System Dynamics

- Outline
  - History and Motivation
  - The System Dynamics Module of Netlogo
  - Basic elements of System Dynamics: stocks and flows
  - Building System Dynamics Models
    - Exponential growth
    - Logistic growth
    - The dynamics of love affairs
    - Sheep and wolves
- Also want to use this lecture to explore some possible dynamics in higher dimensions

History + Motivation

- So far:
  - have talked about a bit of theory of dynamical systems and some basic numerical techniques how to solve them on a computer
- There are various environments in which such these techniques can be used in an automated way, these include:
  - Building your own models using libraries (e.g. the numerical recipes in C or various python libraries)
  - Matlab/Mathematic/Maple
  - The graphical interface of various commercial “system dynamics” packages like Stella or Vensim or the free system dynamics module of Netlogo
- There will be tutorials on Matlab/Mathematica, but in this lecture we want to focus on system dynamics – probably the most accessible tool which is
  - widely used in management/business/management studies for analysing industrial processes
  - to some extent in natural resource modelling (Club or Rome and limits to growth study)

History

- 1950s and 60s: Jay Forrester analysed industrial processes like business cycles at GE
- First simulation packages SIMPLE and later DYNAMO
- In the 70s Forrester was invited to help with the system dynamics approach to develop models of global resource constraints -> WORLD1,2
- Nowadays software with GUIs to allow easy access to model development around, the most popular are probably STELLA and VENSIM, see
  - http://en.wikipedia.org/wiki/List_of_system_dynamics_software

  for a larger list of available packages

Motivation

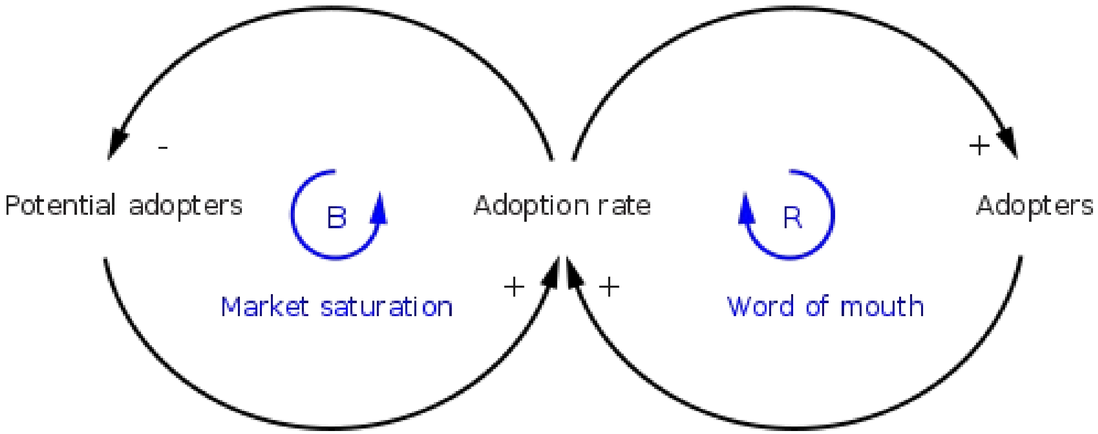
- Why bother?
  - Easy access to model building

    -> Use it to build your own models if you are not too familiar with differential equations
  - “Graphical language” of feedback loops etc.

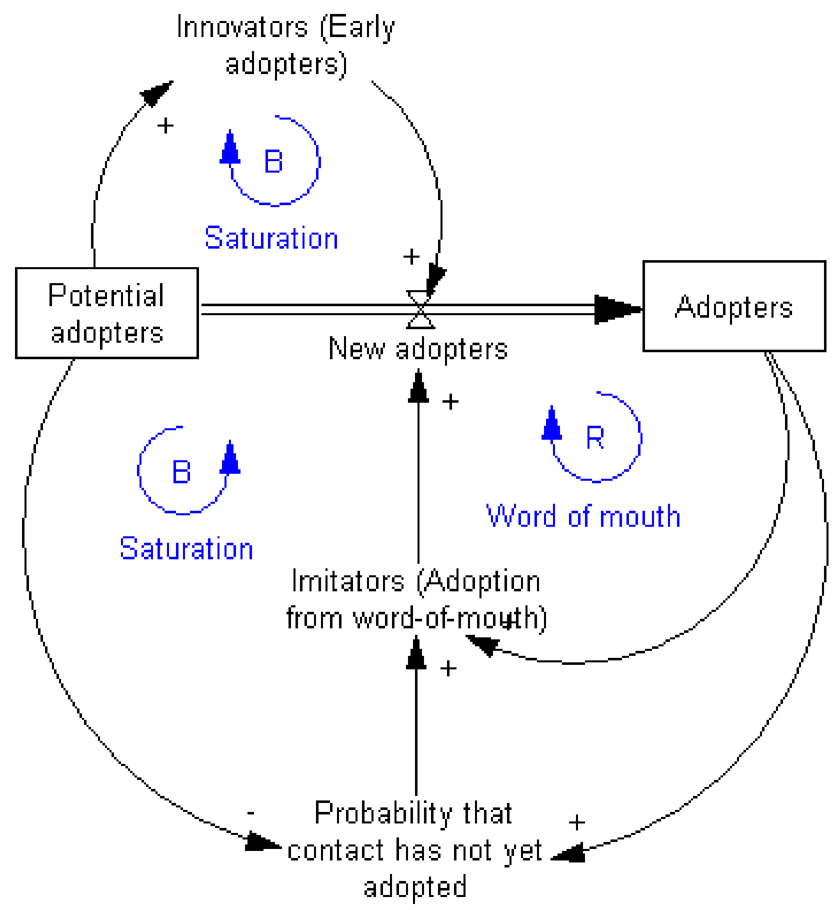
    -> fairly useful to develop conceptional models
  - Fairly widely used in some disciplines

    -> important to understand the language and be able to translate it
- We'll use it to play around with some models to illustrate some possible dynamics of (systems) of ODEs

An Example: Dynamics of New Product Introductions



Causal loop diagram for product introductions



Corresponding system dynamics model
“stock and flow diagram”

The System Dynamics Tool of Netlogo

- I have built the following models in Netlogo (because it is publically available)
  - You can download it from:

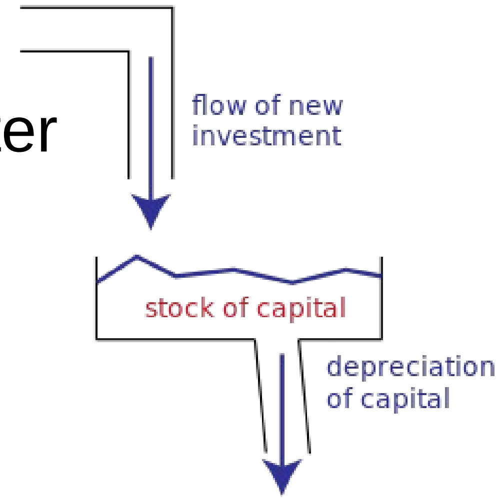
    https://ccl.northwestern.edu/netlogo/download.shtml

    and reimpliment the models and explore them
  - A tutorial on how to use the system dynamics tool of netlogo is available here:

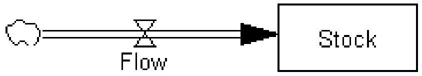
    http://ccl.northwestern.edu/netlogo/docs/systemdynamics.html
  - Models used in this lecture can be downloaded from

    http://users.ecs.soton.ac.uk/mb8/sim.html
  - You can also use Netlogo to construct ABMs

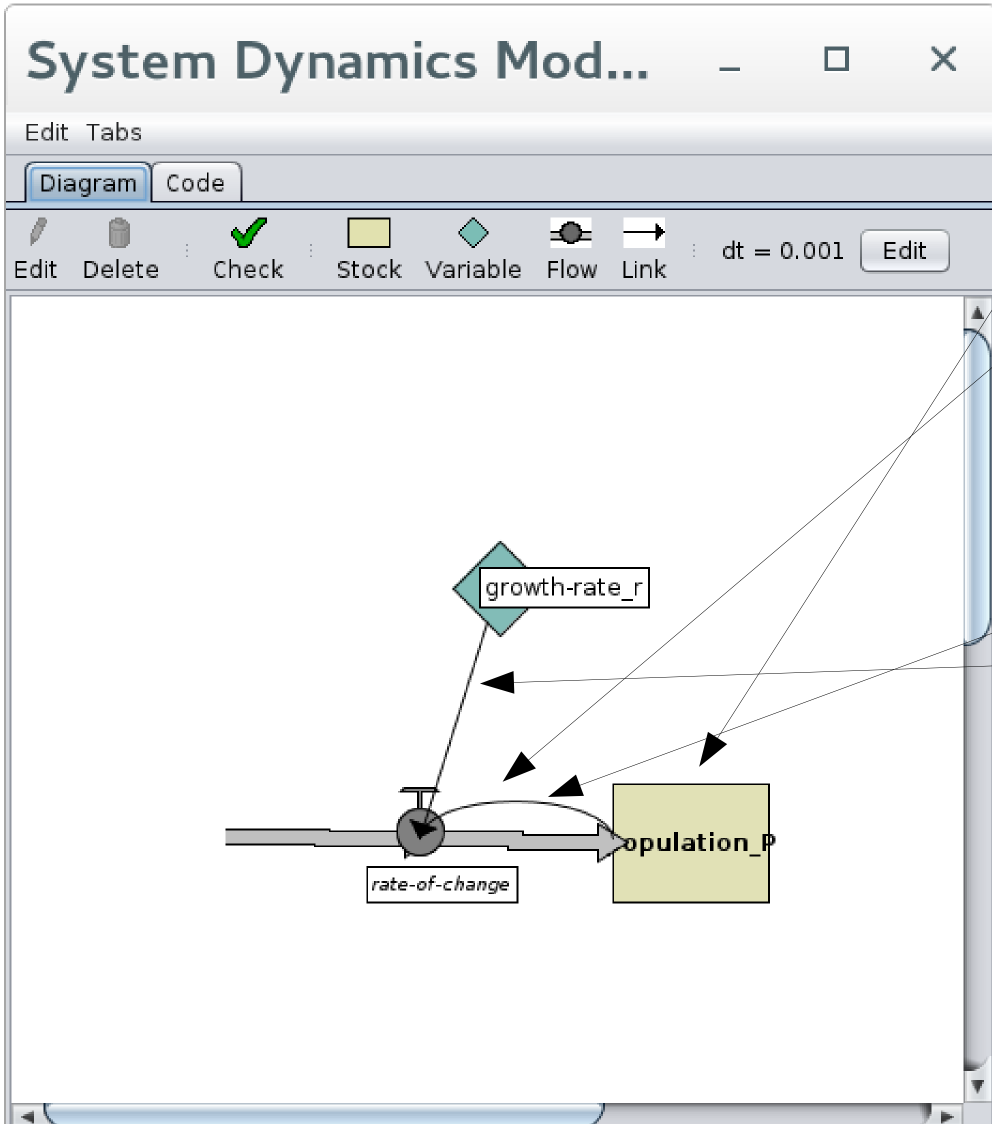
Basic Elements



- Basic elements of SD are **stocks**
  - Collection of stuff, an aggregate. For example: water in a lake, population of sheep, a capital stock ...
- And **flows**:
  - Brings things out of or into a stock. (Modelled as a pipe with a faucet which controls how much runs through it). E.g.: water outflow, sheep births/deaths, investments, ...
- Additional elements are
  - *Variables* = values used in diagrams, can be equations
  - *Links* = makes values of variables available to other elements of the diagram



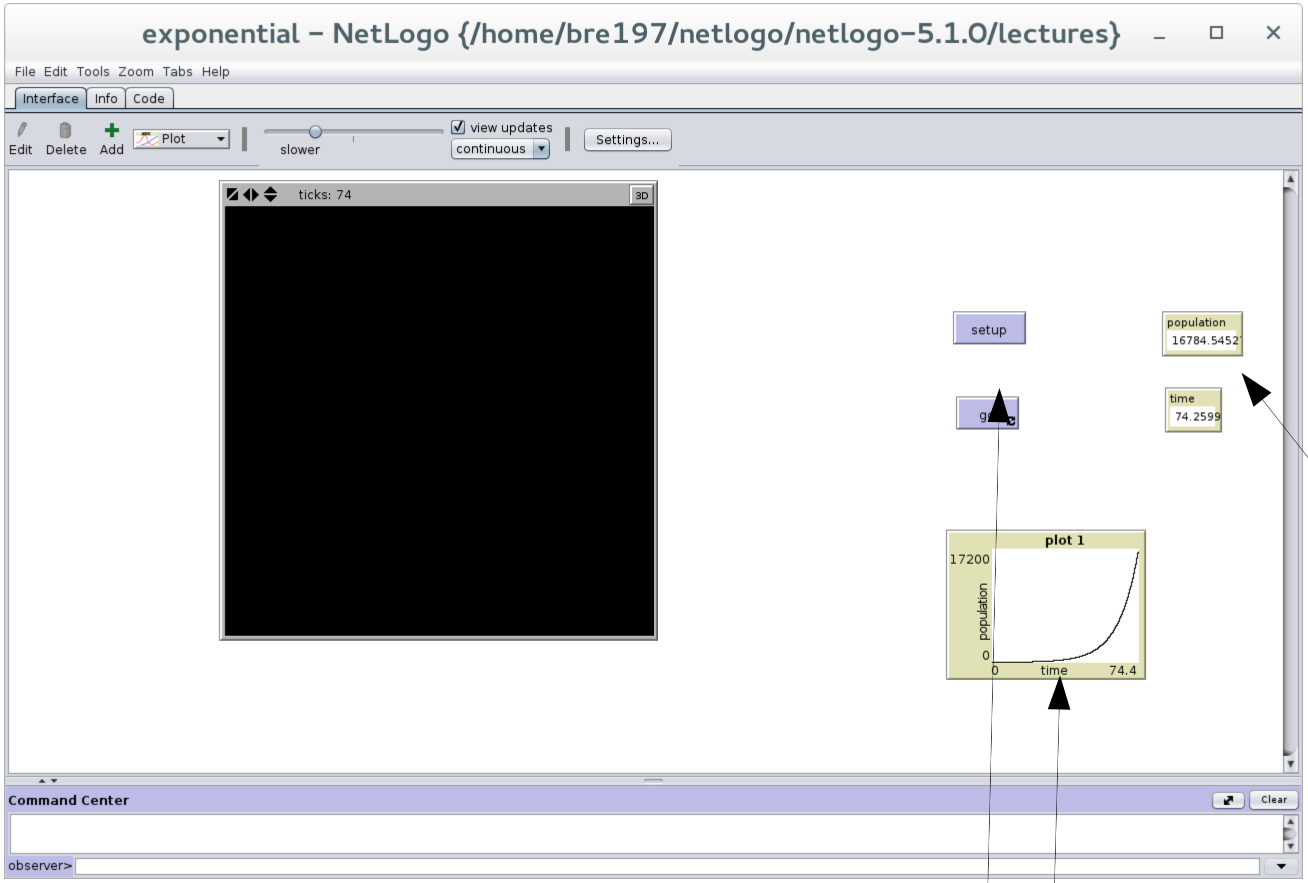
Exponential Growth as an Example of Positive Feedback



$$dP/dt = r\,P$$

- One stock: population P
- There is only an inflow into P, hence we need one “flow pipe”
- Need links, because flow depends on
  - Population
  - Growth rate constant
- In the respective windows:
  - Rate-of-change: growth-rate_r * population_P
  - Growth-rate-r: .1 (change it to expore what happens)

Exponential Growth (2)



file: exponential.nlogo

- In netlogo we also need to create an environment to run the model, i.e.
  - Various buttons – at least a “setup”-button (to initialize) and a “go”-button (to start the model)
  - Some monitors/plots to see what is going on, in this case two monitors to plot population/time and one plot to plot the evolution

What about Logistic Growth?

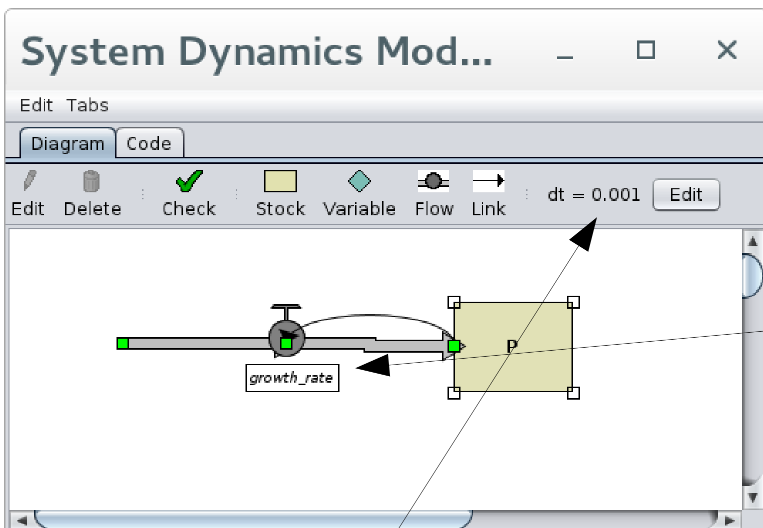
$$dP/dt = r\,P\left(1 - P/K\right)$$

- How does the stocks and flow diagram look like?

What about Logistic Growth?

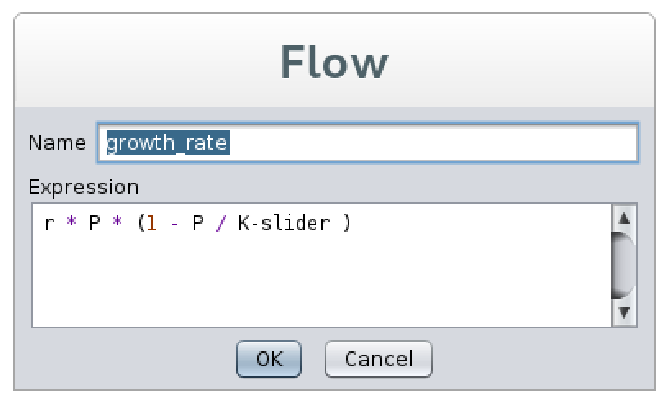
$$dP/dt = r\,P\,(1 - P/K)$$

- How does the stocks and flow diagram look like?



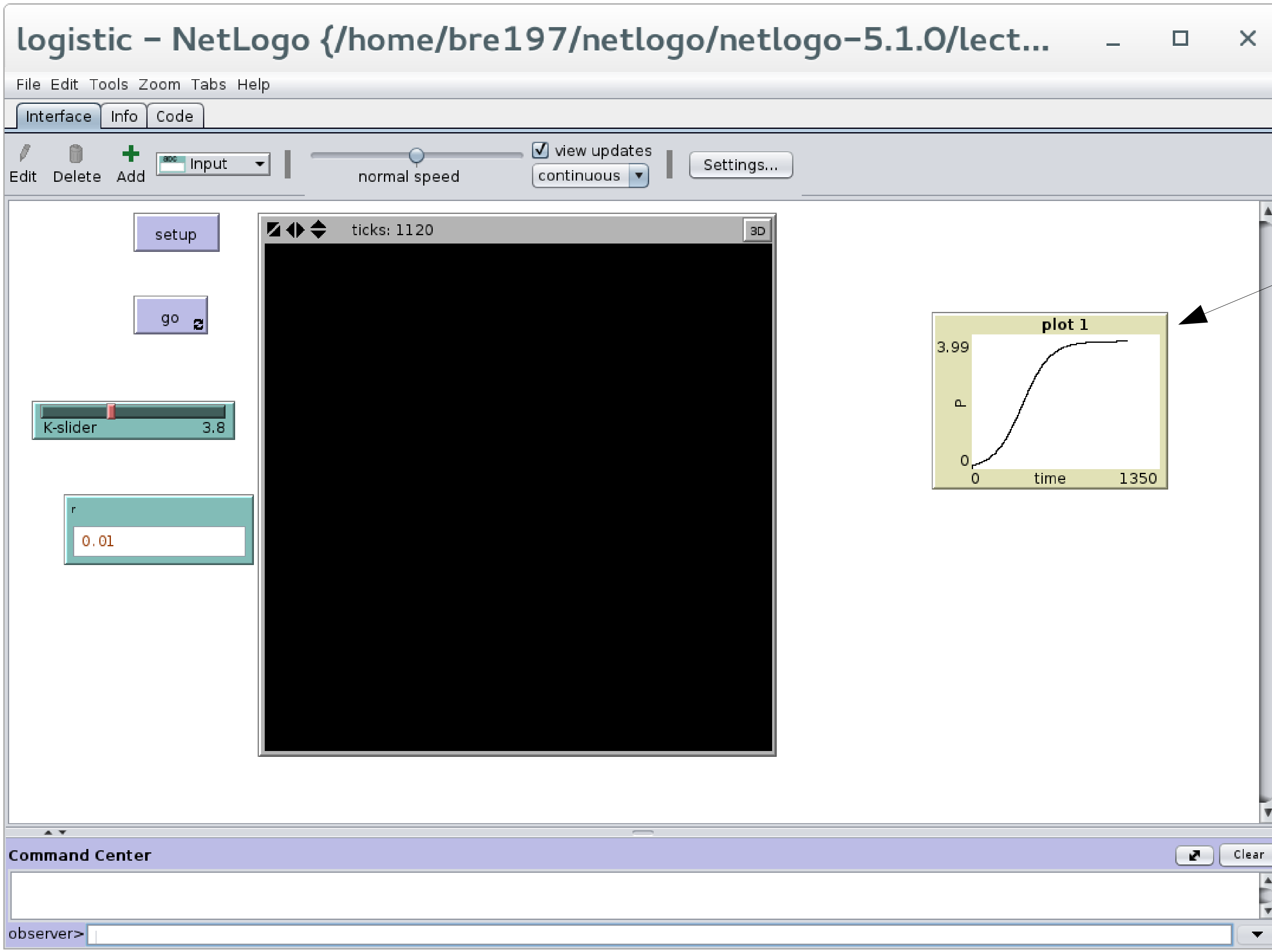
-> essentially same as before

... but the content of the growth_rate pipe has changed and now reads



Also note the small choice of dt

And a Somewhat Fancier Interface



Logistic growth

file: logistic.nlogo – download and play with K and r if you like

The Dynamics of Love Affairs

- Consider Romeo and Juliet and let
  - R(t) be Romeo's love/hate for Juliet (convention being that R<0 is hate, R>0 is love)
  - J(t) be Juliet's love/hate for Romeo
- Consider the following scenario
  - Romeo is in love with Juliet
  - Juliet is “fickle”: the more Romeo loves her, the more she wants to run away and hide
  - When Romeo backs off, Juliet starts to find him attractive again
  - Romeo: mirrors Juliets love, loves her when she loves him and grows cold when she hates him
  - How is this going to end? Can we model it?

The Dynamics of Love Affairs (2)

- More on this in

  Strogatz, S.H. (1988), Love affairs and differential equations, Math. Magazine 61, 35.

- Of course we will use differential equations to solve this problem and then simulate the dynamcis using system dynamics ...
- Equations?

The Dynamics of Love Affairs (2)

- Of course we will use differential equations to solve this problem and then simulate the dynamcis using system dynamics ...
- Equations:

$$dR/dt = a\,J$$

Romeo's love grows in Proportion to Juliet's love

$$dJ/dt = -b\,R$$

Juliet's love grows if R hates her and shrinks the more R loves her

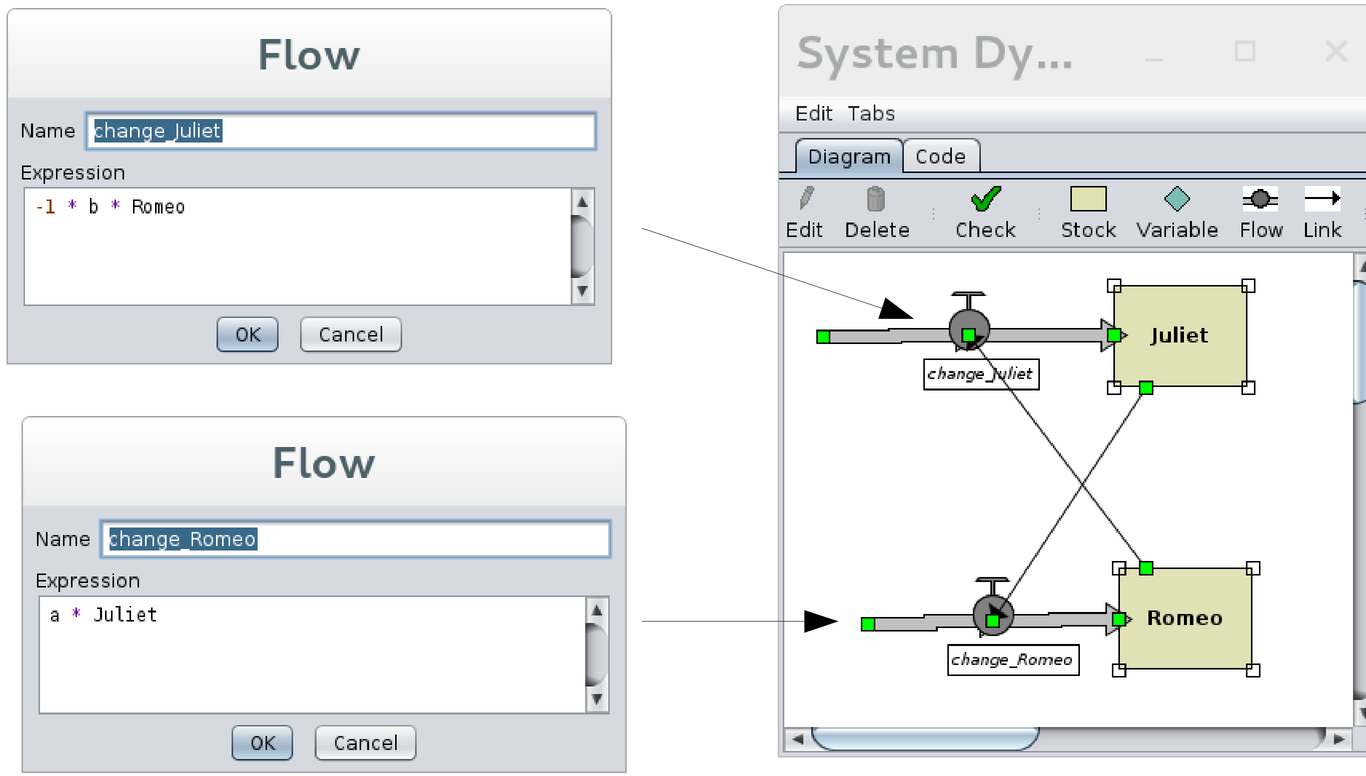
a,b >0: “response coefficients”

- How is this going to end ...

System Dynamics of Love

$$dJ/dt = -b\,R$$

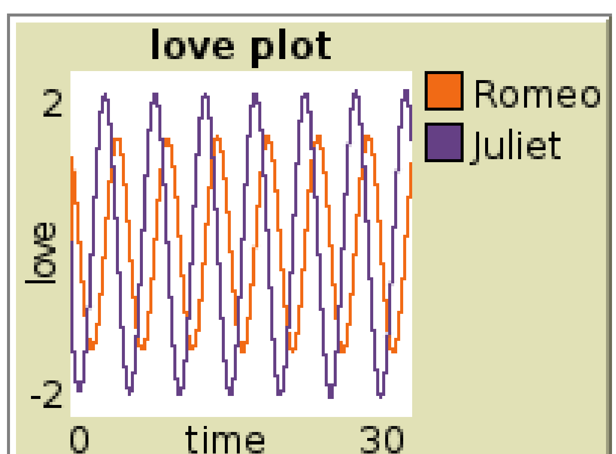
$$dR/dt = a\,J$$



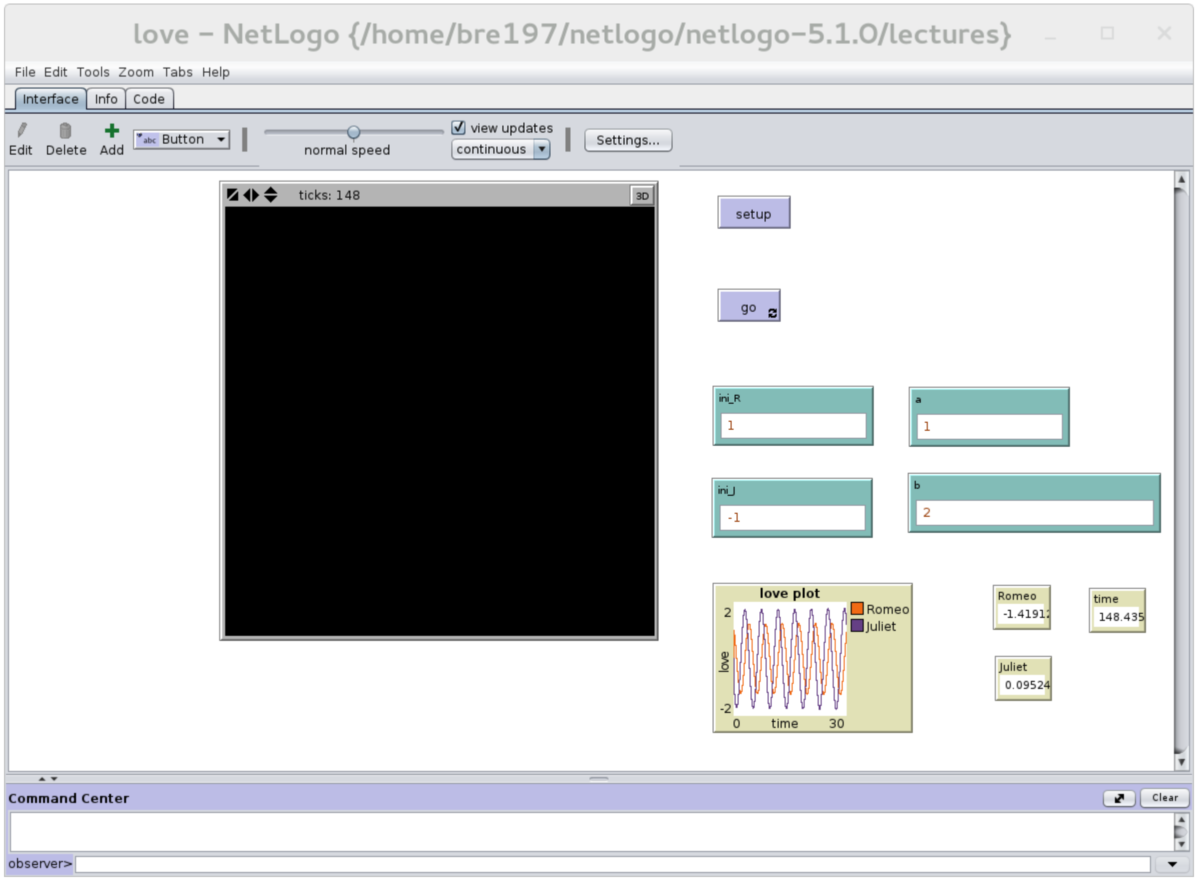
Result for

$$R_0 = 1, J_0 = -1$$

$$a = 1, b = 2$$



The System Dynamics of Love (2)



file: love.nlogo

- What happens if one changes a and b or the initial conditions for J and R?
- How would you have to modify the model if R's (and J's) love would also depend on their own state?
- Play with it and explore ...

Lessons from the Dynamics of Love Affairs

- We investigated a 2d system and found a new type of behaviour: oscillations.
  - When you explore the model you will realize that the structure of these oscillations depends on:
    - The parameters a and b of the model -> frequency
    - The initial conditions -> amplitudes
    - This is typical of linear oscillations, we will briefly revisit this later
- In 2d systems we can find other types of oscillations whose shape only depends on the structure of the equations -> Limit cycles

Wolves and Sheep

- Consider the following system composed of wolves (w) and sheep (s)
- At each time the following processes happen:
  - With a certain probability a sheep gives birth to a new sheep
  - With a certain probability a wolf dies
  - With a certain probability proportional to the number of sheep each wolf “meets” a sheep
    - With probability 1-p this encounter is friendly and both go their ways
    - With probability p the wolf eats the sheep. Upon eating the sheep with probability e the wolf will use the extra calories to give birth to a new wolf

Wolves and Sheep

- Q's:
  - Can wolves and sheep co-exist?
  - Can we understand the dynamics of the population of sheep and wolves over time?
- To explore this it is convenient to write down a system of differential equations that describes the process from the previous slide
  - Let's not worry about averages and just assume populations are large and well-mixed
- Equations?
  - As with Romeo an Juliet we have two variables, s, and w

Equations of Wolves and Sheep

- Sheep:

wolf eats sheep

$$ds/dt = r\,s - p\,s\,w$$

rate of change of sheep

a sheep gives birth to a new sheep with with rate r

“certain probability proportional to the number of sheep”

every wolf has a chance to meet a sheep

- Wolves

$$dw/dt = e\,p\,w\,s - d\,w$$

rate of change of wolves

if a wolf eats a sheep with a certain probability it gives birth to a new wolf

every wolf has a certain chance to die per unit of time

- Parameters: r, p, e, d ... what is their interpretation? Do we need all of them?

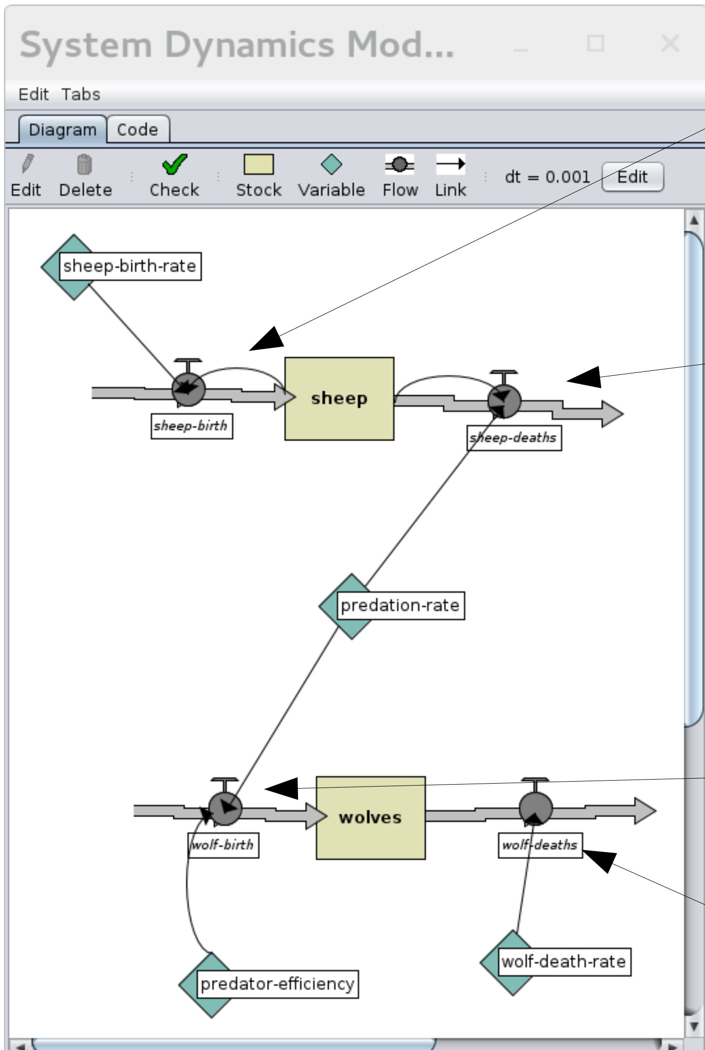
Equations of Wolves and Sheep (2)

- What is the main difference between the equations of wolves and sheep and those of love between Romeo and Juliet?

Equations of Wolves and Sheep (2)

- What is the main difference between the equations of wolves and sheep and those of love between Romeo and Juliet?
  - Equations of R + J are linear – can treat them analytically (later)
  - Equations of W + S are non-linear – analytical treatment much harder (and near impossible if we have many equations)
- Let's solve them numerically with system dynamics to see what is going on ...
  - any ideas?

A System Dynamics Model of Wolves and Sheep



sheep-birth=sheep-birth-rate * sheep

sheep-deaths=p * sheep * wolves

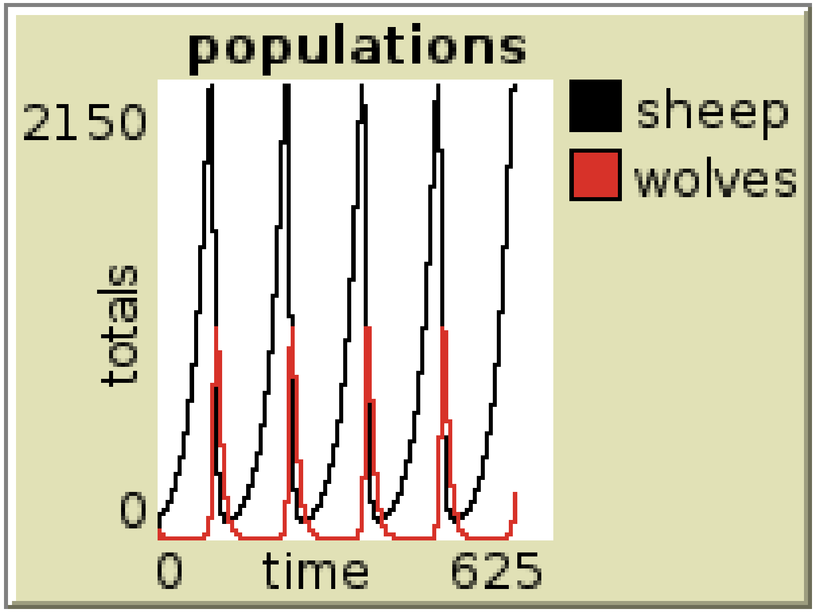
wolf-births=p * e * sheep * wolves

wolf-deaths=d * wolves

Results

file: sheep.nlogo

Example dynamics for r=0.04, p=0.0003, e=0.8, d=0.15, and initial conditions s=100, w=30



- Also in this example we find oscillations
- However, frequency and shape are independent of initial conditions -> this is a self-sustaining oscillation also called a “limit cycle”

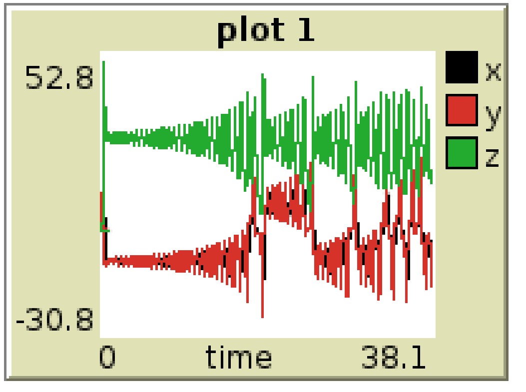
The Lorenz System

- To see what other kinds of dynamics are possible if we increase the dimension and consider 3d systems, have a look at the Lorenz system originally proposed as equations describing the dynamics of atmospheric convection

$$dx/dt = \sigma(y - x)$$

$$dy/dt = x(\rho - z) - y$$

$$dz/dt = x\,y - \beta\,z$$



lorenz.nlogo

Summary

- System dynamics gives an easy-to-use graphical interface to implement systems of ODEs in intuitive language
- Also provides a neat link to graphical output, good tool for model development and problem scoping
- Commercial packages like Stella or Vensim work in a very similar way, but provide some enhanced functionality, e.g. delays, better GUI, etc.
- Word of warning:
  - Integrators in most of these packages are fairly unsophisticated (netlogo only uses Euler!), so artifacts due to numerical instabilities are an issue!