Ordinary differential equations

Numerical Analysis

Proffs. Gianluigi Rozza-Luca Heltai

2018-SISSA mathLab Trieste

Examples and motivations

Example 1. (Biology) Consider a population y of bacteria in a confined environment in which no more than B elements can coexist. Assume that, at the initial time, the number of individuals is equal to $y_0 \ll B$ and the growth rate of the bacteria is a positive constant C. In this case the rate of change of the population is proportional to the number of existing bacteria, under the restriction that the total number cannot exceed B. This is expressed by the differential equation

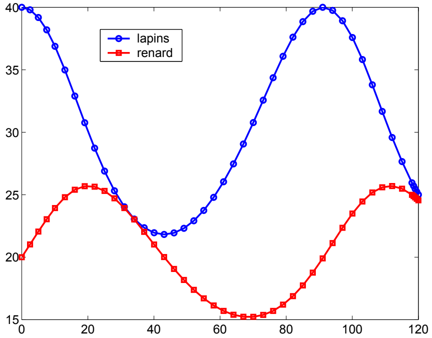
$$y'(t) = Cy(t)\left(1 - \frac{y(t)}{B}\right), \; t > 0, \quad y(0) = y_0. \tag{1}$$

The resolution of this equation allows find the evolution of population over time.

Consider two populations, y_1 and y_2, where y_1 are the prey and y_2 are the predators. The evolution of the two populations is described by the simultaneous differential equations

$$\begin{cases} y_1'(t) = C_1 y_1(t)\left[1 - b_1 y_1(t) - d_2 y_2(t)\right], \\ y_2'(t) = -C_2 y_2(t)\left[1 - b_2 y_2(t) - d_1 y_1(t)\right], \end{cases} \tag{2}$$

where C_1 and C_2 represent the growth rates of the two populations. The coefficients d_1 and d_2 govern the type of interaction between the two populations, while b_1 and b_2 are related to the available quantity of nutrients. The above equation are called the Lotka-Volterra equations.



Introduction

Consider a continuous function $f : \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}$. For given $y_0 \in \mathbb{R}$, we search $y : t \in I \subset \mathbb{R}_+ \to y(t) \in \mathbb{R}$ that satisfies the following problem, called the *Cauchy problem*:

$$\begin{cases} y'(t) = f(t, y(t)) & \forall t \in I \\ y(t_0) = y_0 \end{cases} \tag{3}$$

where $y'(t) = \dfrac{dy(t)}{dt}$.

Examples

- A Cauchy problem can be linear, such as:

$$\begin{cases} y'(t) = 3y(t) - 3t & \text{if } t > 0 \\ y(0) = 1 \end{cases} \tag{4}$$

with $f(t,v) = 3v - 3t$. The solution is $y(t) = (1 - 1/3)e^{3t} + t + 1/3$.

- We have also nonlinear problems, such as

$$\begin{cases} y'(t) = \sqrt[3]{y(t)} & \text{if } t > 0 \\ y(0) = 0 \end{cases} \tag{5}$$

with $f(t,v) = \sqrt[3]{v}$. This problem has got three following solutions : $y(t) = 0$, $y(t) = \sqrt{8t^3/27}$, $y(t) = -\sqrt{8t^3/27}$.

- For the following problem:

$$\begin{cases} y'(t) = 1 + y^2(t) & \text{if } t > 0 \\ y(0) = 0 \end{cases} \tag{6}$$

a solution is a function $y(t) = \tan(t)$ where $0 < t < \frac{\pi}{2}$, i.e. a local solution.

Theorem 1 (Cauchy-Lipschitz, proposition 7.1 in the book)**.** *If a function $f(t, y)$ is*

1. *continuous with respect to both its arguments;*
2. *Uniformly Lipschitz-continuous with respect to its second argument, that is, there exists a positive constant L (named Lipschitz constant) such that*

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2| \quad \forall y_1, y_2 \in \mathbb{R},\ \forall t \in I, \qquad (7)$$

Then the solution $y = y(t)$ of the Cauchy problem (3) exists, is unique and belongs to $C^1(I)$.

Example 2. Consider a problem (4) and we check it exists a unique global solution.

In this case $f(t, v) = 3v - 3t$ and we have:

$$|f(t, y_1) - f(t, y_2)| = |3y_1 - 3t - (3y_2 - 3t)| = |3y_1 - 3y_2| \leq 3|y_1 - y_2|$$

so

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2| \quad \forall y_1, y_2 \in \mathbb{R}, \ \forall t > 0, \quad \text{where} \quad L = 3.$$

So f satisfies the assumptions of Theorem 1 and we can say that the problem (4) has got a unique global solution.

Numerical differentiation

Let $y : [a, b] \to \mathbb{R}$ be C^1 and $t_n \in [a, b]$. The derivative $y'(t_n)$ is given by

$$\begin{aligned} y'(t_n) &= \lim_{h \to 0^+} \frac{y(t_n + h) - y(t_n)}{h}, \\ &= \lim_{h \to 0^+} \frac{y(t_n) - y(t_n - h)}{h}, \\ &= \lim_{h \to 0} \frac{y(t_n + h) - y(t_n - h)}{2h}. \end{aligned}$$

Let $t_0, t_1, \dots, t_{N_h}$, $N_h + 1$ be equidistributed nodes at $[t_0, t_{N_h}]$. Let $h = (t_{N_h} - t_0)/N_h$ be the distance between two consecutive nodes. Let $(Dy)_n$ be an approximation of $y'(t_n)$. We say

- **Forward finite difference** if

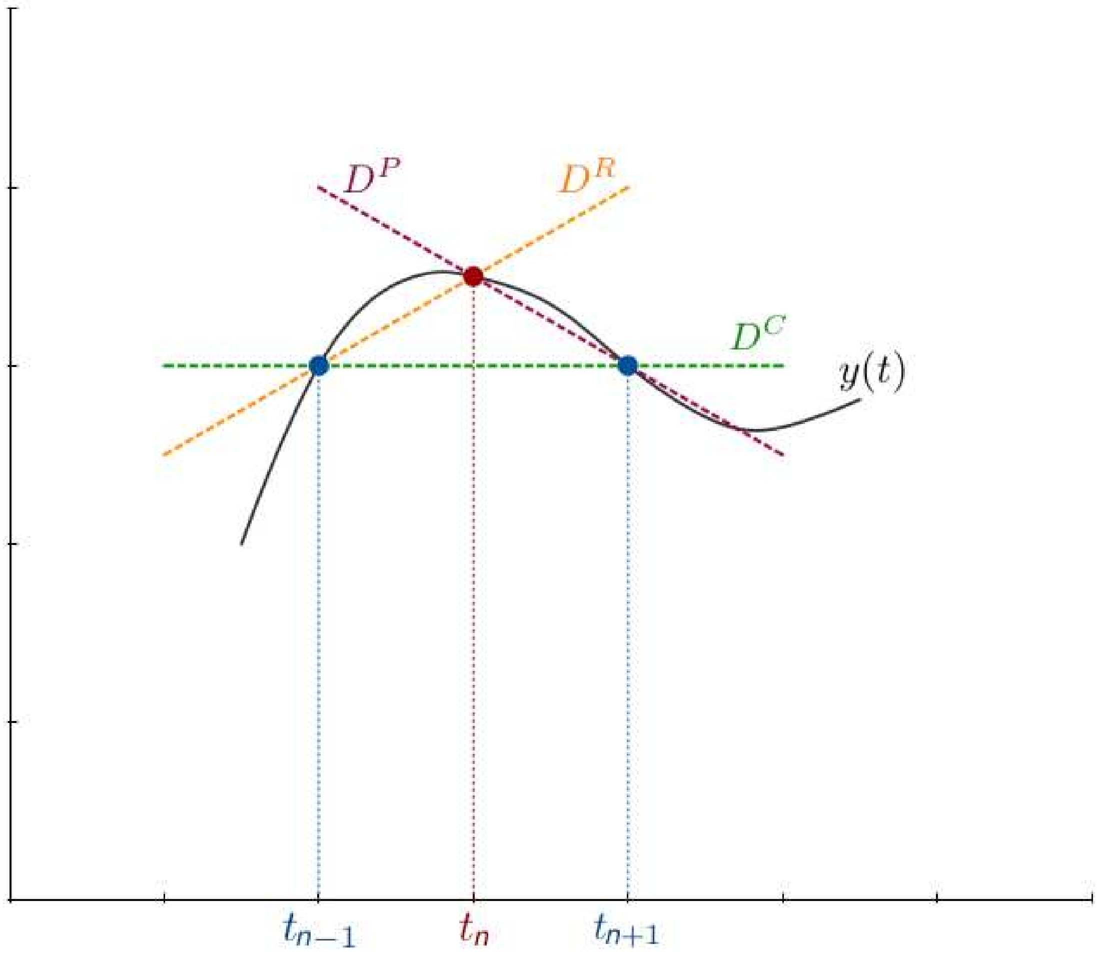
$$(Dy)_n^P = \frac{y(t_{n+1}) - y(t_n)}{h}, \qquad n = 0, \dots, N_h - 1 \qquad (8)$$

- **Backward finite difference** if

$$(Dy)_n^R = \frac{y(t_n) - y(t_{n-1})}{h}, \qquad n = 1, \dots, N_h \qquad (9)$$

- **Centered finite difference** if

$$(Dy)_n^C = \frac{y(t_{n+1}) - y(t_{n-1})}{2h}, \qquad n = 1, \dots, N_h - 1 \qquad (10)$$


D^P
D^R
D^C
$y(t)$
t_{n-1}
t_n
t_{n+1}

The error in the finite difference

If $y \in C^2(\mathbb{R})$ for all $t \in \mathbb{R}$, then there exists ξ_n between t_n and t such that (using the Taylor expansion)

$$y(t) = y(t_n) + y'(t_n)(t - t_n) + \frac{y''(\xi_n)}{2}(t - t_n)^2. \tag{11}$$

- For $t = t_{n+1}$ in (11), we obtain

$$y(t_{n+1}) - y(t_n) = y'(t_n)h + \frac{y''(\xi_n)}{2}h^2,$$

so the forward finite difference is given by

$$(Dy)_n^P = \frac{y(t_{n+1}) - y(t_n)}{h} = y'(t_n) + \frac{h}{2}y''(\xi_n).$$

In particular,

$$|y'(t_n) - (Dy)_n^P| \le Ch, \quad \text{where} \quad C = \frac{1}{2}\max_{t\in[t_n,t_{n+1}]}|y''(t)|.$$

- For $t = t_{n-1}$ in (11), we obtain

$$y(t_{n-1}) - y(t_n) = y'(t_n)(-h) + \frac{y''(\xi_n)}{2}(-h)^2,$$

so the backward finite difference is given by

$$(Dy)_n^R = \frac{y(t_n) - y(t_{n-1})}{h} = y'(t_n) - \frac{h}{2}y''(\xi_n).$$

In particular,

$$|y'(t_n) - (Dy)_n^R| \leq Ch,$$

where $C = \frac{1}{2}\max_{t\in[t_{n-1},t_n]}|y''(t)|$.

- For $t = t_{n+1}$ and $t = t_{n-1}$ with expansion of order 2 (if $y \in C^3(\mathbb{R})$)

$$y(t_{n+1}) = y(t_n) + y'(t_n)\,h + \frac{y''(t_n)}{2}\,h^2 + \frac{y'''(\xi_{n_1})}{6}\,h^3,$$
$$y(t_{n-1}) = y(t_n) - y'(t_n)\,h + \frac{y''(t_n)}{2}\,h^2 - \frac{y'''(\xi_{n_2})}{6}\,h^3,$$

we obtain

$$y(t_{n+1}) - y(t_{n-1}) = 2y'(t_n)h + \frac{y'''(\xi_{n_1}) + y'''(\xi_{n_2})}{6}\,h^3,$$

and

$$(Dy)_n^C = \frac{y(t_{n+1}) - y(t_{n-1})}{2h} = y'(t_n) + \frac{y'''(\xi_{n_1}) + y'''(\xi_{n_2})}{12}\,h^2.$$

It has the following estimation

$$|y'(t_n) - (Dy)_n^C| \leq Ch^2,$$

where $C = \frac{1}{6}\max_{t\in[t_{n-1},t_{n+1}]}|y'''(t)|$.

Definition 1. *The difference* $\tau_n(h) = |y'(t_n) - (Dy)_n^P|$ *is called truncation error in the point* t_n. *We say that* τ_n *is of order* $p > 0$ *if*

$$\tau_n(h) \leq Ch^p,$$

for a positive constant C.

Thanks to the found estimation, the truncation error of the forward and the backward finite difference is of order 1; the truncation error of centered finite difference is of order 2.

The finite difference method

FOR APPROXIMATION THE CAUCHY PROBLEM (Chapt. 7.3 in the book)

Let $0 = t_0 < t_1 < \ldots < t_n < t_{n+1} < \ldots$ be an equidistributed sequence of real numbers and $h = t_{n+1} - t_n$ be the time step. We denote by

$$u_n \quad \text{an approximation of} \quad y(t_n).$$

In the Cauchy problem (3), for $t = t_n$, we have

$$y'(t_n) = f(t_n, y(t_n)).$$

We want to approximate the derivative $y'(t_n)$ in the point t_n. We can use a **finite difference differentiation**.

Forward Euler

$$\begin{cases} \dfrac{u_{n+1} - u_n}{h} = f(t_n, u_n) & \text{for } n = 0, 1, 2 \ldots, N_h - 1 \\ u_0 = y_0 \end{cases} \tag{12}$$

Backward Euler

$$\begin{cases} \dfrac{u_{n+1} - u_n}{h} = f(t_{n+1}, u_{n+1}) & \text{for } n = 0, 1, 2 \ldots, N_h - 1 \\ u_0 = y_0 \end{cases} \tag{13}$$

Centered scheme

$$\begin{cases} \dfrac{u_{n+1} - u_{n-1}}{2h} = f(t_n, u_n) & \text{for } n = 1, 2 \ldots, N_h - 1 \\ u_0 = y_0 \\ u_1 \qquad \text{to determine} \end{cases} \tag{14}$$

Remark 1.

- *The forward Euler is eplicit because u_{n+1} depends on u_n explicitly:*

$$\text{(forwardEuler)} \qquad u_{n+1} = u_n + hf(t_n, u_n).$$

- *The backward Euler is implicit because u_{n+1} is implicitly defined in terms of u_n:*

$$\text{(backwardEuler)} \qquad u_{n+1} = u_n + hf(t_{n+1}, u_{n+1}).$$

In general, for the backward Euler, we have to solve a nonlinear equation at each time step.

Fixed point iterations: Note that (backward Euler) is equivalent to a fixed point problem with

$$u_{n+1} = \phi(u_{n+1}) = u_n + hf(t_{n+1}, u_{n+1}) \tag{15}$$

We can solve this problem thanks to the following iterations

$$u_{n+1}^{k+1} = \phi(u_{n+1}^k), \quad k = 0, 1, 2, \ldots \tag{16}$$

The Newton method: Starting from the equation:

$$F(u_{n+1}) \equiv u_{n+1} - \phi(u_{n+1}) = 0, \tag{17}$$

we use the following iterations:

$$u_{n+1}^{k+1} = u_{n+1}^k - \frac{F(u_{n+1}^k)}{F'(u_{n+1}^k)} = u_{n+1}^k - \frac{F(u_{n+1}^k)}{1 - \phi'(u_{n+1}^k)}, \quad k = 0, 1, 2, \ldots \tag{18}$$

In both cases, we have $\lim_{k\to\infty} u_{n+1}^k = u_{n+1}$.

Example 3. Consider the following differential equation

$$\begin{cases} y'(t) = -ty^2(t), \quad t > 0 \\ y(0) = 2. \end{cases} \tag{19}$$

We want to solve this equation using forward Euler and backward Euler methods, at interval $[0, 4]$ with $N_h = 20$ subintervals (it is equivalent to a time step $h = 0.2$). We approximate the exact solution $y(t_n)$ at times $t_n = nh$, $n = 0, 1, \dots 20$ (therefore $t_n = 0.2, 0.4, 0.6, \dots$) by a numerical solution u_n.

In Matlab/Octave, the forward Euler method can be used by:

```
>> h = 0.2;                                  % the time step
>> u(1) = 2;                                 % the initial value
>> t = [0:h:4];                              % vector of time t(n)
>> for n=1:20;                               % loop ‘‘for’’
     u(n+1) = u(n) + h * ( -t(n) * u(n)^2 );
   end;
>> plot(t,u);                                % we draw the graph
```

We can also use the functions `feuler` and `beuler`:

- *Forward Euler*

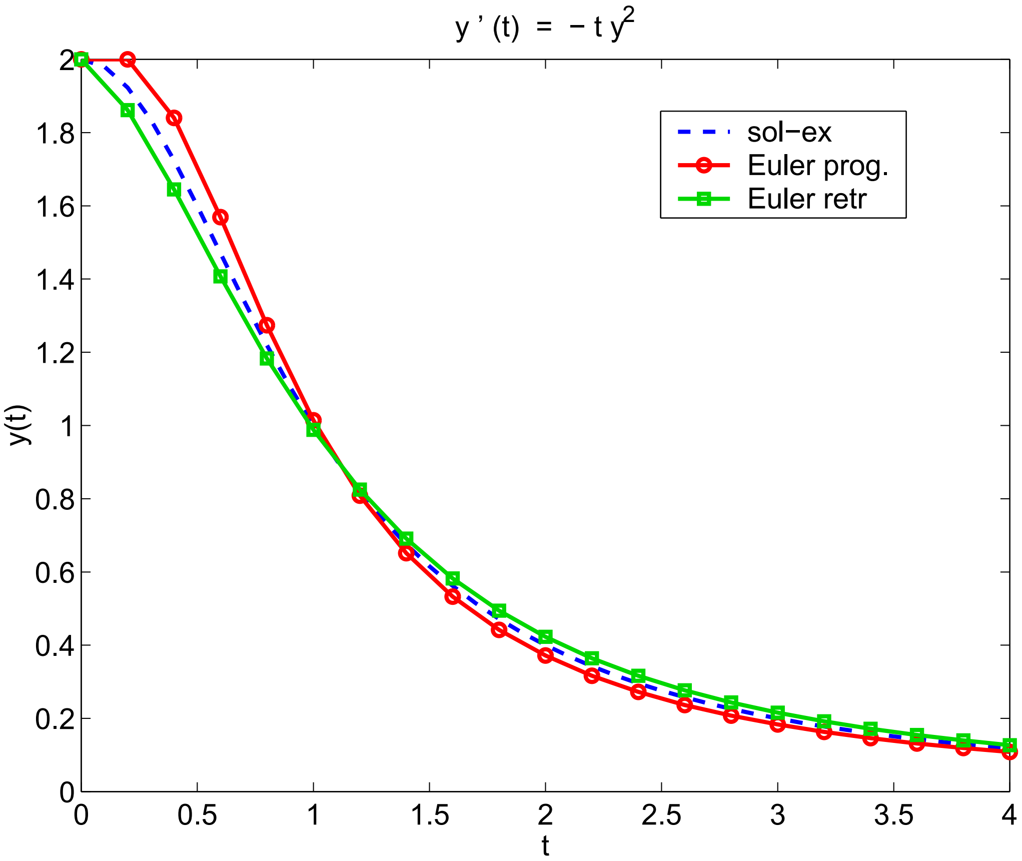
  ```
  >> f = @(t,y) -t.*y.^2;
  >> Nh = 20; tspan = [0 4]; y0 = 2;
  >> [t_EP, y_EP] = feuler(f, tspan, y0, Nh);
  ```

  Output variables `t_EP` and `y_EP` contain sequences of the times t_n and the values u_n respectively.

- *Backward Euler*
  The function `beuler` uses the same syntax:

  ```
  >> [t_ER, y_ER] = beuler(f, tspan, y0, Nh);
  ```

Comparison between the exact solution and those obtained by forward and backward Euler methods.



Stability conditions

The choice of time step h is not arbitrary. For forward Euler, we will see later that if h is not small enough then stability problems may arise.

For example, if we consider the problem

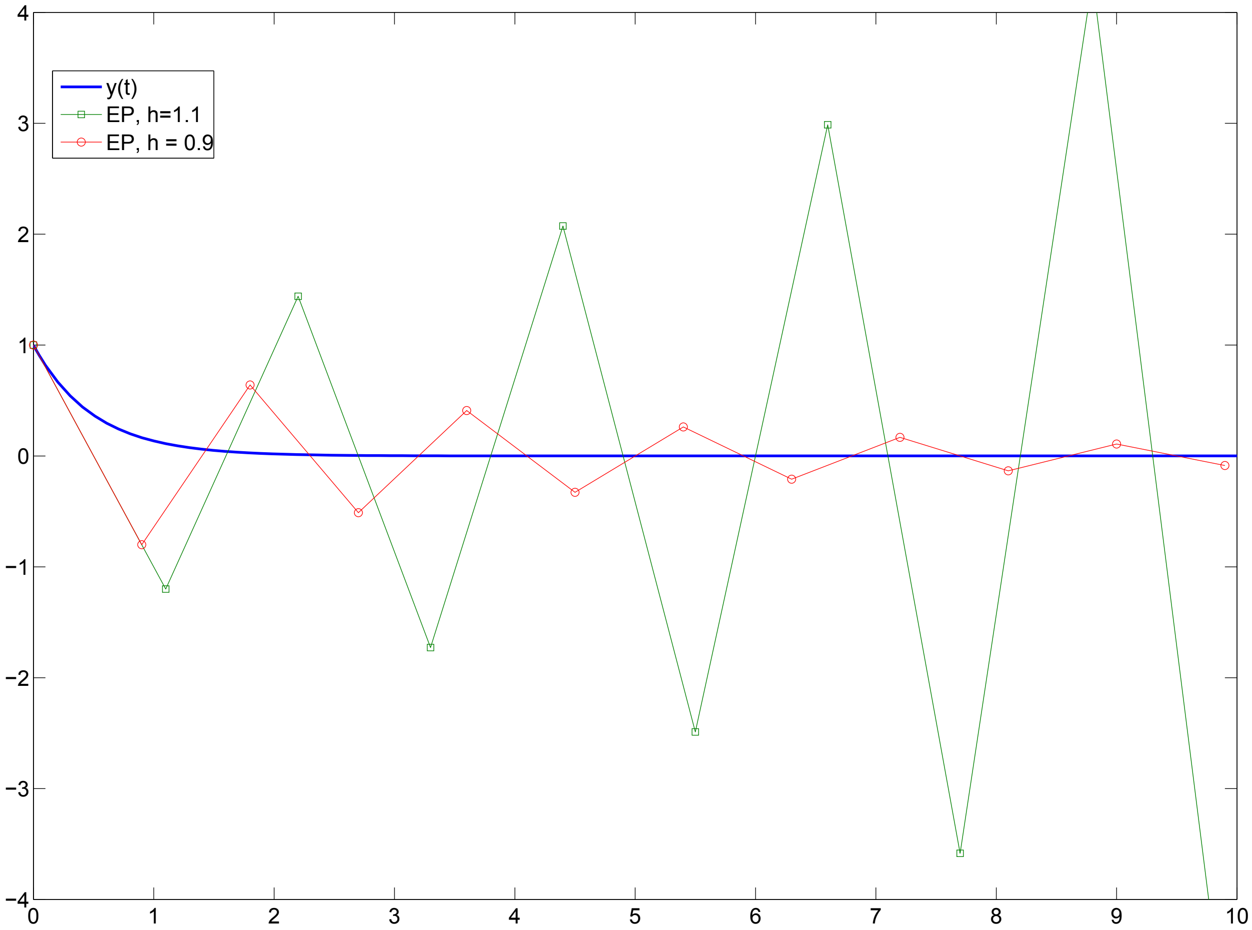
$$\begin{cases} y'(t) = -2y(t) & \text{for } t \in \mathbb{R}_+ \\ y(0) = 1, \end{cases} \tag{20}$$

then the solution is

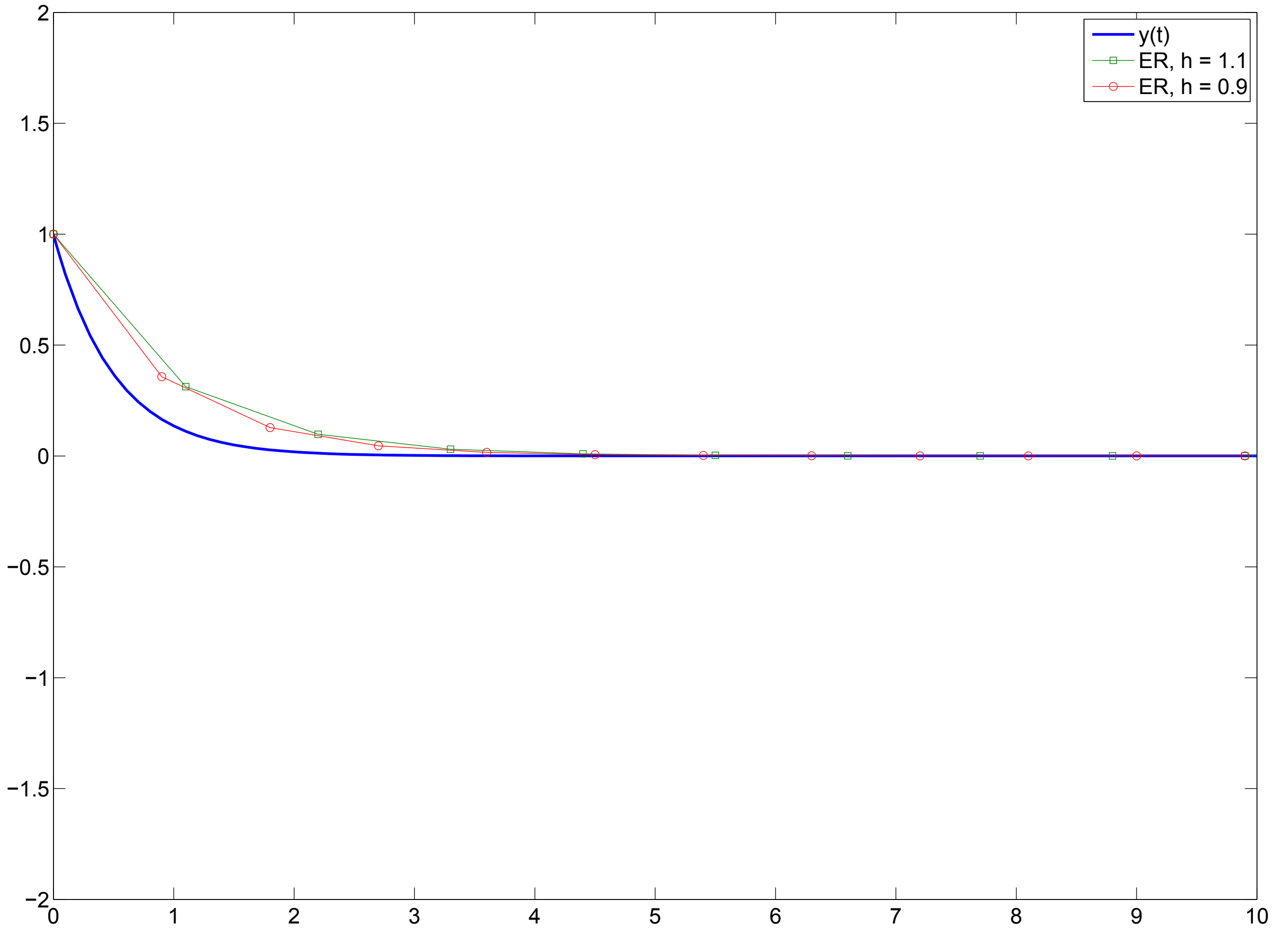
$$y(t) = e^{-2t},$$

We can observe that behavior with respect to h of forward and backwar Euler methods are very different.

Stability conditions (forward Euler)



Stability conditions (backward Euler)



The (absolute) stability properties

(Chapt. 7.6 in the book)

For given $\lambda < 0$, we consider the model problem:

$$\begin{cases} y'(t) = \lambda y(t) & \text{for } t \in \mathbb{R}_+ \\ y(0) = 1 \end{cases} \tag{21}$$

The solution is

$$y(t) = e^{\lambda t}. \qquad \text{In particular,} \quad \lim_{t\to\infty} y(t) = 0.$$

Let $0 = t_0 < t_1 < \ldots < t_n < t_{n+1} < \ldots$ such that $t_n = nh$ and where the *time step* $h > 0$ is fixed.

We say that a numerical scheme associated to the model problem is absolutely stable if $\lim_{n\to\infty} u_n = 0$.

- For the **forward Euler**:

$$u_{n+1} = (1 + \lambda h) u_n, \qquad \text{where} \quad u_n = (1 + \lambda h)^n, \quad \forall n \geq 0. \qquad (22)$$

If $1 + \lambda h < -1$, then $|u_n| \to \infty$ when $n \to \infty$, therefore forward Euler is *unstable*.
To ensure stability, we need to *limit the time step* h, by imposing the **stability condition** :

$$\boxed{|1 + \lambda h| < 1 \text{ hence } h < 2/|\lambda|.}$$

- For the **backward Euler**:

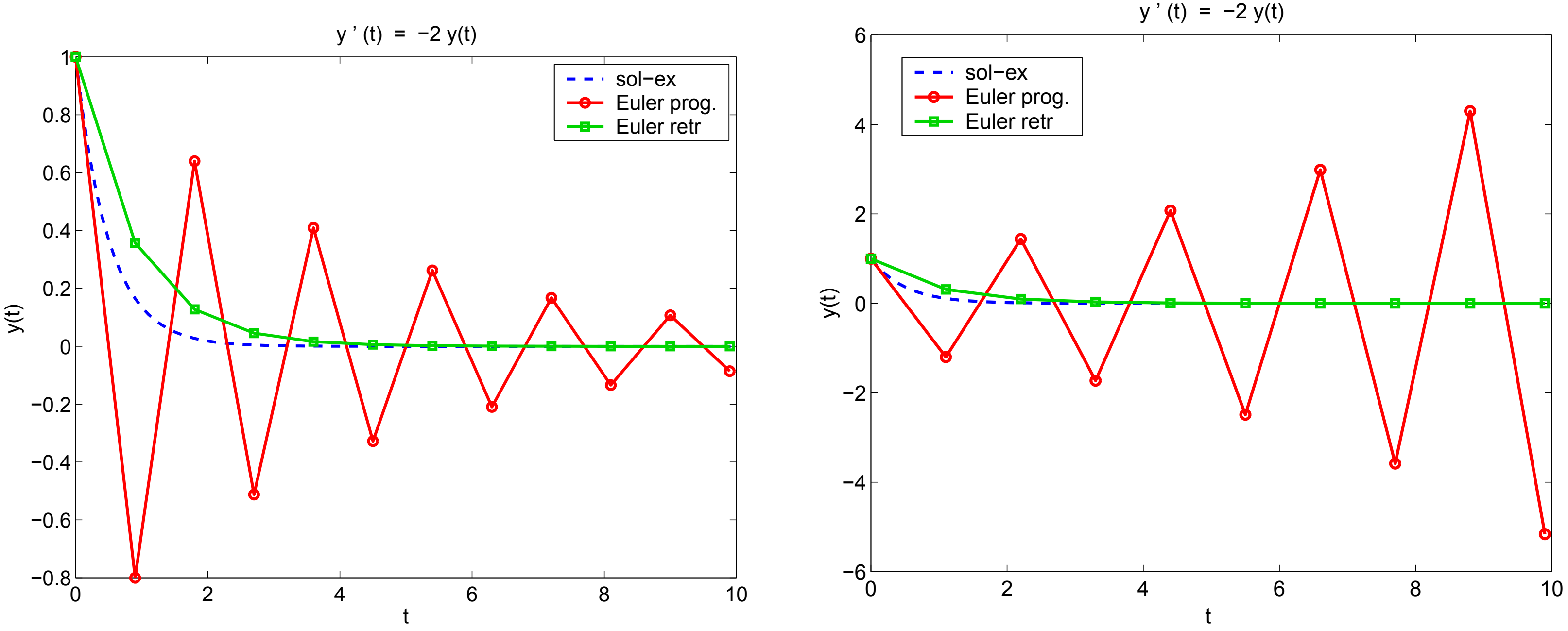
$$u_{n+1} = \left(\frac{1}{1 - \lambda h} \right) u_n \qquad \text{and therefore} \qquad u_n = \left(\frac{1}{1 - \lambda h} \right)^n, \quad \forall n \geq 0.$$

Because $\lim_{n \to \infty} u_n = 0$, it is unconditionally stable (it is stable for any $h > 0$).

Example 4. Let solve the problem (21) for $\lambda = -2$ and $y_0 = 1$ at interval $[0, 10]$ using forward and backward Euler methods with $h = 0.9$ and $h = 1.1$. Here are the Matlab/Octave commands for the case $h = 0.9$. Note that, even if $f(t, y)$ does not depend on t, it must be defined in Matlab/Octave as a function of (t, y).

```
>> f = @(t,x) -2*x; h=0.9; tspan=[0 10]; Nh = 10/h; y0=1;
>> [t_ep, y_ep] = feuler(f, tspan, y0, Nh);
>> [t_er, y_er] = beuler(f, tspan, y0, Nh);
>> t = linspace(0, 10, 11); sol_ex = @(t) exp(-2*t);
>> plot(t, sol_ex(t), ’b’, t_ep, y_ep, ’ro-’, t_er, y_er’, ’go-’)
```

The following figure shows obtained solutions for $h = 0.9$ (on the left) and $h = 1.1$ (on the right) and the exact solution.



Comparison of solutions that we obtain by the forward and backward Euler methods for $h = 0.9$ (on the left, stable) and $h = 1.1$ (on the right, unstable) (stability condition for forward Euler: $|\lambda| = 2 \Rightarrow h < 2/|\lambda| = 1$).

Absolute stability controls perturbations

(Chapt. 7.6.2 in the book)

For a generic problem, it raises the question of stability, i.e. the property that small perturbations on the data induce small perturbations on the approximation.

We want to show the following property.

A numerical method which is absolutely stable on the model problem, guarantees that the perturbations are kept under control as t tends to infinity (is stable in the above sense).

Consider now the following *generalized model problem*:

$$\begin{cases} y'(t) = \lambda(t)y(t) + r(t), \quad t \in (0, +\infty), \\ \\ y(0) = 1, \end{cases} \tag{23}$$

where λ and r are two continuous functions and $-\lambda_{max} \leq \lambda(t) \leq -\lambda_{min}$ with $0 < \lambda_{min} \leq \lambda_{max} < +\infty$. In this case the exact solution does not necessarily tend to zero as t tends to infinity.
For instance if both r and λ are constants we have

$$y(t) = \left(1 + \frac{r}{\lambda}\right) e^{\lambda t} - \frac{r}{\lambda}$$

whose limit when t tends to infinity is $-r/\lambda$. Thus, in general, it does not make sense to require a numerical method to be absolutely stable, i.e. to satisfy (23).

For the sake of simplicity we will confine our analysis to the forward Euler method (23).

We have

$$\begin{cases} u_{n+1} = u_n + h(\lambda_n u_n + r_n), \quad n \geq 0, \\ u_0 = 1 \end{cases}$$

where $\lambda_n = \lambda(t_n)$ and $r_n = r(t_n)$.
Let us consider the following “perturbed” method:

$$\begin{cases} z_{n+1} = z_n + h(\lambda_n z_n + r_n + \rho_{n+1}), \quad n \geq 0, \\ z_0 = u_0 + \rho_0, \end{cases} \tag{24}$$

where $\rho_0, \rho_1, \ldots$ are given perturbations which are introduced at every time step.

This is a simple model in which ρ_0 and ρ_{n+1} represent truncation errors or numerical errors.

Question: Is the difference $z_n - u_n$ bounded for all $n = 0, 1, ...$ independently of n and h?

We will consider two cases:

(i) Let $\lambda_n = \lambda$ and $\rho_n = \rho$ be two constants. We can write the schema for the error $e_n = z_n - u_n$

$$\begin{cases} e_{n+1} = e_n + h(\lambda e_n + \rho), \quad n \geq 0, \\ e_0 = \rho. \end{cases} \tag{25}$$

that the solution is

$$e_n = \rho(1 + h\lambda)^n + h\rho \sum_{k=0}^{n-1} (1 + h\lambda)^k = \rho\psi(h, \lambda), \tag{26}$$

where

$$\psi(h, \lambda) = \left((1 + h\lambda)^n (1 + \frac{1}{\lambda}) - \frac{1}{\lambda} \right)$$

We use equation for the geometric sum

$$\sum_{k=0}^{n-1} a^k = \frac{1 - a^n}{1 - a}. \tag{27}$$

Suppose that $h < h_0(\lambda) = 2/|\lambda|$, *i.e. h ensure the absolute stability of the forward Euler method applied to the problem (21)*.
Therefore $(1 + h\lambda)^n < 1 \; \forall n$ and it follows that the error due to perturbations is bounded by

$$|e_n| \leq \varphi(\lambda)|\rho|, \tag{28}$$

where $\varphi(\lambda) = 1 + |2/\lambda|$. Moreover,

$$\lim_{n\to\infty} |e_n| = \frac{|\rho|}{|\lambda|}.$$

So, the error of perturbations is bounded by $|\rho|$ times a constant that is independent of n and h. Obviously, if $h > h_0$, the perturbations amplifies when n increases because $(1 + h\lambda)^n \to \infty$ when $n \to \infty$.

(ii) In the general case where λ and r depends on t, we have

$$z_n - u_n = \rho_0 \prod_{k=0}^{n-1}(1 + h\lambda_k) + h \sum_{k=0}^{n-1} \rho_{k+1} \prod_{j=k+1}^{n-1}(1 + h\lambda_j) \qquad (29)$$

We require the time step h to satisfy the restriction $h < h_0(\lambda)$, where $h_0(\lambda) = 2/\lambda_{max}$. Then, $|1 + h\lambda_k| \leq \max(|1 - h\lambda_{min}|, |1 - h\lambda_{max}|) < 1$. Let $\rho = \max|\rho_n|$ and λ such that $(1 + h\lambda) = \max(|1 - h\lambda_{min}|, |1 - h\lambda_{max}|)$.

It holds:

$$|z_n - u_n| \leq |\rho_0| \prod_{k=0}^{n-1} |1 + h\lambda_k| + h \sum_{k=0}^{n-1} |\rho_{k+1}| \prod_{j=k+1}^{n-1} |1 + h\lambda_j|$$

$$\leq \rho \prod_{k=0}^{n-1}(1 + h\lambda) + h \sum_{k=0}^{n-1} \rho \prod_{j=k+1}^{n-1}(1 + h\lambda) = \rho\psi(h, \lambda)$$

So, even in this case, $e_n = z_n - u_n$ satisfies (28).

Remark 2. *Consider now the following generalized model problem*

$$\begin{cases} y'(t) = f(t, y(t)) \quad t > 0 \\ y(0) = y_0 \ , \end{cases}$$

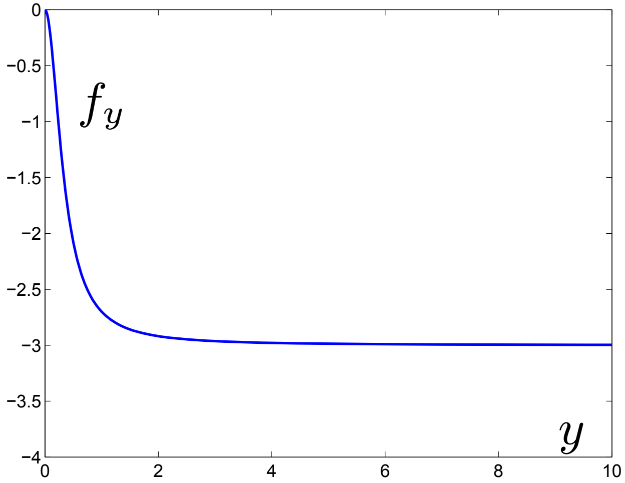
at unbounded interval. We can extend the control of perturbations to generalized model problem (23), in cases where exists $\lambda_{\min} > 0$ *and* $\lambda_{\max} < \infty$ *such that*

$$-\lambda_{max} < \partial f / \partial y(t, y) < -\lambda_{min}, \forall t \geq 0, \ \forall y \in D_y, \tag{30}$$

This allows to get (29) and to obtain the same conclusions as in (ii) if $0 < h < 2/\lambda_{max}$.
Let D_y *be a set that contains the trajectory of* $y(t)$ *(possible values of* u_n*).*

Example 5. Let us consider the Cauchy problem

$$\begin{cases} y'(t) = \arctan(3y) - 3y + t, \quad t \in (0, +\infty), \\ y(0) = 1. \end{cases}$$



For $y = 0$, we have $y'(t) > 0$, $t \in (0, +\infty)$. We deduce that if we draw the graph of the function $y(t)$, it can never go below the axis t and therefore $y(t) > 0$, $\forall t \in (0, +\infty)$. We put $D_y = (0, \infty)$ and we calculate $f_y = \partial f / \partial y = 3/(1 + 9y^2) - 3$. You can choose

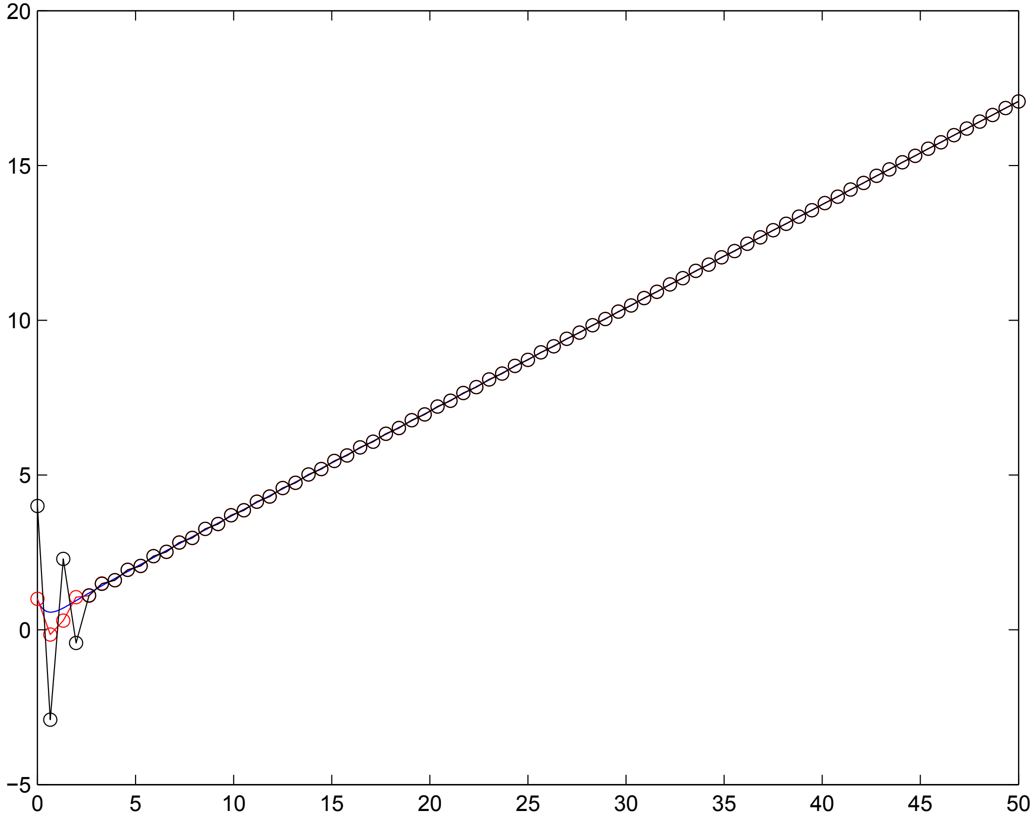
$$\lambda_{\max} = 3 \, (< \infty) \text{ et } \lambda_{\min} = \lambda^* > 0.$$

So,

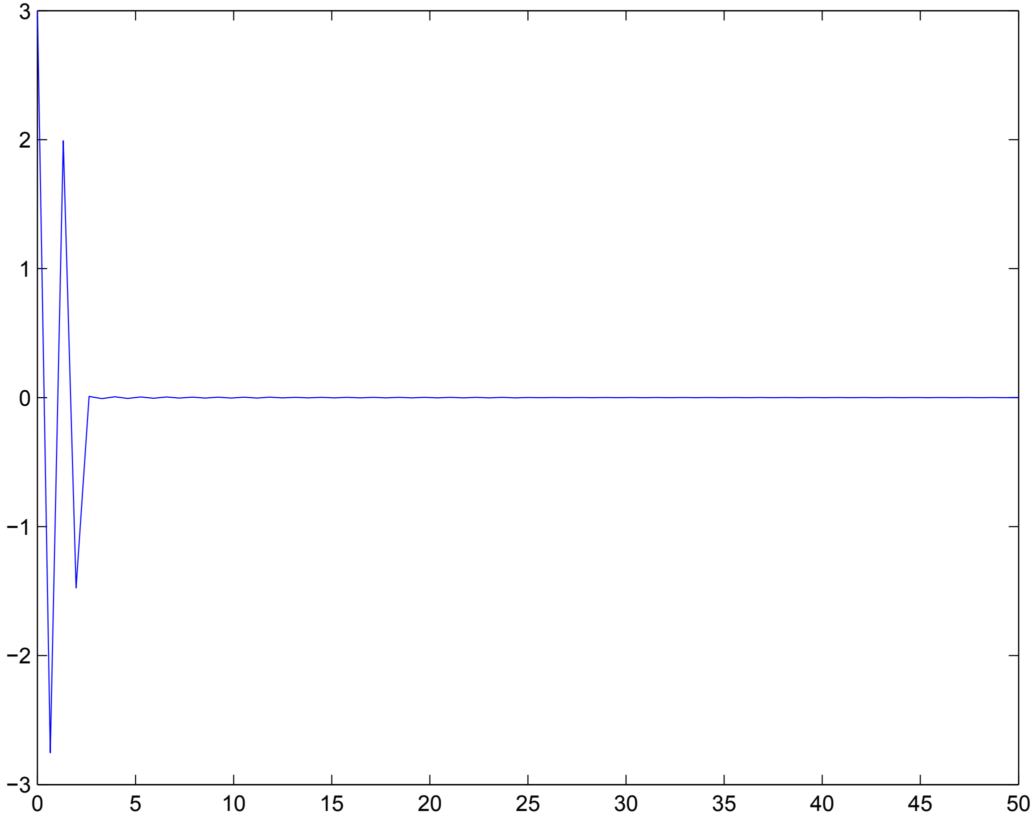
$$-\lambda_{max} < \partial f / \partial y(t, y) < -\lambda_{min}, \forall t \geq 0, \;\; \forall y \in D_y = (0, \infty),$$

and the forward Euler method is stable if $h < 2/\lambda_{\max} = 2/3$.

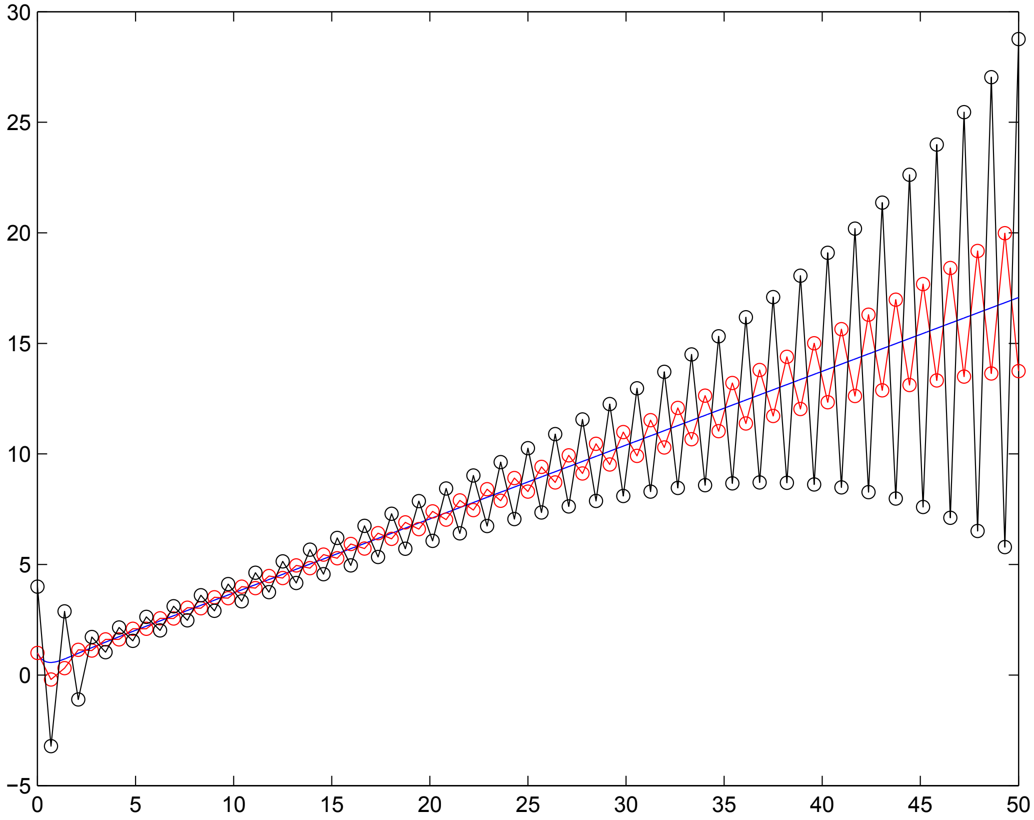
Solution $y(t)$ of the problem in the example 5; numerical solution u_n (red) using the forward Euler ($h = 2/3 - 0.01$, stable) and solution z_n (black) with perturbation $\rho_0 = 3$ (only at the initial value).



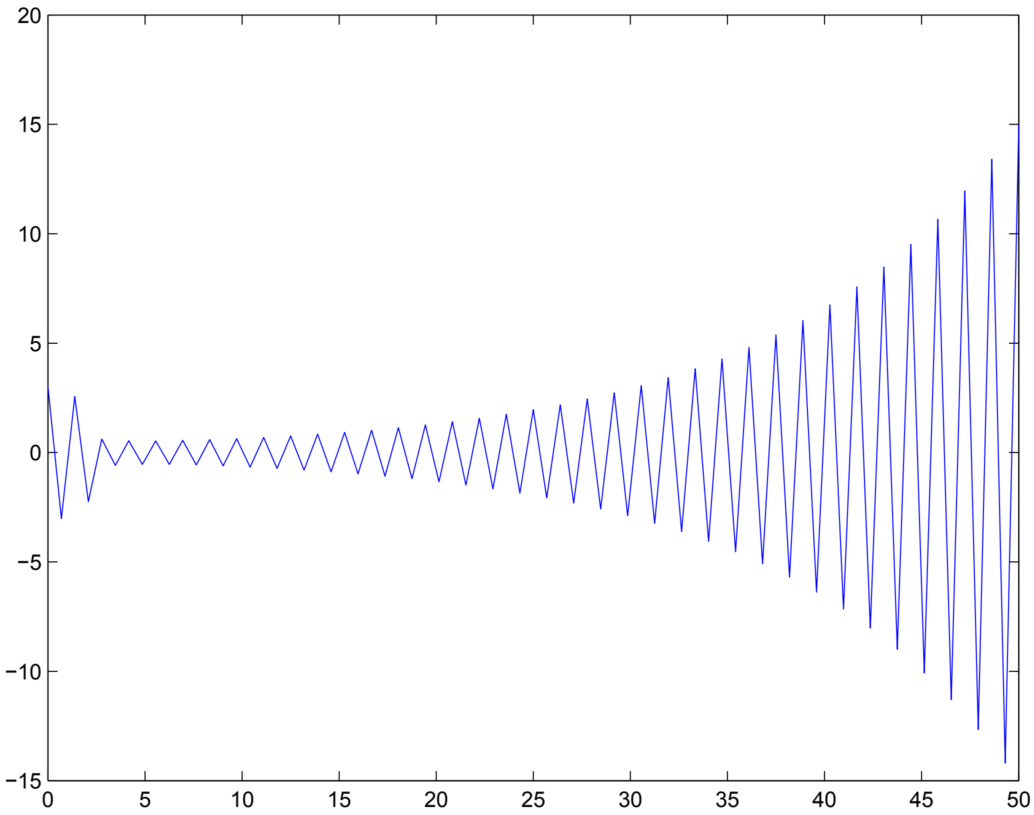
Error $e_n = z_n - y_n$ between the perturbed numerical solution and non-perturbed numerical solution ($h = 2/3 - 0.01$, stable).



Solution $y(t)$ of the problem in the example 5; numerical solution u_n (red) using the forward Euler ($h = 2/3 + 0.02$, unstable) and solution z_n (black) with perturbation $\rho_0 = 3$ (only at the initial value).



Error $e_n = z_n - y_n$ between the perturbed numerical solution and non-perturbed numerical solution ($h = 2/3 + 0.02$, unstable).



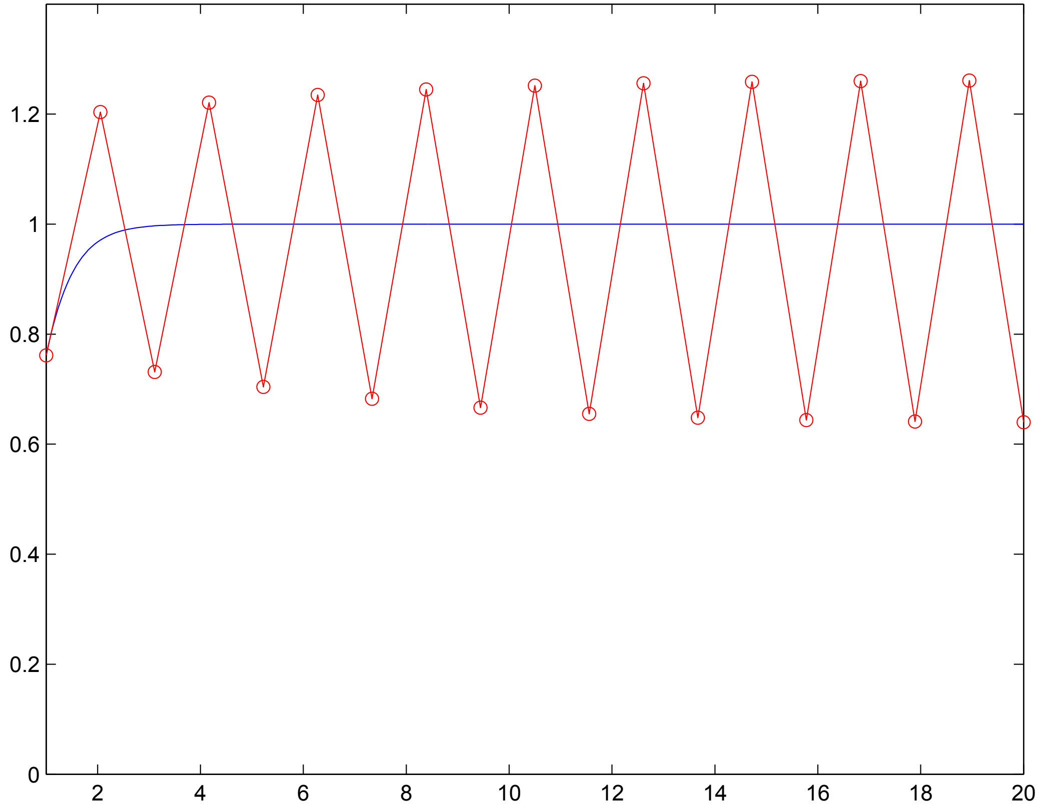
Example 6. We seek an upper bound on h that guarantees stability for the forward Euler method applied to approximate the Cauchy problem:

$$\begin{cases} y' = 1 - y^2, & t > 1, \\ y(1) = (e^2 - 1)/(e^2 + 1). \end{cases} \tag{31}$$

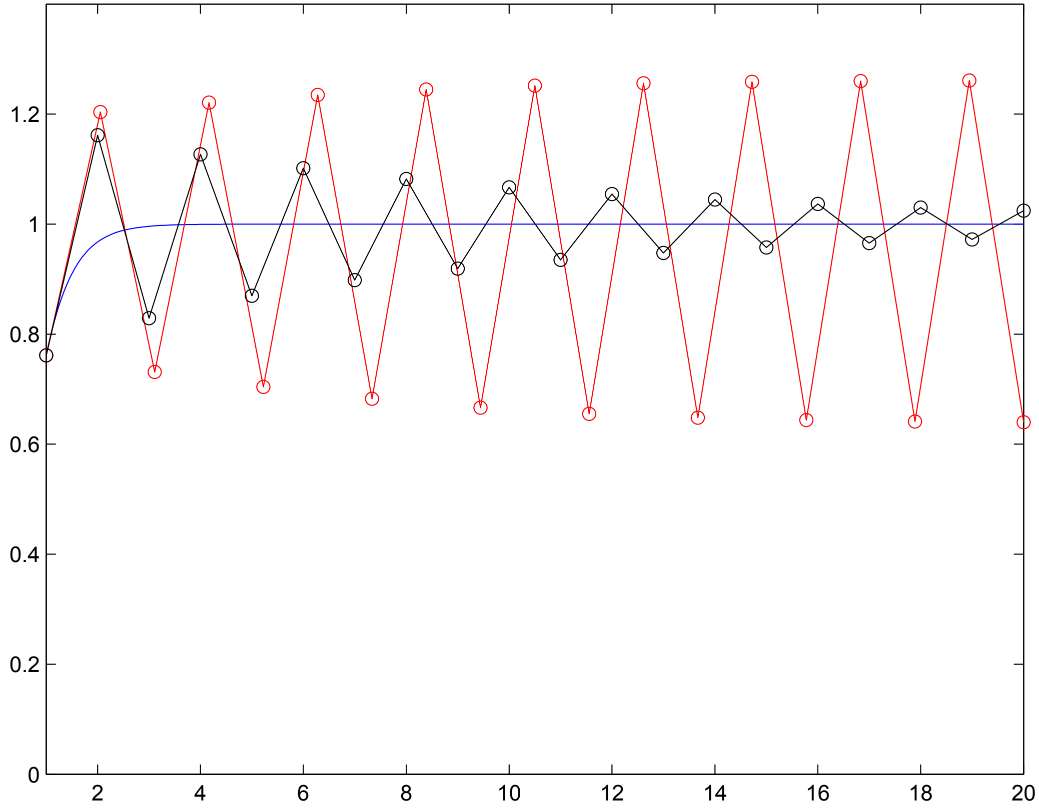
The exact solution is $y(t) = (e^{2t} - 1)/(e^{2t} + 1)$ and $\partial f/\partial y = -2y$. Since $y \in [y(1), 1] = D_y$, $\partial f/\partial y \in (-2, -2y(1))$ for all $t > 1$ we can take $\lambda_{\min} = 2y(1)(> 0)$ and $\lambda_{\max} = 2(< \infty)$.

Thus, for absolute stability of the forward Euler method, h should be smaller than $h_0 = 1$ and it must be checked that $u_n \in [y(1), 1]$.
In the figure below, it represents the exact solution (dotted line) and approximate solutions obtained in the interval $(1, 20)$ with $h = 20/19$ (dashed line) and $h = 20/21$ (solid line). This shows that limitation of h_0 obtained above is accurate.

Numerical solution of the problem (31) obtained using forward Euler method with $h = 20/19$ (red).
The exact solution corresponds to the solid line.



Numerical solutions of the problem (31) obtained using forward Euler method with $h = 20/19$ (red) and $h = 20/21$ (black).
The exact solution corresponds to the solid line.



Convergence of the forward Euler

(Chapt. 7.3.1 in the book)

Definition 2. *Let $y(t)$ be the solution of the Cauchy problem (3) on the interval $[0,T]$; let u_n be an approximated solution at time $t_n = nh$, where $h = T/N_h$ ($N_h \in \mathbb{N}$) is the time step, found by a given numerical method. The method is convergent if*

$$\forall n = 0, \ldots, N_h : \quad |u_n - y(t_n)| \leq C(h)$$

where $C(h) \to 0$ when $h \to 0$.

Moreover, if there exists $p > 0$ such that $C(h) = \mathcal{O}(h^p)$, We say that the method *converges with order* p.

In the following, we will analyze the convergence and the order of the forward Euler method.

Convergence of the forward Euler

We will prove the following convergence result:

Theorem 2. *If $y \in \mathcal{C}^2([0,T])$ and f is uniform Lipschitz continuous on the second variable, L is the Lipschitz constant, then*

$$\forall n \geq 0, \ \ |y(t_n) - u_n| \leq c(t_n)h, \tag{32}$$

where

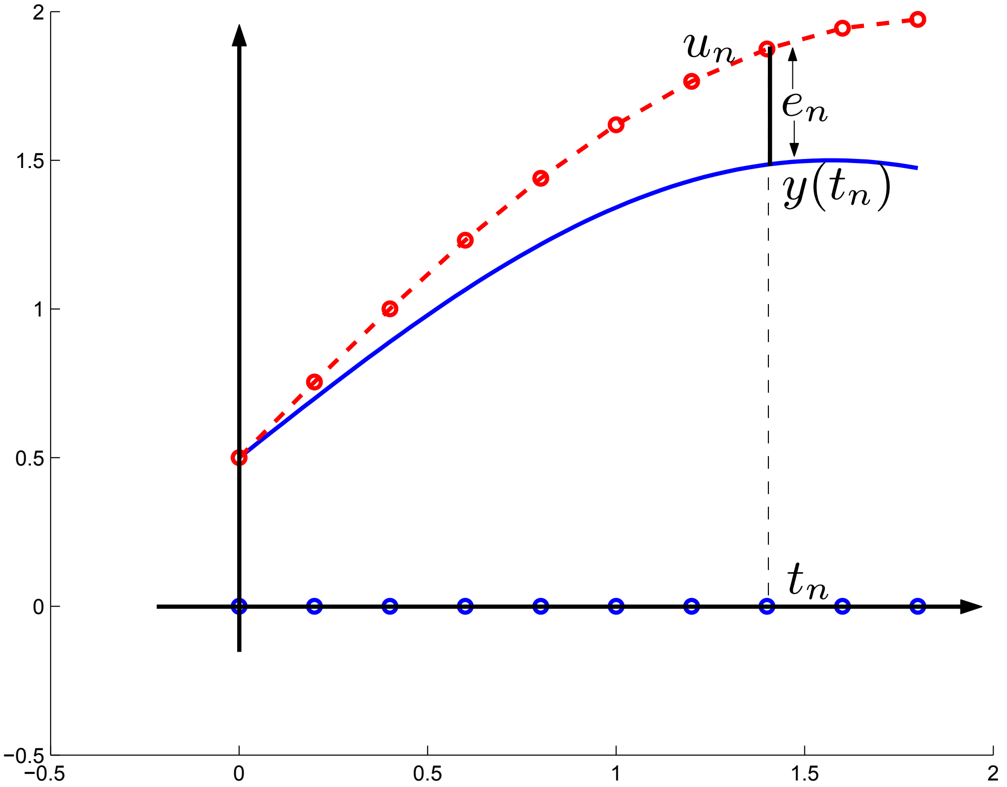
$$c(t_n) = \frac{e^{Lt_n} - 1}{2L} \max_{t\in[0,T]} |y''(t)|.$$

In particular, the method converges with order $p = 1$ according to the previous definition, with

$$C(h) = c(T)h.$$

Proof. We define the error at time n as:

$$e_n = u_n - y(t_n)$$



We define the *local truncation error* of the forward Euler method as

$$\tau_{n+1}(h) = \frac{y(t_{n+1}) - y(t_n)}{h} - y'(t_n), \tag{33}$$

and the *global truncation error*

$$\tau(h) = \max_n |\tau_n(h)|.$$

We know there exists $\xi_n \in (t_n, t_n + h)$ such that

$$\tau_{n+1}(h) = \frac{1}{2} y''(\xi_n) h.$$

So we have the following estimation for the global truncation error:

$$\tau(h) \leq \frac{1}{2} \max_{t \in [0,T]} |y''(t)| h.$$

The following equation for the numerical solution u_n

$$\begin{cases} \dfrac{u_{n+1} - u_n}{h} = f(t_n, u_n) \qquad \text{for } n = 0, 1, 2 \ldots, N_h \\ u_0 = y_0. \end{cases}$$

and the equation (33) for a local truncation error

$$\tau_{n+1}(h) = \frac{y(t_{n+1}) - y(t_n)}{h} - y'(t_n) = \frac{y(t_{n+1}) - y(t_n)}{h} - f(t_n, y(t_n)),$$

we obtain:

$$\begin{cases} \dfrac{e_{n+1} - e_n}{h} = f(t_n, u_n) - f(t_n, y(t_n)) - \tau_{n+1}(h), \\ e_0 = 0. \end{cases} \tag{34}$$

Since the function f is Lipschitz, we have

$$|f(t_n, u_n) - f(t_n, y(t_n))| \leq L|u_n - y(t_n)| \leq L|e_n|.$$

Given this inequality, (34) can be written as:

$$|e_{n+1}| \leq (1 + Lh)|e_n| + h|\tau_{n+1}(h)|.$$

Let $E_j = |e_j|$. Then we have the following inequality:

$$\begin{aligned} E_{n+1} &\leq (1 + hL)E_n + h\tau(h) \\ &\leq (1 + hL)\left[(1 + hL)E_{n-1} + h\tau(h)\right] + h\tau(h) \\ &\leq \left[1 + (1 + hL) + (1 + hL)^2 + \ldots + (1 + hL)^n\right] h\tau(h) \\ &= \frac{(1 + hL)^{n+1} - 1}{hL} h\tau(h) \end{aligned}$$

But

$$1 + hL \le e^{hL},\ \forall h > 0,$$

hence

$$(1 + hL)^n - 1 \le e^{Lhn} - 1 = e^{Lt_n} - 1.$$

Therefore

$$E_n \le \frac{e^{Lt_n} - 1}{L}\tau(h) \le \frac{e^{Lt_n} - 1}{L} h \frac{1}{2} \max_{t\in[0,T]} |y''(t)|,$$

i.e.

$$|u_n - y(t_n)| \le \left[\frac{1}{2}\frac{e^{Lt_n} - 1}{L} \max_{t\in[0,T]} |y''(t)|\right] h \le c(T)h,$$

where $c(T) = \frac{e^{LT}-1}{2L} \max_{t\in[0,T]} |y''(t)|$.

■

Remark 3. *The same type of results can be obtained for the backward Euler method.*

Remark 4. *The estimation of convergence (32) is obtained under the assumption that the function f is Lipschitz continuous. More precisely, the estimation*

$$|u_n - y(t_n)| \le h t_n \frac{1}{2} \max_t |y''(t)| \,, \tag{35}$$

is true if f also satisfies the condition $\frac{\partial f}{\partial y}(t, y) \le 0$ for all $t \in [0, T]$ and for all $y \in (-\infty, \infty)$.

We prove (35). Using the Lagrange theorem, there exists ξ_n such that

$$f(t_n, u_n) - f(t_n, y(t_n)) = \frac{\partial f(t, \xi_n)}{\partial y}(u_n - y(t_n)) = \frac{\partial f(t, \xi)}{\partial y} e_n.$$

So, using (34) we find

$$e_{n+1} = \left(1 + h\frac{\partial f(t, \xi_n)}{\partial y}\right) e_n - h\frac{h}{2} y''(\eta_n).$$

If $h < \frac{2}{\lambda_{max}}$ then we have $1 + h\frac{\partial f(t, \xi_n)}{\partial y} \in (-1, 1)$ and therefore

$$|e_{n+1}| \leq |e_n| + h\tau(h).$$

Since $e_0 = 0$, we deduce

$$|e_n| \leq nh\tau(h) = t_n\tau(h),$$

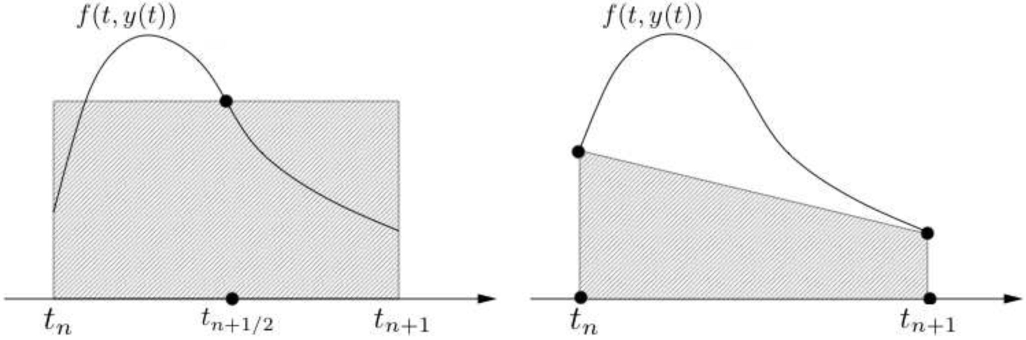
which was to be demonstrated (35).

Runge-Kutta methods of order 2

If we integrate the equation $y'(t) = f(t, y(t))$ between t_n and t_{n+1}, we obtain:

$$y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} f(t, y(t))dt. \tag{36}$$

Remark 5. *Numerical integration methods (Chapt. 4.2 in the book)*



We want to approximate the integral of the function $f(t, y(t))$. If we use the midpoint formula, we approximate the area below the curve by the area of a rectangle that has as a basis h and as a height the value of the function at time $t_n + h/2$ (see figure on the left). If we use the trapezoidal formula, we approximate the area below the curve by the area of a trapezoid that has as basis both values of the function at times t_n and t_{n+1} and as a height h (see figure on the right).

Using trapezoidal formula, we find the following implicit method, that is called **Crank-Nicolson or trapezoidal method** :

$$\boxed{u_{n+1} - u_n = \frac{h}{2}\left[f(t_n, u_n) + f(t_{n+1}, u_{n+1})\right], \quad \forall n \geq 0.} \tag{37}$$

This method is unconditionally stable when it is applied to the model problem (21).

If we modify the schema (37) (changing to explicit) then we obtain the **Heun method**:

$$\boxed{u_{n+1} - u_n = \frac{h}{2}\left[f(t_n, u_n) + f(t_{n+1}, u_n + hf(t_n, u_n))\right].} \tag{38}$$

Both methods (Crank-Nicolson and Heun) are of order 2 with respect to h.

If we use in (36) midpoint formula, we obtain

$$u_{n+1} - u_n = h\, f(t_{n+\frac{1}{2}}, u_{n+\frac{1}{2}}). \tag{39}$$

If we approximate $u_{n+1/2}$ by

$$u_{n+\frac{1}{2}} = u^n + \frac{h}{2} f(t_n, u_n),$$

we obtain **improved Euler method**:

$$\boxed{u_{n+1} - u_n = h\, f\left(t_{n+\frac{1}{2}}, u_n + \frac{h}{2} f(t_n, u_n)\right).} \tag{40}$$

Heun and improved Euler methods are particular cases of the Runge-Kutta method of order 2. When we apply them to the model problem (21), we have the same stability condition $h < 2/|\lambda|$ for both methods.

In the following table we summarize the characteristics of the methods:

Method	*Explicit/Implicit*	*Stability*	*w.r.to* h
Forward Euler	Explicit	Conditionally	1
Backward Euler	Implicit	Unconditionally	1
Crank–Nicolson	Implicit	Unconditionally	2
Heun	Explicit	Conditionally	2
Improved Euler	Explicit	Conditionally	2
Runge–Kutta	Explicit	Conditionally	4

There are more complicated methods, such as **Runge-Kutta method of order 4**, that is obtained by considering the integration of the Simpson method:

$$u_n \rightarrow \left\{ \begin{array}{c} u_{n+1} = u_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \\ \text{where:} \\ K_1 = f(t_n, u_n), \\ K_2 = f(t_n + \frac{h}{2}, u_n + \frac{h}{2}K_1), \\ K_3 = f(t_n + \frac{h}{2}, u_n + \frac{h}{2}K_2), \\ K_4 = f(t_{n+1}, u_n + hK_3). \end{array} \right.$$

Example 7. Let us consider the Cauchy problem

$$\begin{cases} y'(t) = -y(0.1 - \cos(t)), \quad t > 0 \\ y(0) = 1. \end{cases} \tag{41}$$

We solve this problem by the forward Euler and Heun methods on the interval $[0, 12]$ with a time step $h = 0.4$.

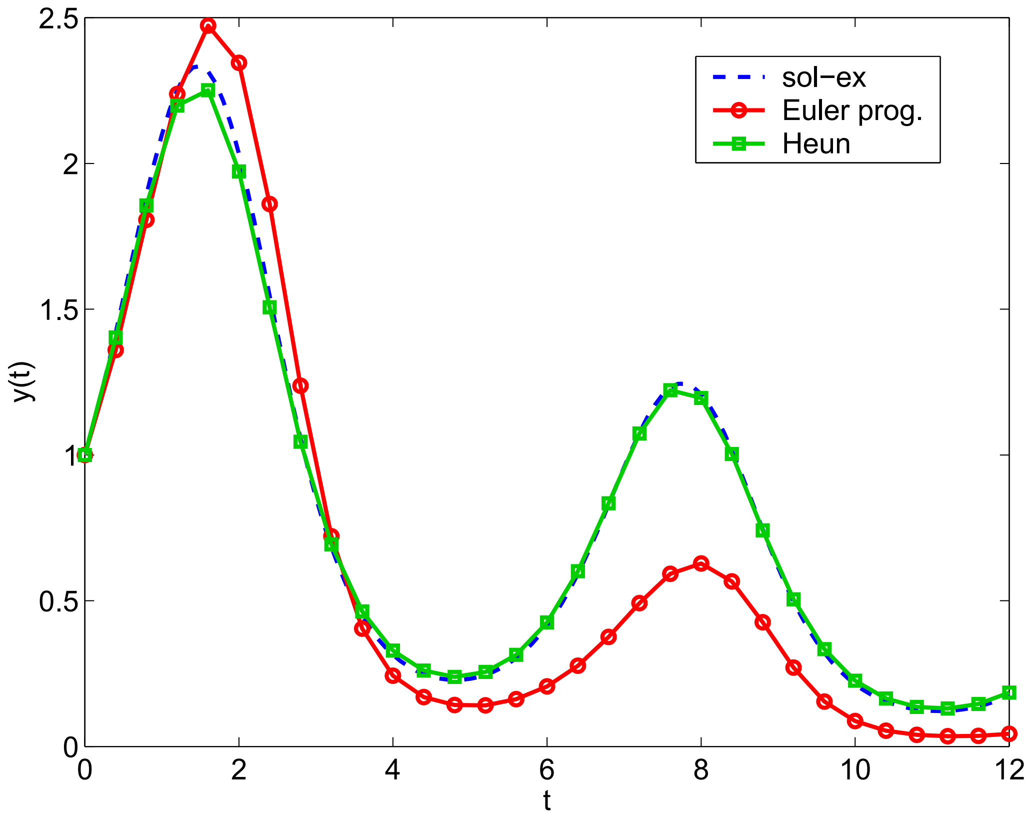
```
>> f = @(t,y) (cos(t) - 0.1)*y;
>> h = 0.4; tspan = [0 12]; y0 = 1; Nh = 12/h;
>> % forward Euler
>> [t_ep, y_ep] = feuler(f, tspan, y0, Nh);
>> % Heun
>> [t_heun, y_heun] = heun(f, tspan, y0, Nh);
```

The first of the following figures shows the solutions obtained by both methods and the exact solution $y(t) = e^{-0.1t+\sin(t)}$. Note that the solution obtained by the Heun method is much more precise than the forward Euler method.

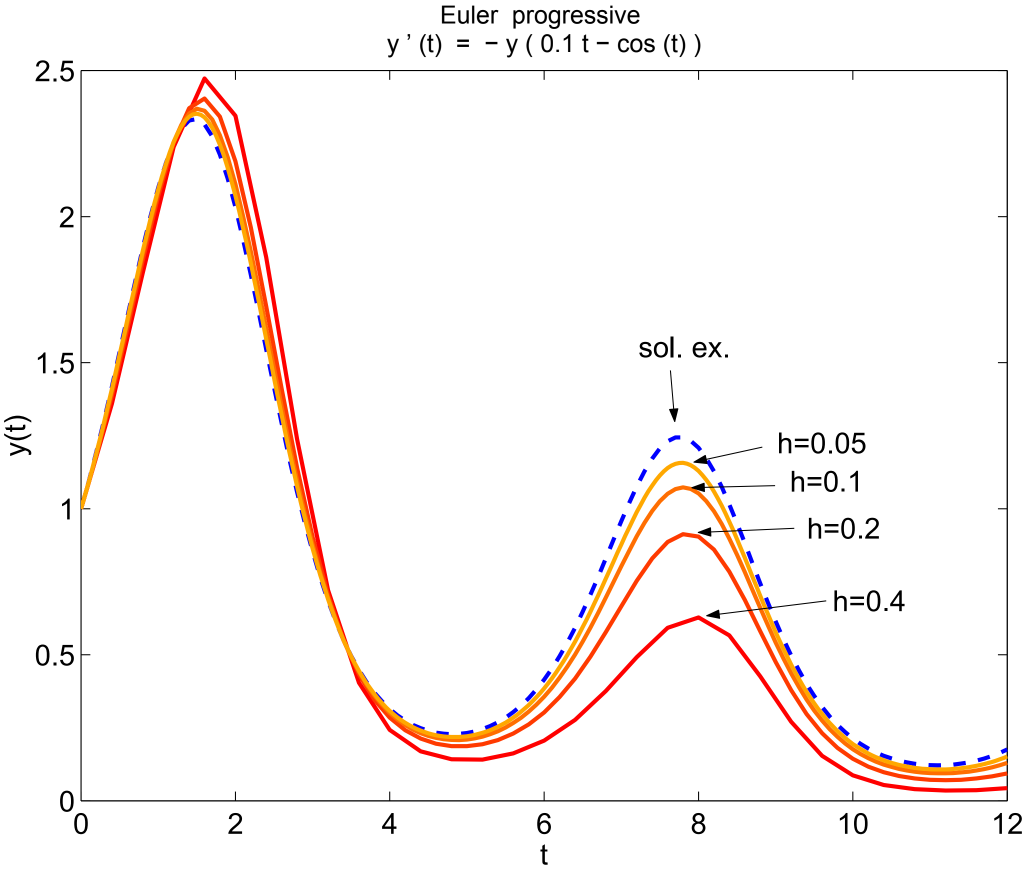
Moreover, we can see that if we reduce the time step, the solution obtained by the forward Euler method approximates the exact solution. The second figure shows the solutions obtained with $h = 0.4, 0.2, 0.1, 0.05$ using the following commands:

```
>> sol_ex = @(t) exp(-0.1*t + sin(t));
>> t = [0:0.01:12];
>> plot(t, sol_ex(t), 'b--'); hold on;
>> h=0.4; Nh = 12/h;
>> for i=1:4
    [t_ep, y_ep] = feuler(f, tspan, y0, Nh);
    plot(t_ep, y_ep)
    Nh = Nh*2;
   end
```

Comparison of solutions obtained by the forward Euler and Heun methods for $h = 0.4$.



Solutions obtained by the forward Euler method for different time steps.

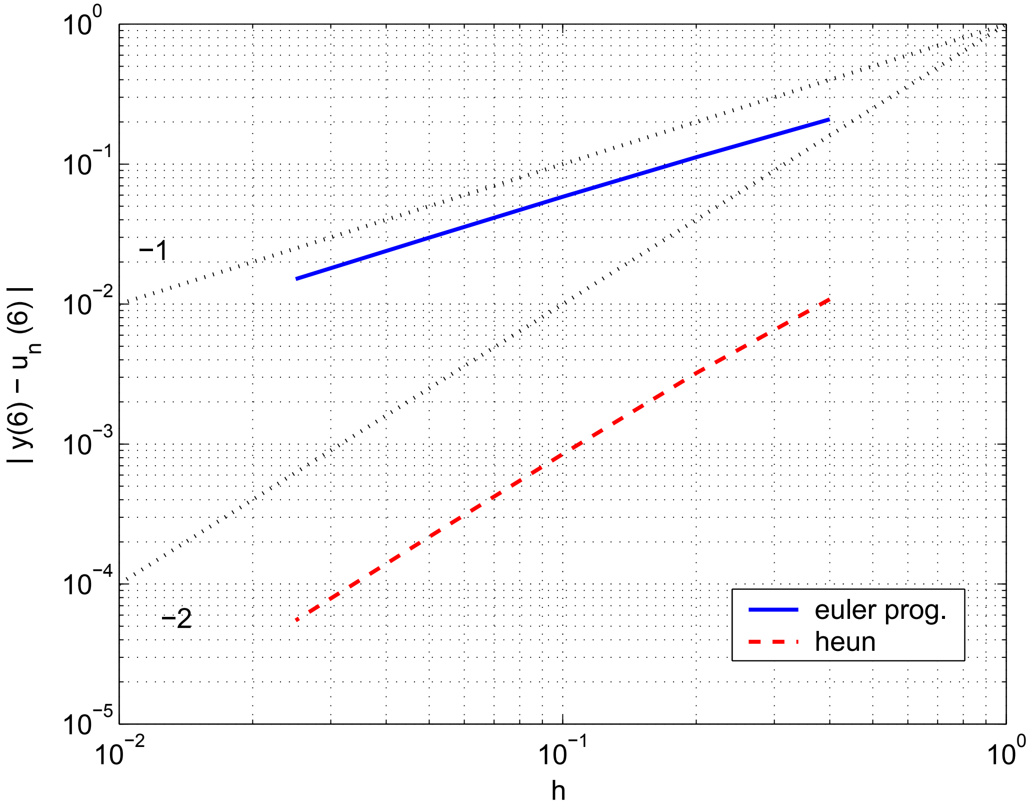


We want to estimate the order of convergence of these two methods. For this we will solve the problem with different time steps and we will compare the results obtained at time $t = 6$ with the exact solution.

```
>> h=0.4; Nh = 12/h; t=6; y6 = sol_ex(t);
>> for i=1:5
     % foreward Euler
     [t_ep, y_ep] = feuler(f, tspan, y0, Nh);
     err_ep(i) = abs(y6 - y_ep(fix(Nh/2)+1));
     % Heun
     [t_heun, y_heun] = heun(f, tspan, y0, Nh);
     err_heun(i) = abs(y6 - y_heun(fix(Nh/2)+1));
     Nh = Nh*2;
   end
>> h=[0.4, 0.2, 0.1, 0.05, 0.025];
>> loglog(h,err_ep,'b',h,err_heun,'r')
```

The following figure shows, in logarithmic scale, errors of both methods depending on h. Clearly, the forward Euler method converges with order 1 and Heun method converges with order 2.

Errors of the forward Euler and Heun methods in the calculation of $y(6)$. Note that a scale is logarithmic.



Systems of differential equations

(Chapt. 7.9 in the book)

Let us consider the following system of non-homogeneous ordinary differential equation with constant coefficients.

$$\begin{cases} \mathbf{y}'(t) = A\mathbf{y}(t) + \mathbf{b}(t) \quad t > 0, \\ \mathbf{y}(0) = \mathbf{y}_0, \end{cases} \tag{42}$$

where $A \in \mathbb{R}^{p \times p}$ and $\mathbf{b}(t) \in \mathbb{R}^p$, where we assume that A has got p distinct eigenvalues λ_j, $j = 1, \ldots, p$.

From the numerical point of view, the methods introduced in the scalar case can be extended to systems of differential equations. For example, the forward Euler method (12) becomes:

$$\begin{cases} \dfrac{\mathbf{u}_{n+1} - \mathbf{u}_n}{h} = A\mathbf{u}_n + \mathbf{b}_n \qquad \text{for } n = 0, 1, 2, \ldots, N_h - 1 \\ \mathbf{u}_0 = \mathbf{y}_0 \ , \end{cases} \tag{43}$$

while the backward Euler method (13) becomes:

$$\begin{cases} \dfrac{\mathbf{u}_{n+1} - \mathbf{u}_n}{h} = A\mathbf{u}_{n+1} + \mathbf{b}_{n+1} \qquad \text{for } n = 0, 1, 2, \ldots, N_h - 1 \\ \mathbf{u}_0 = \mathbf{y}_0 \ , \end{cases} \tag{44}$$

Regarding stability, if $\mathbf{b} \equiv \mathbf{0}$ and the eigenvalues λ_j $(j = 1, \ldots, p)$ of matrix A are strictly negative: $\lambda_j < 0$, $j = 1, \ldots, p$, then $\mathbf{y}(t) \to \mathbf{0}$ when $t \to +\infty$, and the forward Euler method is stable (i.e. $\mathbf{u}_n \to \mathbf{0}$ if $n \to +\infty$) if

$$h < \frac{2}{\max_{j=1,\ldots,p} |\lambda_j|} = \frac{2}{\rho(A)} \ , \tag{45}$$

where $\rho(A)$ is the spectral radius of A, while the backward Euler method is unconditionally stable.

Example 8. Linear system

The system

$$\left\{\begin{array}{lcl} y_1'(t) & = & -2y_1(t) + y_2(t) + e^{-t} \\ y_2'(t) & = & 3y_1(t) - 4y_2(t) \end{array}\right. \tag{46}$$

with initial conditions $y_1(0) = y_{10}$, $y_2(0) = y_{20}$, can be written as (42), where

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}, \quad A = \begin{bmatrix} -2 & 1 \\ 3 & -4 \end{bmatrix}, \quad \mathbf{b}(t) = \begin{bmatrix} e^{-t} \\ 0 \end{bmatrix}, \quad \mathbf{y}_0 = \begin{bmatrix} y_{10} \\ y_{20} \end{bmatrix}.$$

Let $h > 0$ be the time step; for $n \in \mathbb{N}$, we set $t_n = nh$, $\mathbf{b}_n = \mathbf{b}(t_n)$ and we denote by $\mathbf{u}_n$ an approximation of the exact solution $\mathbf{y}(t_n)$ at time t_n.

The forward Euler, backward Euler and Crank-Nicolson methods for approximating the solution $\mathbf{y}(t)$ of (46) are written respectively:

forward Euler
$$\left\{\mathbf{u}_{n+1} = \mathbf{u}_n + hA\mathbf{u}_n + h\mathbf{b}_n = (I + hA)\mathbf{u}_n + h\mathbf{b}_n\right.$$

backward Euler
$$\begin{cases}(I - hA)\mathbf{u}_{n+1} = \mathbf{u}_n + h\mathbf{b}_{n+1}\\ \mathbf{u}_0 = \mathbf{y}_0\end{cases}$$

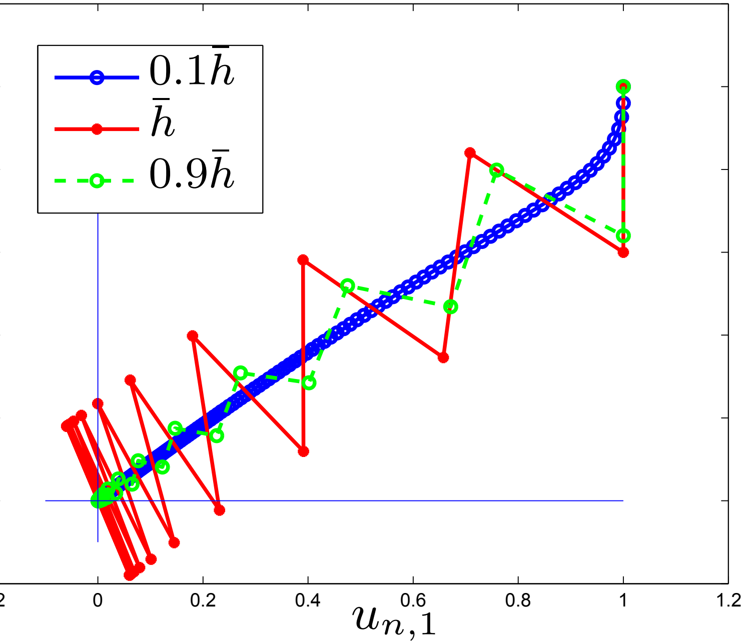
Crank-Nicolson
$$\begin{cases}(I - \frac{h}{2}A)\mathbf{u}_{n+1} = (I + \frac{h}{2}A)\mathbf{u}_n + \frac{h}{2}\left(\mathbf{b}_n + \mathbf{b}_{n+1}\right)\\ \mathbf{u}_0 = \mathbf{y}_0\end{cases}$$

It should be noted that at each step of the methods of BE and CN, we must solve a linear system with the matrix $I - hA$ and $I - \frac{h}{2}A$, respectively (these are implicit methods).

The forward Euler method is explicit (there is not linear system to solve), but it is stable conditionally. In this case, the eigenvalues of A are $\lambda_1 = -1$ and $\lambda_2 = -5$; they are strictly negative, so the condition (45) on h is satisfied. $\rho(A) = 5$, so the stability condition is

$$h < \bar{h} = \frac{2}{5}.$$

Behavior of the forward Euler method for the system (46) with initial condition $\mathbf{y}_0 = [1, 1]^\top$ and different values of the time step h.



```
A = [-2 1; 3 -4];
dy = @(t,y,A) A*y + [exp(-t); 0];
h_bar = 2 / max(abs(eig(A)));
[t,y]=feuler(dy,[0,10],[1;1],10/(0.1*h_bar),A);
plot(y(1,:), y(2,:)); hold on;
[t,y]=feuler(dy,[0,10],[1;1],10/(h_bar),A);
plot(y(1,:), y(2,:), 'ro');
[t,y]=feuler(dy,[0,10],[1;1],10/(0.9*h_bar),A);
plot(y(1,:), y(2,:), 'go--');
```

We could also consider the case of a nonlinear system of the form

$$\mathbf{y}'(t) = \mathbf{F}(t, \mathbf{y}(t)),$$

(for example the system (2)). If $\frac{\partial \mathbf{F}}{\partial \mathbf{y}}$ is a matrix with real and negative eigenvalues, then the backward Euler method is unconditionally stable, while the forward Euler method is stable under condition (45), where $A = \frac{\partial \mathbf{F}}{\partial \mathbf{y}}$.

Example 9. Nonlinear system
The nonlinear system

$$\begin{aligned} y_1'(t) &= -2y_1(t) + \sin(y_2(t)) + e^{-t}\sin(t), \\ y_2'(t) &= \cos(y_1(t)) - 4y_2(t), \end{aligned} \tag{47}$$

with initial conditions $y_1(0) = y_{10}$, $y_2(0) = y_{20}$, can be written as

$$\mathbf{y}'(t) = \mathbf{F}(t, \mathbf{y}(t)),$$

where

$$\mathbf{F}(t, \mathbf{y}(t)) = \begin{bmatrix} -2y_1(t) + \sin(y_2(t)) + e^{-t}\sin(t) \\ \cos(y_1(t)) - 4y_2(t) \end{bmatrix}.$$

Let $h > 0$ be the time step; for $n \in \mathbb{N}$, we set $t_n = nh$ and we denote by $\mathbf{u}_n$ an approximation of the exact solution $\mathbf{y}(t_n)$ at time t_n.

The forward Euler, backward Euler and Crank-Nicolson methods for approximating the solution $\mathbf{y}(t)$ of (47) are written respectively:

forward Euler
$$\begin{cases} \mathbf{u}_{n+1} = \mathbf{u}_n + h\mathbf{F}(t_n, \mathbf{u}_n), \\ \mathbf{u}_0 = \mathbf{y}_0, \end{cases}$$

backward Euler
$$\begin{cases} \mathbf{u}_{n+1} - h\mathbf{F}(t_{n+1}, \mathbf{u}_{n+1}) = \mathbf{u}_n, \\ \mathbf{u}_0 = \mathbf{y}_0, \end{cases}$$

Crank-Nicolson
$$\begin{cases} \mathbf{u}_{n+1} - \frac{h}{2}\mathbf{F}(t_{n+1}, \mathbf{u}_{n+1}) = \mathbf{u}_n + \frac{h}{2}\mathbf{F}(t_n, \mathbf{u}_n), \\ \mathbf{u}_0 = \mathbf{y}_0. \end{cases}$$

It should be noted that at each step of the methods of BE and CN, we must solve a nonlinear system.

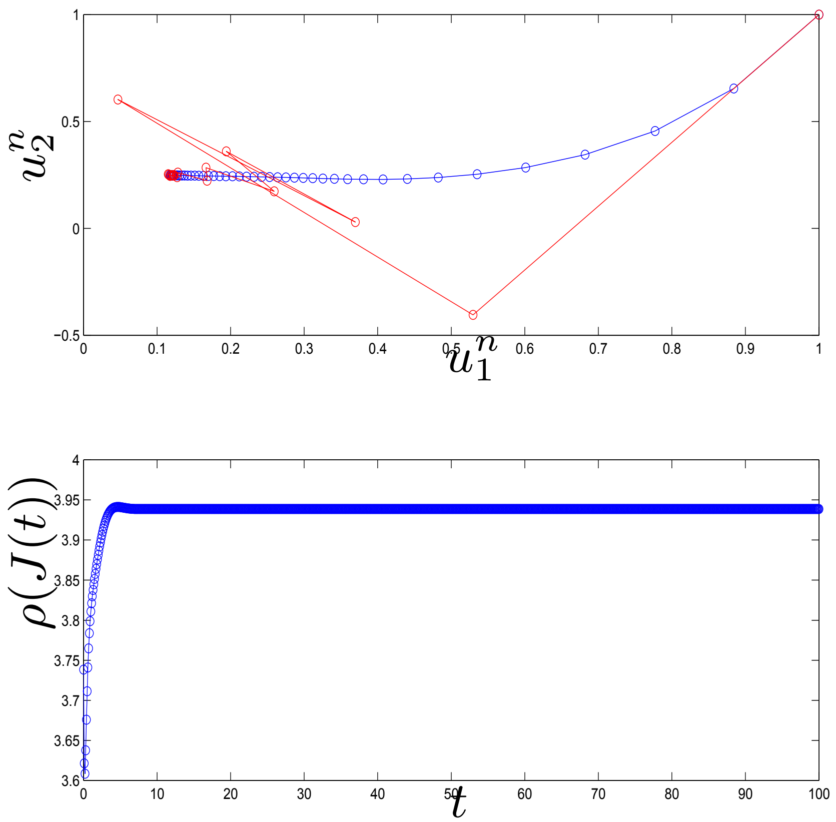
The forward Euler method is explicit (there is not system to solve), but it is stable conditionally. In this case, the jacobian of $\mathbf{F}$ is given by

$$J = \frac{\partial \mathbf{F}}{\partial \mathbf{y}} = \begin{bmatrix} -2 & \cos y_2 \\ -\sin y_1 & -4, \end{bmatrix}$$

and the eigenvalues are $\lambda_{1,2} = -3 \pm \sqrt{1 - \sin y_1 \cos y_2}$; They are strictly negative, in particular $-3 - \sqrt{2} < \lambda_{1,2} < -3 + \sqrt{2} < 0$, and $\rho(J) < 3 + \sqrt{2}$. Therefore the stability condition is

$$h < \bar{h} = \frac{2}{\rho(J)}, \quad \text{for example if} \quad h < \frac{2}{3+\sqrt{2}} \simeq 0.453.$$

Behaviour of the forward Euler method for the system(47) with initial condition $\mathbf{y}_0 = [1,1]^\top$: $h = 0.1$ (blue) and $h = 0.8\bar{h}$ (red). If we take $h \geq \bar{h}$, we can observe the instability of the method.



We used the following commands:

```
dy = @(t,y) [-2*y(1) + sin(y(2)) + exp(-t)*sin(t); ...
              cos(y(1)) - 4*y(2)]
[t,y] = feuler(dy, [0,100],[1; 1], 100/0.1);
subplot(2,1,1); plot(y(1,:), y(2,:),’o-’); hold on;

J = @(y) [-2, cos(y(2)); -sin(y(1)), - 4];
for i = 1:size(t,2);
  rho(i) = max(abs(eig(feval(J, y(:,i)))));
end

subplot(2,1,2);
plot(t,rho, ’o-’);

h_bar = 2/max(rho);

[t,y] = feuler(dy, [0,100],[1; 1], 100/(0.8*h_bar));
subplot(2,1,1);
plot(y(1,:), y(2,:), ’ro-’);
```

Here is a summary of the stability:

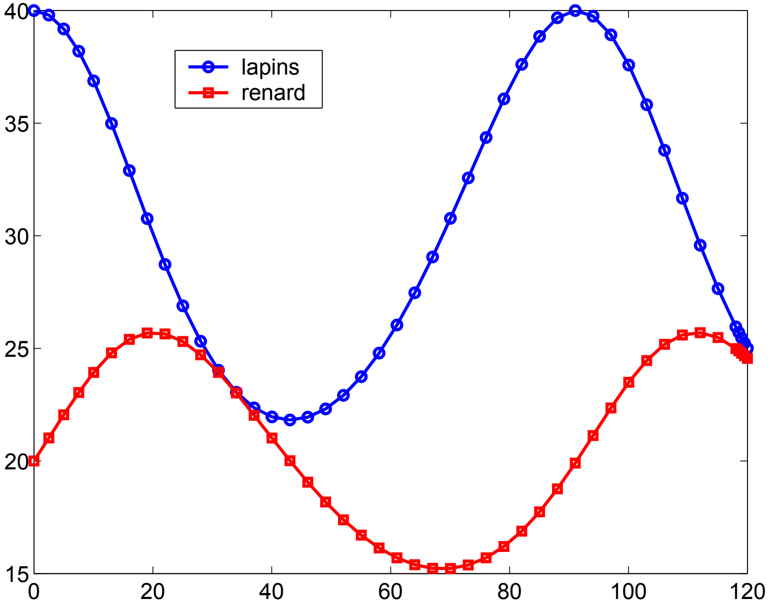
Problem		Stability of the explicit methods
Model	$y' = \lambda y$	$h < 2/\|\lambda\|$
Cauchy	$y' = f(t, y(t))$	$h < 2/\max \left\| \frac{\partial f}{\partial y} \right\|$
System of linear eq.	$\mathbf{y}' = \mathbf{A}\mathbf{y} + \mathbf{b}$	$h < 2/\rho(\mathbf{A})$
System of nonlinear eq.	$\mathbf{y}' = \mathbf{F}(t, \mathbf{y}(t))$	$h < 2/\rho(\mathbf{J})$

for

- $\rho(\mathbf{A}) = \max_i |\lambda_i(\mathbf{A})|$, for a system of linear equations;
- $\rho(\mathbf{J}) = \max_i |\lambda_i(\mathbf{J})|$, for a system of nonlinear equations, where $\mathbf{J}(t, \mathbf{y}) = \dfrac{\partial \mathbf{F}}{\partial \mathbf{y}}$, for $\lambda_i(\mathbf{J}) < 0$, $\forall i$.

Applications

We return to the example given at the beginning of the chapter.



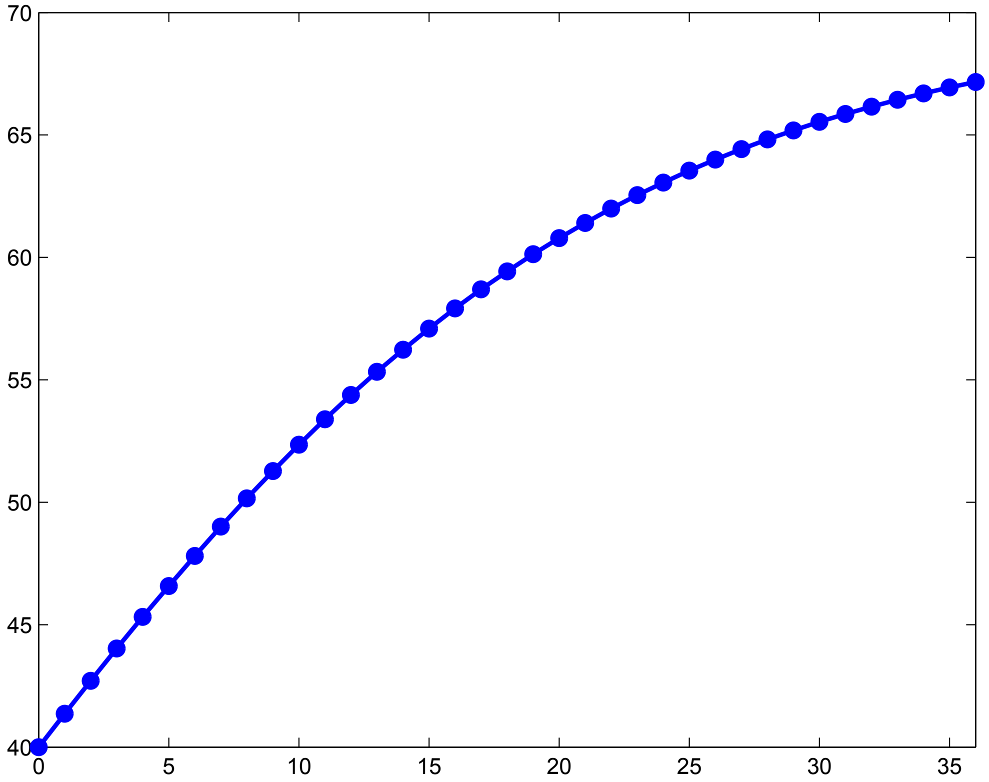
Example. 1 (suite) At first, we consider the scalar equation (1):

$$y'(t) = Cy(t)\left(1 - \frac{y(t)}{B}\right), \ t > 0, \quad y(0) = y_0.$$

Let us take an initial population of 40 rabbits whose growth factor is $C = 0.08$ (the unit of time is one month) and the maximum population is equal B $B = 70$ rabbits. We solve the equation using the Heun method with $h = 1$ month over a period of three years:

```
>> f=@(t,y) 0.08*y.*(1-(y/70));
>> tspan = [0 36]; y0=40; h = 1; Nh = 36/h;
>> [t, y] = heun(f,tspan,y0,Nh); plot(t,y)
```

Evolution of the population of rabbits over a period of 3 years.



Now, we consider the system (2). Let us take an initial population $y_1(0)$ of 40 rabbits, a population $y_2(0)$ of 20 foxes and the Lotka-Volterra equations:

$$\begin{cases} y_1'(t) = 0.08\, y_1(t) - 0.004\, y_1(t)y_2(t), \\ y_2'(t) = -0.06\, y_2(t) + 0.002\, y_1(t)y_2(t). \end{cases} \tag{48}$$

We want to study the evolution of the two populations over a period of 10 years. Let us define the vectors:

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}, \qquad \mathbf{F}(t,\mathbf{y}) = \begin{bmatrix} 0.08\, y_1(t) - 0.004\, y_1(t)y_2(t) \\ -0.06\, y_2(t) + 0.002\, y_1(t)y_2(t) \end{bmatrix},$$

We can rewrite the system (48) in the general form:

$$\mathbf{y}'(t) = \mathbf{F}(t,\mathbf{y}), \;\; t > 0, \qquad \mathbf{y}(0) = [y_1(0), y_2(0)]^T. \tag{49}$$

All the methods that we have seen so far are applicable to the system(49). For example, the forward Euler method can be written as

$$\frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{h} = \mathbf{F}(t_n, \mathbf{u}_n),$$

which is equivalent to the system of equations

$$\begin{cases} \dfrac{u_{n+1,1} - u_{n,1}}{h} = 0.08\, u_{n,1} - 0.004\, u_{n,1} u_{n,2}, \quad n \geq 0 \\ \dfrac{u_{n+1,2} - u_{n,2}}{h} = -0.06\, u_{n,2} + 0.002\, u_{n,1} u_{n,2}, \quad n \geq 0 \\ u_{0,1} = y_1(0), \quad u_{0,2} = y_2(0). \end{cases}$$

The command **heun** can solve system of differential equations. First we must write a function that define the system:

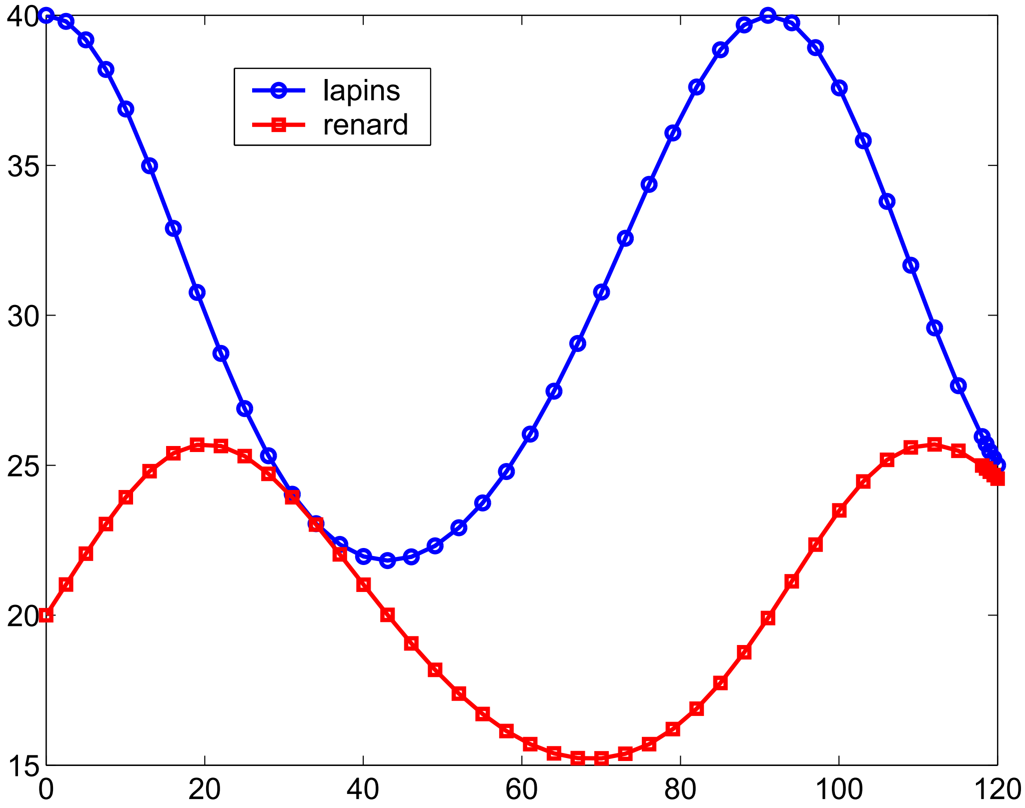
```
>> fun2 = @(t,y) [ 0.08*y(1) - 0.004*y(1)*y(2);
                  -0.06*y(2) + 0.002*y(1)*y(2) ]
```

Then we can solve the system using:

```
>> y0=[40 20]; tspan=[0 120]; Nh=40;
>> [t,y] = heun(fun2, tspan, y0, Nh);
>> plot(t,y(:,1),'b', t,y(:,2),'r')
```

The first column of `y` contains the solution y_1, while the second column contains y_2. The following figure shows the evolution of the two populations.

Evolution of populations of rabbits and foxes over 10 years.


lapins
renard