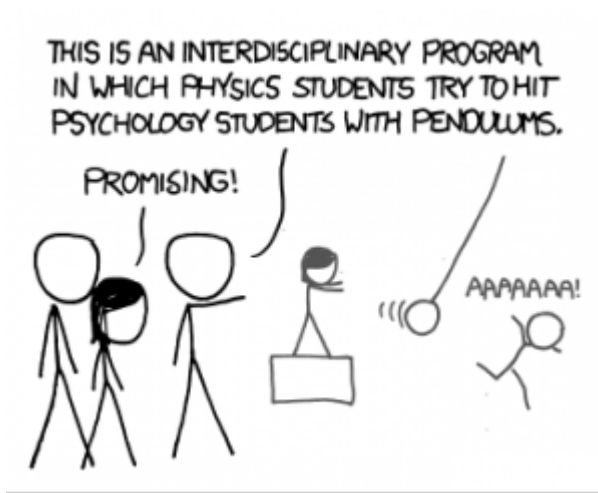


- [Em Sage \(Gráficos online\)](#)
- [Em R](#)
- [In R, english](#)

Stability in dynamical systems - A tutorial in R



In ecology equilibrium and stability are very important concepts, but ecologists have defined them in many different ways. One of the definitions most commonly used was brought from the branch of physics and mathematics called analysis of [dynamical systems](#).

It is this approach that brought to ecology the differential equations used to describe the dynamics of populations, such as [logistic](#) and the [Lotka-Volterra system](#).

There are techniques to assess whether these systems of equations have stationary points, and if they are stable. Robert May (1972) used these tools to prove something surprising: the stability in food webs decreases with the increase of species and webcomplexity.

This exercise is an informal demonstration of the stability analysis of equations that represent dynamical systems, and a simplified version of the procedure used by Robert May. The goal is that you understand the concepts of equilibrium and stability in dynamical systems, which will enable you to distinguish them from other definitions of balance and stability used in ecology. With this you will also be able to critically evaluate the results of May (1972).

R setup

This exercise is ran in R (R Core Team 2011), but you don't need to know the R language, because we'll provide you R codes with the commands ready to run. The codes are in the light-blue boxes in this page, and also in a script file, below in this section. The only thing you need to know is how to send commands written in this file to R. For this you can copy the commands in this page and paste them on the R command line. But it is more straightforward to use the script file. To do this:

1. Install R in your computer, with additional packages `deSolve` and `rootSolve`. The [R site](#) has installation instructions.
2. Create a directory on your computer for this exercise.
3. Copy to this directory the following files:
 1. [eq_funcoes.r](#)

2. [eq_comandos.r](#)

4. Run R from this directory.
5. Open the script file `eq_comandos.r`. For Windows and Mac the R basic installation includes a simple front-end, that allows you to open script files, and send the commands to R. Search the main menu for something like "File > Open Script File". Alternatively, or if you are a Linux user, you can use your favorite [IDE or Code editor](#).
6. Load the required R functions and packages with the commands:

```
library (deSolve)
library (rootSolve)
source ("eq_funcoes.r")
```

That's it! The commands in the script file are in the same order of this page. Run the tutorial by sending the commands to the R command line.

Single population model

Let's start with the stability analysis of the well-known logistic equation of population growth:

$$V = rN \left(\frac{1-N}{K} \right)$$

Where V is the velocity, or rate, of population growth¹⁷¹, r is the intrinsic rate of population growth, and K the carrying capacity.

With the provided function `plota.logist` you can plot the population dynamics described by the logistic model:

```
plota.logist (n = 2, r = 0.1, K = 50, time = 200)
```

The arguments of this function in R allows you to change the parameters of the logistic function:

- `n`: initial population size
- `r`: the intrinsic growth rate
- `K`: carrying capacity
- `time`: end of the time interval to run the dynamic

To understand the effects of each parameter on this dynamical system, you can try changing the parameter values and run the function again.

Technical details

To do this tutorial you need not know how the functions used here to simulate population dynamics work. But if you are curious, in a nutshell these functions use numeric integration routines to calculate the population sizes at small time intervals, and then add these small quantities to get population sizes predicted by the model at each interval.

In R, the package [simecol](#) provides a very amenable wrapper to these computational routines, for simulations of ecological systems.

Stationary point in the logistic

Can a dynamic system reach equilibrium and will it remain there? This is the basic question of stability analysis. But first we have to define equilibrium for differential equations of population dynamics, also known as a stationary state:

An equilibrium, or stationary state, is a population size at which the growth rate is zero, that is, where

$$\frac{dN}{dt} = 0$$

There are two population sizes that satisfy this condition for the logistic equation:

- $N = K$
- $N = 0$

These stationary points make biological sense: the population is not growing when it reaches the carrying capacity or, trivially, when it has no individuals.

Check out that these two population sizes are in equilibrium with the commands:

```
## Logistic starting at N = K
plota.logist (n = 50, r = 0.1, K = 50, time = 200)

## Logistic starting at N = 0
plota.logist (n = 0, r = 0.1, K = 50, time = 200)
```

Stability in the logistic equation

Are these stationary points stable? We'll use R to check this, but first we have to define stability:

A stationary point is stable if the population returns to it after a small perturbation.

A small disturbance is a slight increase or decrease in population size. Hence, we are interested in **local stability**, since it is defined only for small perturbations. This definition and the forthcoming analyses does not evaluate **global stability**, a much more complicated issue.

Our R function to plot the logistic has two arguments to allow disturbances:

- `perturb`: value of the disturbance
- `t.perturb`: when the disturbance occurs

Use these arguments to add one individual to a population at the carrying capacity and to a population with no individuals:

```
## Disturbing when N = K
plota.logist (n = 50, r = 0.1, K = 50, time = 200, perturb = 1, t.perturb =
```

```
100)  
## The same with the population starting at n = 2  
plota.logist (n = 2, r = 0.1, K = 50, time = 200, perturb = 1, t.perturb =  
100)  
  
## Disturbing when N = 0  
plota.logist (n = 0, r = 0.1, K = 50, time = 50, perturb = 1, t.perturb =  
10)
```

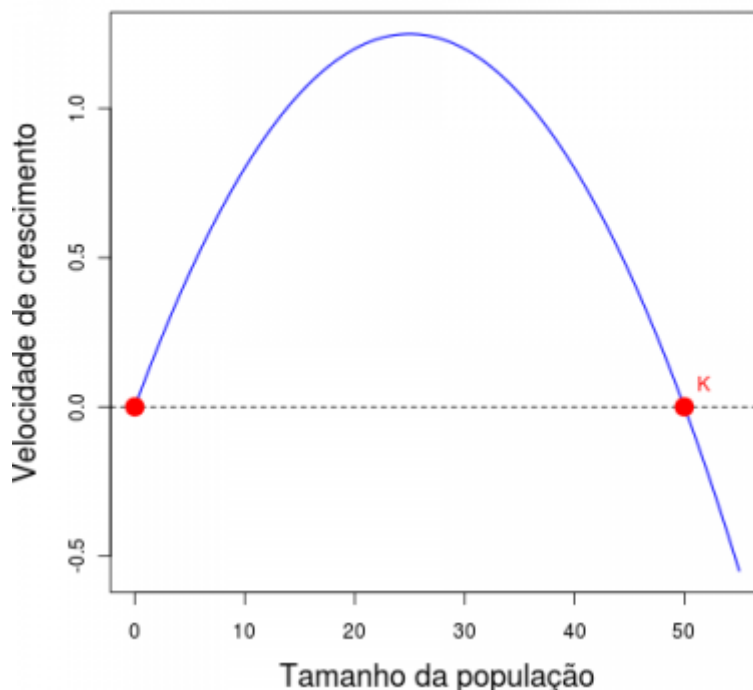
Questions

- Are these points stable?
- What is the biological interpretation?

Mathematical Interpretation

The stability criterion used here evaluates the behavior of the population growth rate when the population size varies slightly around a stationary point. We can state the conditions for stability using simple mathematical concepts. To do this, let's take a look at the relationship between the growth rate and population size in the logistic equation.

Take a look at the figure below: the growth rate, or growth speed, is a quadratic function of population size, tracing a parabola. The red points are the stationary population sizes, where the growth rate is zero ¹⁷²⁾.



When the population is small it increases and the growth rate increases. The population growth is

accelerated by population size.

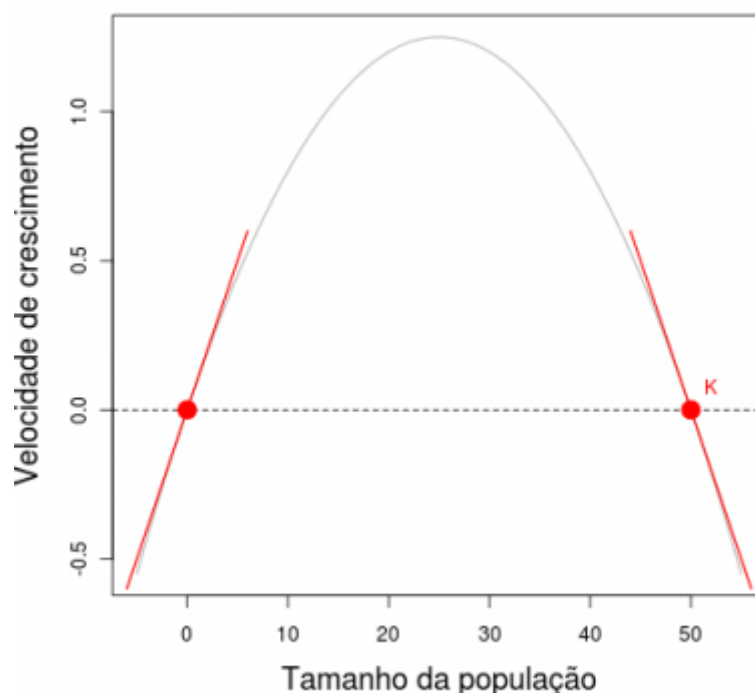
But above a certain population size the increase in population size decreases the growth speed. This is the inflection point of the logistic curve, and from this point on the increase in population size slows ¹⁷³⁾ its growth.

This is the well-known behavior of the logistic growth: it is similar to exponential growth when the population is small, and slows down as the population approaches the carrying capacity. So in this model the growth speed has a positive relationship with population size near the stable point $N = 0$. Therefore, a small increase above zero increases the growth rate, which increases the population size, which further increases the growth rate, and so on. This is an unstable stationary point, because a small perturbation drives the population away from it.

The opposite happens at the point $K = N$, where the growth velocity has a negative relationship with population size. If we reduce the population size a little below K , it will grow, but such growth will slow growth until the growth rate is zero. If we increase the population above the carrying capacity, the growth rate is negative ¹⁷⁴⁾, and thus population size decreases until it reaches K , because the negative growth rate also slows down as population size approaches the carrying capacity. Thus, disturbances in the vicinity of carrying capacity are drawn back to this stationary point.

In short, what defines stability around an equilibrium point is the sign of the relationship between growth rate and population size. This corresponds to the sign of the derivative of velocity with respect to population size at these points.

In a plot of a function, derivatives at a given point can be depicted as the slope of a line tangent to the graph of the function at this point. Below it is the same parabola of the previous figure, now with tangents to the stationary points. As you would expect, the slope of the line is positive at the point $N = 0$ and negative point $N = K$.



Now we can state a general criterion of stability for a population in mathematical terms:

A population size at a stationary point is stable if the derivative of the growth rate with respect to population size at this point is negative.

In mathematical notation this criterion is:

$$\frac{dV}{dN} \Big|_{\hat{N}} < 0$$

which stands for “the derivative of V in relation to N at point \hat{N} is less than zero.”

Two populations

How to generalize the stability criteria for multiple interacting populations?

To answer this we will use the system of [equations of Lotka-Volterra for competition](#):

$$V_1 = r_1 N_1 \left(\frac{K_1 - N_1 - \alpha N_2}{K_1} \right)$$

$$V_2 = r_2 N_2 \left(\frac{K_2 - N_2 - \beta N_1}{K_2} \right)$$

Where, to species 1 and 2:

- V_1 , V_2 are the rates of population growth;
- r_1 , r_2 are the intrinsic growth rates;
- K_1 , K_2 are the carrying capacities;
- α , β are the competition coefficients, that express the effect of each species on the other.

Use the function `plots.LV` to plot the abundances of two competing species that coexist ¹⁷⁵. The arguments are similar to those of the function `plots.logist`:

```
plots.LV (n = c (n1=1, n2 = 1), r1 = 0.2, K1 = 150,  
          r2 = 0.2, K2 = 100, alpha = 0.2, beta = 0.1,  
          time = 150)
```

Stable points with coexistence

In this system of equations the stable points are (Gotelli 2007):

$$\hat{N}_1 = \frac{K_1 - \alpha K_2}{1 - \alpha \beta}$$

$$\hat{N}_2 = \frac{-K_2 + \beta K_1}{1 - \alpha \beta}$$

Function `lv.neq` calculates these values, given the parameter values. You can use it to check the stationary points of the previous plot.

```
(lv.n1 <- lv.neq (r1 = 0.2, K1 = 150, r2 = 0.2, K2 = 100,
  alpha = 0.2, beta = 0.1))
```

Stability

There are two conditions for the coexistence of competitors in the Lotka-Volterra system with two species:

$$\frac{1}{\beta} > \frac{K_1}{K_2} > \alpha$$

$$\frac{1}{\beta} < \frac{K_1}{K_2} < \alpha$$

Let's evaluate the stability of the equilibrium population sizes under these two conditions. The function `plots.LV` also has arguments to change the population size at any time.

We have already calculated the stationary population sizes for the first condition. This values were kept in a object called `lv.n1`. To evaluate the stability at this point just use this object as the argument that defines the initial population size (`n`). In doing so, the dynamics will start from the stationary point. To add a perturbation, use the argument `perturb` to add or subtract a small quantity from each population.

The following command subtracts one individual from population 1 and adds one individual to population 2, at time 50:

```
## Disturbing at the stationary point
## Graphic
plots.LV (n = lv.n1, r1 = 0.2, K1 = 150, r2 = 0.2, K2 = 100, alpha = 0.2,
  beta = 0.1, time = 150, c = perturb (-1.1), t.perturb = 50)
```

Try the same for a combination of parameters that fulfill the second condition of coexistence:

```
## Coexistence of two species, 1/beta > K1/K2 > alpha ##
## Calculation of population sizes at the stationary point
(lv.n2 <- lv.neq (r1 = 0.2, K1 = 150, r2 = 0.2, K2 = 100, alpha = 1.8, beta
  = 0.9))
## Plot
plots.LV (n = lv.n2, r1 = 0.2, K1 = 150, r2 = 0.2, K2 = 100, alpha = 1.8,
  beta = 0.9, time = 150)

## Disturbing at the stationary point
plots.LV (n = lv.n2, r1 = 0.2, K1 = 150, r2 = 0.2, K2 = 100, alpha = 1.8,
  beta = 0.9, time = 200, perturb= c(-1,1), t.perturb = 50)
```

What happens if you reverse the disturbance?

```
## Reverse disturbance
plots.LV (n = lv.n2, r1 = 0.2, K1 = 150, r2 = 0.2, K2 = 100, alpha = 1.8,
  beta = 0.9, time = 200, perturb= c(1, -1), t.perturb = 50)
```

Mathematical Interpretation

The analysis of stability of this two-species system follows the same logic used for the single-species model. A stationary point will be stable if the growth rate (or growth speed) at the point's vicinity has a negative relationship with population size. But now we have two coupled equations, and growth rates of each species depend on the population sizes of the two species.

Partial derivatives

One way to solve the problem is to evaluate the effect of each population on each growth rate, keeping the other population at a fixed size. This is done with **partial derivatives**, which are represented with the symbol ∂ .

For instance:

$$\frac{\partial V_1}{\partial N_1}$$

is the derivative of the growth rate V_1 of population 1 relative to its size N_1 , but with population 2 constant. This derivative expresses the effect that species 1 has on its own population growth. The effect of species 1 on the growth rate of species 2 is

$$\frac{\partial V_2}{\partial N_1}$$

The jacobian matrix

And now the trick is to assemble these partial derivatives in a matrix, which summarizes partial effects of each species on itself and on the competitor:

$$\begin{matrix} \frac{\partial V_1}{\partial N_1} & \frac{\partial V_1}{\partial N_2} \\ \frac{\partial V_2}{\partial N_1} & \frac{\partial V_2}{\partial N_2} \end{matrix}$$

This is the **Jacobian matrix** of the system of differential equations. The diagonal of this matrix has the effect of the populations of each species on themselves. Outside the diagonal we have the inter-specific effects.

The Community matrix

In the study of stability of the logistic equation we evaluated the sign of the derivative $\frac{dV}{dN}$ at the stationary points. Here we have to do the same for each partial derivative. Thus we have the matrix of partial derivatives evaluated at a stationary point. In ecology, the Jacobian of a Lotka-Volterra system evaluated at the stationary points is known as the **community matrix** (May 1972).

Stability criterion

We know that the equilibrium is stable if the relationship between population size and speed in their vicinity is negative. But how to evaluate it now? The generalization of the criterion of local stability for a system of Lotka-Volterra equations is the following:

For a system of Lotka-Volterra equations, a stationary state is stable if the real parts of all eigenvalues of the community matrix are negative.

Eigenvalues are a property of matrices, and a basic concept in linear algebra. They have many applications and implications, but here it is enough to stick to a very informal and intuitive notion:

In our case the eigenvalues express the dominant vectors of the inter and intra-specific effects in the community matrix.

Eigenvalues can have real and imaginary parts. By now, we'll need only the real parts, so do not worry about the other part, if imaginary numbers scare you ¹⁷⁶. The number of eigenvalues of a matrix of communities is equal to the number of species. The calculation of eigenvalues without a computer is laborious, but there are many computer numeric routines to do the job. In R, we can use the function `eigen`.

Calculations in R

Let's calculate the matrix of communities for the first combination of parameters that we had tried above, with the function `jacob.lv`. To do this input the parameter values, and the function returns the population sizes at equilibrium and the Jacobian matrix evaluated at these points:

```
## First combination of parameters: 1/beta < K1/K2 < alpha
## Community matrix
j1 <- jacob.lv (r1 = 0.2, K1 = 150, r2 = 0.2, K2 = 100, alpha = 0.2, beta = 0.1)
```

Now we apply the function `eigen` to get the eigenvalues of this matrix:

```
## Eigenvalues
eigen (j1, only.values == TRUE)$values
```

Now repeat the calculations, using the second combination of parameters we tried:

```
## Second combination of parameters: 1/beta > K1/K2 > alpha
## Community Matrix
j2 <- jacob.lv (r1 = 0.2, K1 = 150, r2 = 0.2, K2 = 100, alpha = 1.8, beta = 0.9)
## Eigenvalues
eigen (j2, only.values == TRUE) $ values
```

Question: The results are consistent with what you have observed disturbing the system?

The adventures of Baron May

The community matrix can be extended to as many species as we want. Robert May (1972) did this to assess whether species diversity increases stability. If so, large stable community matrices would be easier to assemble at random than small ones.

To test this hypothesis, May outlined the following simulation:

1. Set the number of species in the system, S
2. Define the fraction of potential interactions that occur. This is the connectance of the community matrix C
3. Create a matrix with S rows by S columns
4. Fill the diagonal of this matrix with the value -1
5. Fill the remaining cells with random values until the desired connectance is reached
6. Check if the eigenvalues of this matrix are all negative. If so, count this system as stable.

This algorithm generate a system in which intraspecific competition is equal for all species, and there is a fixed proportion of interspecific associations, distributed at random, and with random magnitudes. The sign of these interactions is also drawn at random. May drew inter-specific effects from a normal distribution with zero mean, which makes positive and negative interactions equally likely. In this case, there is another important parameter of the simulation, which is the standard deviation of normal distribution from which the effects are drawn. Larger standard deviations increases the average strength effect. May called this parameter **the interaction strength** of the random community matrix.

Simulating

The provided function may run the original May's simulation, and has the arguments:

- S : number of species
- C : connectance
- f : average strength of interactions
- $nsim$: number of repetitions of the simulation

The function returns the proportion of simulations that produced stable systems ¹⁷⁷⁾.

Let's start with 20 species, connectance of 0.3 ¹⁷⁸⁾ and interaction strength of 0.2:

```
(sim.1 <- may (S = 20, C = 0.3, f = 0.2, nsim = 100))
```

The above command saves the results as a table. Proceed increasing the number of species, but keeping the other parameters constant:

```
(sim.1 <- rbind (sim.1, may (S = 40, C = 0.3, f = 0.2, nsim = 100)))  
(sim.1 <- rbind (sim.1, may (S = 60, C = 0.3, f = 0.2, nsim = 100)))  
(sim.1 <- rbind (sim.1, may (S = 80, C = 0.3, f = 0.2, nsim = 100)))  
(sim.1 <- rbind (sim.1, may (S = 90, C = 0.3, f = 0.2, nsim = 100)))  
(sim.1 <- rbind (sim.1, may (S = 100, C = 0.3, f = 0.2, nsim = 100)))  
(sim.1 <- rbind (sim.1, may (S = 110, C = 0.3, f = 0.2, nsim = 100)))
```

```
(sim.1 <- rbind (sim.1, may (S = 120, C = 0.3, f = 0.2, nsim = 100)))
```

Can you perceive a pattern? A plot may help:

```
## Plot
plot (p.estab ~ S , data = sim.1, xlab = "N species", ylab = "Proportion of
stable matrices")
```

The connectance can be interpreted as a measure of system complexity. Does greater complexity increases the probability of a system being stable? You can test this with:

```
## Fixed species number and intercation strength, increased connectance ##
(sim.2 <- may (S = 120, C = 0.3, f = 0.2))
(sim.2 <- rbind (sim.2, may (S = 120, C = 0.28, f = 0.2)))
(sim.2 <- rbind (sim.2, may (S = 120, C = 0.26, f = 0.2)))
(sim.2 <- rbind (sim.2, may (S = 120, C = 0.24, f = 0.2)))
(sim.2 <- rbind (sim.2, may (S = 120, C = 0.22, f = 0.2)))
(sim.2 <- rbind (sim.2, may (S = 120, C = 0.20, f = 0.2)))
(sim.2 <- rbind (sim.2, may (S = 120, C = 0.18, f = 0.2)))
(sim.2 <- rbind (sim.2, may (S = 120, C = 0.16, f = 0.2)))
## Plot
plot (p.estab ~ C, data = sim.2, Xlab = "connectance", ylab = "Proportion of
stable matrices")
```

Question

Does diversity and complexity lead to stability?

References

- **Gotelli, N. 2007. A primer of ecology. Sinauer** , 3rd Ed. (A basic reference on dynamic models for ecologists).
- **May, RM 1972. Will a large complex system be stable? Nature, 238, 413-414.** (The classical paper which established the concept of stability of food webs as a property of a system of generalized Lotka-Volterra equations.)
- **May, RM 2001. Stability and complexity in model ecosystems. Princeton, Princeton University Press.** (In this influential monograph May develops the ideas of his 1972 paper. The book was issued in 1973, and due its importance to theoretical ecology was reprinted in the collection [Princeton Landmarks of Biology](#) in 2001.)
- **Sarah P. Otto & Troy Day 2007. A Biologist's Guide to Mathematical Modeling in Ecology and Evolution. Princeton, Princeton University Press.** (Good introduction to mathematical modelling, written by biologists to biologists. As in this exercise, has a less traditional and more intuitive approach. A great book for those who do not know linear algebra and want to better understand the details of stability analysis and matrix algebra we use here. see also [the companion site](#).)
- R Development Core Team (2011). R: A language and environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL

<http://www.R-project.org/>.

[R](#), [cálculo](#), [derivada](#), [equação diferencial](#), [lotka-volterra](#), [crescimento logístico](#)

171)

the same as dN/dt

172)

They are the roots of the quadratic equation.

173)

in the terminology of physics, it is negative acceleration

174)

make sure you saw this in the figure

175)

in this case the parameter values were chosen to result in stable coexistence of the two competitors

176)

imaginary numbers are not that scary, anyway. In stability analysis they are also important, because they carry information about oscillations in the system

177)

that is, community matrices with all eigenvalues with negative real parts

178)

30% of the cells of the community matrix has non-zero elements

From:

<http://ecovirtual.ib.usp.br/> -

Permanent link:

<http://ecovirtual.ib.usp.br/doku.php?id=ecovirt:roteiro:math:stabilityenglishr>



Last update: **2016/05/10 07:19**