

Numerical methods for Dynamical systems

INF656I

Practical work n° 1

Goal(s)

- ★ Implement a simple simulation engine
- ★ Explicit Euler method
- ★ Fixed step-size and variable step-size

Exercise 1

The goal of this exercise is to implement a simple numerical simulation engine based on Euler's method as a numerical integration scheme.

We recall that an initial value problem for ordinary differential equations is described by

$$\dot{\mathbf{y}} = f(t, \mathbf{y}) \quad \text{avec} \quad \mathbf{y}(0) = \mathbf{y}_0 . \quad (1)$$

The application of Euler's scheme for this equation generates a recurrent equation of the form

$$\mathbf{y}_{n+1} = \mathbf{y}_n + hf(t_n, \mathbf{y}_n) \quad \text{et} \quad t_{n+1} = t_n + h .$$

A simple simulation engine offers a way to simulate a dynamical system or to compute its trajectory starting from a known initial conditions. The pseudo code is

```

Data :  $f$  the dynamic,  $\mathbf{y}_0$  initial condition,  $t_0$  initial time,  $t_{\text{end}}$  final time,  $h$  integration step size
 $t \leftarrow t_0$ ;
 $\mathbf{y} \leftarrow \mathbf{y}_0$ ;
while  $t < t_{\text{end}}$  do
    Print( $t, \mathbf{y}$ );
     $\mathbf{y} \leftarrow \text{Euler}(f, t, \mathbf{y}, h)$ ;
     $t \leftarrow t + h$ ;
end

```

Question 1

Implement this algorithm in Python

Question 2

Use this simulation engine to solve

NB : Think to automatize !!

— A1

$$\dot{y} = -y$$

with $y(0) = 1$

— A3

$$\dot{y} = y \cos(t)$$

with $y(0) = 1$

— B1

$$\begin{aligned}\dot{y}_1 &= 2(y_1 - y_1 y_2) \\ \dot{y}_2 &= -(y_2 - y_1 y_2)\end{aligned}$$

with $y_1(0) = 1$ and $y_2(0) = 3$

— E1

$$\begin{aligned}\dot{y}_1 &= y_2 \\ \dot{y}_2 &= 5(1 - y_1^2)y_2 - y_1\end{aligned}$$

with $y_1(0) = 2$ and $y_2(0) = 0$. The duration of the simulation is 1 second and the integration step size is $h = 10^{-3}$. For all the problem (almost), the simulation time is 20s. By default, $h = 0.01$.

Question 3

For Problem 2, you can consider different values of h . What is the behavior of the simulation engine with h equals to 0.0001, 0.001, 0.01, 0.1, 0.2, 0.5?

Exercise 2

A limitation of fixed step integration methods is that they consider a priori temporal discretization which may not be well adapted to the adapted to the studied dynamics.

One way to implement variable step size with Euler's explicit method is to make several integration step. The idea is to compute for a finite horizon t_n two approximations of the solution $\mathbf{y}(t_n)$ denoted by $\mathbf{y}_n^{(0)}$ and $\mathbf{y}_n^{(1)}$. Hence, we can compute an error $e_n = \mathbf{y}_n^{(1)} - \mathbf{y}_n^{(0)}$ which represents an estimation of the distance to the true solution.

In practice, a first approximation is calculated $\mathbf{y}_n^{(0)}$ with on step size h and a second approximation $\mathbf{y}_n^{(1)}$ with a step size equals to $h/2$. Note that you need 2 simulations steps with $h/2$ to get the approximation for the horizon h .

If the error e_n is greater than a tolerance tol defined by the user then the approximation \mathbf{y}_n should not be considered (we say that the simulation step is rejected) and the simulation should be redo with a smaller step size h . Otherwise we keep the approximation (we says that we accept the simulation step) and we can try to increase h for the next simulation step.

A general formula to make h variable is

$$h_{n+1} = 0.9h_n \min \left(\max \left(\frac{tol}{|e_n|}, 0.3 \right), 2 \right) .$$

The values 0.9, 0.3, 2 limit the variation of h in order to avoid too big or too small step size.

Remark that the initial step size h_0 is given by the user.

Question 1

Implement this method

Question 2

solve problem of Exercice 1 with tol equals to 10^{-2} . Compute plots with fixed-step results.