ENSTA Paris
2020-2021

# Numerical methods for dynamical systems

## Homework n° 2

## Goal(s)

- Implementation of Runge-Kutta methods
- Application of Runge-Kutta methods on different ODE

For the next exercises, we will consider different dynamical systems to test the Runge-Kutta methods. In particular, we will consider

**Non-stiff problems**

— A1 :

$$\dot{y} = y \quad \text{with}$$

with $y(0) = 1$, the final simulation time is 20 seconds.

— 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 $y_2(0) = 3$, the final simulation time is 20 seconds.

**Stiff problems**

— A1 :

$$\begin{aligned}\dot{y}_1 &= -0.5 y_1\\ \dot{y}_2 &= -y2\\ \dot{y}_3 &= -100 y_3\\ \dot{y}_4 &= -90 y_4\end{aligned}$$

with $y_1(0) = y_2(0) = y_3(0) = y_4(0) = 1$, , the final simulation time is 20 seconds and the initial step-size is $h_0 = 10^{-2}$.

— F1 : (chemical reaction)

$$\begin{aligned}\dot{y}_1 &= 1.3(y_3 - y_1) + 10400 k y_2\\ \dot{y}_2 &= 1880(y_4 - y_2(1+k))\\ \dot{y}_3 &= 1752 - 269 y_3 + 267 y_1\\ \dot{y}_4 &= 0.1 + 320 y_2 - 321 y_4\end{aligned}$$

with $k = \exp(20.7 - 1500/y_1)$, initial conditions $y_1(0) = 761$ and $y_2(0) = 0$, $y_3(0) = 600$, $y_4(0) = 0.1$ and a final simulation tome of 1000 seconds, an initial steps size $h = 10^{-4}$

**Other problems**

— Orbit (3 body problems)

$$\begin{aligned}\dot{y}_1 &= y_3\\ \dot{y}_2 &= y_4\\ \dot{y}_3 &= y_1 + 2y_4 - \mu_h \frac{y_1 + \mu}{D_1} - \mu \frac{y_1 - \mu_h}{D_2}\\ \dot{y}_4 &= y_2 - 2y_3 - \mu_h \frac{y_2}{D_1} - \mu \frac{y_2}{D_2}\end{aligned}$$

with $\mu = 0.012277471$ and $\mu_h = 1 - \mu$, $D_1 = ((y_1 + \mu)^2 + y_2^2)^{3/2}$, $D_2 = ((y_1 - \mu_h)^2 + y_2^2)^{3/2}$.
Initial conditions $y_1(0) = 0.994$, $y_2(0) = 0$, $y_3(0) = 0$ et $y_4(0) = -2.00158510637908252240537862224$.
The final simulation time is 35 seconds. Remark that the solution of this system should be perdiodic (period around 17.0652).

## Exercise 1

The goal of this exercise is to implement a simulation engine based on Runge-Kutta methods.
We recall that an initial value probem for ordinary differential equations is defined by

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

Runge-Kutta methods applied on Equation (1) generate an iteration scheme of the form

$$\begin{aligned}
\mathbf{k}_i &= f\left(t_n + c_i h, \mathbf{y}_n + h\sum_{j=0}^{i} a_{ij}\mathbf{k}_j\right) \\
\mathbf{y}_{n+1} &= \mathbf{y}_n + h\sum_{i=0}^{s} b_i \mathbf{k}_i \\
\mathbf{z}_{n+1} &= \mathbf{y}_n + h\sum_{i=0}^{s} b'_i k_i
\end{aligned}$$

Note : $\mathbf{k}_i$ have the same dimension than $f$.

### Question 1

The simplest adaptive explicit Runge–Kutta method combines the Heun's method with the Euler's method. Its Butcher tableau is

$$\begin{array}{c|cc}
0 & & \\
1 & 1 & \\
\hline
 & \frac{1}{2} & \frac{1}{2} \\
 & 1 & 0
\end{array}$$

The integration methods is defined by

$$\begin{aligned}
\mathbf{k}_1 &= f(t_n, \mathbf{y}_n) \\
\mathbf{k}_2 &= f(t_n + h_n, \mathbf{y}_n + h_n\mathbf{k}_1) \\
\mathbf{y}_{n+1} &= \mathbf{y}_n + h_n\left(\frac{1}{2}\mathbf{k}_1 + \frac{1}{2}\mathbf{k}_2\right) \\
\mathbf{z}_{n+1} &= \mathbf{y}_n + h_n(\mathbf{k}_1)
\end{aligned}$$

Implement this method.

### Question 2

We consider a second adaptive Runge-Kutta method which is the Bogacki-Shampine method. Its Butcher tableau is

$$\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \\
\frac{3}{4} & 0 & \frac{3}{4} & & \\
1 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & \\
\hline
 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & \\
 & \frac{7}{24} & \frac{1}{4} & \frac{1}{3} & \frac{1}{8}
\end{array}$$

The integration methods is defined by

$$\begin{aligned}
\mathbf{k}_1 &= f(t_n, \mathbf{y}_n) \\
\mathbf{k}_2 &= f(t_n + \frac{1}{2}h_n, \mathbf{y}_n + \frac{1}{2}h_n\mathbf{k}_1) \\
\mathbf{k}_3 &= f(t_n + \frac{3}{4}h_n, \mathbf{y}_n + \frac{3}{4}h_n\mathbf{k}_2) \\
\mathbf{y}_{n+1} &= \mathbf{y}_n + h_n\left(\frac{2}{9}\mathbf{k}_1 + \frac{1}{3}\mathbf{k}_2 + \frac{4}{9}\mathbf{k}_3\right) \\
\mathbf{k}_4 &= f(t_n + h_n, \mathbf{y}_{n+1}) \\
\mathbf{z}_{n+1} &= \mathbf{y}_n + h_n\left(\frac{7}{24}\mathbf{k}_1 + \frac{1}{4}k_2 + \frac{1}{3}\mathbf{k}_3 + \frac{1}{8}\mathbf{k}_4\right)
\end{aligned}$$

Implement this method

## Question 3

Define a function to handle the adaptive step-size method (see slides 28 and 29 in the lecture). An infinite norm will be used by default

Some constants have to be also defined for this purpose.

- maximal step size $h$
- minimal step size $h$
- relative tolerance
- absolute tolerance

Note that in case of Bogacki-Shampine $\mathbf{y}_{n+1}$ is 3-order approximation while $\mathbf{z}_{n+1}$ is 2-order approximation. This method uses the "Local extrapolation approach"

## Question 4

Try to solve the given problems with the following methods :

- explicit Heun's method
- implicit trapeziodal method
- variable Euler-Heun method
- Bogacki-Shampine's method

Note that appropriate step sizes have to be defined in case of explicit fixed-step methods.

**Remark :** You reuse the source code which is given with the lecture.

## Question 5

Change the simulation engine to compute statistics as the number of accepted and rejected steps during the simulation.

## Question 6

Use different norms in the adaptibe step-size algorithm :

- the Euclidian norm
- the weigthed 2-norm defined by

$$\mathrm{err} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\mathbf{y}_{n+1,i} - \mathbf{z}_{n+1,i}}{sc_i}\right)^2}$$

  with $sc_i = \max(\mathrm{atol}, \mathrm{rtol} \times \max(|\mathbf{y}_{n+1,i}|, |\mathbf{y}_{n,i}|))$

Observe the differences in terms of accepted and rejected steps using these norms.

## Question 7

- Use different tolerances (atol and rtol) in variable step-size methods to detect the limit of numerical stability, *i.e.* for which value the simulation result seems to diverge.
- for fixed-step size methods, play with the step-size to detect the value for which the simulation result diverge

## To submit

- A small report should be sent summarize the answers to the questions.
- This report should be associated to the source code.

Send the archive containing the report and the source codes in a mail which title is

[numerical methods for dynamical systems] FIRSTNAME LASTNAME

to `alexandre.chapoutot@ensta-paris.fr`
**before the next lecture, Friday October 2, 2020.**