models

A simple approach to complex problems

Course objectives, structure, & expectations

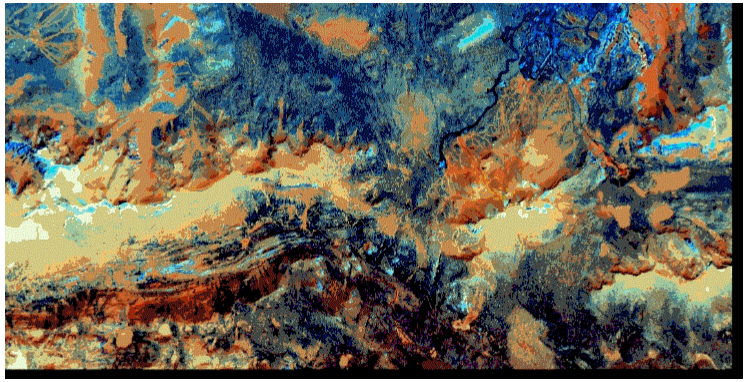
- Develop strong foundation for agent-based ecological modeling through hands-on learning
- Learn to conceptualize agent-based systems, then prototype a model of your own design

- Structure is designed as lecture then hands-on learning. You will be working through some exercises in the textbook as well as developing your own model.

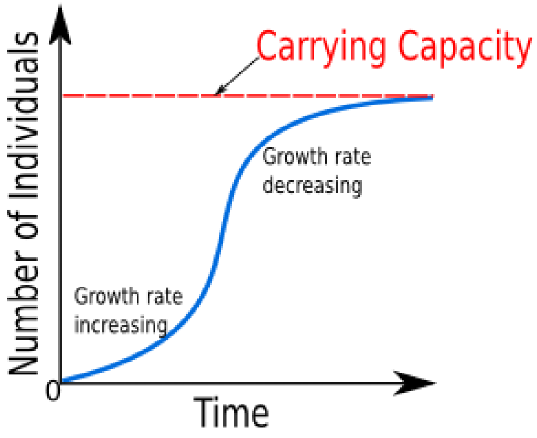
- Students are expected to be able to troubleshoot using common tools including each other. This course is intense and we want to have a truly collaborative environment.

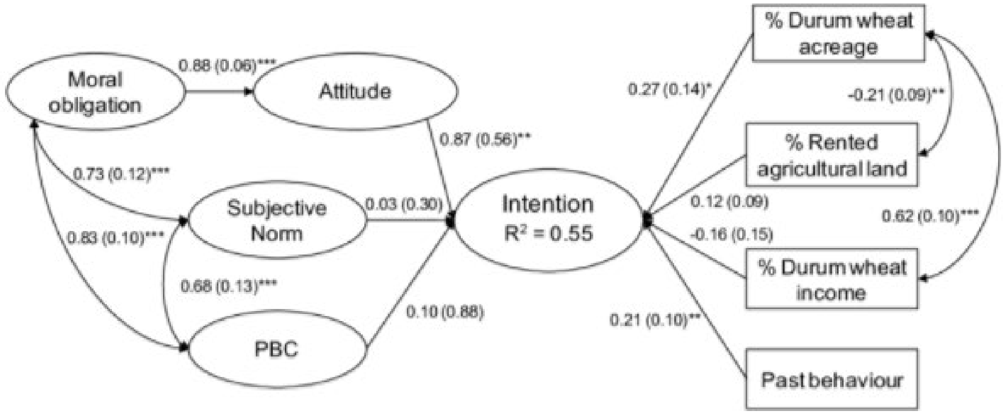
Overview

- What is an “ecological model”?
- Types of models & how agent-based models are different

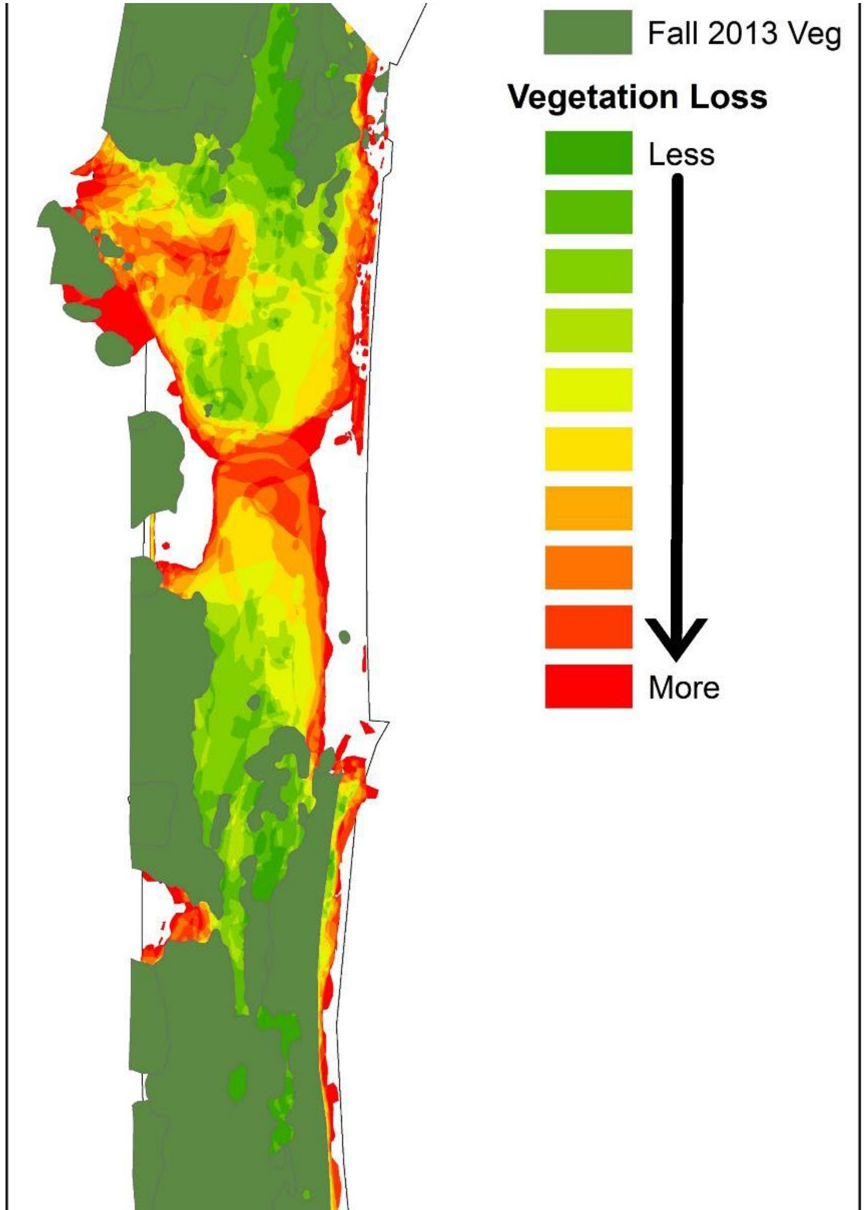


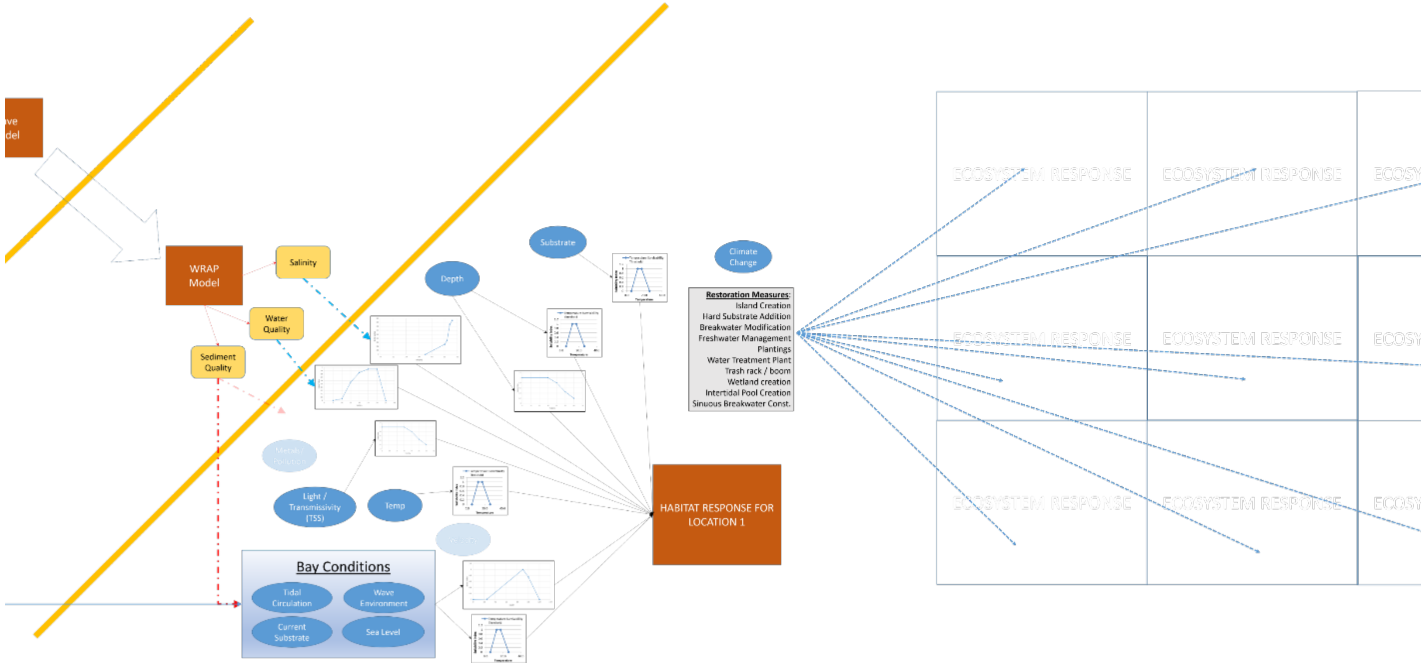
$$
\begin{aligned}
f'(3) &= \lim_{h\to 0}\frac{(3+h)^2-3^2}{h} \\
&= \lim_{h\to 0}\frac{9+6h+h^2-9}{h} \\
&= \lim_{h\to 0}\frac{6h+h^2}{h} \\
&= \lim_{h\to 0}(6+h) \\
&= 6
\end{aligned}
$$





$$
\frac{\partial \tilde{u}}{\partial t}+\tilde{u}\frac{\partial \tilde{u}}{\partial x}+\tilde{w}\frac{\partial \tilde{u}}{\partial z} = -\frac{\tau_{wx}}{\rho_o b}-g\frac{\partial \eta}{\partial x}-\frac{g}{\rho_o}\int_z^{\eta}\frac{\partial \rho}{\partial x}\mathrm{d}\zeta + \frac{1}{b}\frac{\partial}{\partial x}\left(bA_x\frac{\partial \tilde{u}}{\partial x}\right)+\frac{1}{b}\frac{\partial}{\partial z}\left(bA_z\frac{\partial \tilde{u}}{\partial z}\right) \tag{2}
$$





Definition of models

You are a modeler.

- Commonly used in the nerd world:
  - “abstractions of reality”
  - “a representation of a system for a purpose”
  - “a way to represent a system for the purposes of reproducing, simplifying, analyzing, or understanding it”

- How would you tell your family what a model is?

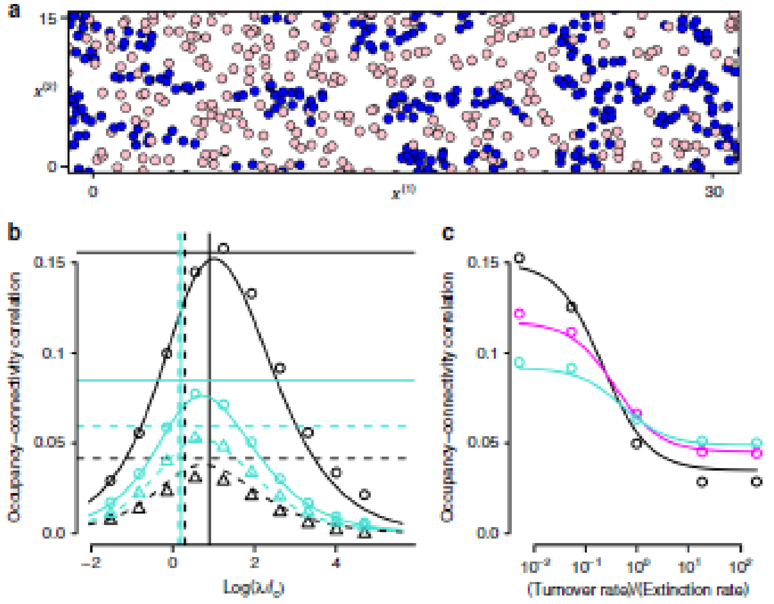
Definition of model for this class

Numerical representation of an ecological system implemented in software

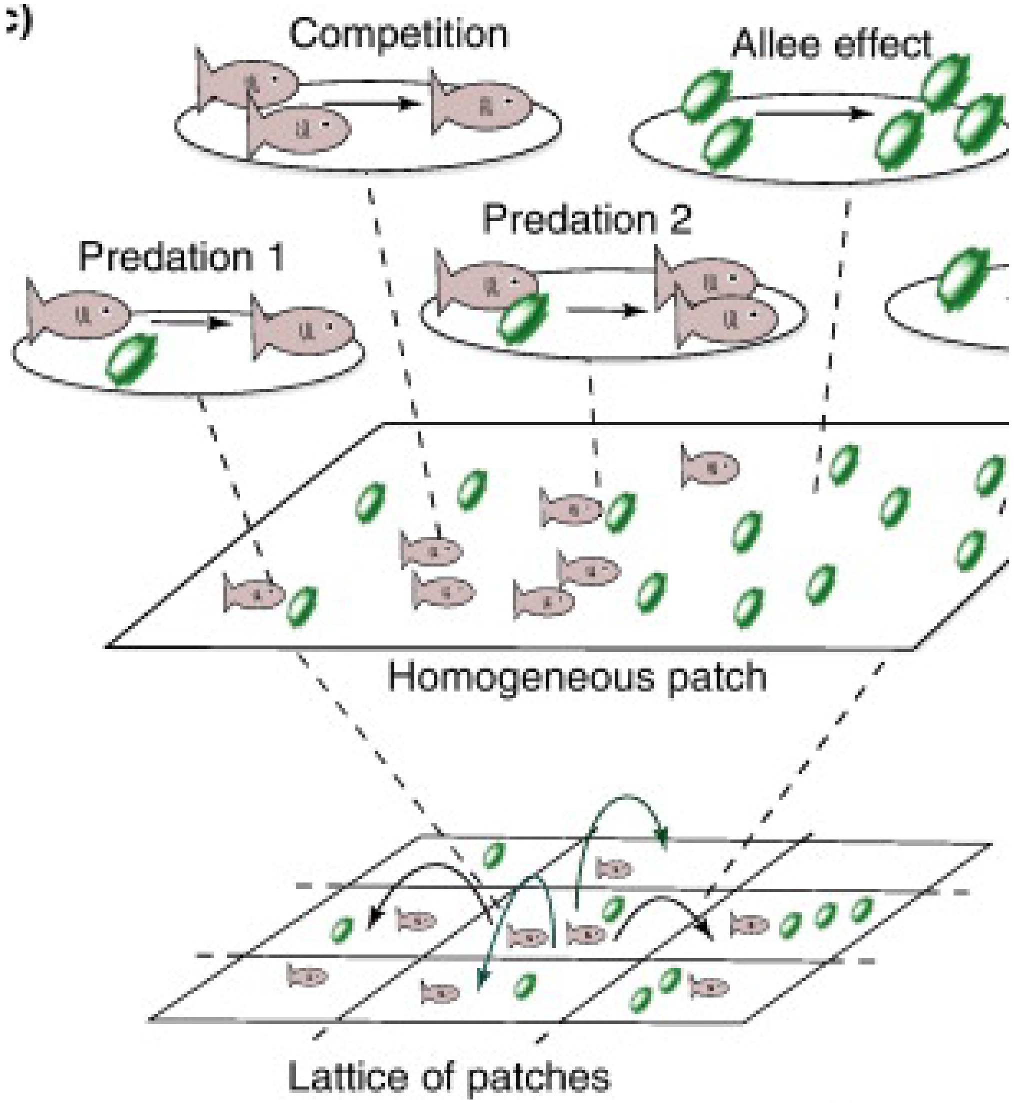
SIDE NOTE: **not** software applicable to any situation

Ecological modeling

Represents environment based on point of view of model builders



- Ecology is a question-driven discipline
- Ecosystems are inherently complex, interdependent systems with lots of uncertainty
- Models are developed to explore complexity, interactions, forecasts
  - Each species reacts differently to stimuli
  - Multiple approaches for a single problem
    - Habitat v. population v. meta-population v. community v. ecosystem
    - Different types of models can be used for same question
  - Trade-offs: detail, scale, expense



Why do we develop models?

Models!

To increase understanding & develop theory

To organize thinking

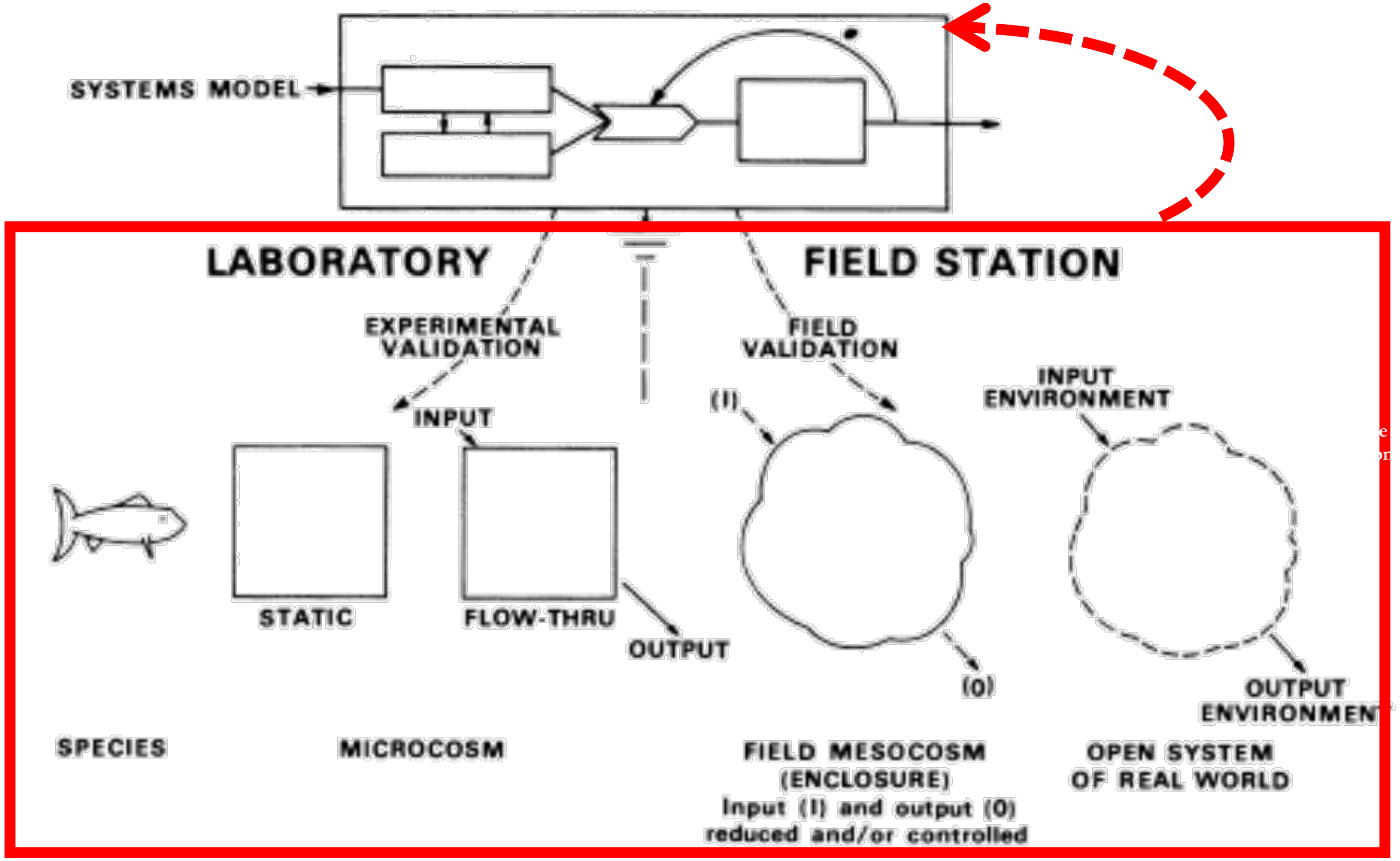
To forecast future conditions

To inform decision-making

Provide a platform for critical thinking

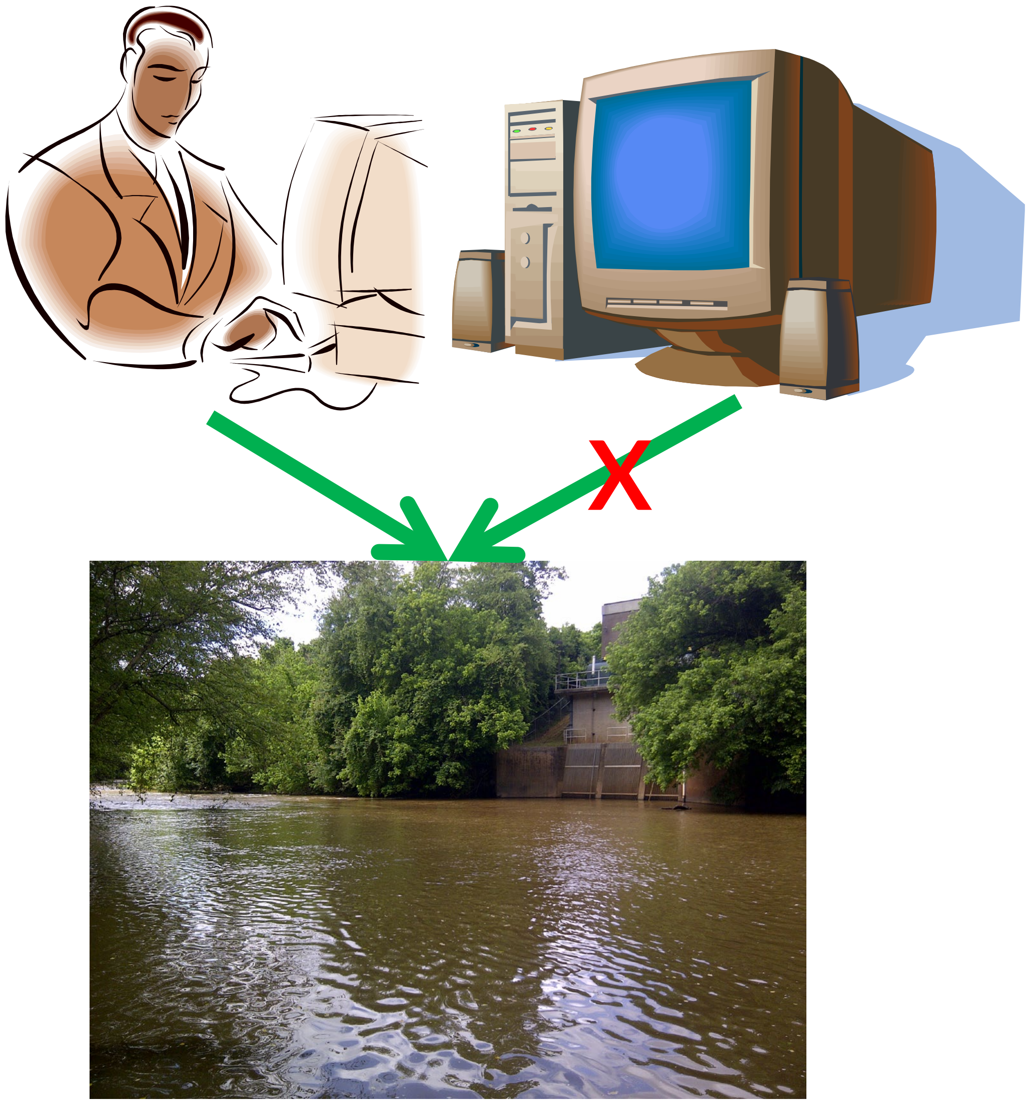
Is modeling science?

Is science modeling?



- Modeling process is based on scientific method
  - A hypothesis is a mental/written model
- Odum (1984) describes models as an important part of scientific method
- Software has made it easy to ignore scientific practice
- Standard techniques and documentation are important for scientific models

Models are never



Answers or Decisions

People make decisions. Models inform people.

Reality

Inherently a model is an abstraction of reality.

Table 1. Description of model types often used for modeling environmental benefits.

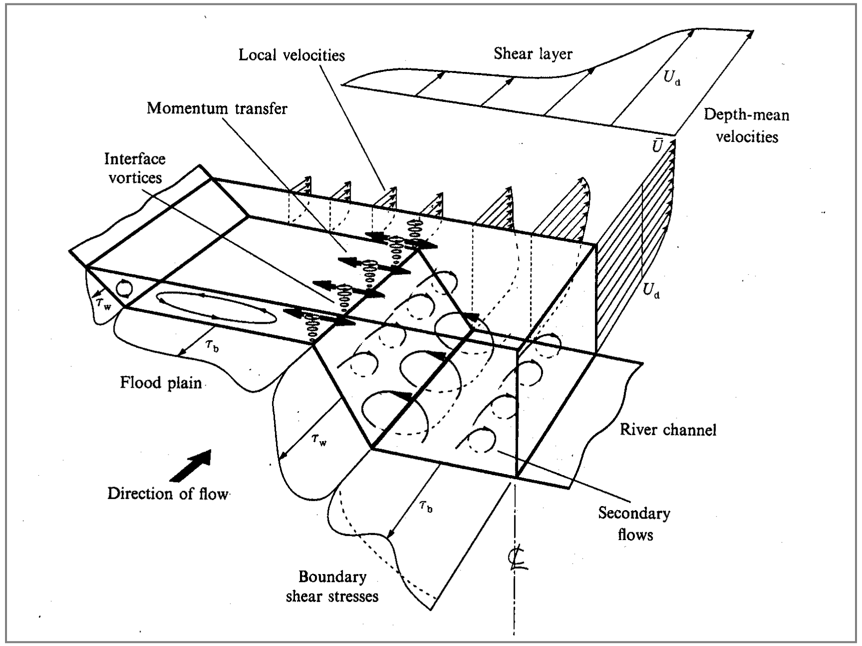
Model	General Use	Example
Analytical	Systems where solution to closed form equations represent system	Population growth, Lotka-Volterra models
Conceptual	Diagramming relationships among components, organizing information, determining data needs	CEMCAT (see Fischenich 2008, for more examples)
Index	Determining habitat quality across a landscape, relates species presence to environmental variables	HSI, HGM
Simulation	Modeling dynamics of complex systems that have multiple factors interacting across scales, often have spatial components	Agent-based models, ADH-CASM, ELAM, ICM, system dynamic models
Statistical	Analysis of datasets to determine distributional properties of the data	ANOVA, goodness-of-fit, regression, t-test,
Spatial	Projects where particular spatial attributes are important can be incorporated into simulation models	GIS, EDYS

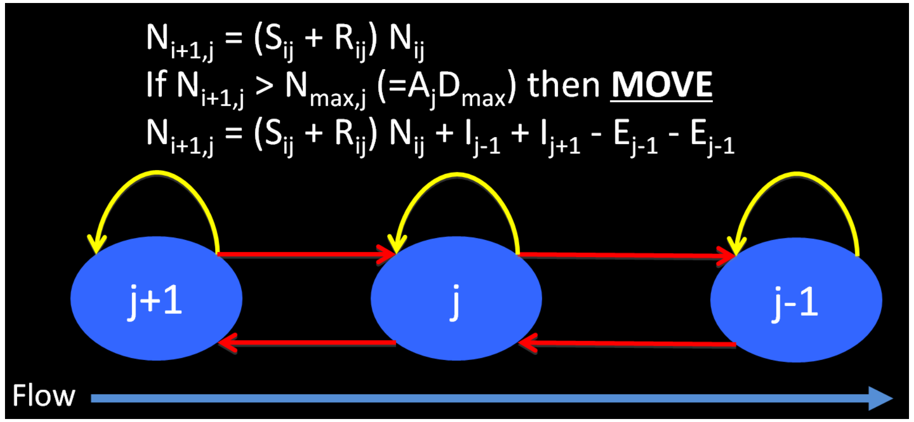
Analytical Models

Solution of closed form equations representing the system

$$\begin{bmatrix} n_0 \\ n_1 \\ \vdots \\ n_{\omega-1} \end{bmatrix}_{t+1} = \begin{bmatrix} f_0 & f_1 & f_2 & f_3 & \dots & f_{\omega-1} \\ s_0 & 0 & 0 & 0 & \dots & 0 \\ 0 & s_1 & 0 & 0 & \dots & 0 \\ 0 & 0 & s_2 & 0 & \dots & 0 \\ 0 & 0 & 0 & \ddots & \dots & 0 \\ 0 & 0 & 0 & \dots & s_{\omega-2} & 0 \end{bmatrix} \begin{bmatrix} n_0 \\ n_1 \\ \vdots \\ n_{\omega-1} \end{bmatrix}_{t}$$

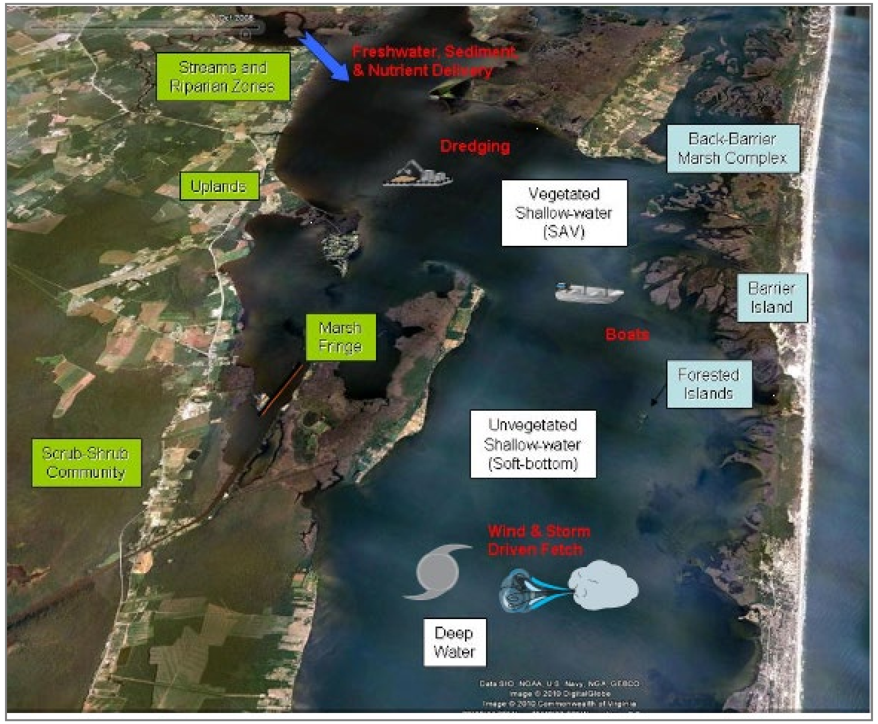
$$\rho g H S_0 + \frac{\partial}{\partial z}\left(\rho\lambda\sqrt{\frac{f}{8}}H^2 U_d \frac{\partial U_d}{\partial z}\right) - \rho U_d{}^2 \frac{f}{8}\sqrt{1+S_{0z}{}^2} = \Gamma$$

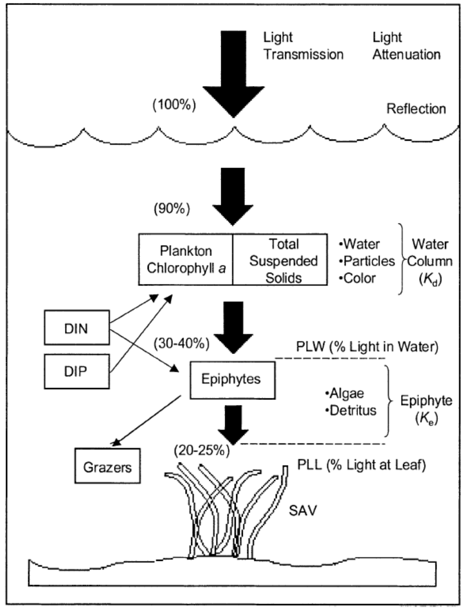


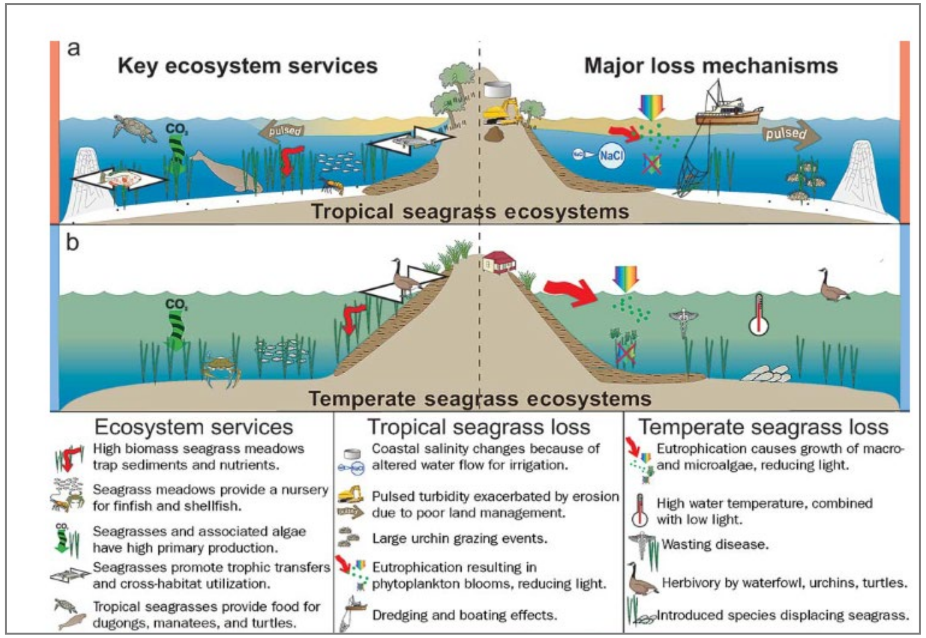


Conceptual Models

Diagramming relationships among components, organizing information, determining data needs





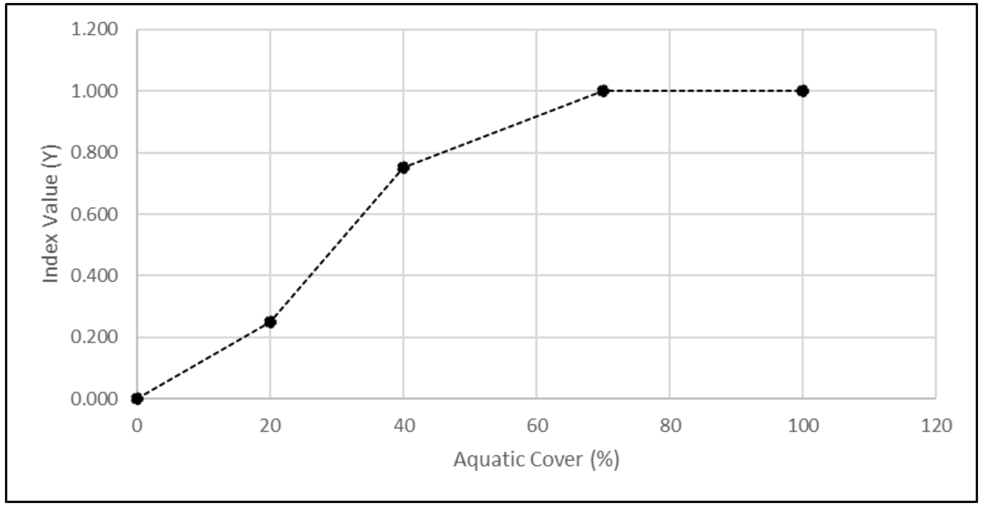


Index Models

Determining ecosystem quality relative to environmental variables

Used by agencies for restoration planning

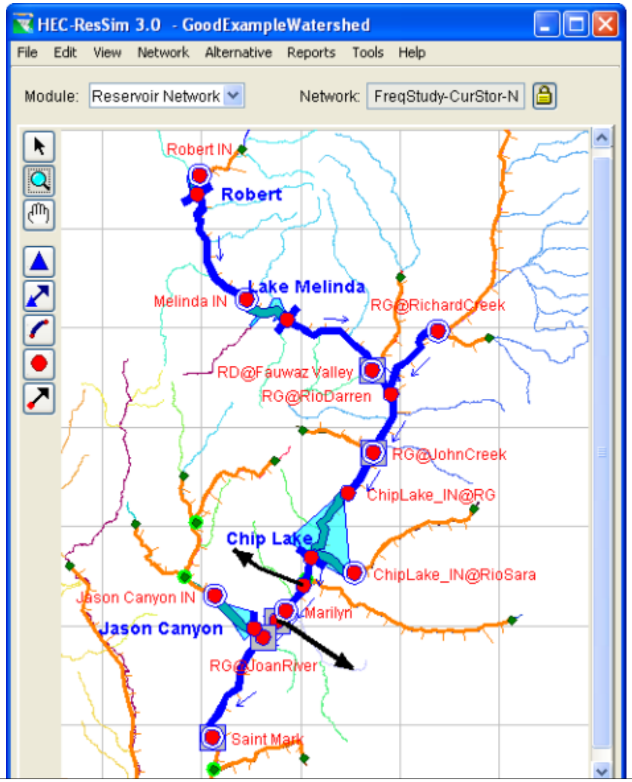
Simple models based on linear step functions

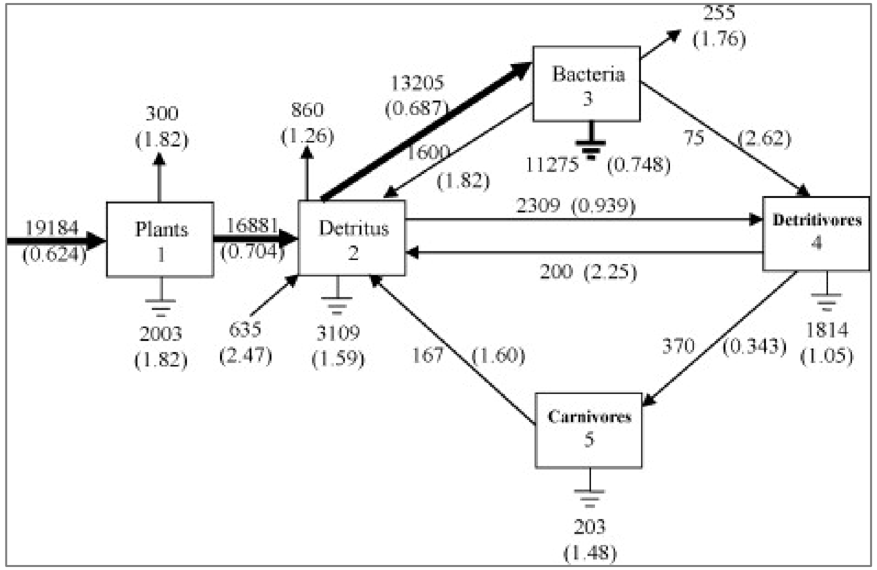


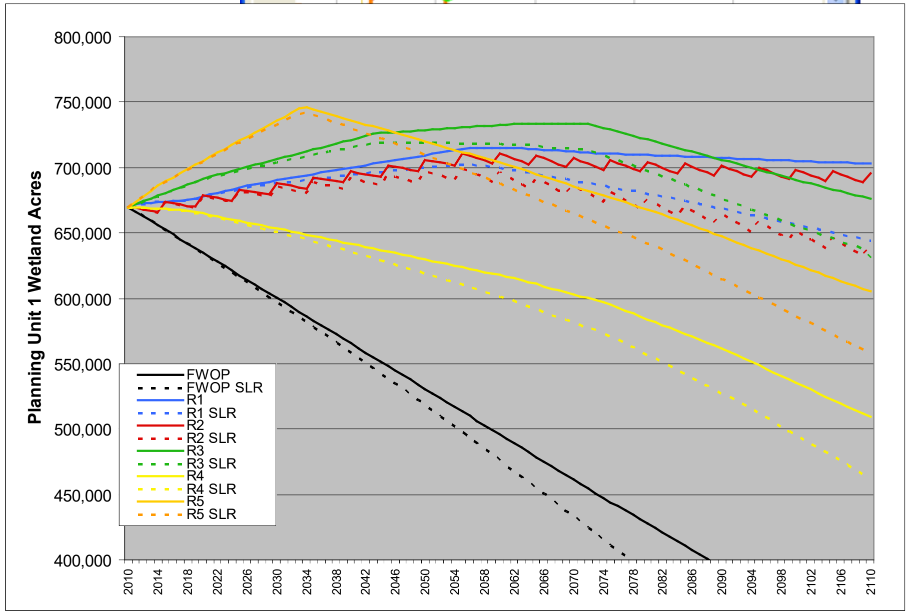
Values	Intercept	Slope	Equation
0 - 20	0.00	0.0125	Y= 0 + (0.0125 * Aquatic Cover (%))
20 - 40	-0.25	0.0250	Y= -0.25 + (0.025 * Aquatic Cover (%))
40 - 70	0.42	0.0083	Y= 0.42 + (0.0083 * Aquatic Cover (%))
70 - 100	1.00	0.0000	Y= 1 + (0 * Aquatic Cover (%))
100 -	0.00	0.0100	Y= 0 + (0.01 * Aquatic Cover (%))
-			

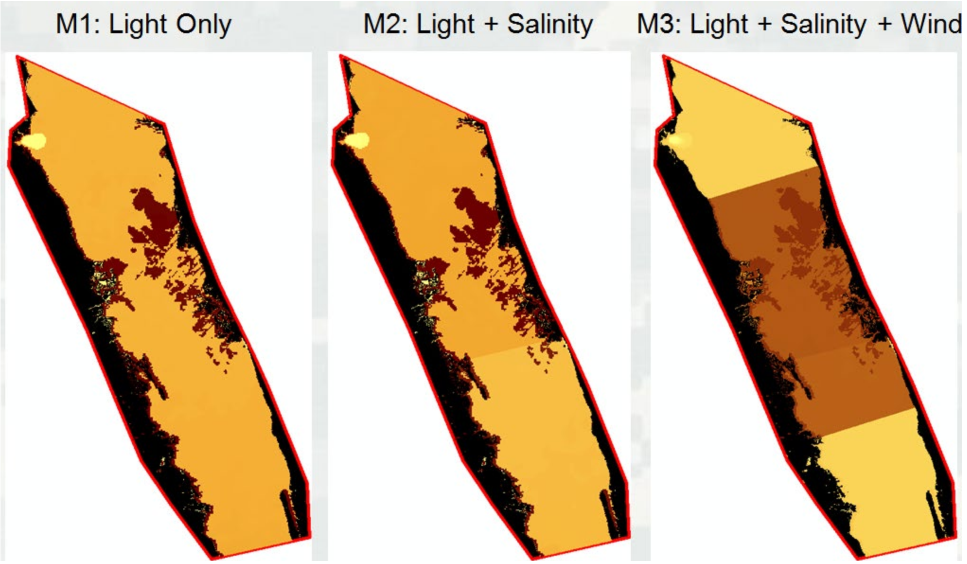
System Dynamics

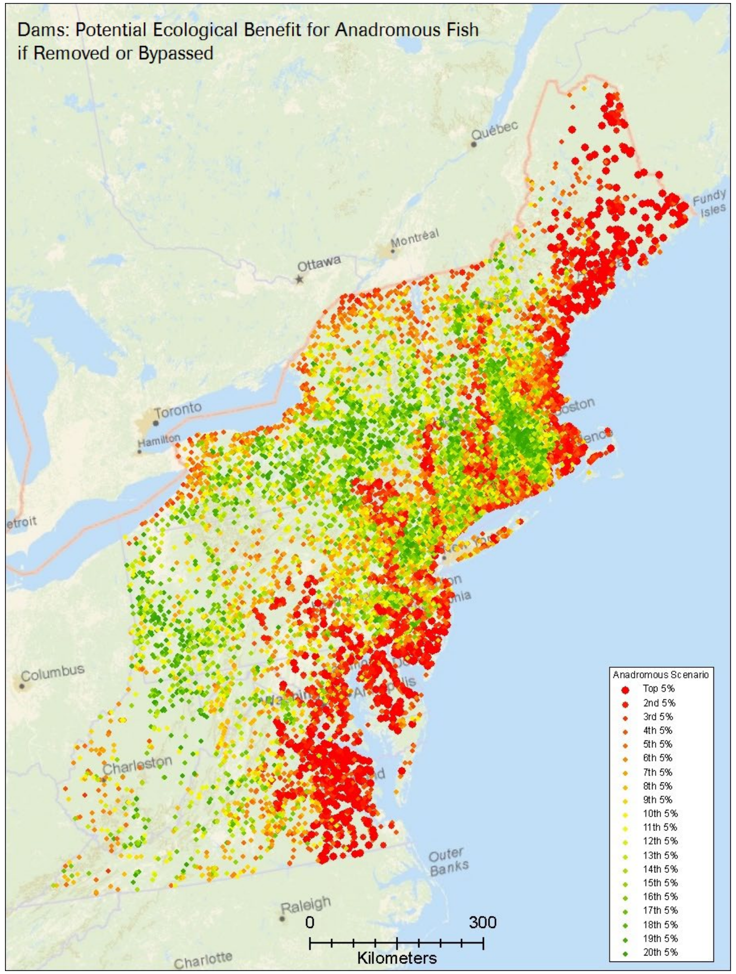
Modeling dynamics of complex systems by focusing on state of system as summarized variables (e.g., overall biomass, population size, etc)









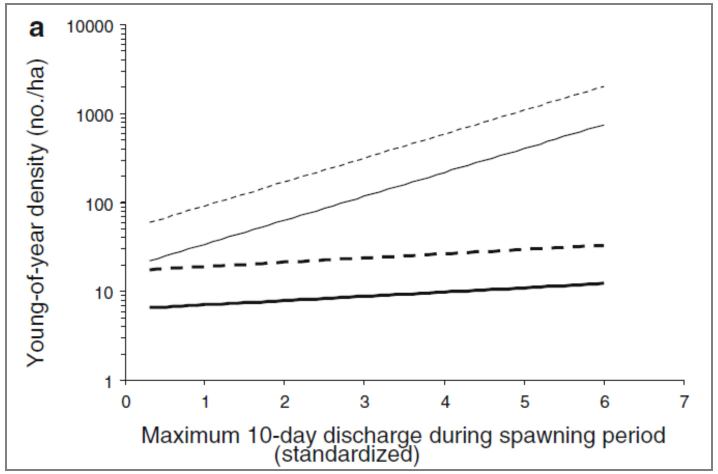


Spatial Models

Combination of spatial attributes often coupled with simulation

Statistical Models

Analysis of datasets to determine distributional properties of the data



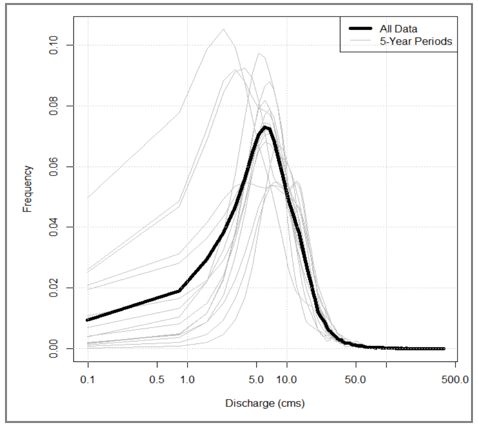
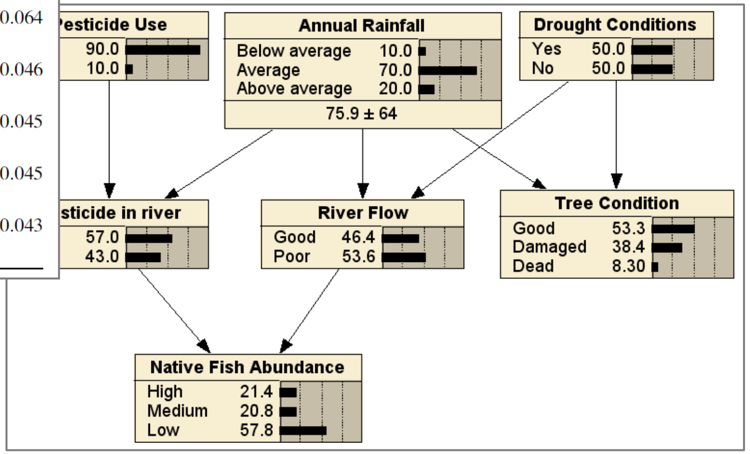
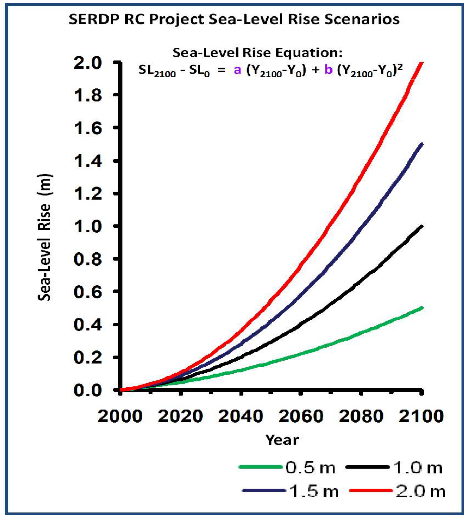


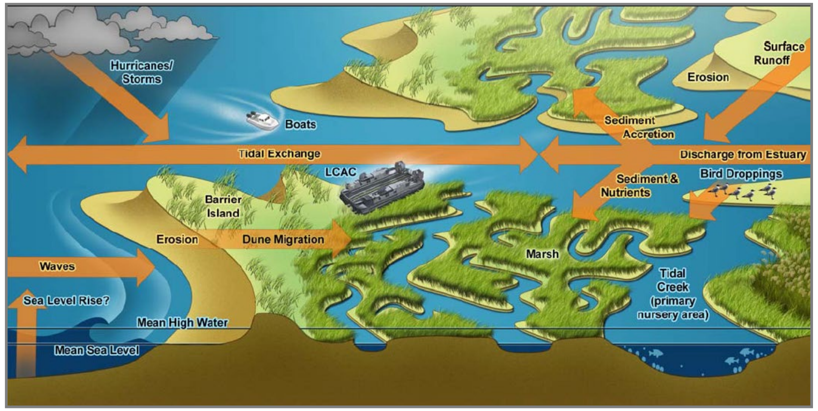
Table 3 Model selection criterion for the top ten Pradel reverse time, tag-recapture models of mussel survival (*Phi*), capture probability (*p*) and recruitment (*f*)

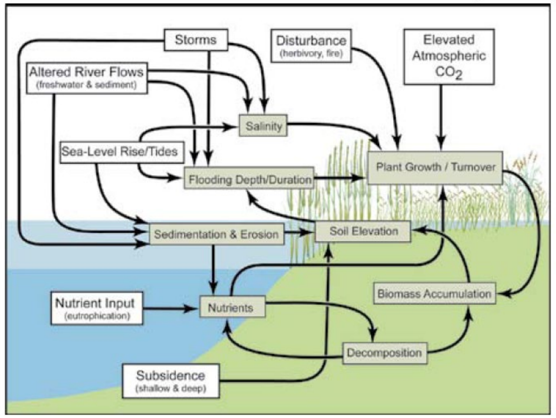
Model	−2Log *L*	*K*	Δ AIC$_c$	w_i
Phi(Maximum 10-day summer discharge), *p*(Discharge during sampling, species) *f*(Median summer discharge lag 2)	4574.1	8	0.000	0.071
Phi(Maximum 10-day summer discharge), *p*(Discharge during sampling, species) *f*(Maximum 10-day summer discharge lag 2)	4574.1	8	0.007	0.070
Phi(Maximum 10-day summer discharge), *p*(Discharge during sampling, species) *f*(Minimum 10-day spring discharge lag 2)	4574.1	8	0.056	0.069
Phi(Maximum 10-day summer discharge), *p*(Discharge during sampling, species) *f*(Minimum 10-day summer discharge lag 2)	4574.1	8	0.065	0.068
Phi(Maximum 10-day summer discharge), *p*(Discharge during sampling, species) f(Median spring discharge lag 2)	4574.2	8	0.120	0.067
Phi(Maximum 10-day summer discharge), *p*(Discharge during sampling, species) *f*(Maximum 10-day spring discharge lag 2)	4574.3	8	0.204	0.064
Phi(Median summer discharge), *p*(Discharge during sampling, species) *f*(Median spring discharge lag 2)	4574.9	8	0.848	0.046
Phi(Median summer discharge), *p*(Discharge during sampling, species) *f*(Maximum 10-day spring discharge lag 2)	4575.0	8	0.896	0.045
Phi(Median summer discharge), *p*(Discharge during sampling, species) *f*(Minimum 10-day spring discharge lag 2)	4575.0	8	0.897	0.045
Phi(Median summer discharge), *p*(Discharge during sampling, species) *f*(Maximum 10-day summer discharge lag 2)	4575.0	8	0.988	0.043

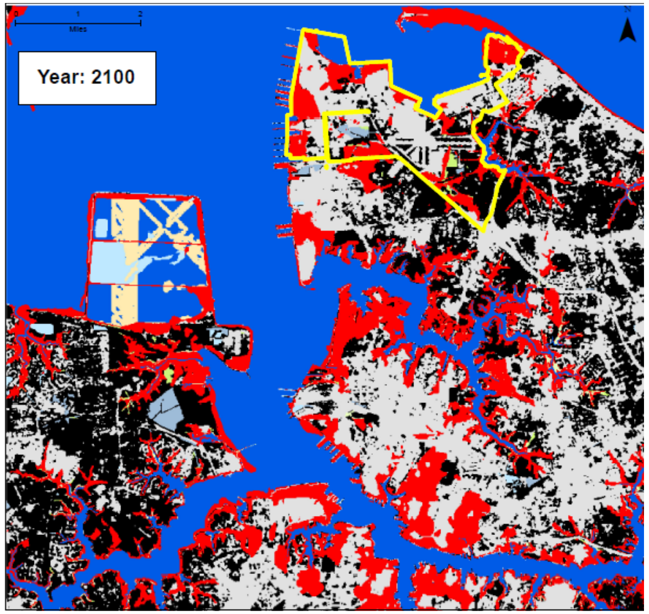


Integrated Modeling









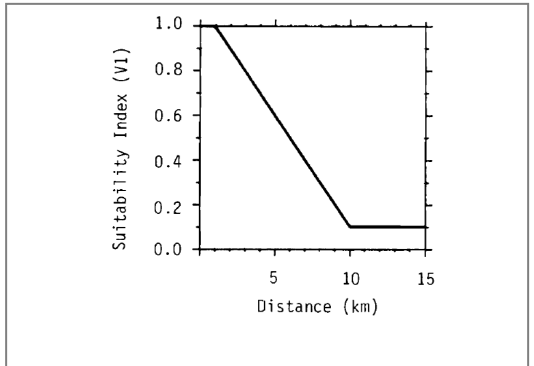
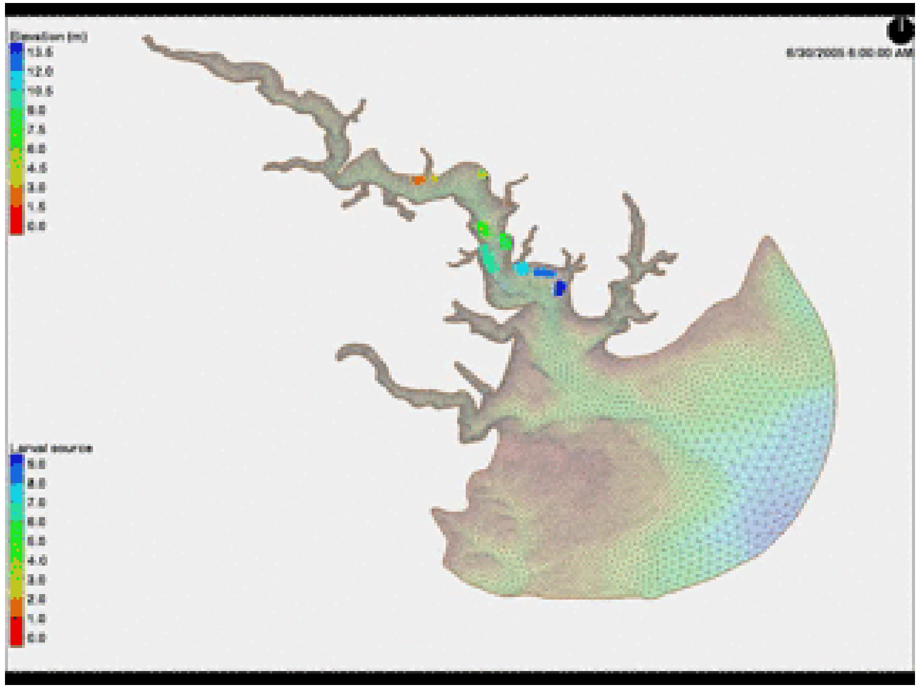


Figure 2. Distance between foraging areas and heronry sites modifies SI values.

Agent-based Models



Tries to mimic real system by capturing complexity. Builds on concepts of individual interactions creating emergent system properties.

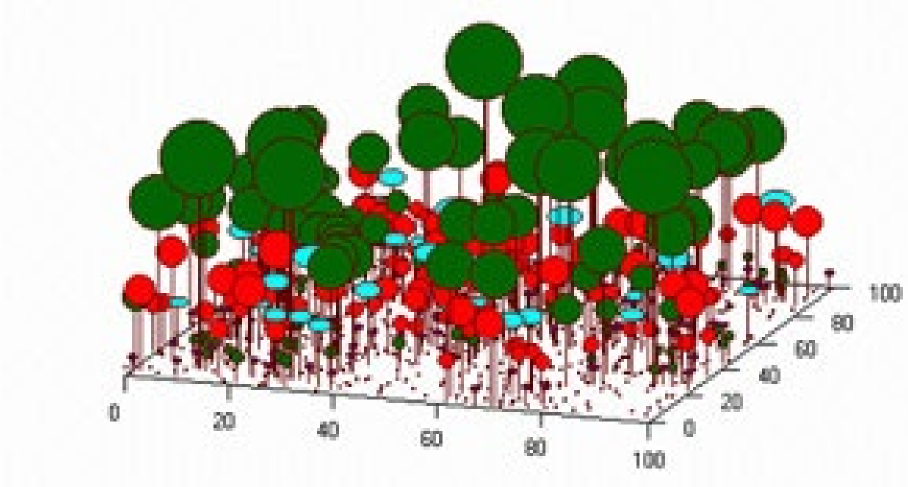
Different from each other & independent

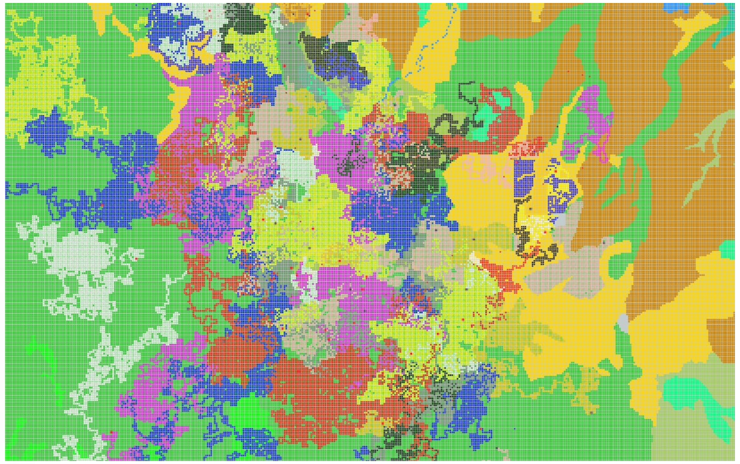
Model across levels!

Species-Species
Species-Environment

Characteristics

- *Unique* agents are *autonomous* and interact with *others* and their *environment*
- System-level properties *emerge* from low-level behaviors





19

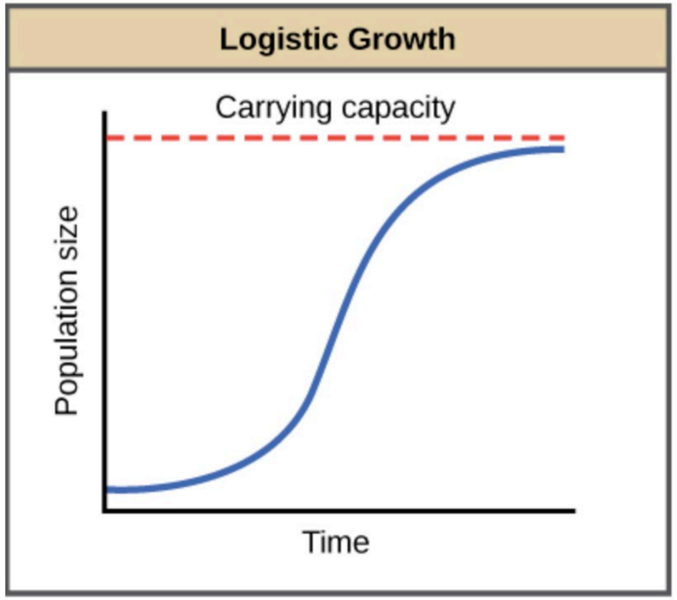
Differences between agent- & equation-based

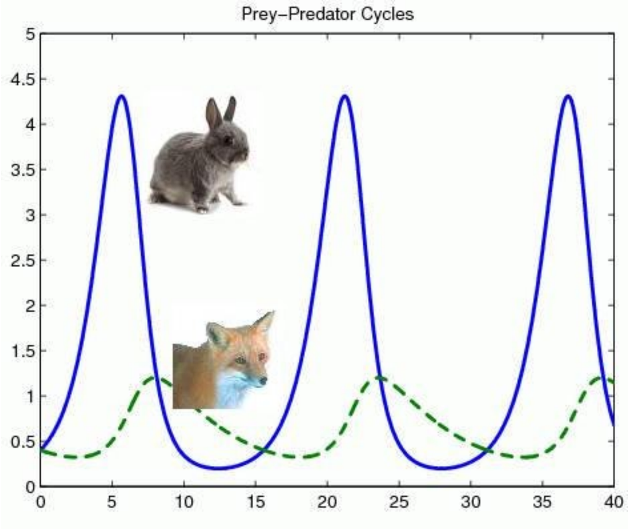
Equation-based modeling: assumes homogeneity and continuity.

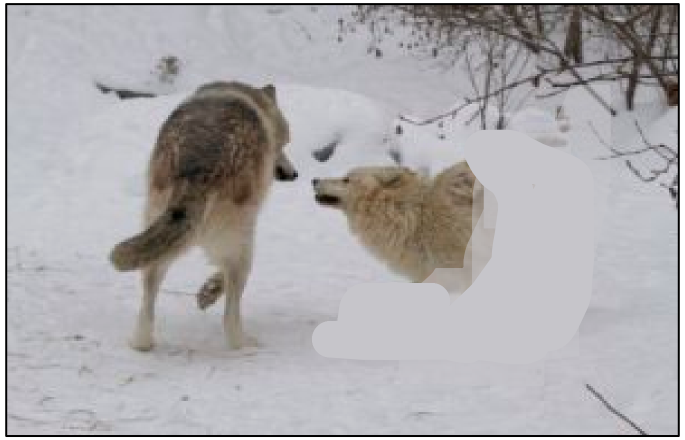
$$\frac{dN}{dt} = rN\left(\frac{K-N}{K}\right)$$

Lotka-Volterra predator-prey equations:

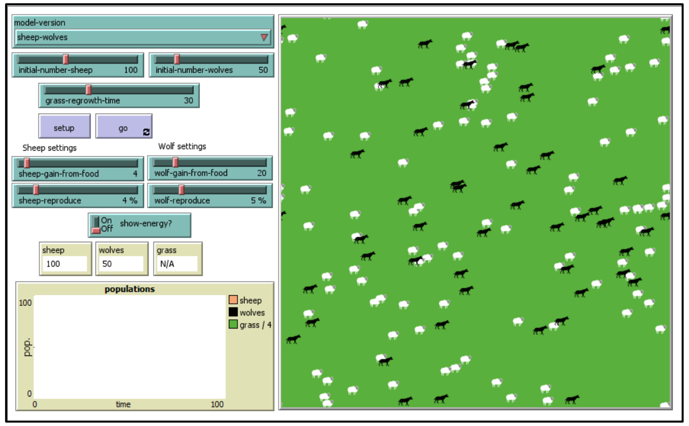
$$dN/dt = rN - aNP$$
$$dP/dt = baNP - mP$$







Agent-based: assumes variability and randomness, discrete entities (i.e., can't have ¼ a wolf)



Why Agent-based Models?

Agent-based models are computationally expensive, require some programming skills, and a completely novel way to analyze them.

So why would we use them?

- We think of the world more complexly and consider interactions
- Can explore what happens to systems because of individuals, and to individuals because of systems (feedbacks and thresholds)
- Can use rules instead of equations to explain phenomena (e.g, don’t have to know differential equations). Language is much closer to natural language.
- Enables analysis of big data and integrated approaches.

21

Main uses for ABMs

Description – describes real world system

Explanation – identifies essential mechanisms, explicates mechanisms of emergence

Experimentation – create a virtual laboratory for testing hypotheses

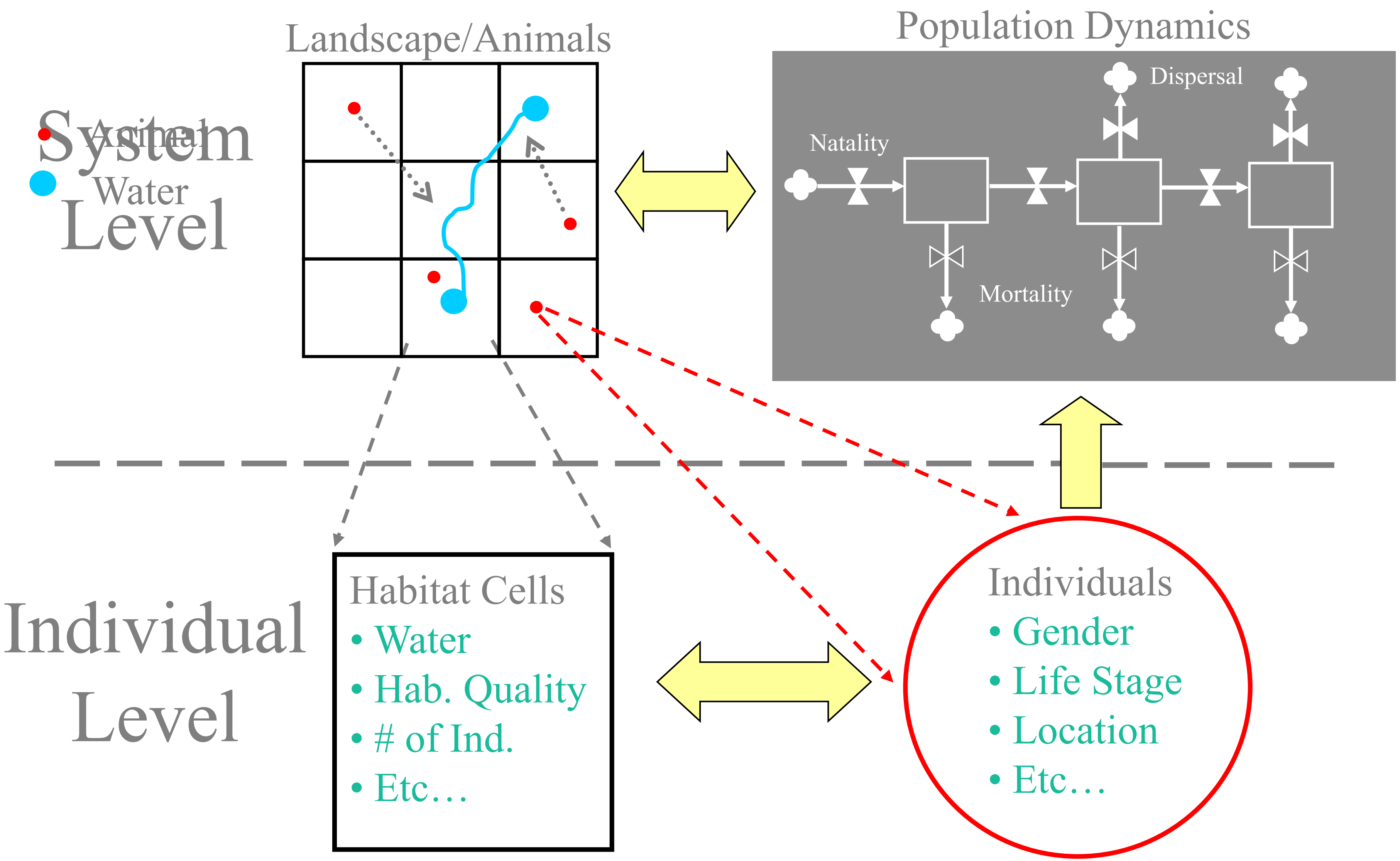
Analogy – relates to other systems/processes

Dialog – focused discussions on what's most important for generating patterns

Communication/Education – tools for explaining system dynamics

Prediction – forecast system dynamics (depends on accuracy of input data)

Conceptual Overview



Emergence

Arising of novel and coherent structures, patterns and properties though interaction of elements

Can’t be deduced from the properties of the elements

Pattern can feedback on individuals…*across levels*!

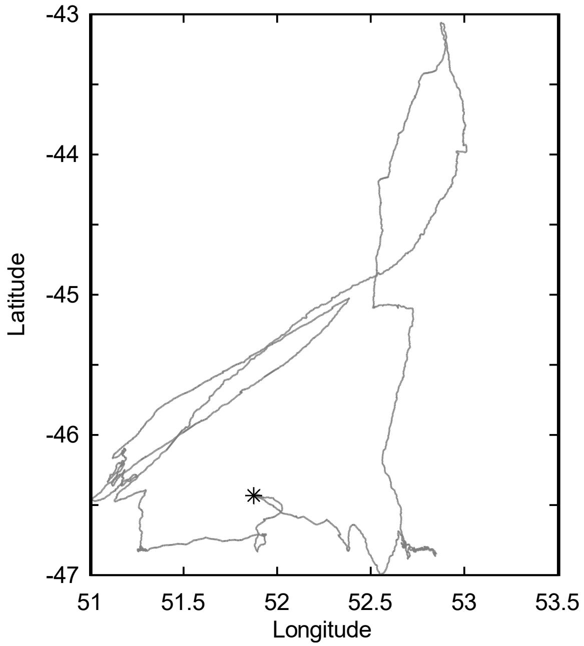
Key is to find simplest ruleset to generate emergent patterns

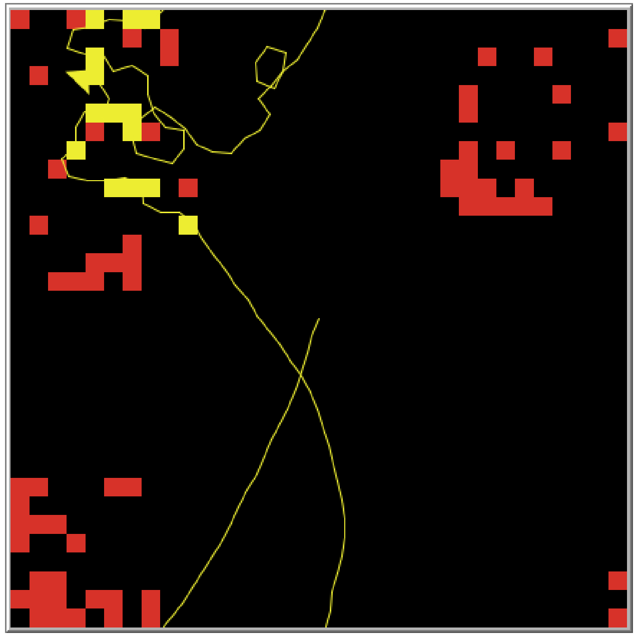
Example (show model)

- Imagine a system of arrows where each faces the same direction along a defined radius of a circle
  - Assign 1 rule: Each second move forward 0.35 units and then turn right 1°. What will happen?
  - Scenario 2: They move 0.5 units. What will happen?

24

Adaptive Behavior



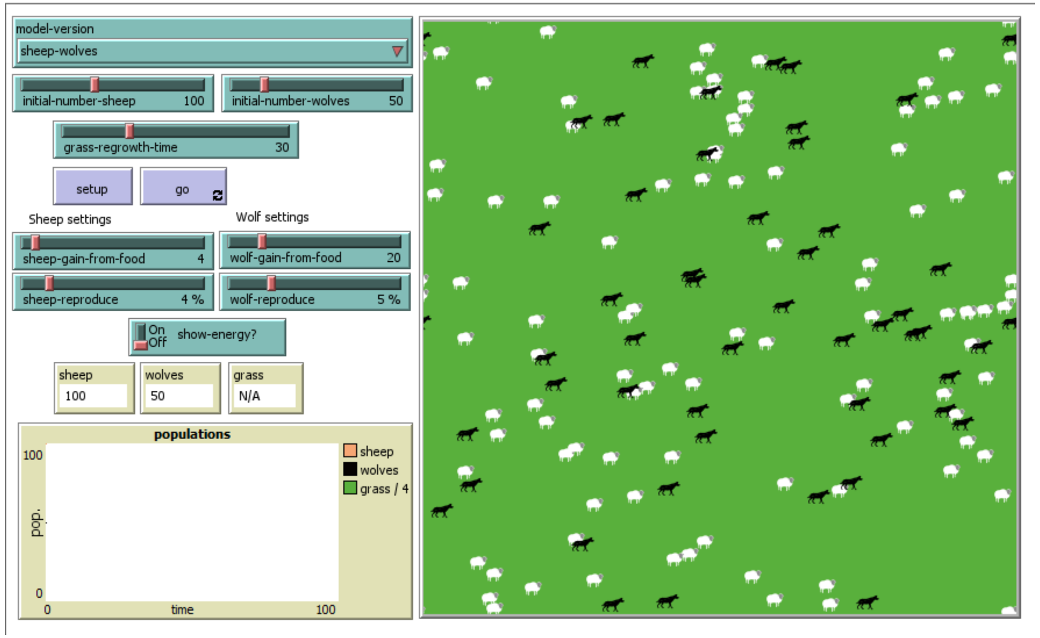


Predator-prey

Lotka-Volterra predator-prey equations:

$$dN/dt = rN - aNP$$
$$dP/dt = baNP - mP$$

Agent-based model



Intro to Netlogo: Let’s get it!

Netlogo: platform that recognizes that modeling is different from programming

- Agent-based modeling platform (designed by Uri Wilensky at MIT & Northwestern), based on Logo language, which was designed to help children think logically (Feurzeig, Solomon, & Papert 1967)
- Spatially explicit by nature
- Low threshold designed to narrow gap between modeler and programmer
- High ceiling—models can be scientific enough to be published in literature.
- Open-source and extendable (GIS, R, among others)

- NOT CASE-SENSITIVE (because letters are letters)

Basics of Netlogo

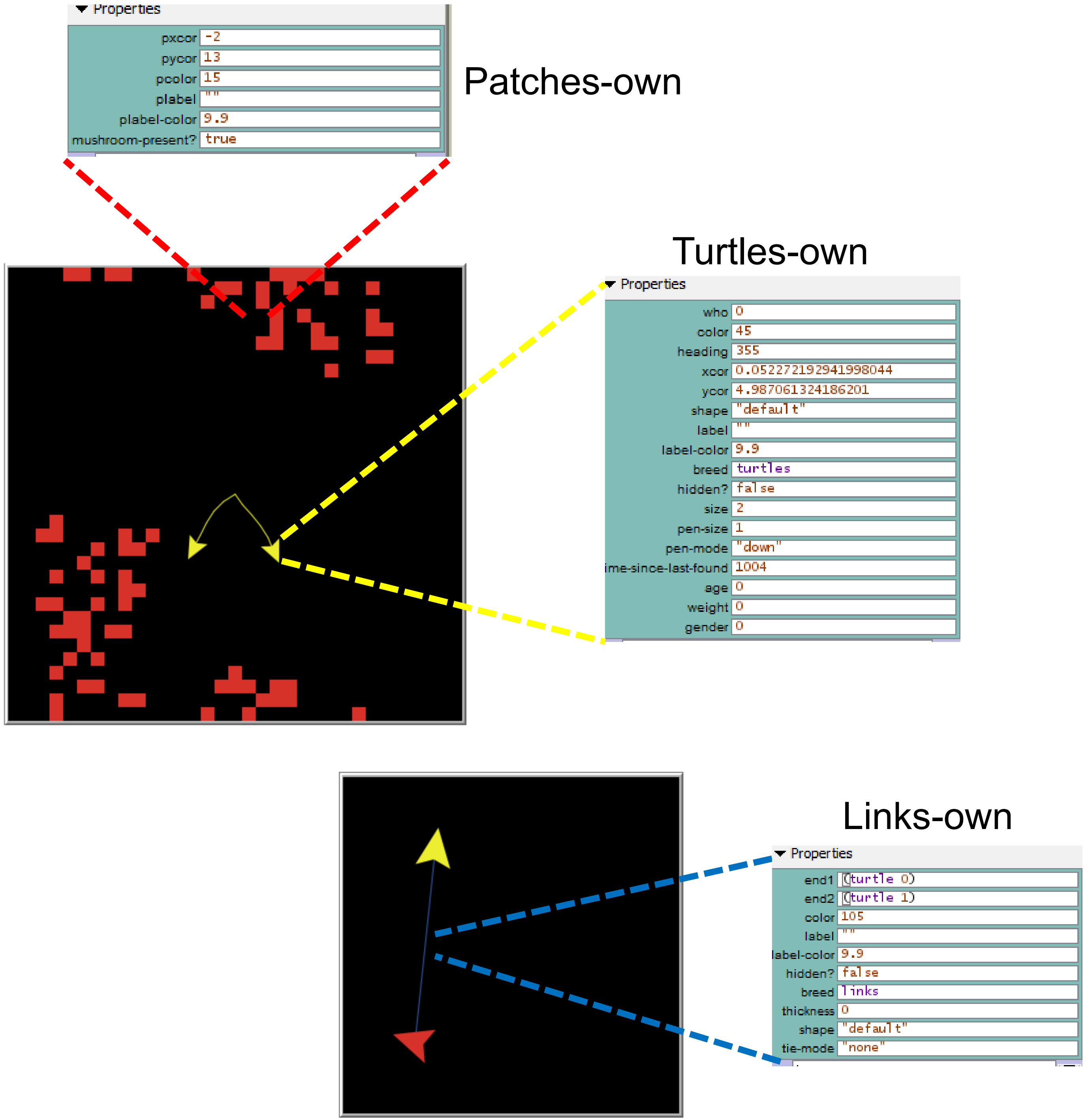
Grid-based

Composed of 3 agentsets

- Turtles (mobile)
- Patches (grid)
- Links (relates two or more turtles)

Each **agentset** can have variables (**turtles-own, patches-own**). Can be different for each patch/turtle.

Variables can be shared across turtles and patches (**globals**). Can be modified by any patch and/or turtle.



27

Primitives

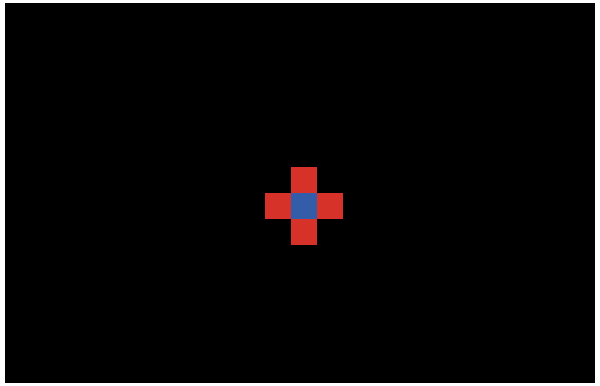
Heart of Netlogo

Built in procedures/functions

High-level language to facilitate modeling

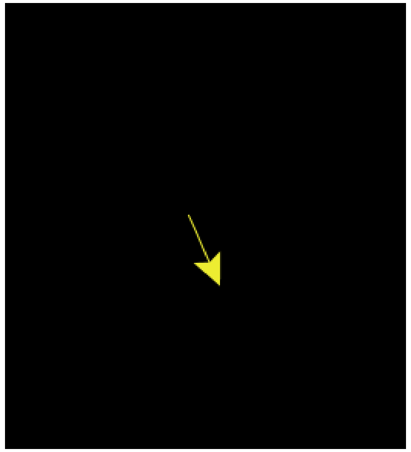
Two types

- *Commands:* tell an agent to do something (**ask**, **move**, …)
- *Reporters:* calculate a value and report it back to program (**sum**, **mean**, …)

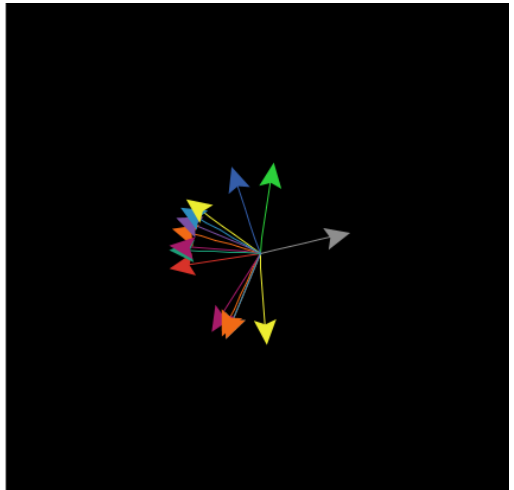


Ask patch 0 0
[
Ask neighbors4
[
set pcolor red
]
]

Ask is most important primitive



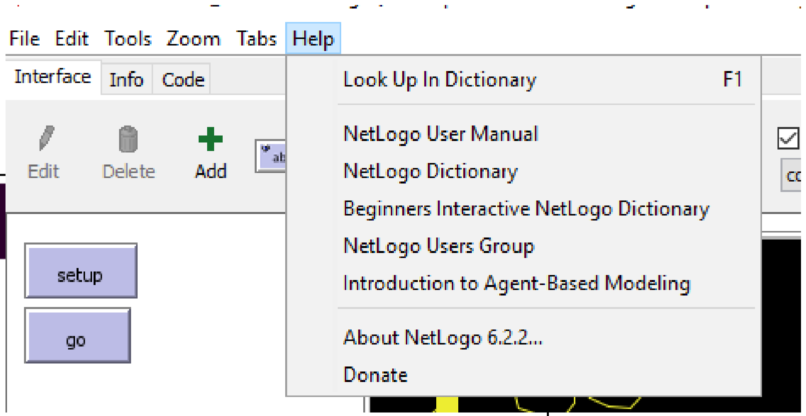
Ask turtles
[
fd 3
]



Let *total-turtles* **count** turtles
Set *total-turtles* **count** turtles

28

Netlogo dictionary



NetLogo Dictionary

Alphabetical: A B C D E F G H I J L M N O P R S T U V W X Y ->

Categories: Turtle - Patch - Links - Agentset - Color - Anonymous Procedures - Control/Logic - World Perspective - Input/Output - File - List - String - Math - Plotting - System - HubNet

Special: Variables - Keywords - Constants

Categories

This is an approximate grouping. Remember that a turtle-related primitive might still be used by patches or the observer, and vice versa. To see which agents (turtles, patches, links, observer) can actually run a primitive, consult its dictionary entry.

Turtle-related

back (bk) *<breeds>*-at *<breeds>*-here *<breeds>*-on can-move? clear-turtles (ct) create-*<breeds>* create-ordered-*<breeds>* create-ordered-turtles (cro) create-turtles (crt) die distance distancexy downhill downhill4 dx dy face facexy forward (fd) hatch hatch-*<breeds>* hide-turtle (ht) home inspect is-*<breed>*? is-turtle? jump layout-circle left (lt) move-to myself nobody no-turtles of other patch-ahead patch-at patch-at-heading-and-distance patch-here patch-left-and-ahead patch-right-and-ahead pen-down (pd) pen-erase (pe) pen-up (pu) random-xcor random-ycor right (rt) self set-default-shape __set-line-thickness setxy shapes show-turtle (st) sprout sprout-*<breeds>* stamp stamp-erase stop-inspecting subject subtract-headings tie towards towardsxy turtle turtle-set turtles turtles-at turtles-here turtles-on turtles-own untie uphill uphill4

Patch-related

clear-patches (cp) diffuse diffuse4 distance distancexy import-pcolors import-pcolors-rgb inspect is-patch? myself neighbors neighbors4 nobody no-patches of other patch patch-at patch-ahead patch-at-heading-and-distance patch-here patch-left-and-ahead patch-right-and-ahead patch-set patches patches-own random-pxcor random-pycor self sprout sprout-*<breeds>* stop-inspecting subject turtles-here

Link-related

both-ends clear-links create-<breed>-from create-<breeds>-from create-<breed>-to create-<breeds>-to create-<breed>-with create-<breeds>-with create-link-from create-links-from create-link-to create-links-to create-link-with create-links-with die directed-link-breed hide-link in-<breed>-neighbor? in-<breed>-neighbors in-<breed>-from in-link-neighbor? in-link-neighbors in-link-from inspect is-directed-link? is-link? is-link-set? is-*<link-breed>*? is-undirected-link? layout-radial layout-spring layout-tutte <breed>-neighbor? <breed>-neighbors <breed>-with link-heading link-length link-neighbor? link links links-own <link-breeds>-own link-neighbors link-with my-<breeds> my-in-<breeds> my-in-links my-links my-out-<breeds> my-out-links no-links other-end out-<breed>-neighbor? out-<breed>-neighbors out-<breed>-to out-link-neighbor? out-link-neighbors out-link-to show-link stop-inspecting tie undirected-link-breed untie

Agentset

all? any? ask ask-concurrent at-points *<breeds>*-at *<breeds>*-here *<breeds>*-on count in-cone in-radius is-agent? is-agentset? is-patch-set? is-turtle-set? link-set max-n-of max-one-of member? min-n-of min-one-of n-of neighbors neighbors4 no-links no-patches no-turtles of one-of other patch-set patches sort sort-by sort-on turtle-set turtles turtles-at turtles-here turtles-on up-to-n-of with with-max with-min

29

Procedures

Can write two types of functions: procedures and reporters with multiple primitives

to <name>
<commands>
End

```
to setup
  clear-all
  set-default-shape turtles "bug"
  create-turtles population
  [ set size 2         ;; easier to see
    set color red  ]   ;; red = not carrying food
  setup-patches
  reset-ticks
end

to go  ;; forever button
  ask turtles
  [ if who >= ticks [ stop ] ;; delay initial departure
    ifelse color = red
    [ look-for-food  ]       ;; not carrying food? look for it
    [ return-to-nest ]       ;; carrying food? take it back to nest
    wiggle
    fd 1 ]
  diffuse chemical (diffusion-rate / 100)
  ask patches
  [ set chemical chemical * (100 - evaporation-rate) / 100  ;; slowly evaporate chemical
    recolor-patch ]
  tick
end
```

to report <name>
<commands>
report X
end

```
to-report nest-scent-at-angle [angle]
  let p patch-right-and-ahead angle 1
  if p = nobody [ report 0 ]
  report [nest-scent] of p
end
```

patch-left-and-ahead
patch-right-and-ahead

patch-left-and-ahead ***angle distance***
patch-right-and-ahead ***angle distance***

Reports the single patch that is the given distance from this turtle, in the direction turned left or right the given angle (in degrees) from th

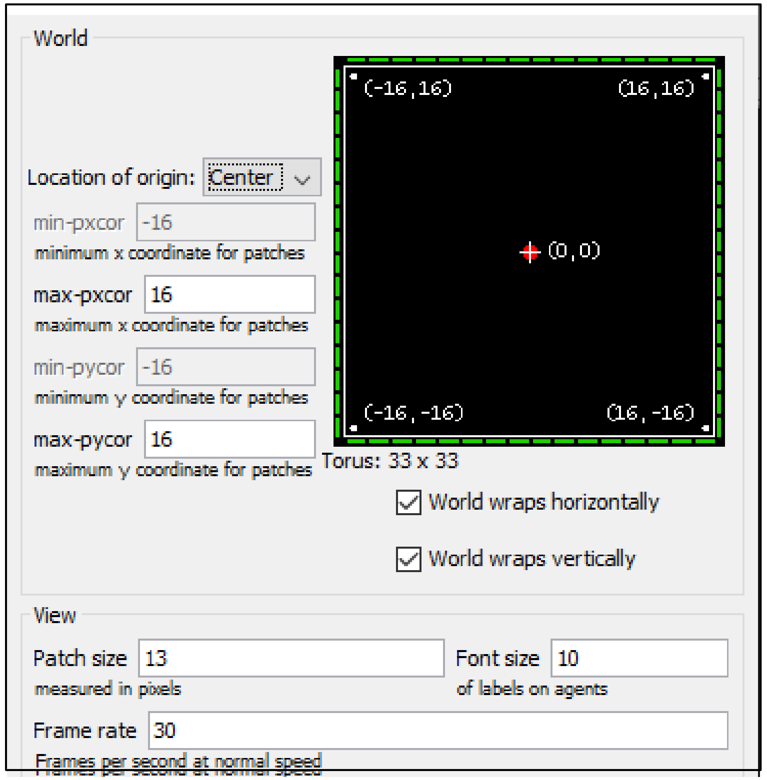
(If you want to find a patch in a given absolute heading, rather than one relative to the current turtle's heading, use patch-at-heading-and

```
ask patch-right-and-ahead 30 1 [ set pcolor green ]
;; this turtle "looks" 30 degrees right of its
;;   current heading at the patch 1 unit away, and turns
;;   that patch green; note that this might be the same
;;   patch the turtle is standing on
```

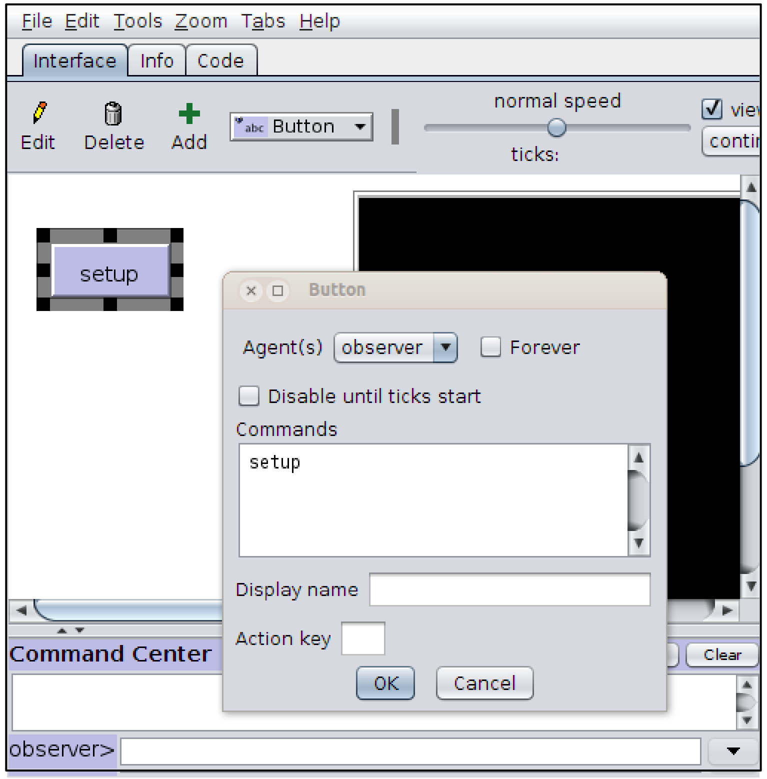
Let's get to modeling!

Adaptive Behavior – Mushroom Hunt, Ch. 2.

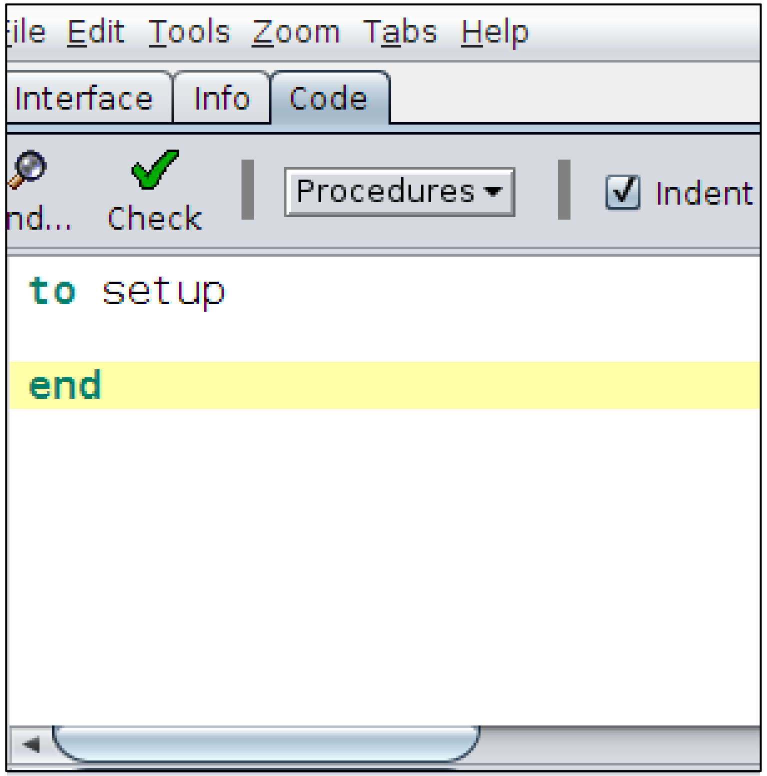
Model objective: Have a turtle exhibit adaptive behavior while searching for patches with mushrooms



1) Check settings



2) Create setup on GUI

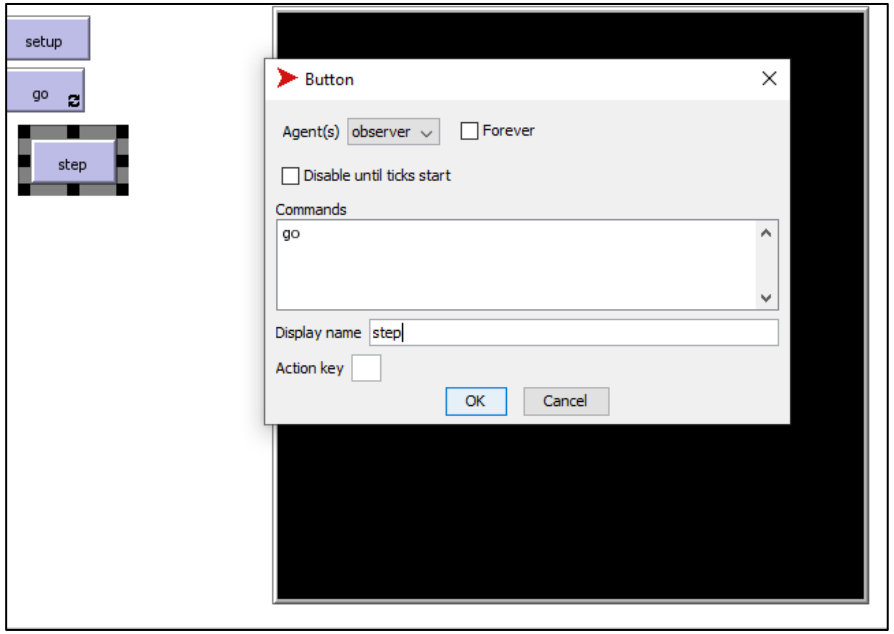


3) Create setup procedure & add line numbers

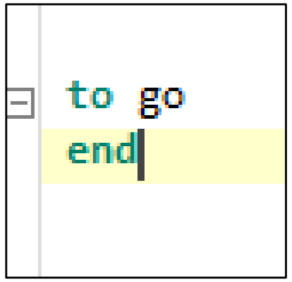
Let’s get to modeling!

Adaptive Behavior – Mushroom Hunt, Ch. 2.

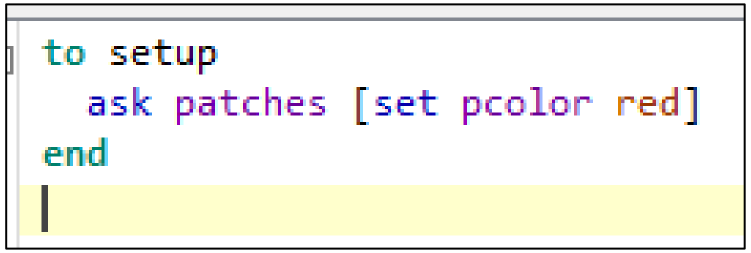
Model objective: Have a turtle exhibit adaptive behavior while searching for patches with mushrooms

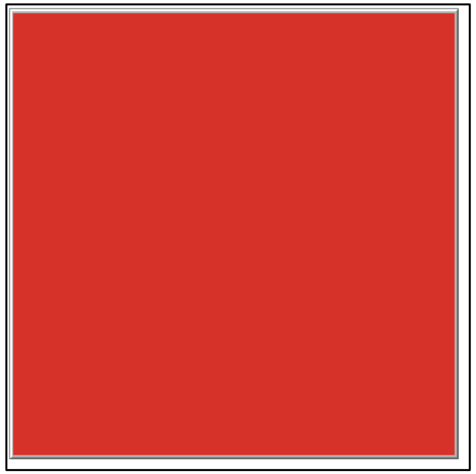


4) Create Go and Step



5) Create Go procedure





6) Turn patches red & test

Document Code!!!!

- All actions must have a comment
- Helps with documentation and with troubleshooting.
- Semicolon indicates comment

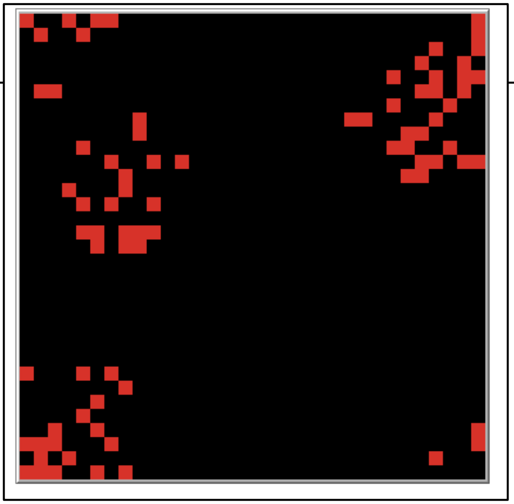
```
ask patches [set pcolor red]  ;turns all patches red
```

Let’s get to modeling!

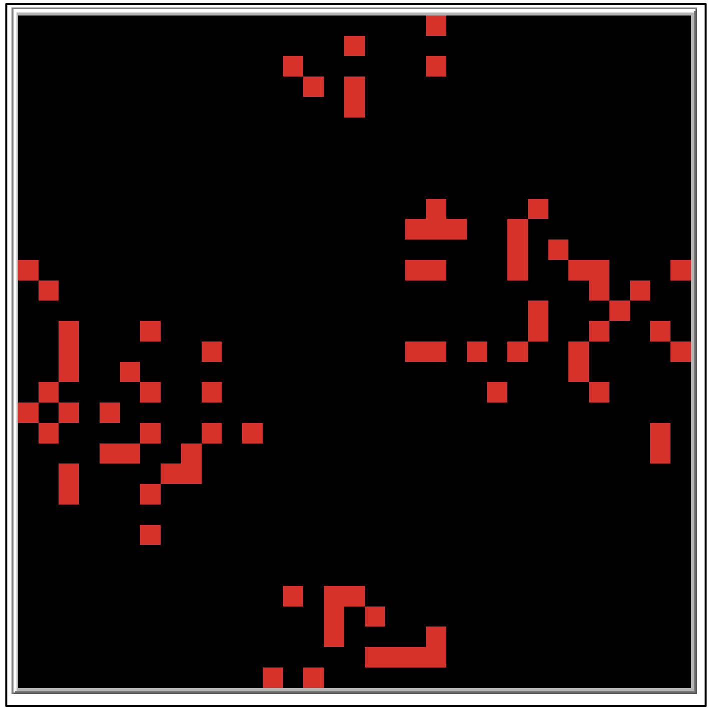
Adaptive Behavior – Mushroom Hunt, Ch. 2.

Model objective: Have a turtle exhibit adaptive behavior while searching for patches with mushrooms

```
to setup
  ask n-of 4 patches
  [
    ask n-of 20 patches in-radius 5
    [
      set pcolor red
    ]
  ]
end
```



7) Create clusters of red patches & test



8) Run it again

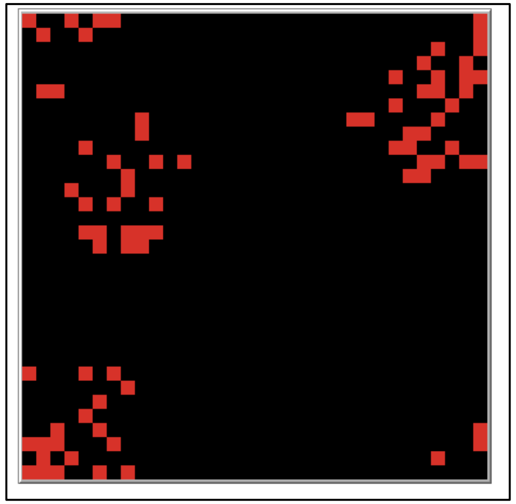
In class assignment:

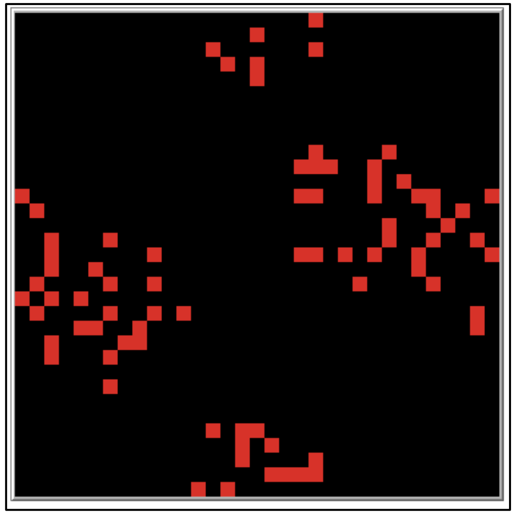
Create two turtles that search the landscape for red patches (i.e., mushrooms), and vary their behavior when they find mushroom patches.

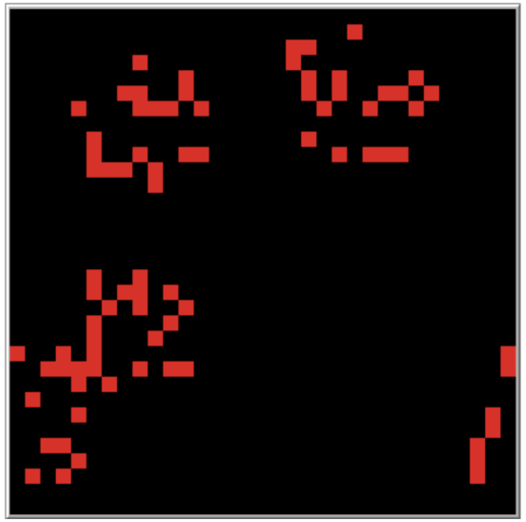
Follow the example in Ch. 2.
Use the dictionary.
Ask questions!

Random Processing

Netlogo processes “ask” statements in random order to prevent bias in order of processing







35

Primitives used (look up as you use them)

ask
clear-all (ca)
n-of
in-radius
set/let
size
color/pcolor
reset-ticks
tick
ifelse
right
forward (fd)
random