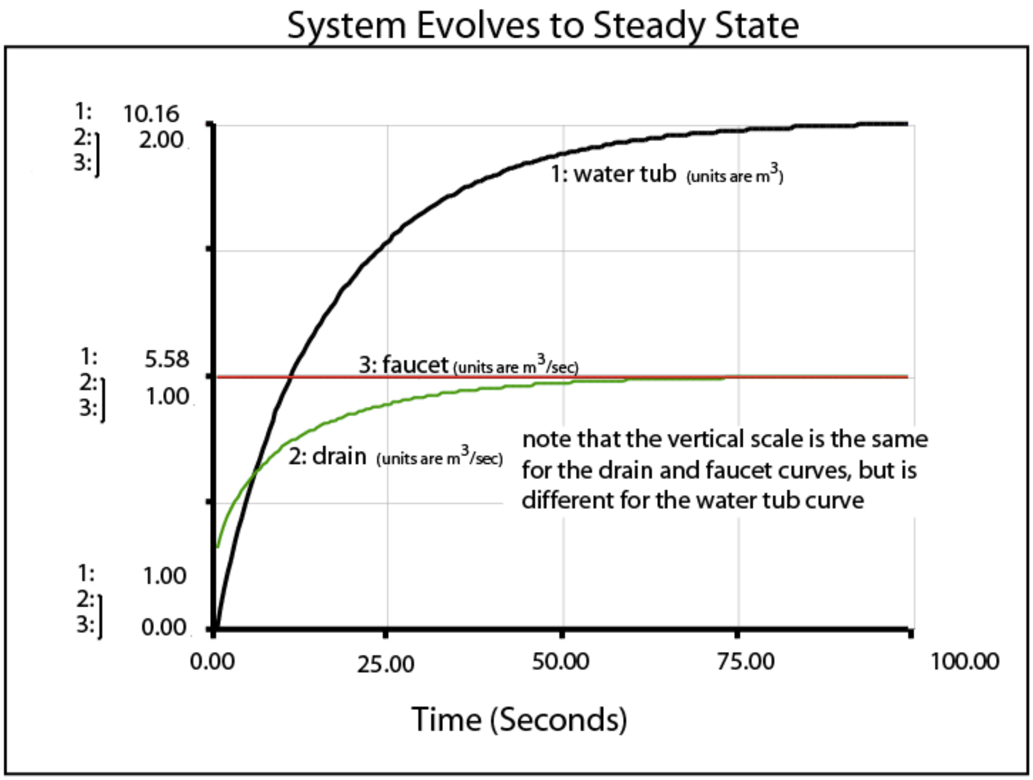


Figure 3. Results of running the water tub model for 100 seconds using the initial conditions given in Figure 2. After about 75 seconds, the system is in a steady state.

The inflow (faucet) stays constant over time, but the outflow (drain) undergoes a type of exponential change, increasing until it approaches the same value as the inflow. Initially, the inflow is greater than the outflow, and so the amount of water, and therefore the depth of the water in the tub increase, thus increasing the drain velocity and the outflow; this continues until the inflow and outflow match, at which point water continues to move through the system, but the amount of water in the tub remains the same.

2. Residence Time

When the system is in the steady state, we can define another concept — the **residence time**. The residence time is effectively the average length of time that an entity — in this case a water molecule — remains in a reservoir. It is really only meaningful for a reservoir that is at or near a steady state condition. By definition, the residence time is the amount of material in the reservoir, divided by either the inflow or the outflow (they are equal when the reservoir is at steady state). If there are multiple inflows or outflows, then we use the sum of the outflows or inflows to determine the residence time. For the water tub system shown here, the residence time is:

$$t_{residence} = \frac{\text{amount in tub}}{\text{outflow}} = \frac{10.16\ \text{m}^3}{1\ \text{m}^3\text{sec}^{-1}} = 10.16\ \text{sec}$$

If we increase the flow rates, the water moves through the reservoir faster, so the residence time decreases. It is possible then, to calculate any of the above three parameters (residence time, reservoir amount, and inflow or outflow) if the other two are known and if we assume the system is in a steady state. For instance, if we assume that the human population is in a steady state, and if we know the average residence time, also known as the life span, we can calculate the number of births and deaths in a year. The population is close to 6 billion, so if we assume an average life span of 70 years, then we can say that 85 million people are born each year and 85 million people die each year, assuming a steady state (which of course is wishful thinking). Residence time is an important concept in problems of pollutants in ground water or surface water reservoirs, and also in understanding the long-term effects of greenhouse gases added to the atmosphere.

3. Response Time

A closely related concept is that of the **response time** of a system, which measures how quickly a system recovers and returns to its steady state after some perturbation. We can illustrate this concept by running several simulations, where we vary the starting amount of water in the reservoir and pay attention to how quickly the system gets to its steady state. The results, shown below, are somewhat surprising; regardless of how great the initial departure from the ending steady state, this system gets to the steady state at about the same time.

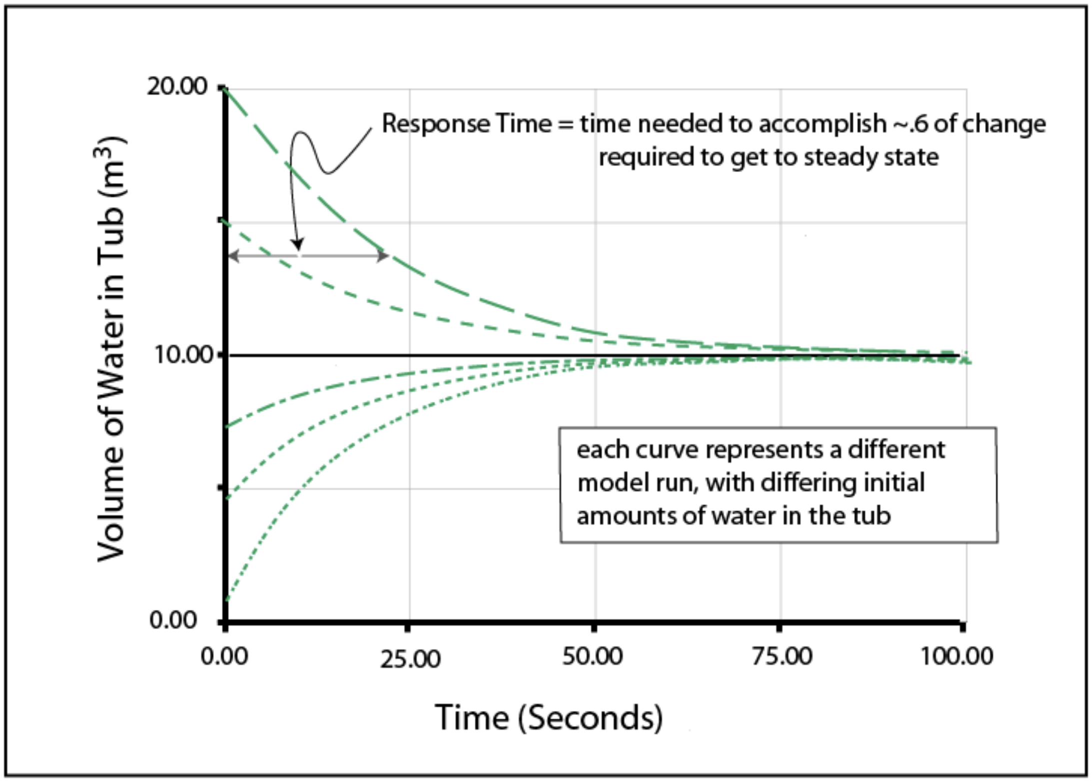


Figure 4. Results of running the water tub model 5 different times, varying the initial aount of water in the tub. In each case, the system returns to its steady state in about 75 seconds, with a response time of around 20-25 sseconds.

In an even simpler system, where the outflow (D) is defined as a simple fraction (k) of the amount in the reservoir (W) such that at each instant in time,

$$D = kW \qquad (4)$$

In this case, the response time is defined as $1/k$ — this turns out to be the time needed for the system to accomplish 63% of the return to its steady state. Also note that in a very simple system, the residence time and the response time are numerically the same even though they are conceptually different. The response time is a very useful concept because if it is known (or hypothesized), we can make some predictions about how quickly the system will respond to a change in a reservoir. Alternatively, if we change one of the inflow or outflow processes, we can predict how that will affect the response time of the system. The concept of response time is important in understanding the future of the global carbon cycle — if we halt the anthropogenic alterations to the carbon cycle, the response time of the system tells us how long it would take for the carbon cycle to return to a more natural state.

Another important observation to be made here is that the system evolves to the same steady state in each case, so the steady state of a system (along with the response time) is primarily determined by the nature of the inflows and outflows.

4. Feedback Mechanisms

This particular system returns to a steady state because it contains a **negative feedback mechanism** in the connection between the drain flow rate and the amount of water in the tub. ***A negative feedback mechanism is a controlling mechanism, one that tends to counteract some kind of initial imbalance or perturbation***. A familiar example of another negative feedback mechanism is a simple thermostat in a home that responds to changes from the steady state, returning the home to a specified temperature. Note that the word negative, as used here, does not mean that this is bad feedback; it just means that this feedback mechanism acts to reverse the change that set the feedback mechanism into operation. If our tub is in its steady state, knocking the system out of its steady state by suddenly dumping in more water will cause a response — the drain will increase its flow rate, thus decreasing the amount of water in the tub, bringing back towards the steady state value. If we instead decrease the amount in the tub, the negative feedback associated with the drain forces the amount of water in the tub to increase until the steady state is returned. The important thing to remember is that negative feedback mechanisms tend to have stabilizing effects on systems.

In contrast, ***a positive feedback mechanism is one that exacerbates some initial change from the steady state, leading to a runaway condition — it acts to promote an enhancement or amplification of the initial change.*** A simple way to modify the simple water tub system in order to create a positive feedback system is to alter the system as shown below.

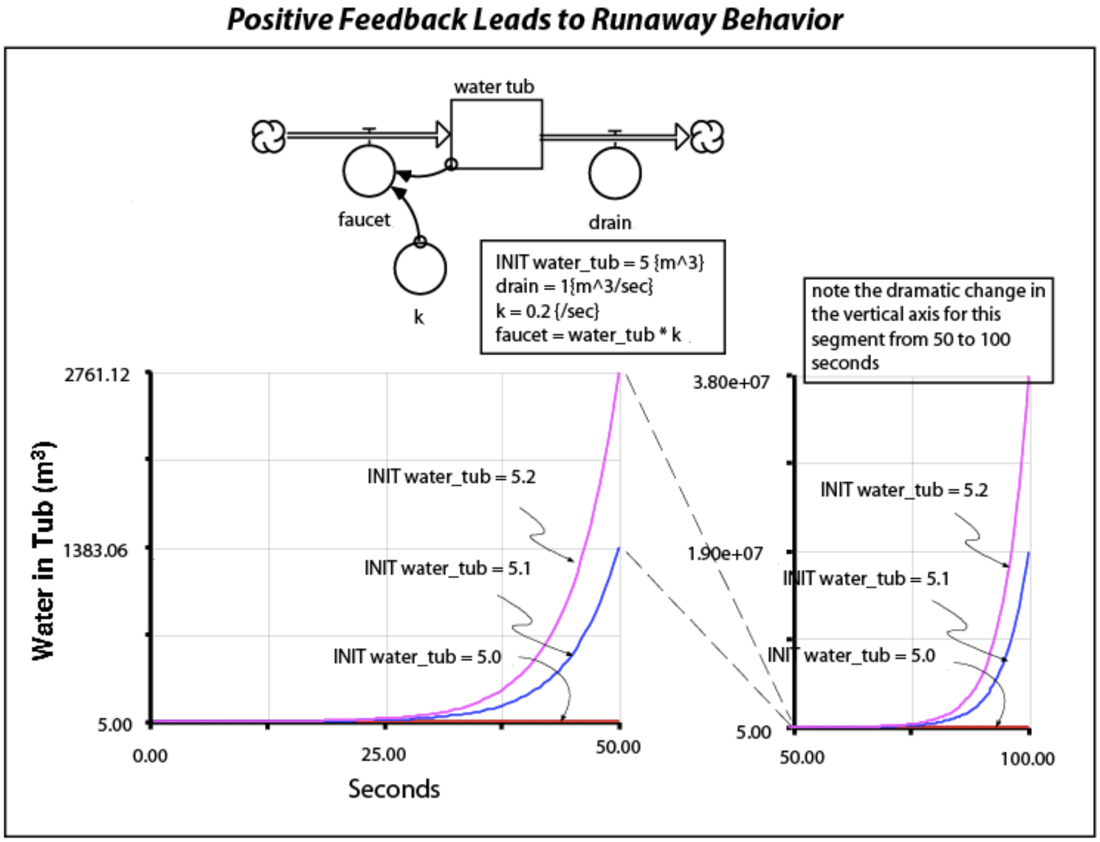


Figure 5. Modified water tub system to illustrate a positive feedback mechanism. In this case, the faucet is defined such that its rate of inflow increases as the amount in the tub increases, while the drain is defined as a constant (physically unreal, of course). The model was run 3 times with different initial amounts of water; initial values greater than 5.0 trigger the positive feedback mechanism, leading to runaway behavior that follows an exponential curve. Comparing the reservoir values at 50 seconds and 100 seconds shows the impressive increases that result from exponential growth — 38 million m^3 of water in the case where the initial value is 5.2 m^3.

This system has one possible steady state, where the initial amount in the water tub is 5 m^3; any departure from this value, however slight, leads to runaway behavior and the amount of water in the tub follows an exponential curve of the form:

$$W(t) = \frac{d}{k} + \left(W_0 - \frac{d}{k}\right)e^{kt},$$

where W(t) is the amount of water in the tub at any time, d is the drain constant (1 in our case), k is the faucet rate constant (0.2), W_0 is the initial amount of water in the tub (variable in the 3 model runs shown in Fig. 5), and t is time. A useful thing to remember with exponential growth is that the doubling time can be easily calculated (after some manipulations of the above equation):

$$t_{doubling} = \frac{.693}{k}.$$

Here, the doubling time works out to be about 3.5 seconds.

If you run this model with an initial water tub value of less than 5, you learn another important lesson, which is that in the real world, there are limits to exponential change, regardless of whether that change leads to growth or decline. In this case, the limit is reached when there is no more water left in the tub. In the case of population growth, the limit to exponential growth is reached when the carrying capacity of the ecosystem is approached — then the population growth decelerates and gradually approaches the carrying capacity (with reference to the human population, see Cohen, 1995, and Meadows et al., 1992 for detailed discussions).

Positive feedback mechanisms, like negative feedback mechanisms are not necessarily good or bad. Epidemics and infections have positive feedback mechanisms associated with them, but so does the growth of money in a bank account with compounded interest. The Earth contains a wide variety of both positive feedbacks and negative feedbacks and depending on the conditions, either kind of feedback may dominate. But — and this is very important — the mere fact that we exist, the fact that our planet has water and an atmosphere is compelling evidence to suggest that ultimately, our Earth system is dominated by negative feedback mechanisms (see Kasting, 1989 or Lovelock, 1988 for more discussion of this). However, it is equally important to realize that human time scales are much shorter than the history of the Earth and over periods of time that interest humans, positive feedback mechanisms may be very important; they have the potential to produce dramatic changes.

5. Lag Time

We next consider a slightly more complex system to illustrate the concept of a **lag time**. In Figure 6, two water tubs have been connected such that the drain from one flows into the adjacent tub. Here, the drains have been simplified greatly — all of the drain parameters in our first model are represented by a rate constants labeled k_1 and k_2; these rate constants get multiplied by the volume in the reservoir at any time to give the volumetric rate of flow out of the drain. Next, we take advantage of a useful feature of STELLA — the ability to define various parameters as graphical functions of other system variables or time. In this case, I want to show how the system responds to a sudden spike in the faucet flow rate, so I first define the faucet rate as being equal to time, then click on a button in the dialog box and a graph appears, enabling you to define the nature of this graphical relationship. The faucet in this case starts out with a value of 1 m^3/sec, which puts the system in a steady state to begin with, then increases to a peak value of 4 m^3/sec, and then quickly returns to 1 m^3/sec again and stays there for the duration of the experiment.

Lag Times Revealed in Response to a Perturbation

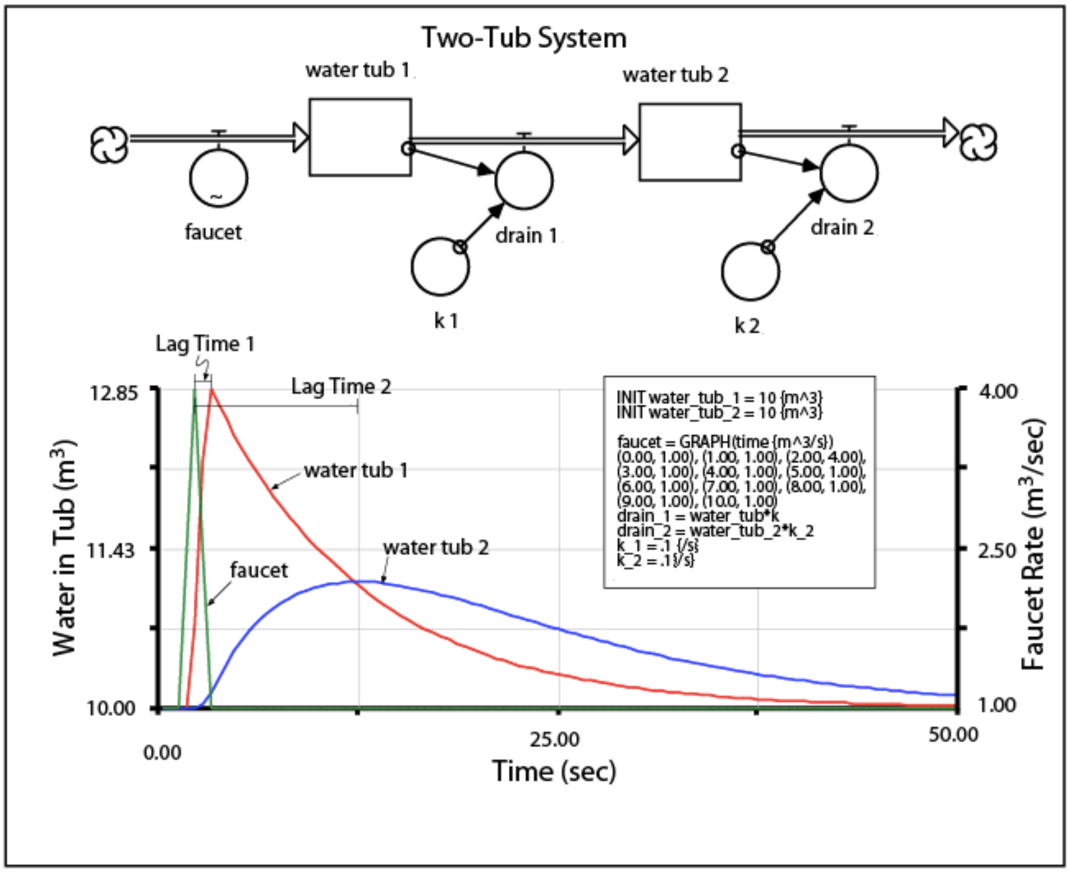


Figure 6. A system with two water tubs connected by a drain illustrates the concept of a lag time. Here, both tubs start out with the same amount and the drains have identical rate constants; the faucet is defined as a graphical function of time, starting at a value of 1 m^3/s, then jumping up to 4 m^3/s at time=2, then returning to 1 for the duration of the time. Water tub 1 peaks at a value of 12.85 m^3 at time=3, so this reservoir has a lag time of 1 second; tub 2 peaks much later, at time=12, so its lag time is 10 seconds behind the faucet peak, and 9 seconds behind the "upstream" reservoir.

The response of the system is shown in Figure 6. The first tub reaches its peak 1 second behind the peak in the faucet — it lags behind the faucet. The second tub peaks at a lower value and much later than the first — the perturbation is buffered by the first water tub. Note that both tubs return to their steady state after variable amounts of time and that the total area under the two curves is equal, meaning that the same volume of water moved through each reservoir. The pulse of extra water, propagating through the system is very similar to the pulse of water moving down a stream system, with the lag time in this analogy being the time between the peak in the rainfall and the flood peak along a certain reach of the stream. The concept of a lag time is also relevant to systems such as the global carbon cycle — anthropogenic additions of CO_2 into the atmosphere are similar to the faucet here; the climatic response involves a certain lag time. This lag time means that if we halt emissions today, the climate will continue to warm — a fact that many policy-makers and citizens should be aware of.

III. Model Simplicity vs. Complexity

Models are clearly meant to represent simplified versions of the real world, yet still complex enough capture the essence of the real system. So, what is too simple, and what is simple enough? One way to understand this is by way of three variations on the water tub model, shown in Figure 7. In the simplest version of this system, the faucet and drain flows are simple constants — they do not change over time. The more realistic model uses Torricelli's Law to express the rate of flow out of the drain. The intermediate model simply represents the drain flow as the product of a rate constant times the volume in the tub.

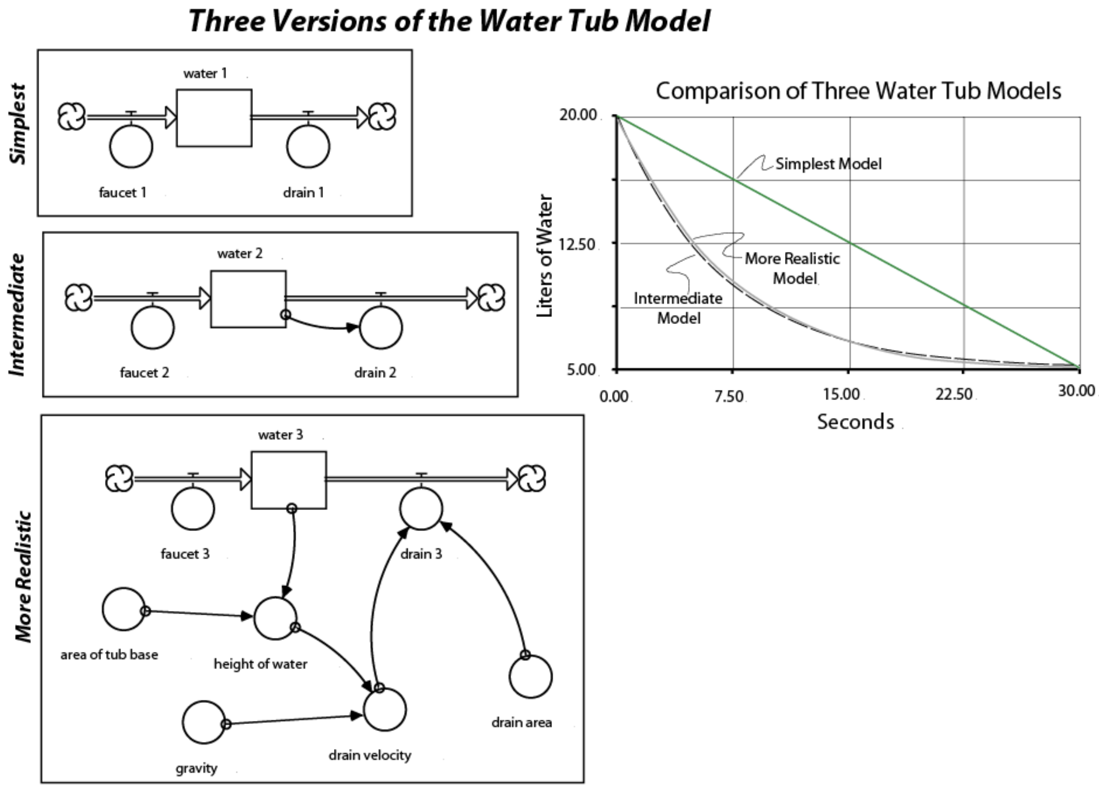


Figure 7. Three variations of the water tub model, with varying degrees of complexity. The simplest model has a constant drain, while in the more realistic model, the drain flow rate is calculated using Torricelli's Law. The model of intermediate complexity just represents that drain flow as a rate constant multiplied by the amount of water in the tub. comparing the models shows virtually no difference between the intermediate complexity model and the more realistic one. The intermediate model is thus an acceptably simple model — not so simple that it fails to capture the essence of the system.

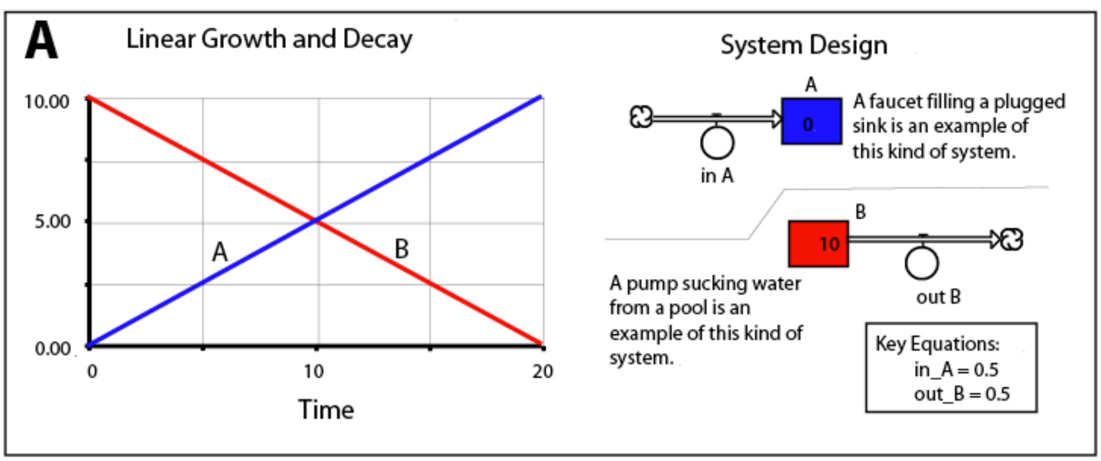
From the results (Fig. 7), it is clear that the simplest model does a poor job of representing the real behavior of this system; it drops off at a constant rate and does not achieve a steady state except in the special case where the inflow is set equal to the outflow. This simple system does not have any negative feedback associated with it either. So, this is a good example of a model that is too simple. In contrast, the intermediate model does a remarkably good job of matching the behavior of the more complex model. It is fair to say, then, that the intermediate model is simple enough and yet not too simple — it captures the essence of the more complex model using a more parsimonious mathematical representation. Nevertheless, the simple model is valuable as a starting point in the modeling process; its shortcomings provide suggestions for improvements and added complexity

It is worth noting that even the more complex model shown in Figure 7 is not overly complex. An overly complex model might, for instance, try to represent the friction, viscosity, and 3-D turbulent flow in the water, which would clearly represent overkill in this case.

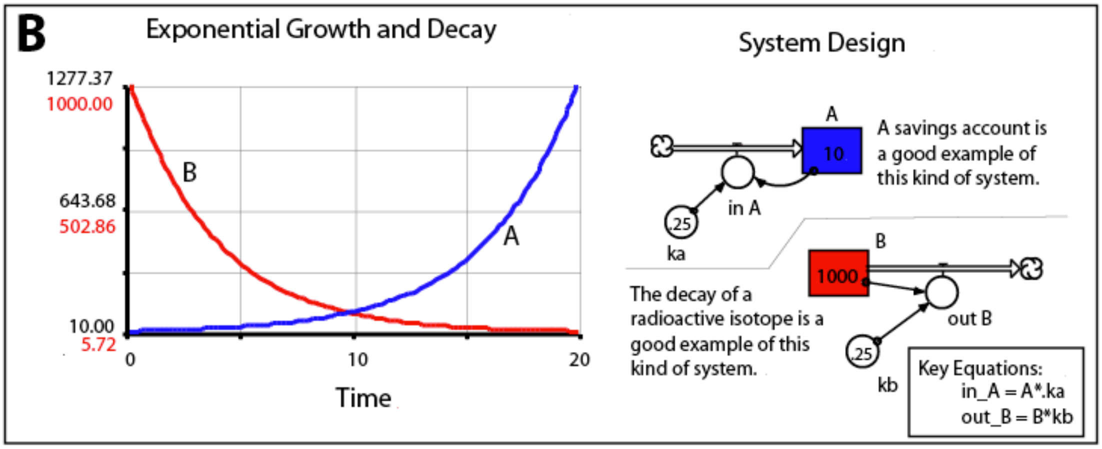
IV. Common System Designs and Behaviors

There is certainly a vast number of models of dynamics systems, but some common design elements or system structures are found in a great number of models. When we venture off into world of modeling, it will be helpful to be familiar with some of these common designs and it will be especially helpful develop a sense of what types of behaviors are associated with these designs.

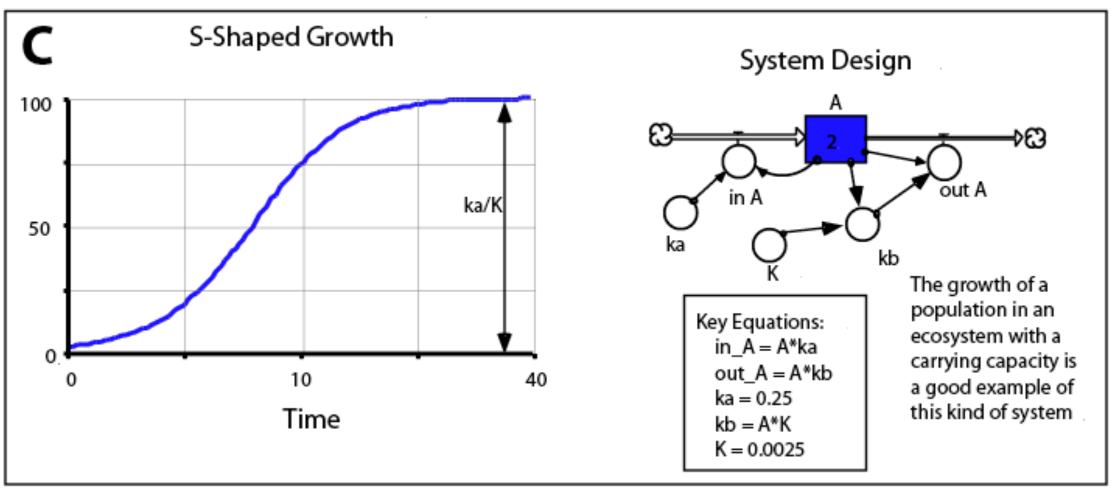
The series of figures below illustrate these common system designs in the form of very simple STELLA models, accompanied by graphs that show the behavior or evolution of these systems over time, and the basic equations used in the models. In each case, I mention a real-life system that is similar to these models.



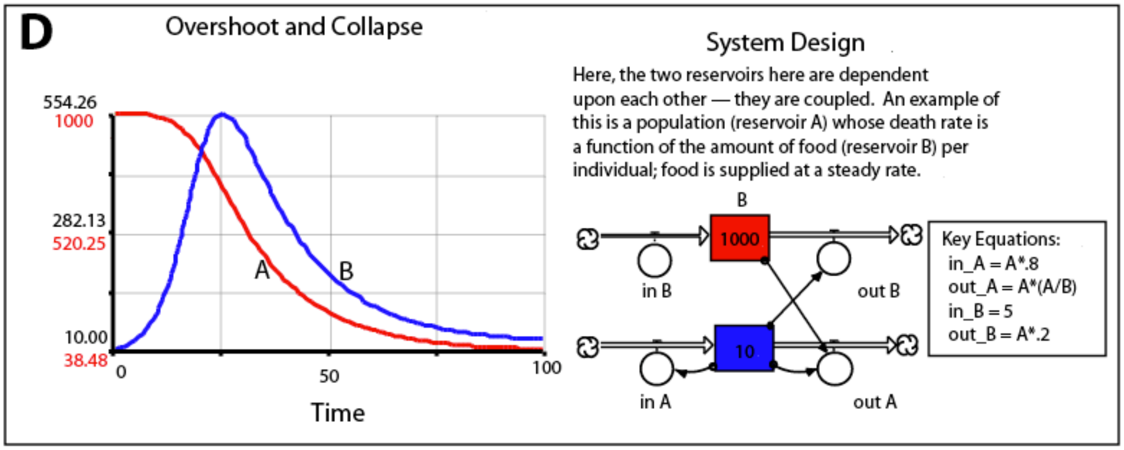
A. Linear Growth and Decay. The key to this system design is that the flows are defined as constants; this results in very simple, easy-to-predict behavior characterized by constant rates of change.



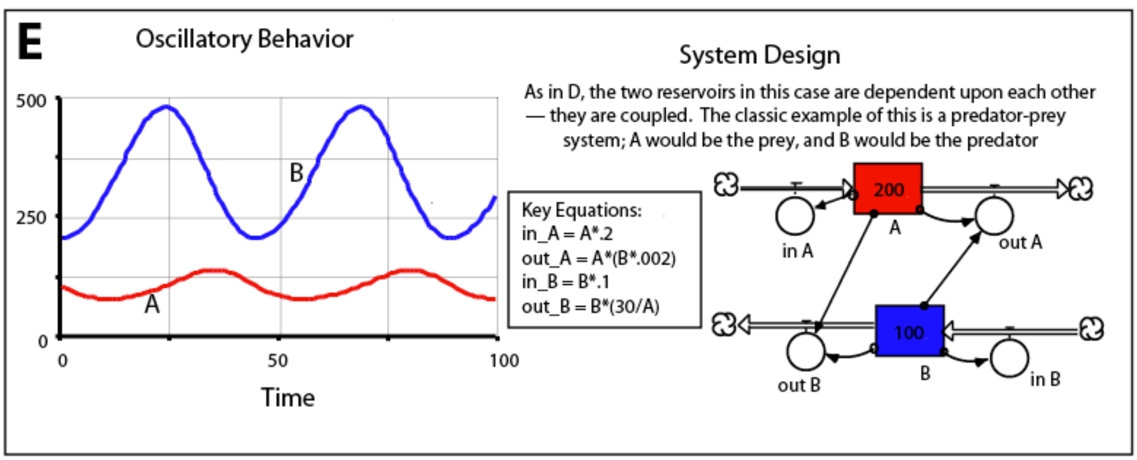
B. Exponential Growth and Decay. These are extremely common design elements that are sometimes referred to as first-order kinetic equations, but in simpler terms, they represent growth or decay (draining) processes where the rate of change is a fixed percentage of the reservoir involved in the flow. Exponential growth represents a classic form of positive feedback, yielding a runaway behavior, while exponential decay represents a negative feedback mechanisms that has a stabilizing effect.



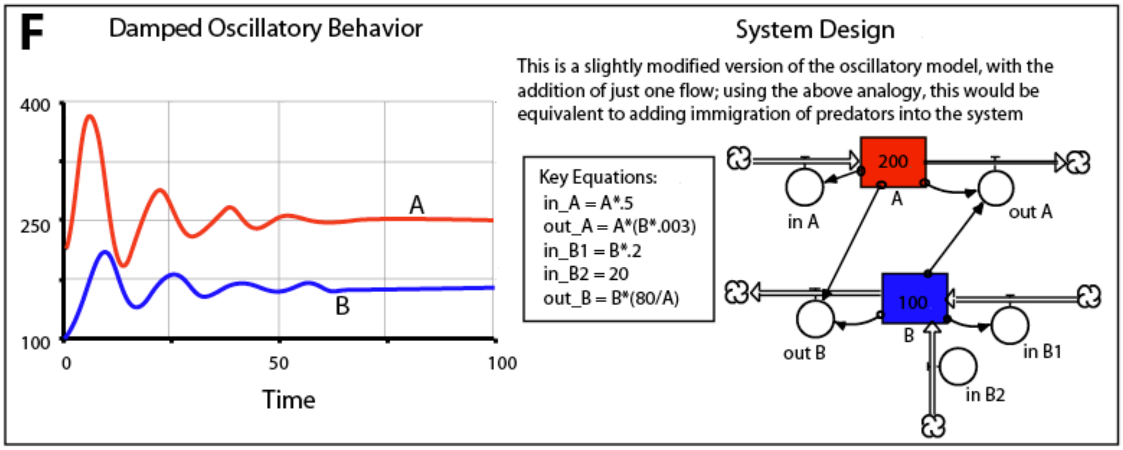
C. S-Shaped Growth. This system design is especially common in models of population growth that is limited by some resource. In this type of a system, one of the flows is defined as a percentage of the reservoir, but that percentage changes as the amount in the reservoir changes. This kind of a system design has a sort of built-in limit, determined by the rate constants, as shown in the figure. Interestingly, this sytem structure can lead to chaotic behavior as we will see in a later chapter on population growth.



D. Overshoot and Collapse. This system design represents a variation on the system shown in C above. In this case, the limit to growth, represented by reservoir B is declining as reservoir A increases. Reservoir A grows exponentially and shoots past its limit and as a result, the limit decreases more and more, fueling a collapse of A until it reaches a steady-state at a very low level.



E. Oscillatory Behavior. This type of system design represents what is called a coupled system since the change of each reservoir is dependent on how its companion reservoir is changing. These systems often lead to an oscillation, creating cycles that do not have an external control. The oscillation arising from these coupled reservoirs is very different from the kind of oscillation forced onto a system by some external control — something that is a sinusoidal function of time.



F. Damped Oscillatory Behavior. This is a variation of the system shown in E above and is common in any environment where friction or some other form of energy loss is a factor. Interestingly, it is also a consequence of migrations in systems of coupled populations.

V. Validation, Tuning, and the Significance of Computer Models

The purpose of a model is not to replicate the real world — it is clearly impossible to put the complexity of the real world into a computer. Instead, the goal is usually to understand something about the behavior of a system, including how the system responds to changes. We create and use these models because their real-life versions are so complex, large, and often slow that we cannot generally understand them without some kind of controlled experimentation. But, the obvious simplification of models commonly creates leads to skepticism that ranges from complete rejection of anything the model reveals to a milder form in which the results are accepted as a good possibility for the way the real world behaves. So, how does one develop a sophisticated, nuanced appreciation for the significance of model results?

Computer models such as the ones shown in this paper are nothing more than a set of differential equations that are integrated over time. The significance of the model results are therefore dependent on the nature of the equations. Just as there is a range in the quality of equations from highly abstract to highly realistic, there is a range in the significance of the model results. If the equations were pulled out of thin air, then the results are not significant relative to any real world system (but they are nevertheless meaningful in a purely mathematical sense). If the equations are designed to express the general relationships of a real world system (e.g., the intermediate model of Fig.7), the model results might be only qualitatively meaningful. If the equations are designed and tested such that they mimic the important processes of the real world system, then the model results may be quantitatively significant, and the model might have some important predictive capabilities. A model such as this last type could be tested against known histories of the real world system; models that pass these tests are sometimes said to be ***validated***, and their results can be considered to be reliable, at least within a certain range of conditions. Many modelers shy away from the term validation since it implies a sense that the model is perfect (which is highly unlikely) — perhaps it is better to say that these models are "***tuned***" to match observations from the real world.

The task of model tuning is simple in some cases. For instance, the water tub model is easily tested by constructing a real version of the model (see Moore and Derry, 1995) and collecting some real data to compare with the computer model results, then adjusting the drain rate constant until the model matches the observations. If this succeeds, then our model is tuned (again, some would say *validated*) and we could say that it captures the essence of the system. Torricelli's Law, used to construct the more realistic version of the water tub model in Figure 7, is essentially derived from observations and so it generally gives quite reliable results.

But for larger systems, such as the global carbon cycle (Figure 11), the process of model tuning or validation is a bit trickier — you clearly can't create a lab-version of such a vast, global-scale model. Instead, we have to rely on some kind of natural experiment in which we have some knowledge of an imposed change and some record of the model's performance or state over time. Unwittingly, humans have been conducting such an experiment and the state of the carbon cycle is partly available in the form of instrumental measurements since the late 1950's and ice core records of atmospheric CO_2 concentrations before then — thus giving us a data set to use in tuning global carbon cycle models.