Dynamical Models in Systems Biology

Samuel Bernard bernard@math.univ-lyon1.fr

September 3th, 2019

What is a dynamical model?

A dynamical model is a mathematical or computer model where the variables, or quantities of interest, vary in time. They usually do so according to a causal mechanism, i.e. the values of the variables at a given time depend on their values in the past.

Perhaps the simplest dynamical models are **difference equations**

$$\overbrace{x_{t+1}}^{\text{dynamical variable}} = \overbrace{f(x_t)}^{\text{update rule}} .$$

Time is discrete, and the causal mechanism is clear: x_{t+1} is a function of x_t. For example, the famous *logistic map* is

$$x_{t+1} = \overbrace{r}^{\text{growth rate}} x_t \overbrace{(1-x_t)}^{\text{limiting factor}} .$$

The variable x is the size of the population of a certain species, expressed as a fraction of a maximal size. The logistic equation was used to show how the size of a population can follow a complex dynamics in absence of environmental fluctuations (May, 1976). A small populations grows with a multiplication factor r (factor rx). When it gets larger, the growth is impeded by the factor $1-x$, which could represent competition for resources. We will see later how such a simple equation can lead to complex dynamics.

The most important class of dynamical models are **ordinary differential equations** (ODEs). The time t varies continuously, and the causal mechanism $f(x(t))$ describes how variables $x(t)$ should vary (this is the derivative on the

left-hand side) as a function of their current state:

$$\overbrace{\frac{dx}{dt}}^{\text{rate of change}} = \overbrace{f(x)}^{\text{rule of change}} .$$

The **state variable** x is a function of time, $x(t)$, but it is usual to drop the explicit dependence on time to highlight the fact that changes in the system depend on the state variable rather than time. Many physical laws can be formulated is such a way, and big parts of biology also have their laws, which we express as **motifs**. The *Lotka-Volterra model* was developed to study predator-prey interaction. The mechanism is reminiscent of the logistic equation. The prey density y grows at a constant rate by, and is harvested by the predator at a rate proportional to the predator density xy, while the predator density grows at the harvest rate and dies at a constant rate ax. The equation can be expressed as

$$\begin{aligned}
\frac{dx}{dt} &= \overbrace{yx}^{\text{harvest term}} - \overbrace{a}^{\text{death rate}} x, \quad \text{predator,} \\
\frac{dy}{dt} &= \overbrace{b}^{\text{growth rate}} y - \overbrace{xy}^{\text{harvest term}}, \quad \text{prey.}
\end{aligned}$$

The term dx/dt denotes the derivative of x with respect to time. The derivative is the rate or the speed at which the variable changes. Difference equations and ODEs may have analytic, or closed form solutions, but these are the exception, not the rule. These solutions, when we can find them, are not as useful as one might think; they are often opaque and do not provide additional insight into the behavior of the model.

In general, dynamical models include constant **model parameters**, that do not vary with time. Parameters are important because they can affect the behavior of the solution, and they often have biological significance. Estimating parameter values given a model and experimental data is the subject of the session Inferring model parameters.

Rather than looking for analytical solutions, we will use a combination of computer simulations and stability analysis to characterize the behavior of the dynamical models. `R` codes necessary to simulate the logistic equation and the Lotka-Volterra model are available on the [Systems Biology Class webpage].

Other dynamical modeling formalisms include *stochastic processes*, *partial differential equations*, *individual-based models*. These are out of scope of this introduction to dynamical models.

Why is it important in systems biology? Dynamic models provide mechanisms, and mechanisms provide understanding, which provide ground for validating results.

How to use dynamical models?

Here is a modelling-centric workflow for using dynamical models. Each step may, and will, fail; this is normal. Then go back to previous step and start over.

- **What biological question do I want to model?**

  Two favorite questions of mine are: Can we reproduce these strange looking data with a very simple model (sufficient mechanism), and what are the conditions for my treatment to work (necessary mechanisms). These questions are best studied with dynamical models because they relate to mechanisms. When the mechanism is well understood, we can try to estimate model parameters.

- **What would be the appropriate model for it?**

  Choose the variables

  The choice mostly depend on the availability of experimental data and the question asked.

  Select the mechanisms

  Here the devil is in the details. Which of the known mechanism should we include? Here there is no fixed method, but everybody would agree that given two models with similar solution, the simplest model should be favored. This is the Occam's razor, or parsimony principle. By simple, we mean few parameters, dynamical variables and nonlinear terms (in that order).

- **What type of data do I need?**

  Are data available, or should I collect new data?

- Find an existing model, or develop a new one

  Translating the mechanisms into equations has many pitfalls. Ambiguous language or imprecise wording makes it difficult to define equations uniquely. Once interpreted, the mechanism must satisfy physical and biological constraints, which may be easily overlooked.

- Get an intuition of the behavior of the model

  Very important step. How should the solution look like? This step is useful to detect any error in the equations or in the numerical implementation of the model (they are not the same! See below).

- Implement the model and run simulations to confirm intuition

  The numerical implementation of the model is not the same as the equations. Furthermore the equations might not represent well the mechanisms.

- **Compare to experimental data**

  Before any attempt to fit the model to experimental data, it is important to look whether the model reproduces the important features of the data. Only then fitting the model makes sense.

- Perform analysis of the model

  Models can reproduce experimental data very well for some sets of parameters, but may be fragile in the sense that small change of parameters can lead to vastly different dynamical behavior. This can be a weakness when the system is expected to be robust, but may also provide testable predictions: can the different behaviors be observed experimentally?

- Answer the question Once the model is validated and the parameters are known, do not forget to answer the initial question!

- Find a new question

Modeling with motifs

Unlike chemical and physical systems, biological systems are not easy to reduce to simple parts. Whether we look at the gene expression, protein interaction, cell fate, metabolism, tissue or organ physiology, *in vivo* systems are complex and interrelated. This does not mean that it is impossible to isolate single mechanisms, but that there is no fundamental rules on how to express them. For example, how gene expression is affected by a transcription factor depends on the availability of the binding site, which depends on the DNA conformation, which depends on histone acetylation, and so on. **Motifs** are small blocks of regulation that can be used to distill all the complexity of biology into simple parametric term.

In the following list, the variable x denote the species of interest. This can be gene expression level, mRNA or protein concentration, cell density, drug concentration.

- Loss/death/degradation rate
  - Linear. The species dies or is removed at a rate proportional to its level, with a constant k: kx.

  Example: a protein with initial concentration x_0 is degraded at a rate $k = 0.1$ per hour, and is not synthesized. The equation for the concentration of x is $dx/dt = -kx$, $x(0) = x_0$. This ODE has an explicit solution $x(t) = x_0 e^{-kt}$, the concentration decreases exponentially.

  - Saturated. The loss rate is linear with constant k_0/K when x is small, but saturate to a fixed value k_0 when x is large. The simplest model for saturated kinetics is the Michaelis-Menten model: $k_0 x/(K + x)$.

- Constant production rate. Production refers to a supply of the species that does not depend directly on its concentration: b.
- Proliferation/reproduction/synthesis rate
  - Linear: rx,
  - Logistic (competition) $rx(1 - x/K)$.
  - Negative feedback: $r_0/(K^h + x^h)$.
  - Positive feedback: $r_0 x^h/(K^h + x^h)$.

The parameter h is a cooperativity coefficient, called Hill coefficient. It defines the strength of the feedback. High Hill coefficient will make the feedback quite sensitive to small variations of x. This can lead to complex dynamics such as **oscillations** and **bistability**.

Examples

Examples are implemented in `R`, with the package `deSolve`. All major programming languages offer some numerical solvers: `matlab`, `python`. Here we use `R` because it offers many functionalities to deal with complex datasets as found in systems biology. `Python` and `matlab` also offer similar functionalities but user-friendliness may vary.

Example 1 Birth/death model

Here is the simplest ODE model we can think of, the linear birth-death ODE model,

$$\frac{dx}{dt} = \overbrace{b}^{\text{immigration}} + \overbrace{rx}^{\text{proliferation}} - \overbrace{kx}^{\text{loss}}.$$

The species has a constant production rate b (immigration), a linear growth rate r (proliferation) and a linear death or loss rate k.

```
birthDeath <- function(Time, State, Pars) {
  with(as.list(c(State, Pars)), {

    # ------------------------------------------------
    # Define the equations here
    # ------------------------------------------------
    deathRate         <- k * x
    productionRate    <- b
    proliferationRate <- r * x
```

```
    dxdt     <- productionRate + proliferationRate - deathRate

    return(list(c(dxdt)))
    # ------------------------------------------------
  })
}

pars  <- c(k  = 0.5,     # per day, death rate
           b  = 0.2,     # individuals per day, production from outside source
           r  = 0.1 )    # per day proliferation (or reproduction) rate

y0  <- c(x = 1.0)
timespan <- seq(0, 20, by = 0.1)
birthDeath.sol <- ode(y0, timespan, birthDeath, pars)
summary(birthDeath.sol)
```

To plot the result

```
plot(birthDeath.sol)
```

Exercises on the birth/death model

(a) With the parameters given above, the solution $x(t)$ converges to a constant value, which one?

(b) The solution does not always converge to a constant. Find conditions on the parameters so that the solution always converge to a positive value given a positive initial condition.

(c) How can the equation be modified to ensure that the solution will always remain bounded given positive initial conditions? Write down the modified birth/death model and try to guess to which value the solution will converge.

(d) Implement the modified birth/death model in `R` and run simulations to verify your intuition.

Example 2 Lotka-Volterra

We have seen above the equations for the Lotka-Volterra model

$$\frac{dx}{dt} = \overbrace{yx}^{\text{harvest term}} - \overbrace{a}^{\text{death rate}} x, \quad \text{predator},$$
$$\frac{dy}{dt} = \overbrace{b}^{\text{growth rate}} y - \overbrace{xy}^{\text{harvest term}}, \quad \text{prey}.$$

The `R` code for the Lotka-Volterra equations

```
LotkaVolterra <- function(Time, State, Pars) {
  with(as.list(c(State, Pars)), {

    # ------------------------------------------------
    # Define the equations here
    # ------------------------------------------------
    killingRate       <- Prey * Predator
    preyGrowthRate    <- rGrowth * Prey
    predatorDeathRate <- rDeath * Predator

    dPreydt     <- preyGrowthRate - killingRate
    dPredatordt <- killingRate - predatorDeathRate

    return(list(c(dPreydt, dPredatordt)))
    # ------------------------------------------------
  })
}

pars  <- c(rGrowth  = 0.5,    # per day, growth rate of prey
           rDeath   = 0.2 )   # per day, death rate of predator

y0  <- c(Prey = 10, Predator = 2)
timespan <- seq(0, 200, by = 1)
LotkaVolterra.sol <- ode(y0, timespan, LotkaVolterra, pars)
summary(LotkaVolterra.sol)
```

The solution can be plotted with

```
plot(LotkaVolterra.sol)
```

Exercises on the Lotka-Volterra model

(a) Run the simulations with different initials conditions. What do you observe?

(b) Modify the code above so that the growth rate of the prey also include competition between the preys for resources.

Example 3 A negative feedback loop (Goodwin model)

Negative feedback loops occur everywhere were the product inhibits its own production. This can be through limited food or space, or through homeostatic regulation, to control body temperature or blood pressure for instance. In most cases negative feedback loops have a the effect of making steady state more stable, i.e. after external perturbation the system quickly returns to its natural state. This useful for instance for rapid red blood cell mobilisation after blood loss. However too much of a good thing can have unintended effects, and negative loop

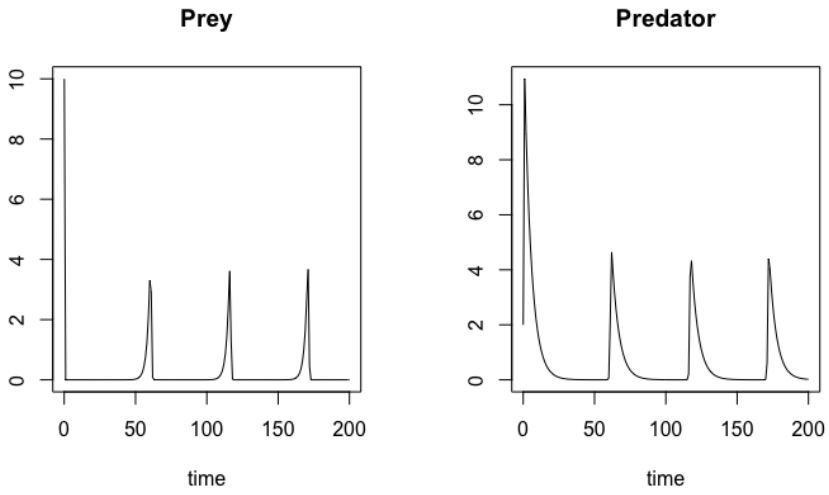

Figure 1: Lotka-Volterra dynamics

can destabilize an otherwise stable steady state. This occurs in the Goodwin model below.

The Goodwin model is the prototype of the negative feedback loop that occurs in many gene regulation networks. To work properly, we need three species. Here we take mRNA concentration X, a protein product concentration Y and a modified protein complex concentration Z. All species have linear degradation rates. The protein is produced at a rate proportional to the mRNA concentration, and the protein complex is produced at a rate proportional to the protein concentration. For simplicity, we set the degradation and production rates of the protein and the complex to the value β, and the mRNA degradation rate to α. To construct the negative feedback loop, we will assume that the protein complex binds to the gene promoter to repress mRNA synthesis is a concentration-dependent manner. Moreover we will assume that in absence of the repressor, mRNA is transcribed (produced) at a constant rate.

Using the negative feedback motif, we can write down the mRNA synthesis rate as

$$f(Z) = k_0 \frac{K^h}{K^h + Z^h}.$$

When there is no repressor (Z=0), the synthesis rate is k_0, and when the repressor expression is $Z = K$, the synthesis rate is reduced by half $k_0 K^h/(K^h + K^h) =$

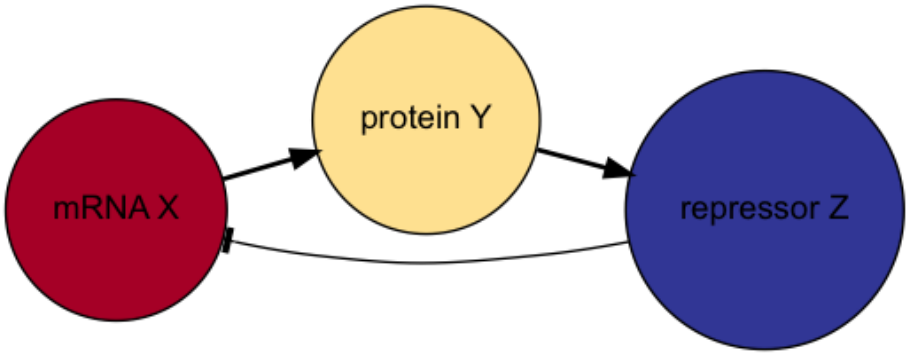

Figure 2: Diagram of the Goodwin network. Arrow heads denote positive effect and "tee" heads denote negative effects.

$k_0/2$. We obtain a set of three ODEs

$$\frac{dX}{dt} = f(Z) - \alpha X,$$
$$\frac{dY}{dt} = \beta(X - Y),$$
$$\frac{dZ}{dt} = \beta(Y - Z).$$

The `R` code to implement the Goodwin model

```
Goodwin <- function(Time, State, Pars) {
  with(as.list(c(State, Pars)), {

    # ------------------------------------------------
    # Define the equations here
    # ------------------------------------------------
    mRNAproductionRate <- k0*K^h/(K^h + Z^h)
    mRNAdegradationRate    <- alpha * X

    dXdt <- mRNAproductionRate - mRNAdegradationRate
    dYdt <- beta * ( X - Y )
    dZdt <- beta * ( Y - Z )

    return(list(c(dXdt, dYdt, dZdt)))
    # ------------------------------------------------
  })
}

pars  <- c(k0      = 2,    # transcripts per hour, max mRNA synthesis rate
           alpha   = 1.0,  # per hour, mRNA degradation rate
```

```
            beta    = 1.0,  # per hour, kinetic rate
            K       = 1,    # mmol, half-repression concentration
            h       = 20 )  # no unit, Hill coefficient

y0  <- c(X = 1, Y = 0, Z = 0)
timespan <- seq(0, 20, by = 0.1)
Goodwin.sol <- ode(y0, timespan, Goodwin, pars)
summary(Goodwin.sol)
```

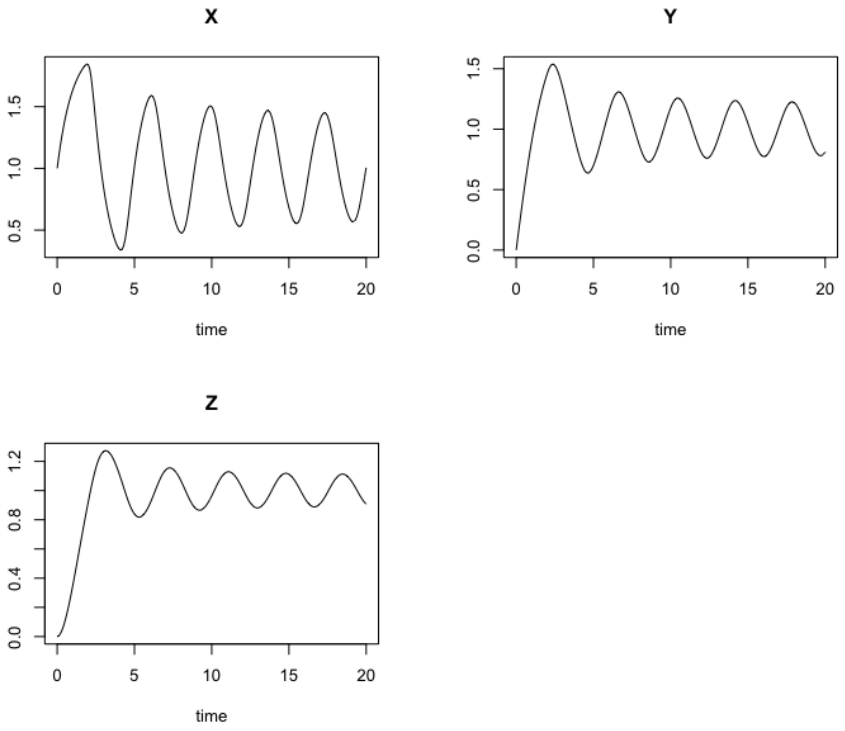

Figure 3: Solution of the Goodwin model. Solutions are oscillatory.

Note There is a Goodwin model in the field of economics as well, and to make thing confusing, the economic Goodwin model is mathematically equivalent to the Lotka-Volterra model. Thus although both economic and biological Goodwin model can display oscillations, they are completely unrelated to each other.

Exercises on the Goodwin model

(a) There is a unique positive steady state. Using the parameters values in the `R` code, find it (set all derivative to zero in the ODE system and solve the three equations for X, Y and Z).

(b) Vary the Hill coefficient h until the steady state becomes stable. What is the value of h? At this value, we say that the Goodwin model undergoes a bifurcation, i.e. a change in the qualitative behavior of the system. Many diseases are associated to qualitative changes in physiology, and dynamical models are used to devise therapeutic strategies to reverse bifurcations. The most successful ones are for heart arrhythmia, such as calcium channel blockers or pacemakers.

Example 4 A positive feedback loop

Positive feedback loop are inherently unstable. They do occur in irreversible events such as mitosis, birth giving, differentiation and lineage choice, etc.

The positive feedback loop rests on the positive loop motif for the production of the species with concentration X.

$$g(X) = k_0 \frac{X^h}{K^h + X^h},$$

where the Hill coefficient $h > 1$. The production depends strongly on X. For low concentrations, the production is low. However for high concentrations, the production is much higher. This nonlinear production curve leads to possible low and high concentrations stable steady states. The ODE reads

$$\frac{dX}{dt} = g(X) - aX.$$

When they exist, the two stable steady states are always separated by a third steady state, which is unstable. They can be found by setting $dX/dt = 0$. This leads to an fixed point equation on X: $aX = g(X)$.

The `R` code to obtain bistability is

```
Bistability <- function(Time, State, Pars) {
  with(as.list(c(State, Pars)), {

    # ----------------------------------------------
    # Define the equations here
    # ----------------------------------------------
    mRNAproductionRate <- k0*X^h/(K^h + X^h)
    mRNAdegradationRate    <- a * X

    dXdt <- mRNAproductionRate - mRNAdegradationRate

    return(list(c(dXdt)))
    # ----------------------------------------------
  })
}
```

```
pars  <- c(k0      = 2,    # transcripts per hour, max mRNA synthesis rate
           a       = 1.0,  # per hour, mRNA degradation rate
           K       = 1,    # mmol, half-repression concentration
           h       = 20 )  # no unit, Hill coefficient

y0  <- c(X = 1.1)
timespan <- seq(0, 20, by = 0.1)
Bistability.sol1 <- ode( c(X = 0.9), timespan, Bistability, pars)
Bistability.sol2 <- ode( c(X = 1.1), timespan, Bistability, pars)
Bistability.sol3 <- ode( c(X = 1.0), timespan, Bistability, pars)
```

To plot all three solutions on one graph

```
plot(Bistability.sol1, Bistability.sol2, Bistability.sol3)
```

Exercises for the bistable model

(a) With the parameters given in the code above, find (approximately) all three steady states. Which ones are stable, unstable?

Example 5 The logistic map

The logistic map is the difference equation

$$x_{t+1} = rx_t(1 - x_t),$$

for $t = 0, 1, ...,$ with the initial condition x_0 given. The `R` codes to solve the logistic map follow the same lines as ODE models, except that the we use the `iteration` method.

```
LogisticMap <- function(Time, State, Pars) {
  with(as.list(c(State, Pars)), {

    # ---------------------------------------------
    # Define the equations here
    # ---------------------------------------------
    xiter     <- r * x * ( 1 - x )

    return(list(c(xiter)))
    # ---------------------------------------------
  })
}

pars  <- c(r  = 3.76 )     # basal growth parameter

y0  <- c(x = 0.2)
timespan <- seq(0, 50, by = 1)
```

```
LogisticMap.sol <- ode(y0, timespan, LogisticMap, pars, method = "iteration")
summary(LogisticMap.sol)
```

The logistic map is one of the simplest dynamical model displaying chaos, oscillatory solutions but very irregular and sensitive to initial conditions. Chaos arises as the parameter r increases from 1.0 to 4.0. For small values of r, the logistic map has one stable steady state. As the parameter is increased, the steady state becomes unstable and is replaced by a periodic solution with period 2. This is followed by a series of period doubling bifurcations, ultimately ending up in chaos.

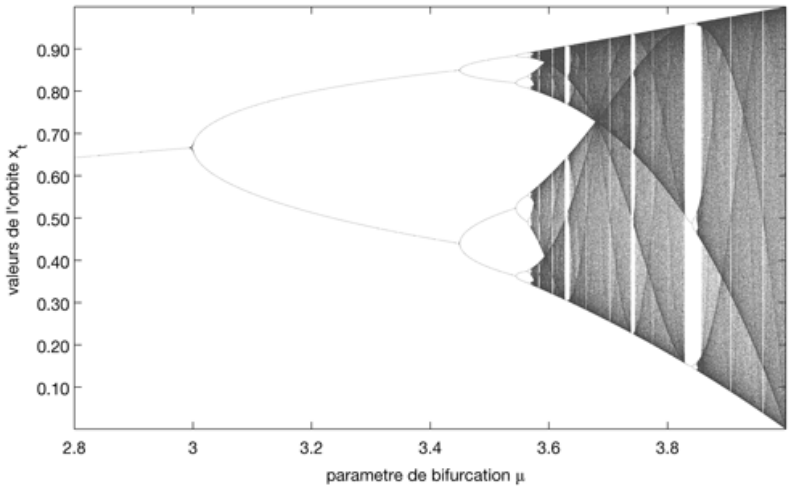

Figure 4: Period doubling bifurcation road to chaos in the logistic map

Exercises for the logistic map

(a) Explore the solutions of the logistic map. Try to find solutions with periods 2, 4, 8... Can you find a solution with period 3? With period 5?

Glossary (English/French)

Dynamical Model/modèle dynamique A dynamical model is a system in which equations describe the time dependence of a set of variables in a geometrical space. Difference equations and ODEs are dynamical models when the independent variable is time.

Dynamical variables/variables dynamiques Dynamical variables are variables that change over time, by opposition to constant parameters. Also called state variables.

Difference equation/équation aux différences A difference equation is an equation that sets a relationship between the values of state variable at finite differences an independent variable (here the indepedent variable is time). It is usual to denote the value of the variable x at time t by x_t for $t = 1, ...$, to indicate that the time t takes discrete values.

Ordinary differential equations/équations différentielles ordinaires An ordinary differential equation is an equation that sets a relationship between a set of variables and their derivatives with respect to continuous independent variable (here the independent variable is time).

Model parameter/paramètre du modèle Model parameters are constant value contained in dynamical models.

Initial conditions/conditions initiales Initial conditions are the values of the state variables at the beginning of the simulation (usually at t=0, but not necessarily). Initial conditions are needed because dynamical models only provide relations between states, not absolute values.

Steady state/état d'équilibre A steady state is a special solution of a dynamical system such that, if the initial conditions are on the steady state, the solution remains on the steady. For an ODE $\bar{x}$ is a steady state if $d\bar{x}/dt = 0$. For a difference equation $x_{t+1} = f(x_t)$, $\bar{x}$ is a steady state if $\bar{x} = f(\bar{x})$. A steady state is stable if solutions with initial conditions close to the steady state will stay close to the steady state.

Oscillations/oscillations Oscillations is a type of non-constant solution where at least one of the variables comes back through a certain value regularly, for any amount of time.

Bistability/bistabilité Bistability is a property of a system where there exists two stable states. Which stable state will attract solution depends on the initial condition. Switch between stable states can be obtained by perturbing the system.

Motifs/motifs Motifs are small blocks of regulation that can be used to distill all the complexity of biology into simple parametric term.

References

- RM May, Simple mathematical models with very complicated dynamics (1976) Nature 261:459–467
- Lotka-Volterra as individual-based model http://www.ahahah.eu/trucs/pp/
- An Introduction to R https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf
- E Klipp et al., Systems Biology, Wiley-VCH Weinheim 2009

- U Alon, An introduction to Systems Biology: Design Principles of Biological Circuits, Chapman & Hall/CRC 2007