

# Numerical methods for dynamical systems

INF656I

## Homework n° 3

### Goal(s)

- ★ Implementation of multi-step methods
- ★ Application of multi-step methods on different ODE

For the next exercises, we will consider different dynamical systems to test the predictor-corrector 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 :

$$\dot{y}_1 = 2(y_1 - y_1 y_2)$$

$$\dot{y}_2 = -(y_2 - y_1 y_2)$$

with  $y_1(0) = 1$  and  $y_2(0) = 3$ , the final simulation time is 20 seconds.

### Stiff problems

— A1 :

$$\dot{y}_1 = -0.5y_1$$

$$\dot{y}_2 = -y_2$$

$$\dot{y}_3 = -100y_3$$

$$\dot{y}_4 = -90y_4$$

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)

$$\dot{y}_1 = 1.3(y_3 - y_1) + 10400ky_2$$

$$\dot{y}_2 = 1880(y_4 - y_2(1+k))$$

$$\dot{y}_3 = 1752 - 269y_3 + 267y_1$$

$$\dot{y}_4 = 0.1 + 320y_2 - 321y_4$$

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 time of 1000 seconds, an initial step size  $h = 10^{-4}$

### Other problems

— Orbit (3 body problems)

$$\dot{y}_1 = y_3$$

$$\dot{y}_2 = y_4$$

$$\dot{y}_3 = y_1 + 2y_4 - \mu_h \frac{y_1 + \mu}{D_1} - \mu \frac{y_1 - \mu_h}{D_2}$$

$$\dot{y}_4 = y_2 - 2y_3 - \mu_h \frac{y_2}{D_1} - \mu \frac{y_2}{D_2}$$

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  $y_1(0) = 0.994$ ,  $y_2(0) = 0$ ,  $y_3(0) = 0$  and  $y_4(0) = -2.00158510637908252240537862224$ .

The final simulation time is 35 seconds. Remark that the solution of this system should be periodic (period around 17.0652).

```

Data :  $f$  dynamics,  $y_0$  initial condition,  $t_0$  starting time,  $t_{end}$  ending time,  $h$  integration step-size
 $t \leftarrow t_0$ ;
 $y \leftarrow y_0$ ;
init  $\leftarrow$  true;
while  $t < t_{end}$  do
  Print( $t, y$ );
  if  $init = true$  then
     $(y, \mathbf{f}_{-1}, \dots, \mathbf{f}_{-p+1}) \leftarrow$  Initialize( $f, t, y, h$ );
    init  $\leftarrow$  false;
  end
   $(y, \mathbf{f}_{-1}, \dots, \mathbf{f}_{-p+1}) \leftarrow$  Solver( $f, t, y, \mathbf{f}_{-1}, \dots, \mathbf{f}_{-p+1}, h$ );
   $t \leftarrow t + h$ ;
end

```

**Algorithm 1** : Pseudo code of simulation engine with multi-step fixed step-size methods

We recall that a numerical simulation engine based on multi-step methods as given in Algorithm 1 where  $\mathbf{f}_{-1}, \dots, \mathbf{f}_{-p+1}$  stand for  $\mathbf{f}$  evaluated at previous  $y_i$  for all  $i = -1, \dots, -p+1$ .

## Exercise 1 – Fixed step-size method

### Question 1

We consider third order Adams-Moulton method defined by

$$\mathbf{y}_{n+1} = \frac{h}{12} (8\mathbf{f}_n + 5\mathbf{f}_{n+1} - \mathbf{f}_{n-1}) + \mathbf{y}_n$$

For the initialization step, we will use the explicit third order Runge-kutta methods defined by

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \left( \frac{1}{6}\mathbf{k}_1 + \frac{2}{3}\mathbf{k}_2 + \frac{1}{6}\mathbf{k}_3 \right)$$

avec

$$\mathbf{k}_1 = f(t_n, \mathbf{y}_n)$$

$$\mathbf{k}_2 = f\left(t_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_1\right)$$

$$\mathbf{k}_3 = f\left(t_n + h, \mathbf{y}_n + h(-\mathbf{k}_1 + 2\mathbf{k}_2)\right)$$

Implement this method

### Question 2

Try solving problems given at the beginning.

## Exercise 2 – Variable step-size method

The goal of this exercise is to implement a simulation engine based on basic multi-step methods based on predictor-corrector approach.

We consider a simple naive predictor corrector method based on second order Adams-Bashforth and Adams-Moulton methods

$$\mathbf{y}_{n+1}^p = \mathbf{y}_n + h \left( \frac{3}{2}\mathbf{f}_n - \frac{1}{2}\mathbf{f}_{n-1} \right)$$

and

$$\mathbf{y}_{n+1} = \frac{h}{2} (\mathbf{f}_n + \mathbf{f}(t_{n+1}, \mathbf{y}_{n+1}^p)) + \mathbf{y}_n$$

The corrector can be iterated a few times to increase accuracy.

### Question 1

Implement this method.

For the initialization step we can use a first-order predictor-corrector approach based on explicit and implicit Euler's methods.

**Question 2**

Define a function to handle the adaptive step-size method (see slides 28 and 29 in Lecture 2). An infinite norm will be used by default.

We have to use Nordsieck vector to fully implement this variable step-size approach.

**Question 3**

Try solving the problems given at the beginning of the document.

**Question 4**

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

**Question 5**

Use different norms in the adaptive step-size algorithm :

- the Euclidean norm
- the weighted 2-norm defined by

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

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

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

**Question 6**

- 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