FOUR PHASES IN SYSTEMS MODELING -
PHASE II: QUANTITATIVE MODEL FORMULATION

OUTLINE

1. SELECT THE GENERAL QUANTITATIVE STRUCTURE FOR THE MODEL
   - Different types of mathematical formats
   - General compartment-model structure
2. DETERMINE THE FUNCTIONAL FORMS OF MODEL EQUATIONS
   - Some general aspects of model equations
   - State variable equations
   - Functional Forms for Other Model Equations
     - Information on which to base the choice of functional forms
     - Selecting types of equations to represent functional forms
     - Estimation of parameter of the model equations
     - Deterministic-model versus stochastic-model parameterization
3. DIMENSIONAL ANALYSIS
   - What is dimensional analysis?
   - Functions of dimensional analysis
   - A simple guide to dimensional analysis
4. CODE THE MODEL EQUATIONS FOR THE COMPUTER
   - Basic steps
   - Flowcharts
   - Computer languages
5. CHOOSE THE TIME STEP FOR THE SIMULATIONS
   - What Is DT?
   - Factors Affecting the selection of DT
   - A Practical Guide to the Choice of DT
   - Examples
6. EXECUTE THE BASELINE SIMULATION
7. PRESENT THE MODEL EQUATIONS
8. EXAMPLES

1. SELECT THE GENERAL QUANTITATIVE STRUCTURE FOR THE MODEL

(1) Different Types of Mathematical Formats
   - Difference equations
   - Differential equations
   - Matrix algebra
   - Partial differential equations
Recall the features of the above different types of mathematics. Why do we need different types of mathematical frameworks for modeling the world? FDEs provide the most flexibility and convenience for representing dynamic systems in terms of a general compartment-model structure, and facilitate numerical simulation.

(2) General Compartment-Model Structure

The compartment-model conceptualization provides a convenient and general framework for building biological and ecological models.

- Compartment models refer to models that describe the flow of physical material between physical or biological storage pools (i.e., compartments). State variables, rate variables, auxiliary variables, and driving variables usually are explicitly identified in compartment models.
- Compartment models represent the rates of change of energy or materials in compartments as the sum of all flow rates into the compartment minus the sum of all the flow rates out of the compartment.
- Each compartment is a state variable, with which one or more rates of flow between compartments are associated.
- The compartment from which the flow is coming is called the donor compartment, and the one to which the flow goes is called the recipient compartment.
- Compartment models can be represented in the form of difference equations, differential equations, or matrices.
- Compartment modeling provides a very general conceptualization that applies to many biological and ecological problems.
- Indeed, this methodology has been dominant in modeling flows of energy and material in ecosystems.
- Systems models usually capitalize on the compartment conceptualization. STELLA, as a simulation package, is well suited for building compartment models.

Example: Compartment models of ecosystem processes

The following system of differential equations may be used as a framework to describe the dynamics of a variety of systems with \( n \) compartments:

\[
\frac{dx_1}{dt} = v_{11}x_1 + v_{12}x_2 + \cdots + v_{1n}x_n
\]

\[
\frac{dx_2}{dt} = v_{21}x_1 + v_{22}x_2 + \cdots + v_{2n}x_n
\]

\[\vdots\]

\[
\frac{dx_n}{dt} = v_{n1}x_1 + v_{n2}x_2 + \cdots + v_{nn}x_n
\]

where \( \frac{dx_i}{dt} \) are the rates of change in the values of state variables \( x_i \) with respect to time, and \( v_{ij} \) are coefficients or some forms of functions of the state variables.
For example, one of the first ecosystem models, developed by H. T. Odum (H. T. Odum. 1957. Trophic structure and productivity of Silver Springs, Florida. Ecological Monographs 27:55-112), simulates how the loss of primary production and alteration in the pools or fluxes of energy in higher trophic levels affect the energy flow in the Silver Spring ecosystem, Florida, and takes the following system of nonlinear differential equations:

\[
\begin{align*}
\frac{dX_1}{dt} &= F_{10} - T_{21}X_1X_2 - M_{51}X_1 - L_{01}X_1 - P_{01}X_1 \\
\frac{dX_2}{dt} &= F_{20} + T_{21}X_1X_2 - T_{32}X_2X_3 - M_{52}X_2 - P_{02}X_2 \\
\frac{dX_3}{dt} &= T_{32}X_2X_3 - T_{43}X_3X_4 - M_{53}X_3 - P_{03}X_3 \\
\frac{dX_4}{dt} &= T_{43}X_3X_4 - M_{54}X_4 - P_{04}X_4 \\
\frac{dX_5}{dt} &= M_{51}X_1 + M_{52}X_2 + M_{53}X_3 + M_{54}X_4 - P_{05}X_5
\end{align*}
\]

where
- \(X_1\) – producers (kilocalories/m\(^2\))
- \(X_2\) - herbivores
- \(X_3\) - carnivores
- \(X_4\) - top carnivores
- \(X_5\) - decomposers
- \(F_{ij}\) - external forcing inputs
- \(T_{ij}\) - trophic level feeding rates
- \(M_{ij}\) - natural mortality rates
- \(P_{ij}\) - respiration rates
- \(L_{10}\) - the rate for downstream losses

The corresponding system of difference equations is:

\[
\begin{align*}
X_{1,t+\Delta t} &= X_{1,t} + F_{10} - T_{21}X_{1,t}X_{2,t} - M_{51}X_{1,t} - L_{01}X_{1,t} - P_{01}X_{1,t} \\
X_{2,t+\Delta t} &= X_{2,t} + F_{20} + T_{21}X_{1,t}X_{2,t} - T_{32}X_{2,t}X_{3,t} - M_{52}X_{2,t} - P_{02}X_{2,t} \\
X_{3,t+\Delta t} &= X_{3,t} + T_{32}X_{2,t}X_{3,t} - T_{43}X_{3,t}X_{4,t} - M_{53}X_{3,t} - P_{03}X_{3,t} \\
X_{4,t+\Delta t} &= X_{4,t} + T_{43}X_{3,t}X_{4,t} - M_{54}X_{4,t} - P_{04}X_{4,t} \\
X_{5,t+\Delta t} &= X_{5,t} + M_{51}X_{1,t} + M_{52}X_{2,t} + M_{53}X_{3,t} + M_{54}X_{4,t} - M_{05}X_{5,t}
\end{align*}
\]
2. IDENTIFY FUNCTIONAL FORMS OF MODEL EQUATIONS

(1) Some General Aspects of Model Equations
- All the equations together should completely describe the relationships among all variables, and thus govern the dynamics of the model.
- Although the mathematical framework may be of different forms, all dynamic equations must be written in the form of difference equations in order to conduct computer simulation.
- Notes on using STELLA:
  - When using a simulation package, such as STELLA, one does not need to choose a mathematical framework because it is already embedded in the software. That framework is the form of a system of first-order differential (or difference) equations.
  - All equations for state variables are automatically written by the software. Thus, only equations for other variables need to be created by the user.
  - Also, because STELLA handles time subscripts of variables internally, the user only needs to focus on the relationship among variables.
  - When a differential equation model is converted into a STELLA model, all equations are automatically translated into the form of difference equations by the software.

(2) State Variable Equations
The general form of state variable equations is the same for all compartment models:

\[
\text{(State variable)}_{t+\Delta t} = \text{(State variable)}_t + (\text{Net change in state variable})_{t, t+\Delta t} \\
= \text{(State variable)}_t + (\text{Transfers into state variable})_{t, t+\Delta t} \\
- (\text{Transfers out of state variable})_{t, t+\Delta t}
\]

This is essentially equivalent to the equation:

\[
y_{t+\Delta t} = y_t + y' \Delta t \\
= y_t + \Delta y_{t, t+\Delta t}
\]

(3) Functional Forms for Other Model Equations
The most challenging task in determining functional forms is for rate variables. Rate equations are critically important for generating system behavior. They may be linear or nonlinear, deterministic or stochastic, and theoretical or empirical.

(i) Information on which to base the choice of functional forms
Four general types of information may be used for choosing functional forms:
1) Quantitative data from observation or experimentation with the real system (firsthand or secondhand)
2) Theoretical relationships or generally acceptable empirical relationships that are appropriate for the system modeled.
3) Qualitative information based on expert opinion or on the literature.
4) Information gained from experimenting with the model itself. Through model simulations with different functional forms of a certain relationship (e.g., linear, exponential, sigmoidal, etc.), for example, one may choose the one that generates most plausible dynamics. This trial-and-error method is certainly not recommended if the previous sources are sufficient to choose all functional forms of model equations.

(ii) Selecting types of equations to represent functional forms

Because many processes or variable relationships in different systems actually are similar mathematically (e.g., snowball effects, vicious cycles, or exponential growth), many well-studied mathematical equations can be used to facilitate model formulation. The following are some particularly useful equation types (see Hall and Day 1977; Haefner 1996, Grant et al. 1997):

(1) Linear increase and decrease
\[ y = ax + b \]
where \( y \) is the dependent variable, \( x \) is the independent variable, and parameters \( a \) and \( b \) are the slope and intercept of the line, respectively. The equation represents a linear increase when \( a > 0 \), and a linear decrease when \( a < 0 \).

(2) Exponential growth and decay
\[ y = ae^{bx} \]
where parameter \( a \) defines the starting value of \( y \), and parameter \( b \) modifies the steepness of the curve. The equation generates an exponential growth curve when \( b > 0 \), and an exponential decay while \( b < 0 \).

(3) Exponential saturation
\[ y = a(1 - e^{-bx}), \quad a, b > 0 \]
where parameter \( a \) sets the maximum value of \( y \) (i.e., the saturation level), and \( b \) determines how fast to get to the saturation level.

(4) Hyperbolic function
\[ y = \frac{ax}{b + x}, \quad a, b > 0 \]
where parameter \( a \) sets the upper limit of the values of \( y \) when \( x \) increases, and \( b \) is conversely related to the rate at which \( y \) approaches the value \( a \).

(5) Hyperbolic power function
\[ y = a + \frac{x^b}{c^x + x^c}, \quad a, b, c > 0 \]
where \( a, b, \) and \( c \) are parameters. Parameter \( a \) sets the upper limit of the values of \( y \), and \( b \) and \( c \) determine the shape of the curve.

(6) Logistic growth
\[ y = \frac{a}{1 + \left(\frac{a}{b} - 1\right)e^{-cx}}, \quad a, b, c > 0 \]
where a, b, and c are parameters. Parameter a is the maximum value of y (e.g.,
the carrying capacity in the logistic population growth equation), b is the starting
value of y corresponding to x equal to 0 (e.g., the starting population size in the
logistic population growth equation), and c determines the rate of change in y
(e.g., the intrinsic increase rate in the logistic population growth equation).

(7) Michaelis-Menten (Substrate Saturation) equation

\[ y = \frac{ax}{x + b} \], \quad a, b > 0

where parameter a is the maximum value of y (the saturation level), and
parameter b is called the half-saturation constant (i.e., when x = b, y = 0.5a).

(iii) Estimation of parameter of the model equations

Parameterizing model equations and choosing functional forms commonly use the same
information. The choice of functional forms usually has more profound implications for the
model development.

- Many statistical techniques available for parameter estimation.
- Regression analysis and curve fitting techniques are frequently used in ecological
  modeling.
- Sometimes, parameter estimation may have to be based on qualitative data, expert
  opinion, or information gained from experimentation with the model.

(iv) Deterministic-model versus stochastic-model parameterization

- Both use the same types of information and follow the same general procedures.
- Deterministic models: Point estimates of parameters
- Stochastic models: Because of random variables, specification of a statistical
distribution or an empirical frequency distribution for the values of parameters and
other input data are necessary. For example,
  - Frequency distribution of historical data
  - Statistical distribution fitted to historical data
  - Statistical distributions generated from regression

3. DIMENSIONAL ANALYSIS

(1) What Is Dimensional Analysis?

- For any mathematical equations with dimensional quantities to be correct, the units of
  measurement (dimension) on the both sides of the equation must agree.
- Dimensional analysis is the procedure of checking physical units of variables to make
  sure that the units on the two sides of an equation are the same.

(2) Functions of Dimensional Analysis

- Checking the validity of a model equation. If the units on the two sides of an
equation are not the same, the equation either has an error in its logic expression of
the relationship among variables (illegitimate formulation), or needs a conversion factor.
- Determining correct conversion factors
- Assisting with the formulation of relationships among variables

Note: Dimensional analysis can detect logic errors in an equation, but it can not guarantee the quantitative correctness of an equation. That is, dimensional correctness does not mean numerical correctness:

\[[\text{apple}] = [\text{apple}], \text{ but } 3 \text{ apples} \neq 2 \text{ apples!}\]

On the other hand, numerical correctness does not guarantee dimensional correctness:

\[2 + 3 = 5, \text{ but } 2 \text{ apples} + 3 \text{ oranges} \neq 5 \text{ apples (or oranges)!}\]

(3) A Guide to Dimensional Analysis
A simple guide to dimensional analysis includes the following steps:
- Replace each variable and parameter with its unit of measurement in bracket, [unit]
- Cancel out units until no further reduction is possible
- Compare the units on the both sides of the equation

Examples

Checking the validity of model equations

(1) The MacArthur-Wision model of species diversity

\[
\frac{dS(t)}{dt} = I - E
\]

\[
\left[\frac{\# sp}{\text{time}}\right] = \left[\frac{\# sp}{\text{time}}\right] - \left[\frac{\# sp}{\text{time}}\right] = \left[\frac{\# sp}{\text{time}}\right]
\]

(2) A population model

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

\[
\left[\frac{\# \text{indiv}}{\text{time}}\right] = \left[\frac{\# \text{indiv}}{\text{time} \cdot \text{indiv}}\right] - \left[\frac{\# \text{indiv}}{\text{time}}\right] = \left[\frac{\# \text{indiv}}{\text{time}}\right]
\]

(3) A simulation model of animal weight dynamics

\[
\text{WEIGHT}(t) = \text{WEIGHT}(t - dt) + (\text{CONSUMPTION} - \text{RESPIRATION}) \ast dt
\]

\[
[W] = [W] + \left[\frac{W}{[T]} - \frac{W}{[T]}\right] [T] = [W]
\]
CONSUMPTION = CON_RATE*WEIGHT
\[
\frac{[W]}{[T]} = \frac{[W]}{[T \cdot W]} \cdot \frac{[W]}{[T]}
\]

RESPIRATION = RESP_RATE*WEIGHT
\[
\frac{[W]}{[T]} = \frac{[W]}{[T \cdot W]} \cdot \frac{[W]}{[T]}
\]

Determining correct conversion factors
(1) A simulation model of animal weight dynamics
\[
\text{RESP RATE} = 0.0025 \times \text{TEMPERATURE}
\]
\[
\frac{[W]}{[W \cdot T]} = \left[ ? \right] \, ^\circ C \quad \Rightarrow \quad [?] = \frac{1}{[^\circ C \cdot T]}
\]
\[
\text{TEMPERATURE} = 20 - 0.167 \times \text{TIME}
\]
\[
[\circ C] = [\circ C] - [?] \, [T] \quad \Rightarrow \quad [?] = \frac{[\circ C]}{[T]}
\]

(2) A predator-prey model
\[
\frac{dN}{dt} = aN - bPN
\]
\[
\frac{dP}{dt} = cPN - dP
\]
\[
\frac{[\# \text{ indiv}]}{[\text{time}]} = \left[ ?_1 \right] \, [\# \text{ indiv}] - \left[ ?_2 \right] \, [\# \text{ indiv}] \, [\# \text{ indiv}]
\]
\[
\Rightarrow \quad \left[ ?_1 \right] = \frac{1}{[\text{time}]} \quad \text{and} \quad \left[ ?_2 \right] = \frac{1}{[\# \text{ indiv} \cdot \text{time}]}
\]
\[
\frac{[\# \text{ indiv}]}{[\text{time}]} = \left[ ?_3 \right] \, [\# \text{ indiv}] \, [\# \text{ indiv}] - \left[ ?_4 \right] \, [\# \text{ indiv}]
\]
\[
\Rightarrow \quad \left[ ?_3 \right] = \frac{1}{[\# \text{ indiv} \cdot \text{time}]} \quad \text{and} \quad \left[ ?_4 \right] = \frac{1}{[\text{time}]}
\]

Assisting with the formulation of relationships among variables
A verbal model goes like this (Bossel 1994):

“The larger the mass, the greater the kinetic energy.”
“The larger the velocity, the larger the kinetic energy.”

This can be expressed as:
\[ E_{\text{kinetic}} \propto m \cdot v \quad \text{or} \quad E_{\text{kinetic}} = ? \cdot m \cdot v \]

where \( E_{\text{kinetic}} \) is the kinetic energy, \( m \) is the mass, \( v \) is the velocity, \( ? \) is an undetermined factor. Carry out a dimensional analysis as follows:

\[
[kg \ m^2 \ s^{-2}] = [?] \ [kg \ m \ s^{-1}]
\]

Therefore,

\[
[?] = [m \ s^{-1}]
\]

The dimension of the undetermined factor suggests to use \( v^2 \) to satisfy the equation. Thus, the equation becomes

\[ E_{\text{kinetic}} = \alpha \cdot m \cdot v^2 \]

where \( \alpha \) has to be a dimensionless (or unitless) parameter, which is necessary to make the two sides of the equation equal in quantity.

4. CODE THE MODEL EQUATIONS FOR THE COMPUTER

In order to conduct simulations, model equations must be translated into some sort of computer languages (or computer codes). This process is typically called programming. The basic steps involved in programming a model for digital computers are:

1. Constructing a flow diagram (or flowchart);
2. Selecting a specific computer language to implement the steps and tasks in the flowchart;
3. Compiling, linking, and debugging the computer code, including correcting syntax, logic, and run-time errors; and
4. Verifying the computer code (often called model verification).

Computer programming can be tedious, time-consuming, and rather boring. But these procedures may be simplified substantially when higher-level simulation languages are used.

1. Flow Diagram
   - A flow diagram (or flowchart) depicts the steps and routes to solve model equations and to simulate model behavior.
   - A flow diagram is different from the conceptual model. It usually outlines the overall simulation scheme and reflects the way how a computer solves a equation or a systems of equation.
   - The following graphs show standard symbols used in flowchart construction and a general flow diagram for systems models.
Some standard symbols used in flowcharts

A general flowchart for systems simulation models

- Input or Output
- A Process or Simulation Step
- Start or Stop
- Check and Decision-Making
- Route and Direction

Input parameters and initial conditions

Calculate driving variables at time t

Calculate auxiliary variables at time t

Calculate rate variables at time t

Calculate state variables at time t

End of simulation? Yes

No

Output

START

STOP
5. CHOOSE THE BASIC TIME UNIT FOR SIMULATION

(1) **What Is DT?**
  - The basic time unit for simulation is also called time step, time interval, or simply DT (or Δt).
  - DT is the interval of time between two consecutive calculations the computer makes.
  - DT determines the units-of-measure for all of the flows in our model.
  - DT defines the smallest time interval over which a change can occur.
  - A smaller DT tends to generate smoother dynamics, whereas a larger DT renders “chunkier” changes in variables of your model.
  - In general, the smaller the DT, the more accurate the numerical solutions are, at the expense of simulation speed. On the other hand, larger ΔTs speed up simulation, but reduce numerical precision due to aggregation error.
  - Importantly, large ΔTs may introduce “artifactual time delay” that are not present in the relationships among variables, resulting in distorted dynamics of the system. Even worse is that artifactual or “pathological” dynamics may occur when ΔT is too large.
  - For continuous systems, a smaller DT make it possible to capture gradual changes which occur over a time interval less than the time unit you are interested in for reporting model results.
  - Note that DT may not be the same as the time interval over which model results are reported. In fact, the simulation DT should always be smaller than the resolution of the data used to parameterize the model for continuous systems.
(2) Factors Affecting the selection of DT:
   1) The nature of the temporal change of the system of interest (e.g., discrete vs. continuous, fast vs. slow).
   2) Feasibility of parameterization of the model
   3) Trade-off between the simulation speed and model accuracy (e.g., precision required for the purpose of model development).

(3) A Practical Guide to the Choice of DT
   1) Usually, an appropriate DT ranges from 0.0625 to 1.0 for modeling continuous systems. But for stiff systems, it may need to be smaller.
   2) The default value for DT of 0.25 in STELLA should work well for modeling many if not most phenomena in real-world systems.
   3) Because the computer operates on the binary arithmetic, choosing a DT from the sequence of \((1/2)^n\) (i.e., 1.0, 0.5, 0.25, 0.125, 0.0625, 0.03125, ...) minimizes round-off errors.
   4) A good first approximation for a proper DT is a time step size that is one-half of the smallest time constant or shortest time delay in the model.
   5) As a check on the appropriateness of DT, a “1/2 test” should be performed. That is, repeatedly run the model with DT set to 1/2 of its previous value each time, until simulation results do not change with a decreasing DT. Then choose the largest DT with which the model still behaves correctly and accurately.

(4) Example: The effect of DT on model behavior -- A logistic population model
   o When DT is small enough (< the smallest time constant, i.e. 1/r), the model generates a smooth logistic curve.
   o When DT is 3, inaccuracy in the numerical solution is apparent -- discrete changes occur for a supposedly continuous system.
   o When DT increases to 4 or 5, artifactual oscillations occur.
   o When DT increases to 7 or bigger, the system crashes.

\[ N(t) = N(t - dt) + (dN/dt) * dt \]

INFLOWS:
\[ dN/dt = r * N * (1 - N/K) \quad \{\text{indiv. / year}\} \]

K = 1000
r = 0.5 \{\text{per capita growth rate, 1/ year}\}

\[ N(t) = N(t - dt) + (dN/dt) * dt \]

DT = 0.5
6. EXECUTE THE BASELINE SIMULATION

- The baseline (or normal) simulation is the model simulation based on a particular set of data to be used as a benchmark or reference against which subsequent simulations are compared.
- A deterministic model needs only one baseline simulation.
- Stochastic models need replicate simulations, each representing a baseline condition. Thus, the minimum number of simulations (replicates, samples) needs to be determined.

How to Determine the Number of Simulations in Stochastic Models?

The following formula can be used to estimate the minimum number of runs required for a stochastic model:

\[
 n \geq 2 \left( \frac{\sigma}{\delta} \right)^2 \left[ t_{a,\gamma} + t_{2(1-P),\gamma} \right]^2
\]

\( n \) = number of samples
\( \sigma \) = true standard deviation, which may be estimated as the square root of the estimated variance within samples
\( \delta \) = smallest true difference that we desire to detect
\( \gamma \) = degrees of freedom of the sample standard deviation with \( b \) groups of samples and \( n \) samples per group, or \( \gamma = b(n-1) \)
\( \alpha \) = significance level
\( P \) = desired probability that a difference will be found to be significant if it is as small as \( \delta \)
\( t_{a,\gamma} \) and \( t_{2(1-P),\gamma} \) = values from a two-tailed t-table with \( \gamma \) degrees of freedom and corresponding to probabilities of \( \alpha \) and \( 2(1 - P) \), respectively.
7. PRESENT THE MODEL EQUATIONS

- Present succinct mathematical expressions whenever possible to represent relationships among variables.
- Present conceptual models and flow diagrams with equations make the model easier to be understood.
- You may order model equations according to their types: driving variable equations, auxiliary variable equations, rate equations, and state variable equations. Initial values of state variables may be specified first with driving variables, or associated with corresponding state variables. This order resembles the sequence of simulation execution of the model.
- In STELLA, you may present your equations in three ways: alphabetically, by sectors (grouping of associated variables), and by order of simulation execution (see the example below).
8. MODEL EXAMPLES

1. A systems model of population dynamics (Wu and Barlas 1989)

In a single-species population system, there are only four basic processes in which population size can change: birth, death, immigration, and emigration. Environmental and genetic properties affect the population dynamics through influencing these processes. However, in the model developed here, a balance between immigration and emigration and a stable average genetic property of populations are assumed.

There are four major feedback loops in the model (see the figure below). Populations are believed to be regulated through a non-linear density-dependent mechanism. The crowding effect, rather than population density or population size itself, directly affects the regulating processes because crowding can occur at different densities. Crowding is determined by both current population density and environmental limitation (e.g., food supply, living space, etc.). A time delay usually exists between crowding in a population and responses in birth and death rates. Environment can significantly alter the carrying capacity and modify the density-dependent relationship.

The response curves of instantaneous per capita birth, death, and growth rates to population density are assumed different from those in exponential or logistic models. Both effects of undercrowding, i.e., lacking cooperation (e.g., less chance of mating and reproducing unsuccessfully) and crowding (e.g., intraspecific competition) are taken into account.
Accordingly, a minimum carrying capacity (Kmin) and maximum carrying capacity (Kmax) are identified. The minimum carrying capacity here may be understood as a deterministic version of minimum viable population (MVP) size. A lower critical density (LCD) is defined as a density below which birth rate declines, and an upper critical density (UCD) is defined as a density beyond which birth rate declines and death rate increases. Below UCD, death rate keeps constant (minimum death rate), and between LCD and UCD neither birth rate nor death rate is affected by the crowding population density. Therefore, the assumption also reflects density-independent effect on the regulation of population growth even though it is principally density-dependent.

**Effect of Population Density on PCBR and PCDR**

![Effect of Population Density on PCBR and PCDR](image)

**Structural diagram in STELLA**

![Structural diagram in STELLA](image)
The mathematical form of the model may look like:

\[
\frac{dN}{dt} = (B(N) - D(N)) N(t - \delta) \left(1 - \frac{N(t - \delta)}{K(t)}\right)
\]

where \(N\) is population size, \(B\) and \(D\) are per capita birth and death rates which are functions of the population itself, \(\delta\) is a delay time, and \(K\) is the carrying capacity, which may vary in time.

<table>
<thead>
<tr>
<th>List of STELLA Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Ordered by sectors)</td>
</tr>
<tr>
<td>Population(t) = Population(t - dt) + (Total_Birth_Rate - Total_Death_Rate) * dt</td>
</tr>
<tr>
<td>INIT Population = 50</td>
</tr>
<tr>
<td>INFLOWS:</td>
</tr>
<tr>
<td>Total_Birth_Rate = Actual_PCBR*Population</td>
</tr>
<tr>
<td>OUTFLOWS:</td>
</tr>
<tr>
<td>Total_Death_Rate = Actual_PCDR*Population</td>
</tr>
<tr>
<td>Actual_PCBR = DELAY(PCBR,0.5)</td>
</tr>
<tr>
<td>Actual_PCDR = DELAY(PCDR,0.5)</td>
</tr>
<tr>
<td>Area = 10</td>
</tr>
<tr>
<td>Crowd = Pop_Density*EnvironLF</td>
</tr>
<tr>
<td>EnvironLF = 5</td>
</tr>
<tr>
<td>Pop_Density = Population/Area</td>
</tr>
<tr>
<td>PCBR = GRAPH(Crowding)</td>
</tr>
<tr>
<td>(0.00, 0.00), (4.00, 0.04), (8.00, 0.1), (12.0, 0.2), (16.0, 0.4), (20.0, 0.54), (24.0, 0.68), (28.0, 0.68), (32.0, 0.68), (36.0, 0.65), (40.0, 0.64), (44.0, 0.62), (48.0, 0.58), (52.0, 0.54), (56.0, 0.5), (60.0, 0.4), (64.0, 0.32), (68.0, 0.14), (72.0, 0.00)</td>
</tr>
<tr>
<td>PCDR = GRAPH(Crowding)</td>
</tr>
<tr>
<td>(0.00, 0.12), (4.00, 0.12), (8.00, 0.12), (12.0, 0.12), (16.0, 0.12), (20.0, 0.12), (24.0, 0.12), (28.0, 0.12), (32.0, 0.12), (36.0, 0.13), (40.0, 0.14), (44.0, 0.16), (48.0, 0.17), (52.0, 0.21), (56.0, 0.28), (60.0, 0.4), (64.0, 0.52), (68.0, 0.68), (72.0, 0.92)</td>
</tr>
</tbody>
</table>
List of STELLA Equations
(Ordered by simulation execution)

{ INITIALIZATION EQUATIONS }
INIT Population = 50
Area = 10
Pop_Density = Population/Area
EnvironLF = 5
Crowding = Pop_Density*EnvironLF
PCBR = GRAPH(Crowding)
(0.00, 0.00), (4.00, 0.04), (8.00, 0.1), (12.0, 0.2), (16.0, 0.4), (20.0, 0.54),
(24.0, 0.68), (28.0, 0.68), (32.0, 0.68), (36.0, 0.65), (40.0, 0.64), (44.0, 0.62),
(48.0, 0.58), (52.0, 0.54), (56.0, 0.5), (60.0, 0.4), (64.0, 0.32), (68.0, 0.14),
(72.0, 0.00)
Actual_PCBR = DELAY(PCBR,0.5)
Total_Birth_Rate = Actual_PCBR*Population
PCDR = GRAPH(Crowding)
(0.00, 0.12), (4.00, 0.12), (8.00, 0.12), (12.0, 0.12), (16.0, 0.12), (20.0, 0.12),
(24.0, 0.12), (28.0, 0.12), (32.0, 0.12), (36.0, 0.13), (40.0, 0.14), (44.0, 0.16),
(48.0, 0.17), (52.0, 0.21), (56.0, 0.28), (60.0, 0.4), (64.0, 0.52), (68.0, 0.68),
(72.0, 0.92)
Actual_PCDR = DELAY(PCDR,0.5)
Total_Death_Rate = Actual_PCDR*Population

{ RUNTIME EQUATIONS }
Population(t) = Population(t - dt) + (Total_Birth_Rate - Total_Death_Rate) * dt
Pop_Density = Population/Area
Crowding = Pop_Density*EnvironLF
PCBR = GRAPH(Crowding)
(0.00, 0.00), (4.00, 0.04), (8.00, 0.1), (12.0, 0.2), (16.0, 0.4), (20.0, 0.54),
(24.0, 0.68), (28.0, 0.68), (32.0, 0.68), (36.0, 0.65), (40.0, 0.64), (44.0, 0.62),
(48.0, 0.58), (52.0, 0.54), (56.0, 0.5), (60.0, 0.4), (64.0, 0.32), (68.0, 0.14),
(72.0, 0.00)
Actual_PCBR = DELAY(PCBR,0.5)
Total_Birth_Rate = Actual_PCBR*Population
PCDR = GRAPH(Crowding)
(0.00, 0.12), (4.00, 0.12), (8.00, 0.12), (12.0, 0.12), (16.0, 0.12), (20.0, 0.12),
(24.0, 0.12), (28.0, 0.12), (32.0, 0.12), (36.0, 0.13), (40.0, 0.14), (44.0, 0.16),
(48.0, 0.17), (52.0, 0.21), (56.0, 0.28), (60.0, 0.4), (64.0, 0.52), (68.0, 0.68),
(72.0, 0.92)
Actual_PCDR = DELAY(PCDR,0.5)
Total_Death_Rate = Actual_PCDR*Population