PART II. BASIC PROCEDURES IN SYSTEMS MODELING

BUILDING BLOCKS FOR SYSTEMS MODELS

Quote of the Day

"To do science is to search for repeated patterns,
not simply to accumulate facts..."

- Robert H. MacArthur (1972)

I. WHY BUILDING BLOCKS?

Building blocks, or simulation modules are simple models that represent some basic system structures and dynamics. These modules are very important for understanding many fundamental processes common in biological, physical, and socioeconomic phenomena. One certainly needs to understand them well before attempting to deal with complex feedback systems. In the same time, the model building blocks demonstrate how these commonly occurring basic processes are represented in systems simulation, and often become convenient and effective for constructing complex systems models.

II. BUILDING BLOCKS FOR SYSTEMS MODELS

(I) FIRST-ORDER SYSTEMS

First-order: One state variable (or 1 stock)

Linear: No non-linear combinations of the state variable of any sort in the algebraic equation of rates.
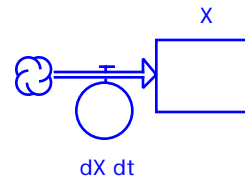
1. Linear Change
- One state variable.
- One rate variable.
- The rate of change is a constant, <u>not</u> dependent on the state variable. Thus, there is no feedback loop.
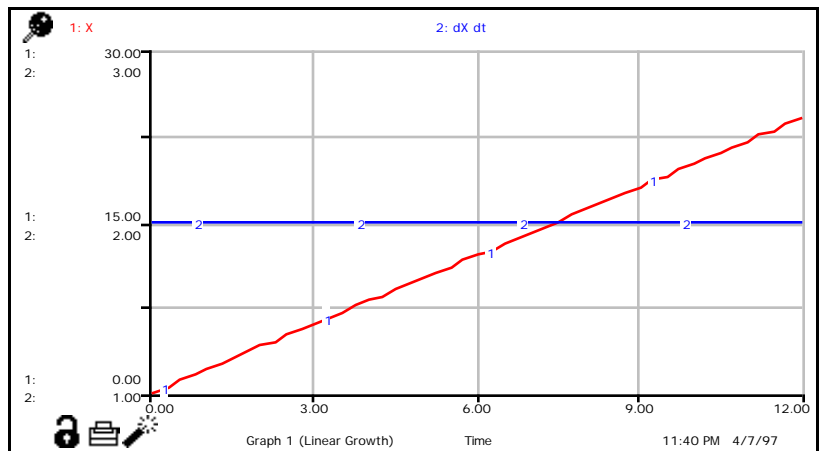
(1) Linear Growth
- The rate of change is positive, and thus represented as an inflow.
- Examples:
    - Increase in water level in a tank with a constant inflow of water into the tank.
    - Plant biomass sometimes is found to increase linearly with certain soil nutrients.

Linear Growth in STELLA


```
X(t) = X(t - dt) + (dX_dt) * dt
INIT X = 0
INFLOWS:
dX_dt = 2
```
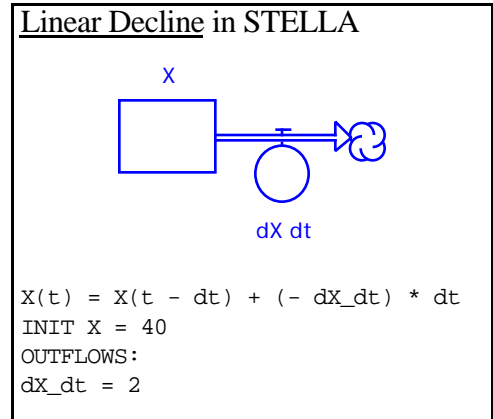
(2) Linear Decline
- The rate of change is negative, and thus represented as an outflow.
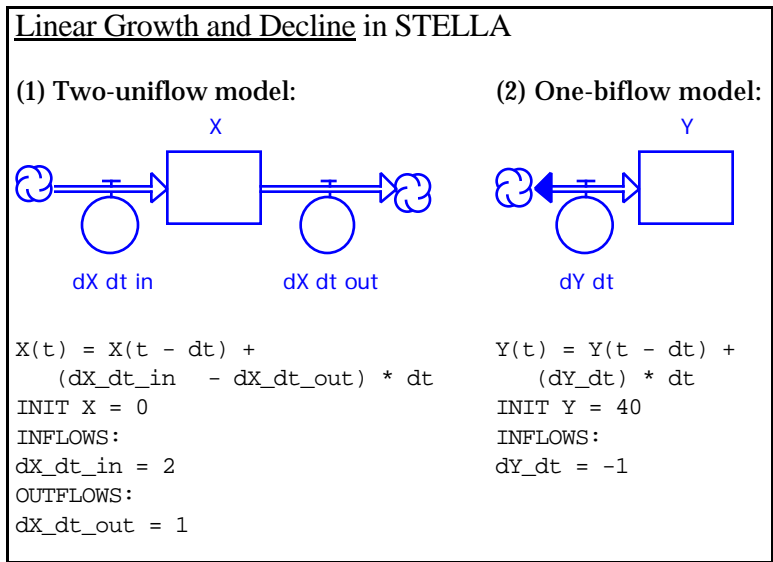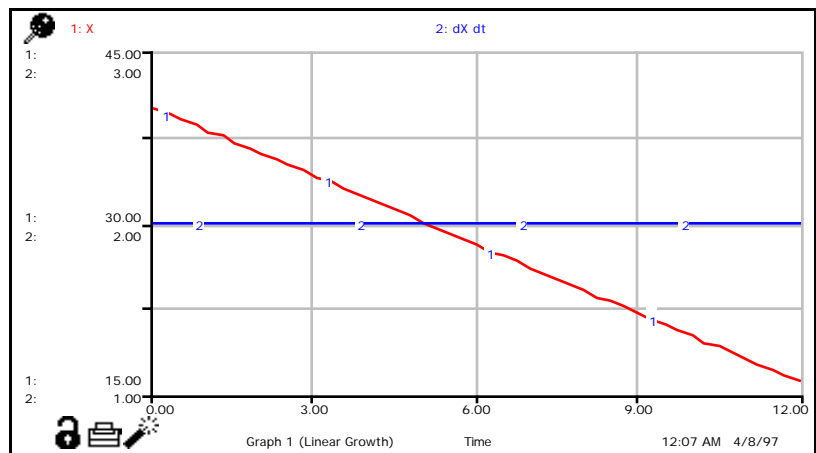- Examples:

    Decrease in the amount of materials left in a stock when the materials are taken out of the stock in the same amount per unit of time.

    Decrease in crop yield with declining annual precipitation sometimes exhibits linear pattern.

Linear Decline in STELLA

X

dX dt

```
X(t) = X(t – dt) + (– dX_dt) * dt
INIT X = 40
OUTFLOWS:
dX_dt = 2
```

(3) Linear Growth and Decline

- Combination of the linear growth and decline module.
- One state variable.
- Either two uniflows (one inflow and one outflow) or one biflow (compare the structural diagrams below).
- The uniflow version gives more details, whereas the biflow version is more concise. The choice of the two forms is dependent on the degree of details is needed.
- Linear growth occurs when the total rate of change

(inflow minus outflow) is positive, and linear decline does otherwise.

1: X			2: dX dt	
1: 45.00				
2: 3.00				
1: 30.00				
2: 2.00				
1: 15.00				
2: 1.00				

0.00   3.00   6.00   9.00   12.00

Graph 1 (Linear Growth)   Time   12:07 AM 4/8/97

Linear Growth and Decline in STELLA

(1) Two-uniflow model:

X

dX dt in   dX dt out

```
X(t) = X(t – dt) +
    (dX_dt_in  – dX_dt_out) * dt
INIT X = 0
INFLOWS:
dX_dt_in = 2
OUTFLOWS:
dX_dt_out = 1
```

(2) One-biflow model:

Y

dY dt

```
Y(t) = Y(t – dt) +
    (dY_dt) * dt
INIT Y = 40
INFLOWS:
dY_dt = -1
```

2. Exponential Growth and Decay

(1) Exponential Growth

- Also known as compounding process
- One state variable.
- One rate variable.
- One auxiliary variable.
- The rate of change is equal to a constant proportion of the state variable.
- Because the rate variable depends on the state variable itself, they form a positive feedback loop, in which the state variable and the rate of change reinforce each other, generating an accelerating, run-away behavior.
- The rate of change is always positive (i.e., the state variable increases monotonically), and is represented it as an inflow.
- **Time Constant T_c**: The reciprocal of the constant r in the exponential model, $T_c = 1/r$, with the dimension of units of time.

  For exponential growth, T_c is the time required for the state variable to become e ($= 2.7183$) times of its current value. This can be seen from the following simple manipulations:

  $$N = N_0 e^{rt}.$$

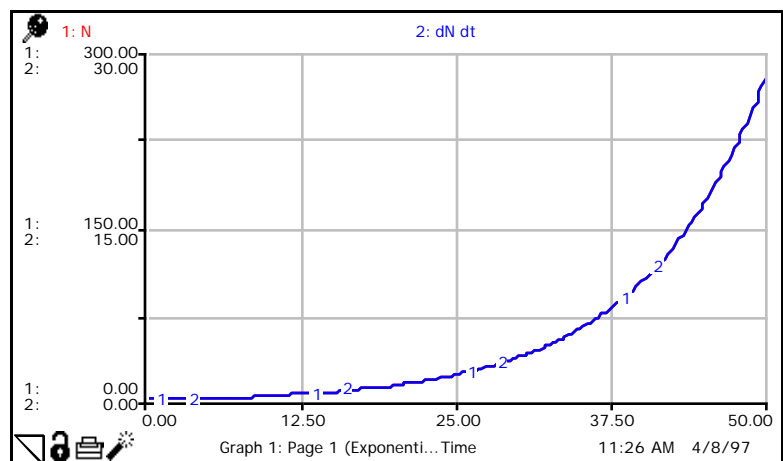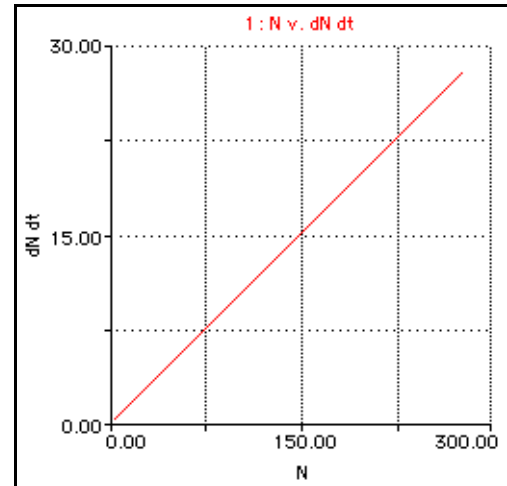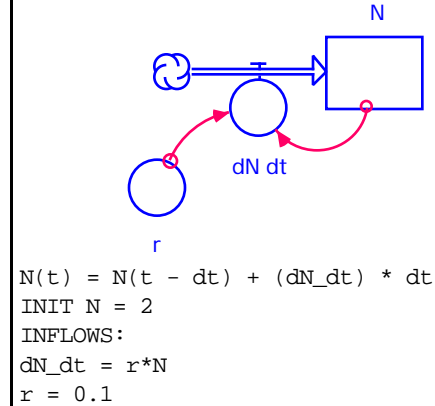 When t is equal to T_c, we have

  $$N_{T_c} = N_0 e^{\frac{1}{T_c} T_c} = N_0 e$$

  In general, if $t = nT_c$,

  $$N_{nT_c} = N_0 e^{\frac{1}{T_c} nT_c} = N_0 e^n$$

  Note: **For continuous systems, the time step, t, in simulation must be smaller than the smallest time constant in the model.**

Exponential Growth in STELLA


```
N(t) = N(t - dt) + (dN_dt) * dt
INIT N = 2
INFLOWS:
dN_dt = r*N
r = 0.1
```


- **Doubling Time T_d**: The time required for the state variable in the exponential model to double in value.

  To calculate the doubling time, let $N = 2N_0$ and let $t = T_d$. Thus, from $N = N_0 e^{rt}$, we have:

$$2N_0 = N_0 e^{rT_d}$$

$$\ln 2 = rT_d$$

$$T_d = \frac{0.69}{r} \quad \text{or} \quad T_d = 0.69\, T_c$$

Therefore, the doubling time for exponential growth is a constant. It is conversely proportional to the rate of change, and is about 70% of the time constant. This is where "the rule of 70" in population demography comes from (i.e., population doubling time = 70 / percent natural growth rate).

Doubling Time
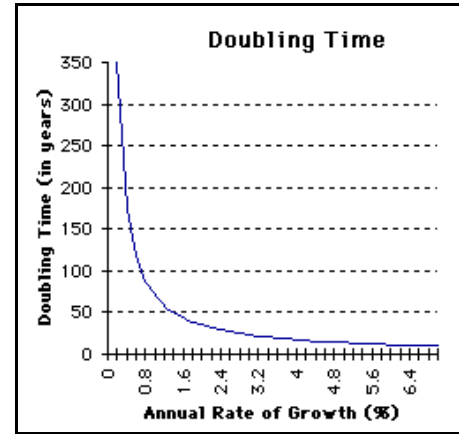
- Exponential growth is a common type of dynamics that exists in all different disciplines. Any phenomena described by words such as "snowball effect", "vicious circles", "virtuous circles", and "bandwagon effect" can be represented as a positive feedback loop structure.

- Examples:

    Savings increase in a bank account due to interest income

    Population growth when resources are not limiting.

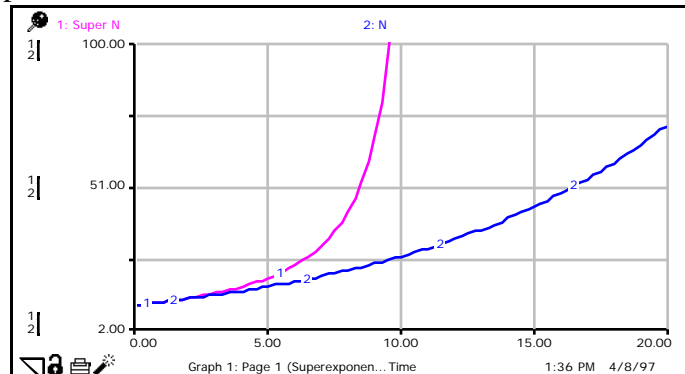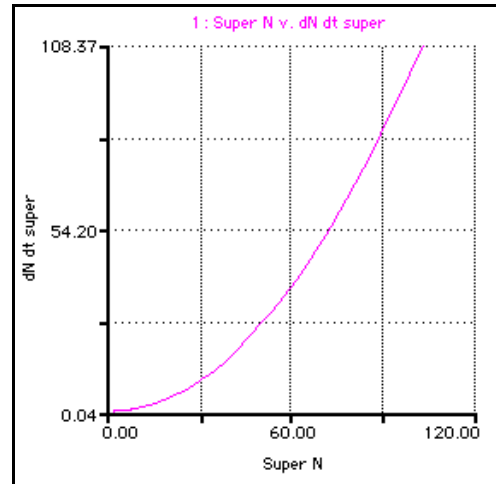   Drug addiction.

    Arms race (proliferation).

- Superexponential (or supraexponential) growth:

    In many positive feedback systems, doubling time decreases, rather than remain constant, as the state variable increases. This means that the rate of change is a nonlinear function of the state variable (see Figure). For example,

$$\frac{dN}{dt} = (rN)N = rN^2$$

    The dynamics generated by this <u>nonlinear</u> system, with a rate of change faster than a fixed proportion of the state variable, is called superexponential growth.

    The superexponential growth has the same feedback structure as the exponential.

(2) Exponential Declay

- Also known as draining process
- One state variable.
- One rate variable.
- The rate of change is equal to a constant proportion of the state variable, but in contrast with exponential growth it is negative.

```
N(t) = N(t - dt) + (dN_dt) * dt
INIT N = 10
INFLOWS:
dN_dt = r*N
Super_N(t) = Super_N(t - dt) + (dN_dt_super) * dt
INIT Super_N = 10
INFLOWS:
dN_dt_super = (r_2/10)*Super_N^2
r = 0.1
r_2 = if (time<10) then 0.1 else 0
```

- The rate variable depends on the state variable itself, and form a feedback loop between them. Because an increase in the value of the state variable increases the rate of change, but an increase in the rate decreases the value of the state variable, the feedback is negative (or goal-seeking).

  - **Time constant T_c**: The reciprocal of the constant r in the exponential model, $T_c = 1/r$, with the dimension of units of time.

    For exponential decay, T_c is the time required for the state variable to become e^{-1} (= 0.368) times of its current value, or the time required for 63% of the contents in the stock to vanish. This is also called the relaxation time, which is sometimes used as a measure of the speed with which a system is absorbing disturbances.

    Mathematically, relaxation time can be derived as follows:

$$N = N_0 e^{-rt},$$

For a time period from t=0 to t=T_c,

$$N_{T_c} = N_0 e^{-\frac{1}{T_c}T_c} = e^{-1} N_0 = 0.368\ N_0,$$

In general, when $t = nT_c$, we have

$$N_{nT_c} = N_0 e^{-\frac{1}{T_c}nT_c} = e^{-n} N_0 = 0.368^n N_0$$

- **Half-life T_h**: The time required for the state variable to reduce its current value by a half. It is an analog of the doubling time in exponential growth. It can be calculated as follows:

$$\frac{1}{2} N_0 = N_0 e^{-rT_h},$$

$$\ln\frac{1}{2} = -rT_h, \text{ or } \ln 2 = rT_h, \text{ thus,}$$

$$T_h = \frac{0.69}{r} \text{ or } T_h = 0.69\ T_c$$

  Therefore, the half-life time for exponential decay is a constant (about 70% of the time constant), independent of the value of the state variable.

- Examples:

  Draining water from a tank

  Population-death rate process

  Redioactive decay process

(3) Exponential Growth and Decay Combined

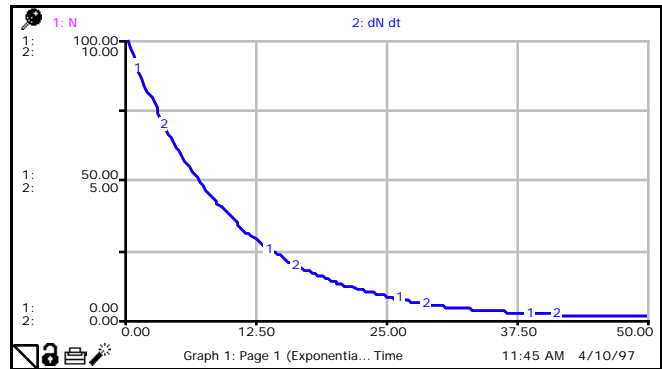 - Simply a combination of exponential growth and exponential decay.
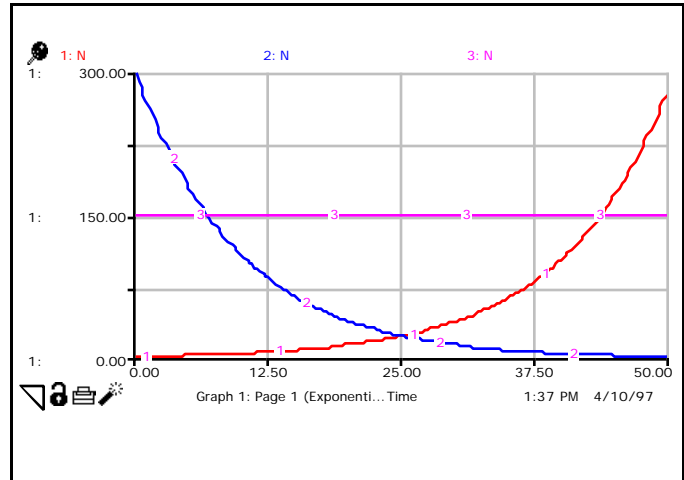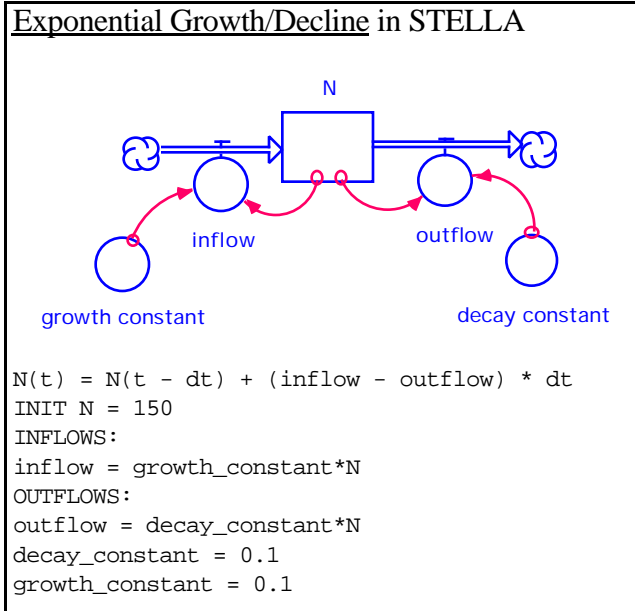

Exponential Decay in STELLA
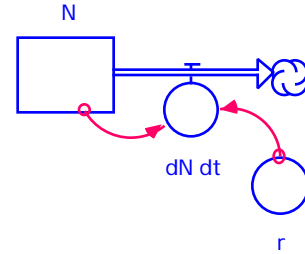

```
N(t) = N(t - dt) + (- dN_dt) * dt
INIT N = 100

OUTFLOWS:
dN_dt = r*N
r = 0.1
```

- The behavior of the first-order linear system exhibits three different patterns:
  1) exponential growth when growth constant is larger than decay constant;
  2) exponential decay when growth constant is smaller than decay constant; and
 3) remaining unchanged (unstable equilibrium) when the two constants are equal.

Exponential Growth/Decline in STELLA


```
N(t) = N(t - dt) + (inflow - outflow) * dt
INIT N = 150
INFLOWS:
inflow = growth_constant*N
OUTFLOWS:
outflow = decay_constant*N
decay_constant = 0.1
growth_constant = 0.1
```

3. Exponential Collapse

- Also known as <u>accelerated decay</u>, indicating that the rate of change gets faster as the level goes down.
- A simple exponential collapse model may consist of 1 state variable, 1 outflow, and 2 constants.
- A positive feedback: increasing level --> increasing rate of change --> increasing level --> ...
- A simple mathematical model for exponential collapse:
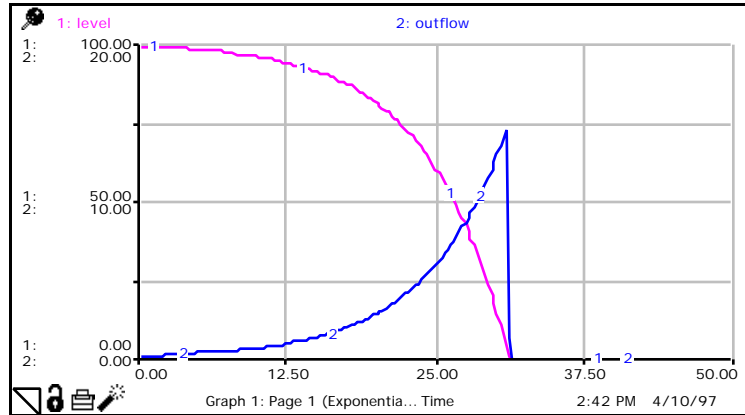
$$\frac{dN}{dt} = -r(M - N)$$

Exponential Collapse in STELLA


```
level(t) = level(t - dt) + (- outflow) * dt
INIT level = 99

OUTFLOWS:
outflow = collapse_constant*(M-level)
collapse_constant = 0.15
M = 100
```

where N ($< M$) is the state variable, and r and M are two constants.

- Examples:
  Population shrinking when smaller than MVP.
  Change in the interactive force between molecules when water is heated up and boils.
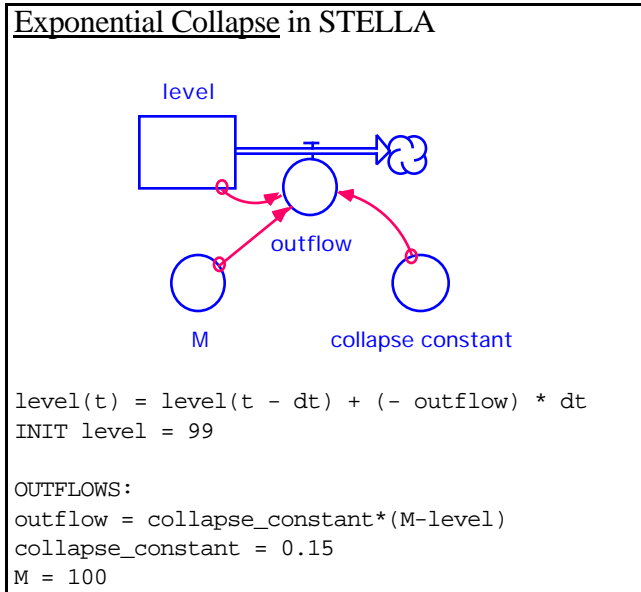
Decline of species diversity with increasing habitat fragmentation.

Some other threshold phenomena in physical and biological processes may exhibit behavior that resembles exponential collapse.
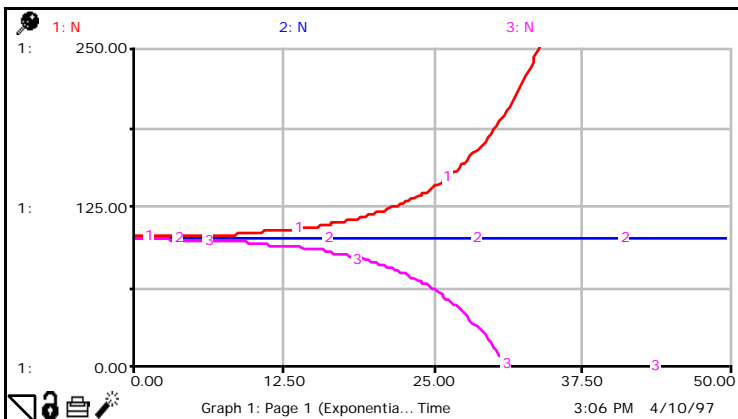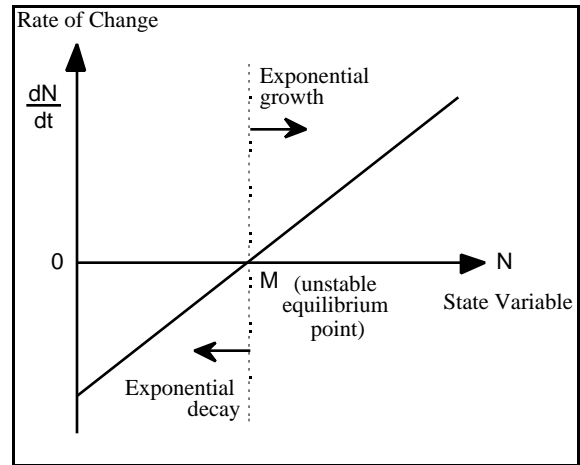

4. Exponential Growth and Collapse Combined

- A simple exponential collapse consists of 1 state variable, 1 bi-directional flow, and two constants.
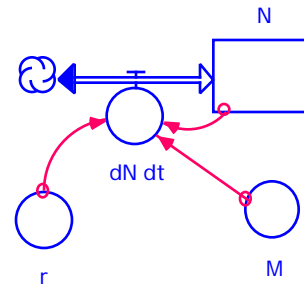- This simple model can be mathematically expressed as:

$$\frac{dN}{dt} = r(N - M)$$

- The simple structure exhibits 3 different behavioral patterns:

  Exponential growth when N > M;
  Exponential collapse when N < M; and
  Remaining constant when N = M.

Exponential Growth/Collapse in STELLA


```
level(t) = level(t - dt) + (-
outflow)*dt
INIT level = 99
OUTFLOWS:
outflow = collapse_constant*(M-level)
collapse_constant = 0.15
```

# 5. Goal-Seeking Behavior

(1) Simple Goal-Seeking

- A simple goal-seeking model may consist of 1 state variable, 1 biflow, and 2 constants.
- The state variable and the flow form a negative feedback.
- A simple goal-seeking model may take the form:

$$\frac{dL}{dt} = c(G - L)$$

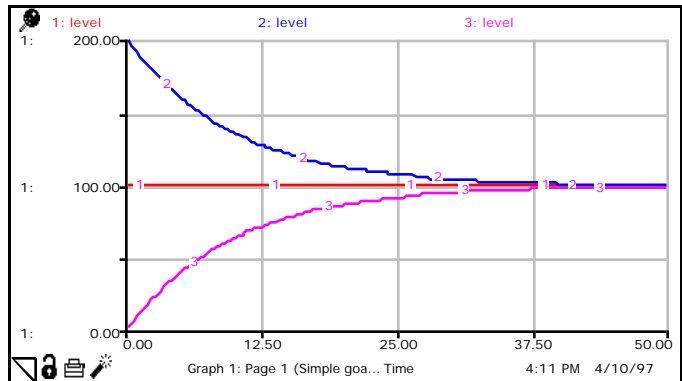where L is the level, G is the goal, and c is a constant.

- Apparently, exponential decay is a special case of goal-seeking behavior, in which the goal is zero!

- Question: What is the time constant T_c for this simple model?
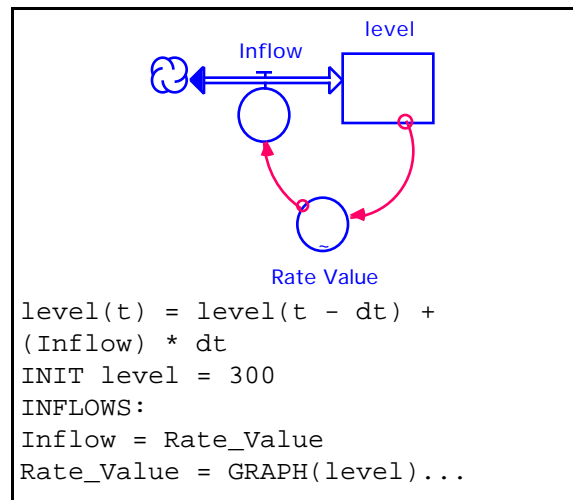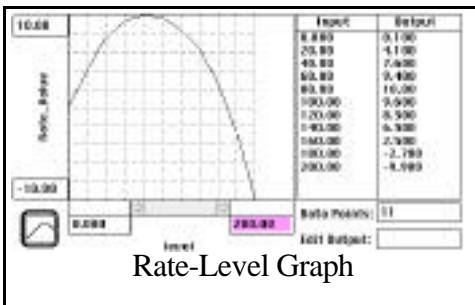
Simple Goal-Seeking in STELLA
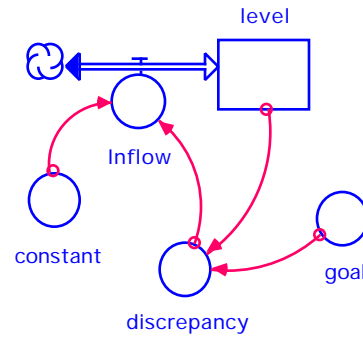

```
level(t) = level(t - dt) + (Inflow) * dt
INIT level = 2

INFLOWS:
Inflow = constant*discrepancy
constant = 0.1
discrepancy = goal-level
goal = 100
```

(2) S-Shaped Growth

- Also known as logistic growth or sigmoidal growth.
- It is a first-order <u>nonlinear</u> system (see the Rate-Level graphs).
- The state variable and the flow form a negative feedback that ultimately gives rise to the goal seeking behavior (see the Forrester diagram).
- It may take different forms (uniflow or biflow versions). [For example, in population regulation, crowding may affect only the per capita birth rate, or only the per capita death rate, or both.]
- The simple structure that generates S-shaped dynamics may look like this:

Rate-Level Graph


```
level(t) = level(t - dt) +
(Inflow) * dt
INIT level = 300
INFLOWS:
Inflow = Rate_Value
Rate_Value = GRAPH(level)...
```

The behavior of this simple structure has three patterns:
1) Increasing and approaching the goal if simulation starts with a value of the level smaller than the goal;
2) Decreasing and approaching the goal if simulation starts with a value of the level larger than the goal;
3) Remaining unchanged if simulation starts with the value of the level equal to the goal.
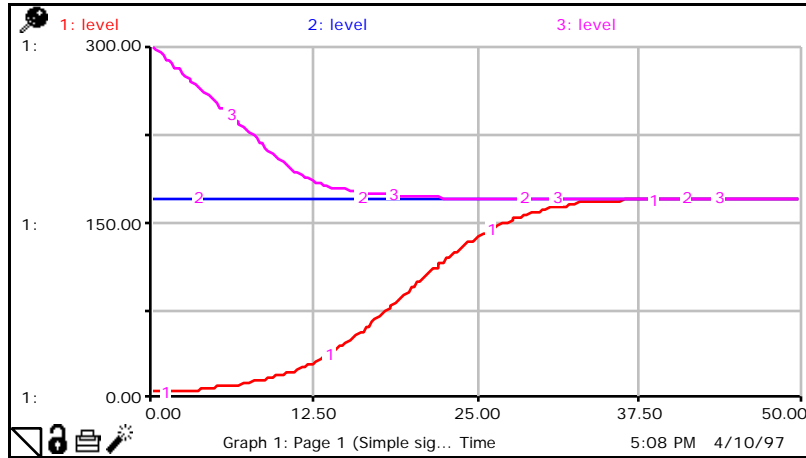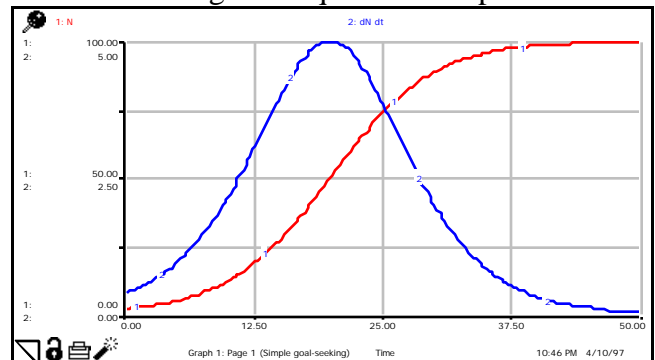

A familiar example of sigmoidal growth is the logistic equation:
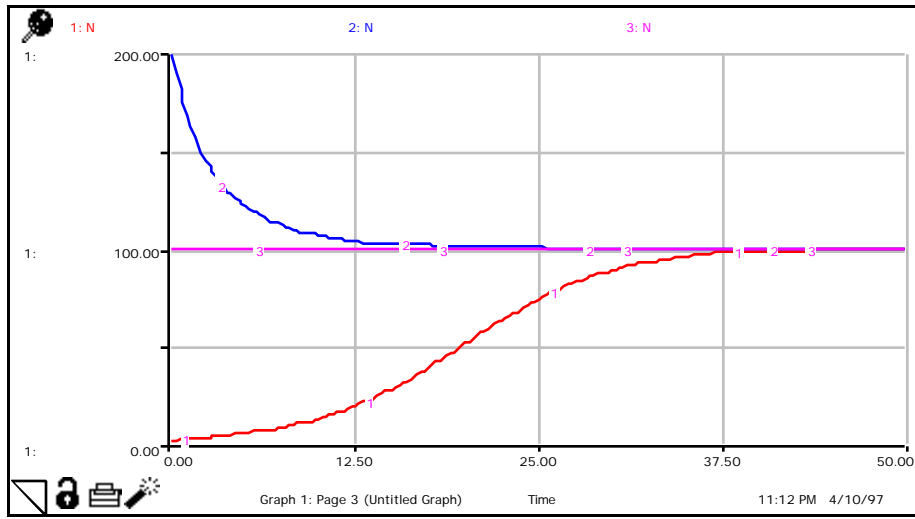
$$\frac{dN}{dt} = rN(1 - N/K)$$

This is a first-order nonlinear differential
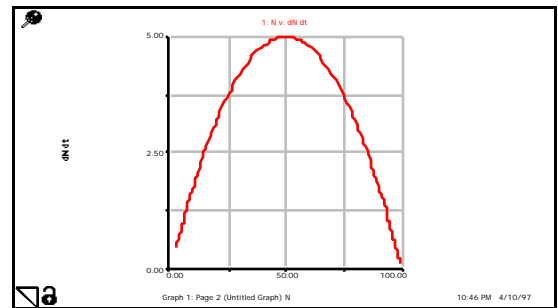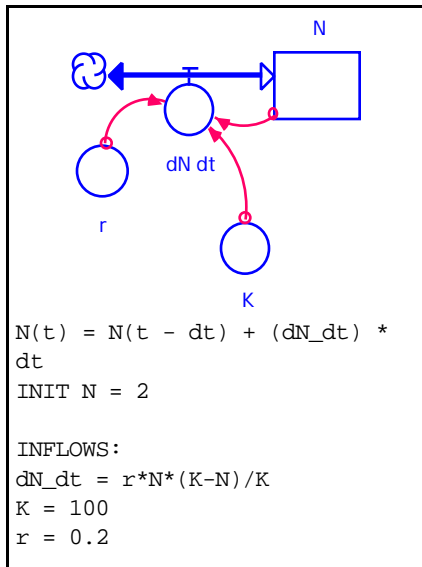
equation. The following is an equivalent simple


```
N(t) = N(t - dt) + (dN_dt) *
dt
INIT N = 2

INFLOWS:
dN_dt = r*N*(K-N)/K
K = 100
r = 0.2
```
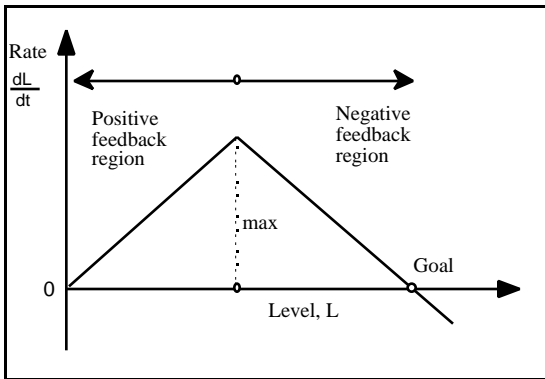
systems model.

Dynamics from the Logistic Equation-Based Module

Feedback loop analysis of the S-shaped growth:

- Positive vs negative feedback regions
- Characteristics of the rate-level curve: several similar, but distinctive forms
- Goal-seeking does NOT always guarantee goal-achieving!
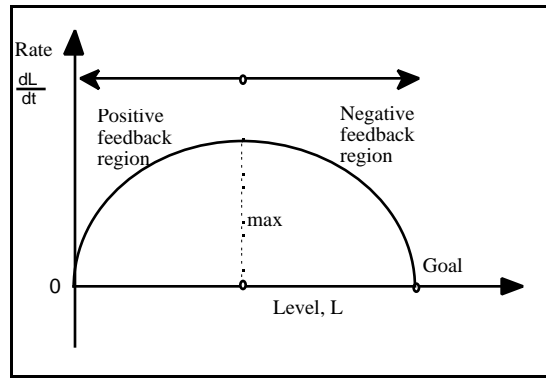
(II) SECOND-ORDER SYSTEMS

First-order (continuous) systems do not generate overshoot and collapse or oscillations unless significant time delays are introduced. Such behavioral patterns are commonly found in second- or higher-order systems (with 2 or more stocks). The following section illustrates some simple systems structures that give rise to non-monotonic dynamics.

6. Overshoot and Collapse

- A simple structure that generates overshoot and collapse behavior includes 2 stocks (e.g., population and food resource) and a few rates associated with them (see the Forrester diagram).
- In this example, population grows exponentially when food is not limited. More and more food resource is consumed by the growing population, resulting in less and less food resource. At some point, death rate will rise above birth rate, and the population eventually collapse due to the lack of food.

Overshoot and Collapse in STELLA


```
food(t) = food(t – dt) + (– consumption_rate) * dt
INIT food = 1000 {tones}
OUTFLOWS:
consumption_rate = per_capita_consumption_rate*population
population(t) = population(t – dt) + (birth_rate –
death_rate) * dt
INIT population = 100
INFLOWS:
birth_rate = per_capita_birth_rate*population
OUTFLOWS:
death_rate = per_capita_death_rate*population
per_capita_birth_rate = 0.1 {1/time}
per_capita_consumption_rate = 0.1  {tones/(individual *
time)}
per_capita_death_rate = GRAPH(food/INIT(food))
(0.00, 0.995), (0.1, 0.645), (0.2, 0.45), (0.3, 0.3),
(0.4, 0.21), (0.5, 0.14), (0.6, 0.095), (0.7, 0.06),
(0.8, 0.03), (0.9, 0.01), (1, 0.00)
```
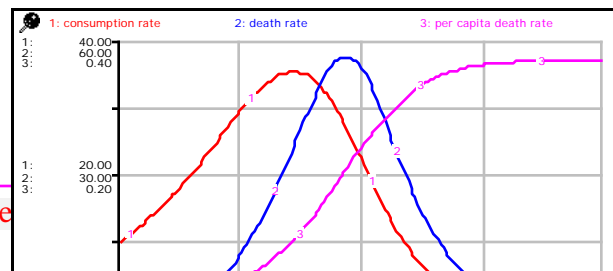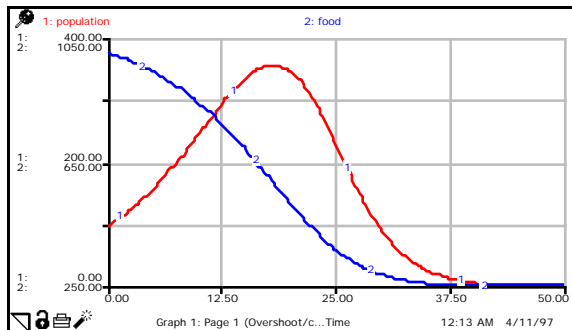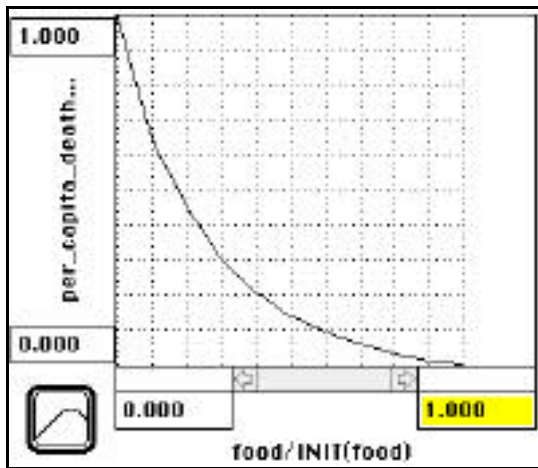
7. Oscillations

- Two stocks and four rates associated.
- The first stock promotes the inflow to the second stock, and the second stock accelerates the depletion of the first stock. This negative feedback is essential for oscillatory behavior.
- With STELLA, you must use one of the Runge-Kutta simulation algorithms.

(1) A Simple General Structure (cf. Richmond et al. 1993).

Oscillation in STELLA


```
Stock_1(t) = Stock_1(t - dt) + (inflow_1 - outflow_1) * dt
INIT Stock_1 = 10
INFLOWS:
inflow_1 = Stock_2*productivity_1
OUTFLOWS:
outflow_1 = 10
Stock_2(t) = Stock_2(t - dt) + (inflow_2 - outflow_2) * dt
INIT Stock_2 = 15
INFLOWS:
inflow_2 = 10
OUTFLOWS:
outflow_2 = Stock_1*productivity_2
productivity_1 = 1
productivity_2 = 1
```


Graph 1 (Oscillation)   Time   2:21 AM   4/11/97

(2) A Predator-Prey Model

Oscillation in STELLA


```
predator(t) = predator(t - dt) + (birth_rate - death_rate) * dt
INIT predator = 10
INFLOWS:
birth_rate = 0.01*predator*prey
OUTFLOWS:
death_rate = 0.2*predator
prey(t) = prey(t - dt) + (prey_births - prey_deaths) * dt
INIT prey = 50
INFLOWS:
prey_births = per_capita_birth_rate*prey
OUTFLOWS:
prey_deaths = 0.03*predator*prey
per_capita_birth_rate = GRAPH(prey)
(0.00, 0.685), (50.0, 0.41), (100, 0.295), (150, 0.215), (200, 0.15), (250, 0.11),
(300, 0.07), (350, 0.04), (400, 0.02), (450, 0.01), (500, 0.00)
```


(From the same model but D-I prey birth rate)
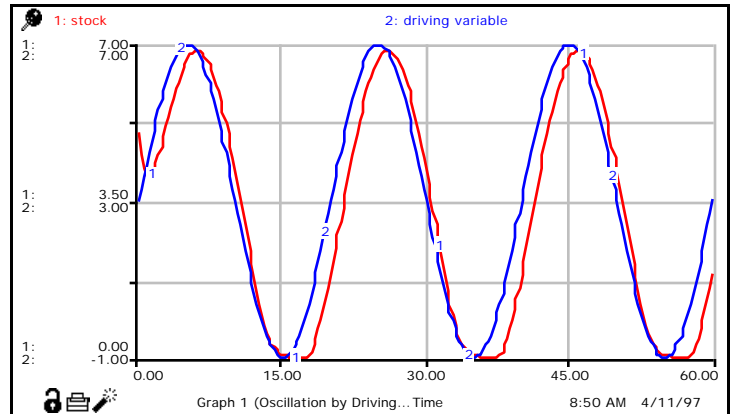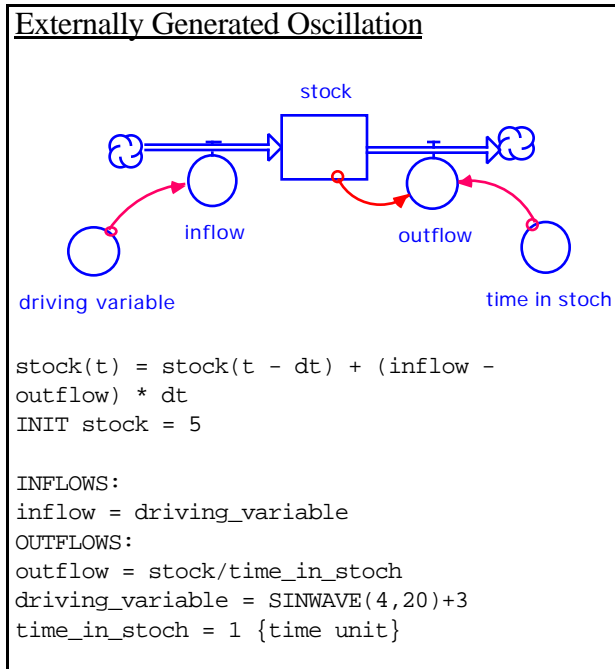
(III) Oscillations Generated by Driving Functions

Driving variables (or functions) that change periodically (e.g., temperature, light intensity) may introduce oscillations into a system of any order.
Oscillations forced by driving variables are called externally generated oscillations.

Externally Generated Oscillation


```
stock(t) = stock(t - dt) + (inflow -
outflow) * dt
INIT stock = 5

INFLOWS:
inflow = driving_variable
OUTFLOWS:
outflow = stock/time_in_stoch
driving_variable = SINWAVE(4,20)+3
time_in_stoch = 1 {time unit}
```

A general equation for modeling periodic driving functions:

$$Y = m + A\cos(\omega(t - \tau))$$

where Y is a driving variable (e.g., temperature, light intensity), m is the mean value of the function, A is the amplitude of the peak above the mean, (t - τ) shifts the peak by τ physical units, and the ω is the angular frequency per physical unit.

For example, suppose we have a time series:
- Mean daily air temperatures: 40 degrees F
- Amplitude: 25 degrees F
- Period: 365 days
- Position of the 1st peak: July 30 (Julian day 211)

The equation becomes:

$$T = 40 + 25\cos\frac{2}{365}(t - 211)$$

(IV) Chain Structure and Time Delay

See modules.