Course title: EOTIST Standard course

Course subject: Modelling

Teacher: Antonello Provenzale

LESSON SM1.2
HOW TO WRITE AN ECOLOGICAL MODEL - EXERCISES

EOTIST project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 952111

[H2020 WIDESPREAD-05-2020 (Twinning)]

TABLE OF CONTENTS

EOTIST project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 952111

[H2020 WIDESPREAD-05-2020 (Twinning)]

1. THE SOFTWARE

GNU-OCTAVE

This tutorial for the downloading and installation of Ocatve software has been edited by Fasma Diele and Carmela Marangi. The same tutorial is recalled in the lesson SM2.2.

A FREE SOFTWARE COMPATIBLE WITH MATLAB

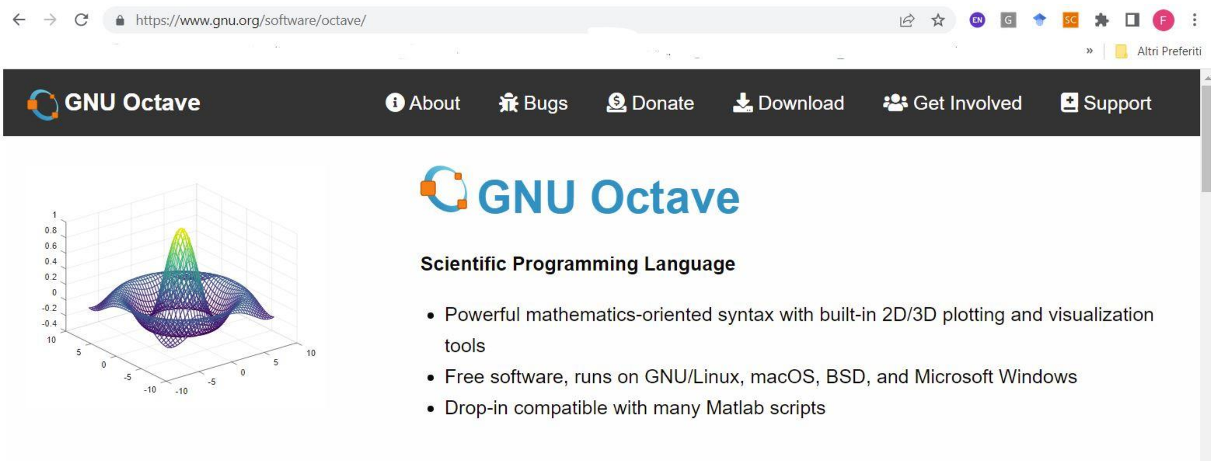
Octave is free software under the GNU General Public License.

GNU Octave is a high-level language, primarily intended for numerical computations. It provides a convenient command line interface for solving linear and nonlinear problems numerically using a language that is mostly compatible with Matlab.

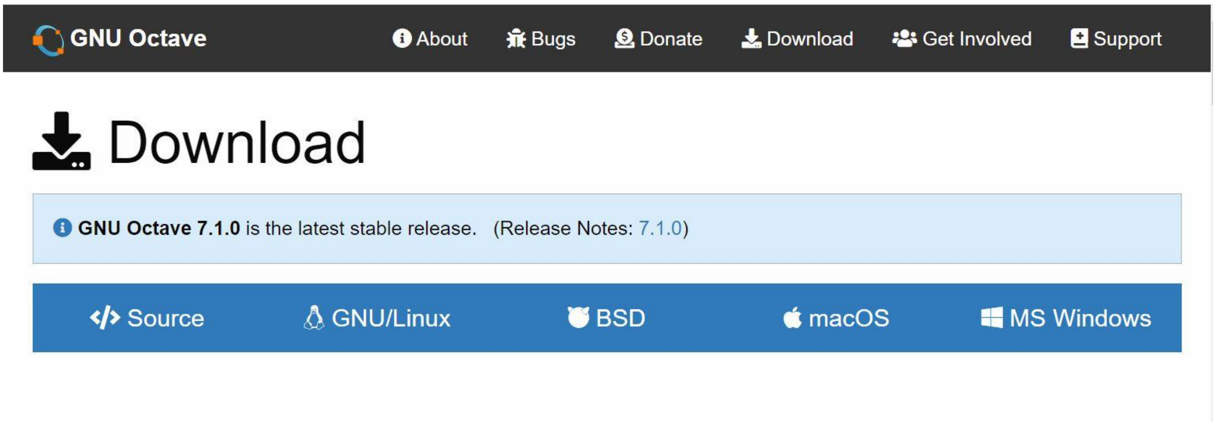
Octave has extensive tools for solving integrating ordinary differential equations.

HOW TO INSTALL OCTAVE

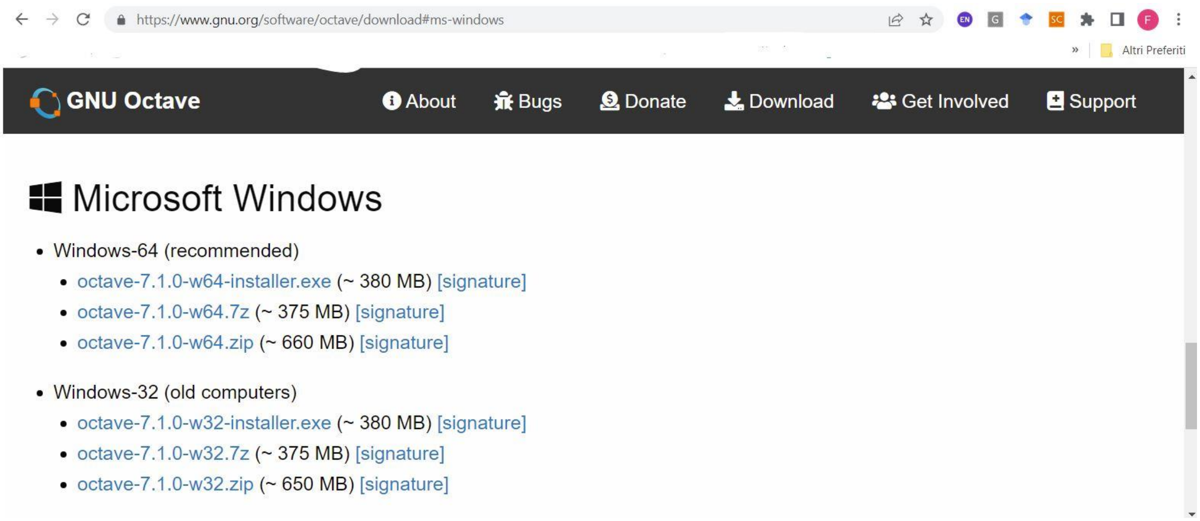
Visit the page https://www.gnu.org/software/octave/



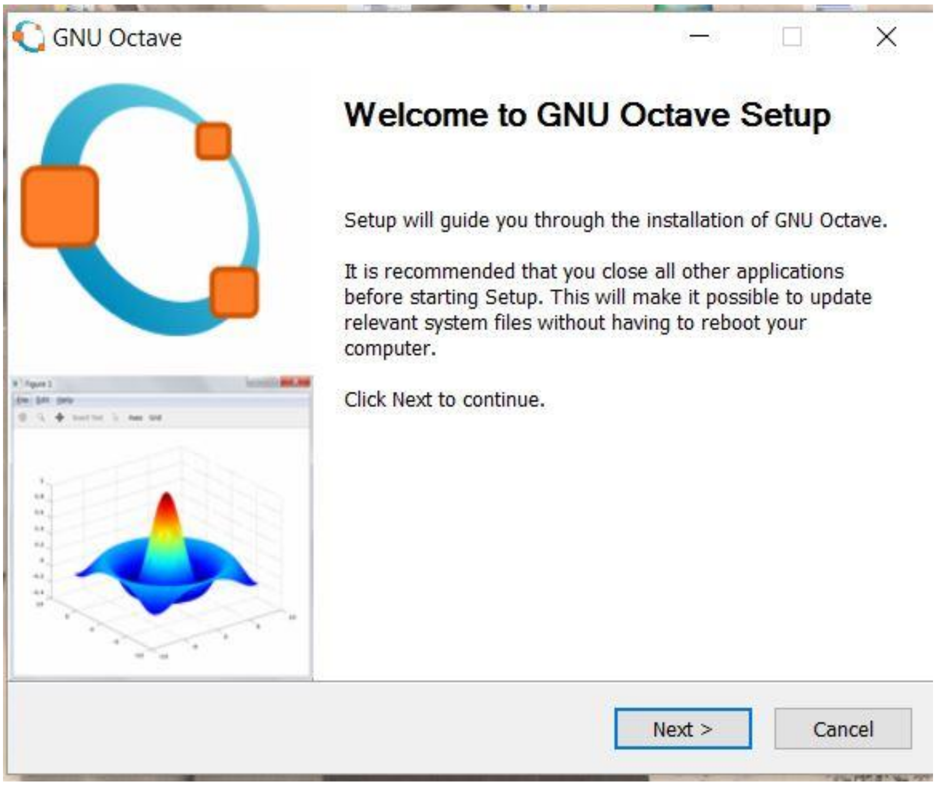
and select ‘Download’



Choose and select the correct framework. For example, by selecting **MS Windows** you will reach the page

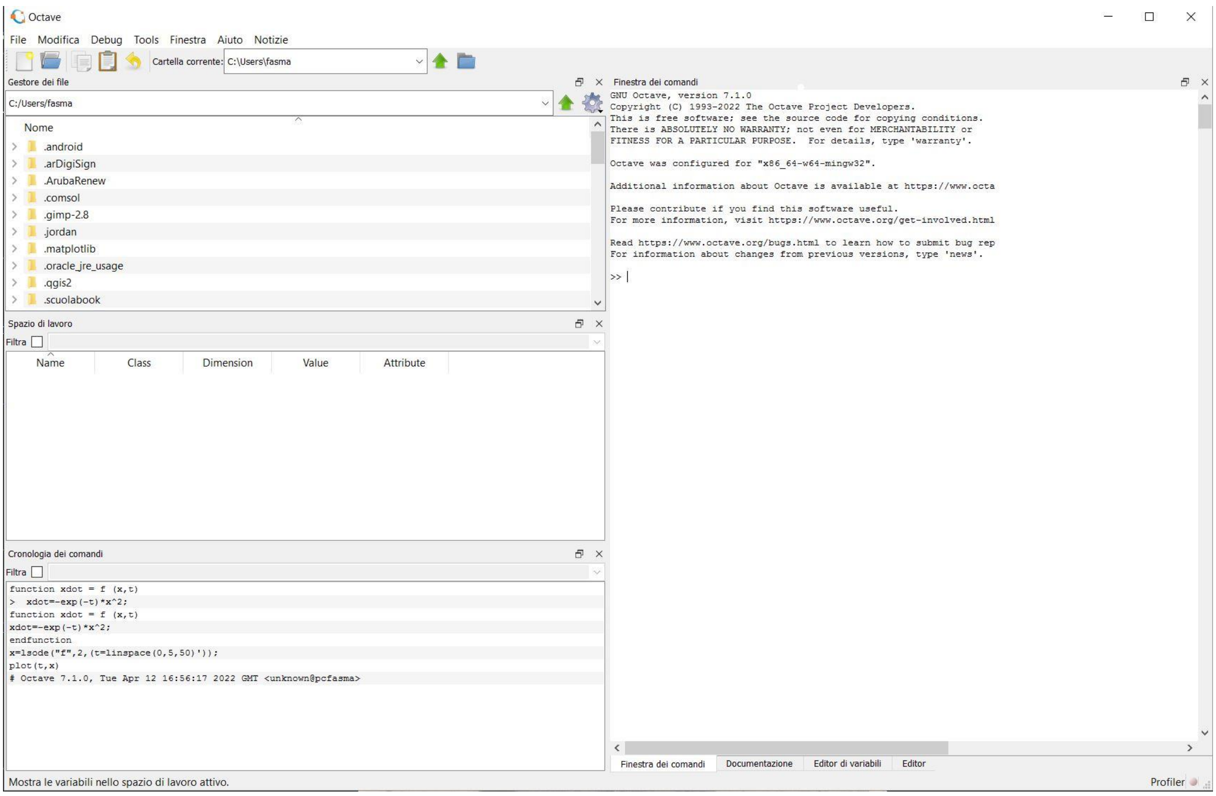


Download and execute the _installer.exe



2. A TUTORIAL FOR THE BASICS OF USING THE SOFTWARE

THE WINDOWS



THE MANUALS

A tutorial for the basics of using GNU Octave is available at

https://wiki.octave.org/Using_Octave.

Before you begin with the ODEs exercises, first read how a system of ODEs has to be implemented in Octave

The online manual at

https://octave.org/doc/v7.1.0/Ordinary-Differential-Equations.html

describes how to implement the ODEs and how to use the functions.

3. WRITE SIMPLE ECOLOGICAL MODELS

In this lesson we will build simple ecological models based on first principles

EXERCISE 1

Using an open programming language (e.g. Octave), integrate the simple logistic map:

1. The logistic map is an iterative mapping expressed as
   $X_{n+1} = 4 A X_n (1-X_n)$
   With $0<A\leq 1$ and $0 \leq X_n \leq 1$
2. Write a script in Octave to find the value of X_{n+1} using a (fixed) value of A and a value of Xn
3. Then, use Xn+1 as X_n (X_{n+1} -> X_n) and find a new value of Xn+1
4. Fix a value of A, take an initial value for X_n between 0 and 1 and iterate the mapping.
5. Explore how the behavior changes for different values of A. Plot, in particular, X_{n+1} versus X_n
6. Discuss some of the behavior types seen for different values of A
7. Take a long series of iterations (at least ten thousand) of the map, and keep only the last 100. Plot such values as a function of the A value (i.e., a point graph with the value of A on the horizontal axis and the corresponding values of the last 100 iterations of the map on the vertical axis).
8. Discuss what happens for different values of A

EXERCISE 2

Using an open programming language (e.g. Octave), integrate in time the Lotka-Volterra model. Use either a Euler time step or a Runge-Kutta-2 time integrator. Both X and Y are functions of time.

1. The Lotka-Volterra system is:

   $dX/dt = X – X Y$
   $dY/dt = -b Y + c X Y$
   where $c < 1$.

2. Look for the fixed points, i.e. those values for which $dX/dt=dY/dt=0$
3. Integrate in time the system above, starting from values of X and Y that are not the fixed points.
4. Plot the solutions (“orbits”) in the plane X-Y, time is the parameter along the orbits. Check whether the system can tend to a fixed point.

EXERCISE 3

Using an open programming language (e.g. Octave), integrate in time the Rosenzweig-MacArthur model. Use either a Euler time step or a Runge-Kutta-2 time integrator. Both X and Y are functions of time.

1. The Rosenzweig-MacArthur model is:

   dX/dt = X (1 – X / K) – C X Y / (1+ R X)
   dY/dt = - b Y + c X Y / (1+ R X)

2. Find the fixed points of the model
3. Integrate in time the system above, starting from values of X and Y that are not the fixed points.
4. Plot the solutions ("orbits") in the plane X-Y, time is the parameter along the orbits. Check whether the system can tend to a fixed point and, if yes, under what conditions.

EXERCISE 4

Using an open programming language (e.g. Octave), integrate in time the NPZ model. Use either a Euler time step or a Runge-Kutta-2 time integrator. The variables N, P and Z are functions of time.

1. The NPZ model is

$$\begin{cases} \dfrac{dN}{dt} = f(N,P,Z) \equiv \Phi_N - \beta \dfrac{N}{k_N + N} P \\ \qquad\qquad + \mu_N \left((1-\gamma) \dfrac{a\epsilon P^2}{a + \epsilon P^2} Z + \mu_P P + \mu_Z Z^2 \right) \\ \dfrac{dP}{dt} = g(N,P,Z) \equiv \beta \dfrac{N}{k_N + N} P - \dfrac{a\epsilon P^2}{a + \epsilon P^2} Z - \mu_P P, \\ \dfrac{dZ}{dt} = h(N,P,Z) \equiv \gamma \dfrac{a\epsilon P^2}{a + \epsilon P^2} Z - \mu_Z Z^2. \end{cases}$$

2. Find the fixed points of the model
3. Integrate in time the system above, starting from values of N, P and Z that are not the fixed points.
4. Plot the solutions N, P and Z as a function of time, for different choices of the parameters. What types of behavior do you see?
5. What happens if you put $\mu_N = 0$?

EXERCISE 5

Using an open programming language (e.g. Octave), integrate in time the vegetation model below. Use either a Euler time step or a Runge-Kutta-2 time integrator. The variables b_1, b_2 and b_3 are functions of time.

1. The vegetation model is

$$\frac{\mathrm{d}b_1}{\mathrm{d}t} = g_1 b_3 - \mu_1 b_1$$

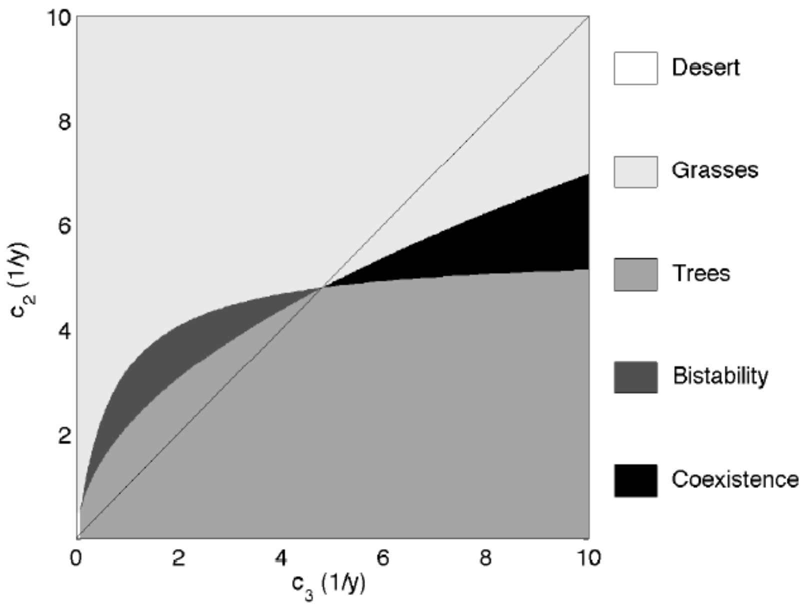
$$\frac{\mathrm{d}b_2}{\mathrm{d}t} = c_2 b_2(1 - b_1 - b_2) - \mu_2 b_2$$

$$\frac{\mathrm{d}b_3}{\mathrm{d}t} = c_3 b_1(1 - b_1 - b_2 - b_3) - \mu_3 b_3 - g_1 b_3 - c_2 b_2 b_3$$

2. Find the fixed points of the model

3. Integrate in time the above model, starting from values of b_1, b_2 and b_3 that are not fixed points.

4. Check whether the solution at long time (i.e. the values of b_1, b_2 and b_3) for some specific values of the parameters are consistent with the plot from Baudena et al:



5. Can you find an oscillating behavior in this model?