Université de Picardie Jules Verne
Cours de l'Ecole Doctorale/Doctoral School's Lectures
Calculs Numériques/Numerical Compuations
2019-2020

ODEs with Scilab

Scilab ODE's solver

In Scilab is implemented a numerical ODE (Ordinary Differential Equation) solver. This solver is rather simple to use and allows efficient simulations and post-treatments of many kind of problems. However it is important to know fiew notions about numerical methods for ODEs, to understand how it works but also to have in mind that any numerical method has its domain of applicability.
Let us consider the so-called Cauchy Problem (initial value problem)

$$\begin{cases} \dfrac{dx(t)}{dt} = f(x(t),t)) \quad t \in)0,T(, \\ \\ x(0) = x_0 \end{cases}$$

We here study only first order ODEs, the higher order equations can be rewritten as first order differential system. The ODE solver is `ode` and is basically used as follows:

- We first define a source or right hand side function: `function [f]= source(t,u)`
- We then generate a sequence of discrete times `t=linspace(t0,T,n)` or `t=t0:dt:T`, `t0` and `dt` being previously defined
- We compute the sequence of approximation `U` at the discrete times as `[U]=ode(u0,t0,t,source)`

Example :

```
t=linspace(0,1,100);
function [f]=oscill(t,theta)
f(1)= theta(2)
f(2)=-sin(theta(1)) -0.5*theta(2)
[U]=ode([0,1],t,oscill);
```

Graphical issues: projection and phase portrait

Projection : we represent the curves $(t, x_i(t)),\ t \in\ [t_0, T]$ for $i = 1, \cdots, n$, if the differential system is defined in $\mathbb{R}^n$.

As an illustration, let us consider the one population dynamical model, including inner competition:

```
t=linspace(0,1,100);
function [f]=popucompet(t,u)
f=u*(2-u);
[U]=ode([0,1],t,popucompet);
plot(t,U)
```

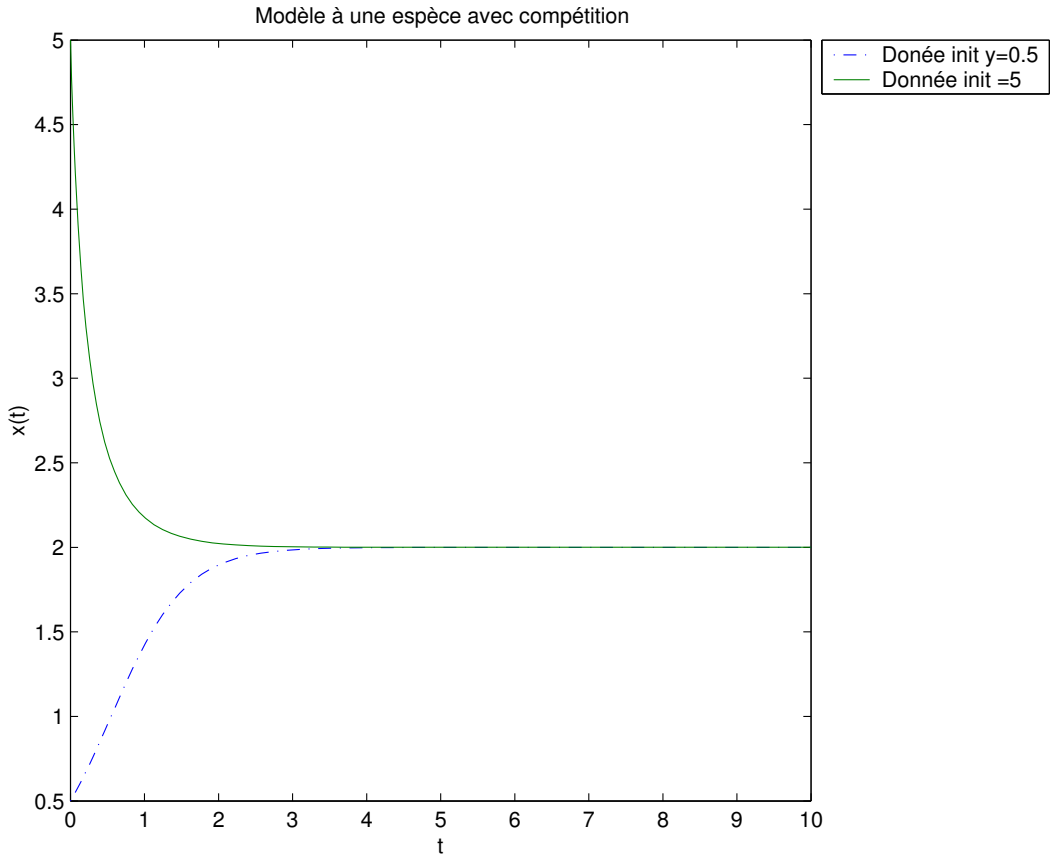

Figure 1: Projection

Phase portrait (following B. Pinçon)

```
// 1/ plot the vector field produced by the Van der Pol equation
n = 30;
delta = 5
x = linspace(-delta,delta,n); // ici y = x
xbasc()
fchamp(VanDerPol,0,x,x)
xselect()

// 2/ solution of the differential equation
m = 500 ; T = 30 ;
t = linspace(0,T,m); // the times at which the solution is approached
u0 = [-2.5 ; 2.5]; // initial condition
[U] = ode(u0, 0, t, VanDerPol);
plot2d(u(1,:)',u(2,:)',2,"000")
```

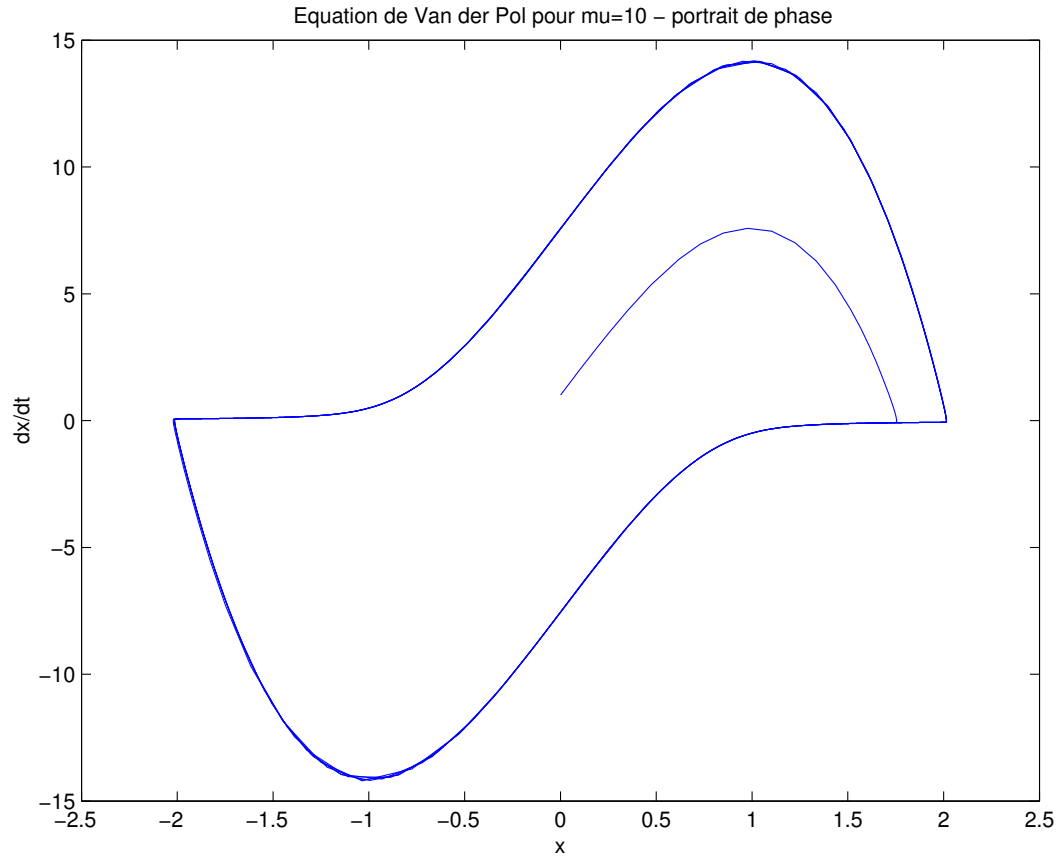

Figure 2: Equation de Van der Pol - Portrait de phase

First numerical schemes in Scilab

1. Two species model

$$\begin{cases} \frac{d}{dt}x(t) = x(1.5 - x - 0.5y), \\ \frac{d}{dt}y(t) = y(2 - 0.5y - 1.5x), \\ x(0) = x_0,\ y(0) = y_0. \end{cases}$$

   Display a phase portrait of the solution

2. The Van der Pol equation describes the vibrations in some physical situations:

$$\begin{cases} \left(\frac{d^2}{dt^2}u(t)\right) - \epsilon(1 - u(t)^2)\left(\frac{d}{dt}u(t)\right) + u(t) = 0,\ t > 0, \\ u(0) = u_0, \\ u'(0) = u_1. \end{cases}$$

   où $\epsilon > 0$.

   (a) Write this equation as a first order differential system

   (b) Solve the system using several methods

   (c) Display a phase portrait of the solution for different values of ϵ, e.g. $\epsilon < 2$, $\epsilon = 1$.