FORWARD EULER METHOD

$$\begin{cases} u_0 \text{ given} \\ u_{n+1} = u_n + hf(t_n, u_n) \quad 0 \leq n \leq N_h - 1 \end{cases} \tag{1}$$

Download the function `feuler.m`

`[tn,un]=feuler(odefun,tspan,y0,Nh)`

INPUT:

`odefun`: function f (function handle)

`tspan=[t0,T]`: 2-components array, t0= initial time, T=final time

`y0`: initial condition

`Nh`: integer number of time steps (Nh is s.t. $T = t_{Nh}$).

OUTPUT:

`tn`: column array holding all discrete times since t_0 up tu t_{Nh}.

`un`: column array holding the numerical solution at times `tn`

Exercise 1 (esode1.m)

Approximate the solution of the Cauchy problem

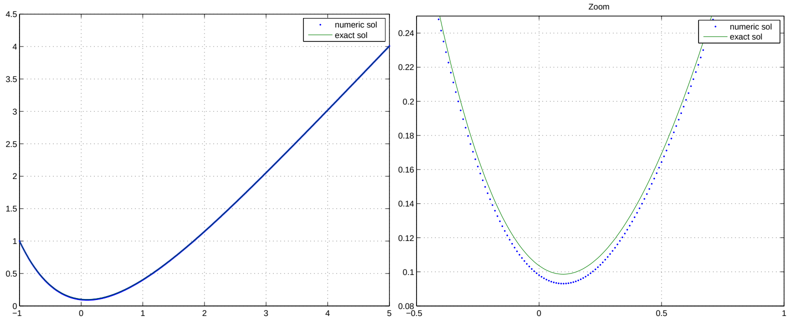
$$\begin{cases} y'(t) = t - y(t) & t \in (-1, 5] \\ y(-1) = 1. \end{cases} \tag{2}$$

Write a script that set the data, calls `feuler`, plots the numerical solution and finally computes the error

$$e_h = \max_{n=0,\dots,N_h} |y(t_n) - u_n|$$

w.r.t. the exact solution $y(t) = t - 1 + 3e^{-(t+1)}$. Set $h = 0.001$.

The plot of both exact and computed solutions



Exercise 2 (esode2.m)

Write a script that solves the Cauchy problem of Ex.1, with $h = 1, .8, .5, .1, .05, .01, .001$.

For each value of h:

- call the function `feuler` to compute `[tn,un]`,
- plot both numerical and exact solution,
- compute the error

$$e_h = \max_{n=0,\dots,N_h} |y(t_n) - u_n|$$

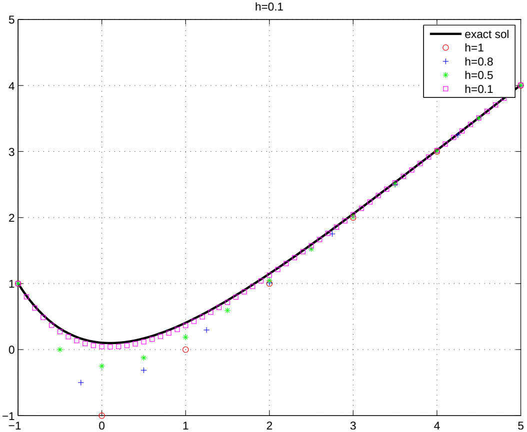
and store these errors inside an array.

Recall that the exact solution is: $y(t) = t - 1 + 3e^{-(t+1)}$.

Once the loop on h is terminated, plot the errors e_h in logarithmic scale and verify that Forward Euler is 1st-order accurate, i.e.

$$\exists C > 0 \text{ s.t. } e_h = C \cdot h \text{ when } h \to 0.$$

Exact and numerical solutions for some values of h:

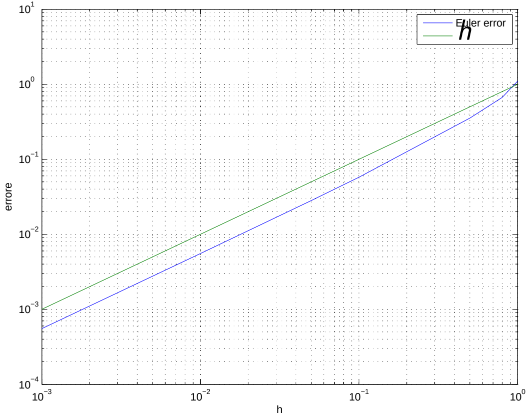


The smaller h is, the closer the numerical solution is to the exact one.

The numerical solution CONVERGES to the exact one.

When $h \to 0$, $\max_n |u_n - y(t_n)| \to 0$.

The error as a function on h is:



The two lines are parallel, then the error behaves like h when $h \longrightarrow 0$.

BACKWARD EULER METHOD

$$\begin{cases} u_0 \text{ given} \\ u_{n+1} = u_n + hf(t_{n+1}, u_{n+1}), \quad 0 \leq n \leq N_h - 1 \end{cases} \tag{3}$$

Download
`http://dm.ing.unibs.it/gervasio/calnum/matlab/beuler.m`:

```
[tn,un]=beuler(odefun,tspan,y0,Nh)
```

INPUT:
`odefun`: function f (function handle)
`tspan=[t0,T]`: 2-components array, t0= initial time, T=final time `y0`: initial condition
`Nh`: integer number of time steps (Nh is s.t. $T = t_{Nh}$).
OUTPUT:
`tn`: column array holding all discrete times since t_0 up tu t_{Nh}.
`un`: column array holding the numerical solution at times `tn`

CRANK-NICOLSON METHOD

$$\begin{cases} u_0 \text{ given} \\ u_{n+1} = u_n + \frac{h}{2}[f(t_n, u_n) + f(t_{n+1}, u_{n+1})], \quad 0 \leq n \leq N_h - 1 \end{cases} \tag{4}$$

Download
`http://dm.ing.unibs.it/gervasio/calnum/matlab/cranknic.m`:

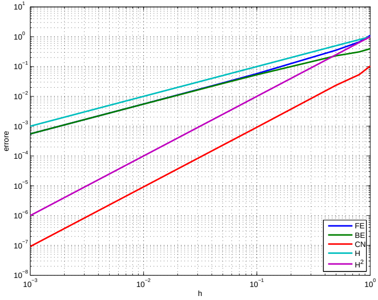
```
[tn,un]=cranknic(odefun,tspan,y0,Nh)
```

INPUT:
`odefun`: function f (function handle)
`tspan=[t0,T]`: 2-components array, t0= initial time, T=final time `y0`: initial condition
`Nh`: integer number of time steps (Nh is s.t. $T = t_{Nh}$).
OUTPUT:
`tn`: column array holding all discrete times since t_0 up tu t_{Nh}.
`un`: column array holding the numerical solution at times `tn`

Exercise 3 (esode3.m)

Repeat Exercise 2, by calling `beuler` first and `cranknic` after. By plotting the errors, verify that BE is 1st order accurate, while CN is 2nd order accurate when $h \longrightarrow 0$.

BE and CN are very expensive: they are implicit methods and non-linear equations have to be solved at each time step.



Vector Cauchy problem

$$\begin{cases} \mathbf{y}'(t) = \mathbf{F}(t, \mathbf{y}(t)) & t \in (t_0, T] \\ \mathbf{y}(t_0) = \mathbf{y}_0 \end{cases}$$

with

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}, \qquad \mathbf{y}'(t) = \begin{bmatrix} y_1'(t) \\ y_2'(t) \end{bmatrix},$$

$$\mathbf{y}_0 = \begin{bmatrix} y_{0,1} \\ y_{0,2} \end{bmatrix}, \qquad \mathbf{F}(t, \mathbf{y}(t)) = \begin{bmatrix} F_1(t, y_1(t), y_2(t)) \\ F_2(t, y_1(t), y_2(t)) \end{bmatrix}$$

Example. Look for $y_1(t), y_2(t)$ solutions of

$$\begin{cases} y_1'(t) = -3y_1(t) - y_2(t) + \sin(t) & t \in (0, 10] \\ y_2'(t) = y_1(t) - 5y_2(t) - 2 & t \in (0, 10] \\ y_1(0) = y_{0,1} = 1, \ \ y_2(0) = y_{0,2} = 1 \end{cases}$$

Vector form of FE, BE, CN

Forward Euler

$$\begin{cases} \mathbf{u}_{n+1} = \mathbf{u}_n + h\mathbf{F}(t_n, \mathbf{u}_n) & n \geq 0 \\ \mathbf{u}_0 = \mathbf{y}_0 \end{cases}$$

Backward Euler

$$\begin{cases} \mathbf{u}_{n+1} = \mathbf{u}_n + h\mathbf{F}(t_{n+1}, \mathbf{u}_{n+1}) & n \geq 0 \\ \mathbf{u}_0 = \mathbf{y}_0 \end{cases}$$

Crank-Nicolson

$$\begin{cases} \mathbf{u}_{n+1} = \mathbf{u}_n + \frac{h}{2}(\mathbf{F}(t_n, \mathbf{u}_n) + \mathbf{F}(t_{n+1}, \mathbf{u}_{n+1})) & n \geq 0 \\ \mathbf{u}_0 = \mathbf{y}_0 \end{cases}$$

All the functions work on vector problems.
The initial datum $\mathbf{y}_0$ now must be a vector (either row or column).
`un` is a two column arrays:

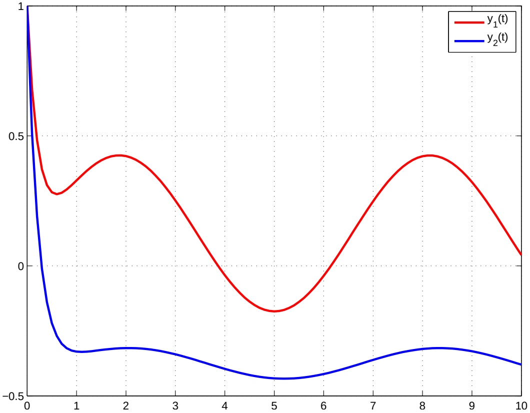
$$
\texttt{un}= \begin{array}{|c|c|}
\hline
y_{0,1} & y_{0,2} \\
u_{1,1} & u_{2,1} \\
u_{1,2} & u_{2,2} \\
 & \\
 & \\
 & \\
u_{1,N_h} & u_{2,N_h} \\
\hline
\end{array}
$$

The fucntion `f=odefun(t,y)` has two inputs: `t` (scalar) and `y` (vector) and it yields the vector `f` with the same dimension as `y`.

Exercise (esodesys1) Solve the system

$$\begin{cases} y_1'(t) = -3y_1(t) - y_2(t) + \sin(t) & t \in (0, 10] \\ y_2'(t) = y_1(t) - 5y_2(t) - 2 & t \in (0, 10] \\ y_1(0) = y_{0,1} = 1, \;\; y_2(0) = y_{0,2} = 1 \end{cases}$$

with FE and $h = 1.e-3$.

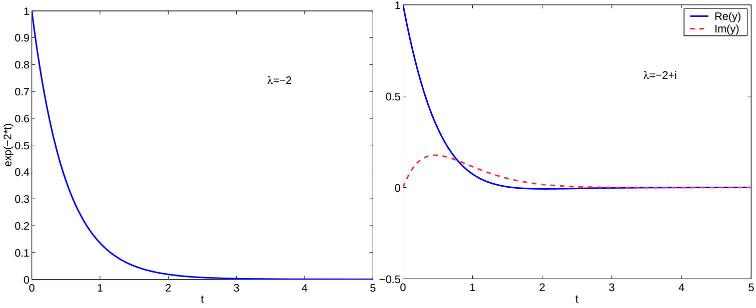


Absolute stability

Let us consider the linear model problem

$$\begin{cases} y'(t) = \lambda y(t) & t \geq 0 \\ y(0) = 1 \end{cases}$$

with $\lambda \in \mathbb{C}$ and $Re(\lambda) < 0$. The solution is an exponential function vanishing to zero when $t \longrightarrow \infty$.



We ask that the nuemrical solution reflects the same behavior of the exact solution, that is it vanishes when $t_n \to \infty$.
FE for the problem $y'(t) = f(t, y(t)) = \lambda y(t)$ reads:

$$u_{n+1} = u_n + hf(t_n, u_n) = u_n + h\lambda u_n = (1 + h\lambda)u_n$$

In order that $u_n \to 0$ when $t_n \to \infty$.

$$\begin{aligned} u_{n+1} &= (1 + h\lambda)u_n \\ &= (1 + h\lambda)^2 u_{n-1} \\ &= .... = (1 + h\lambda)^{n+1} u_0 \end{aligned} \tag{5}$$

Therefore,

$$|u_n| \to 0 \text{ per } t_n \to \infty \iff |1 + h\lambda| < 1$$

Se $\lambda \in \mathbb{R}$,

$$|1 + h\lambda| < 1 \iff 0 < h < \frac{-2}{\lambda}$$

Se $\lambda \in \mathbb{C}$,

$$|1 + h\lambda| < 1 \iff 0 < h < \frac{-2Re(\lambda)}{|\lambda|^2}.$$

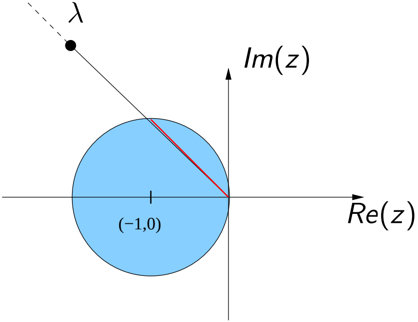
The Absolute Stability Region of a generic scheme is the set

$$\mathcal{A} = \{z = h\lambda \in \mathbb{C} : u_n \to 0 \text{ per } t_n \to \infty\}$$

ASR of FE is

$$\mathcal{A}_{FE} = \{z = h\lambda \in \mathbb{C} : \ |1 + z| < 1\}$$

i.e., the circle (without boundary) of the complex plane centerd in $(-1, 0)$ and with radius 1.



λ is given, we must to determine which values of h guarantee absolute stability

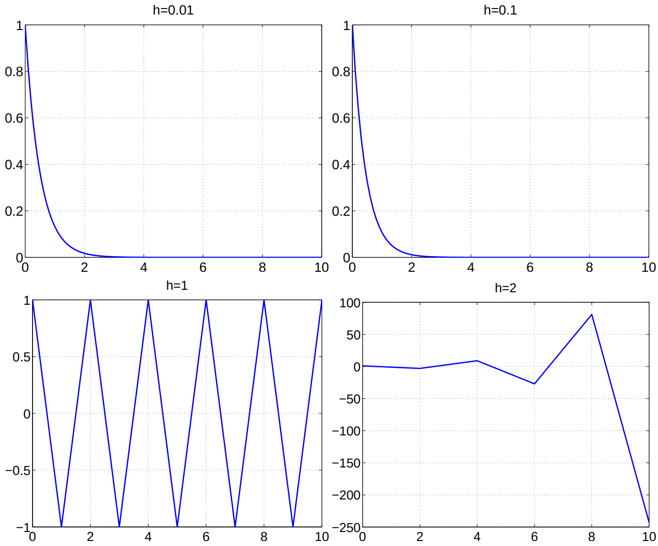
Exercise 4 (esode4.m)

Let us consider the linear problem

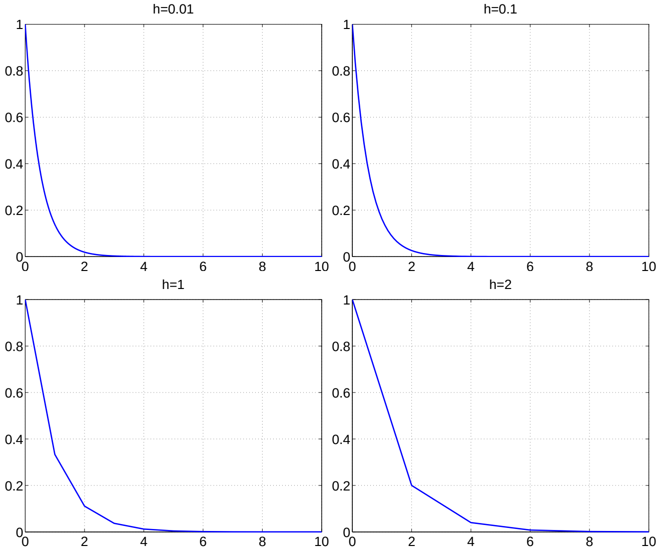
$$\begin{cases} y'(t) = -2y(t) & t \in [0, 10] \\ y(0) = 1 \end{cases}$$

1. Look for bounds on h such that FE is absolutely stable.
2. Compute and plot the numerical solution by FE with $h = 0.01$, $h = 0.1$, $h = 1$, $h = 2$. Set $h_0 = \frac{-2Re(\lambda)}{|\lambda|^2}$, verify that when $h < h_0$ the numerical scheme is absolutely stable, i.e. $u_n \to 0$ when $n \to \infty$, while when $h \geq h_0$ the numerical solution provides oscillations that do not vanish when $t_n \to \infty$.
3. Compute and plot the numerical solution by both BE and CN with $h = 0.01$, $h = 0.1$, $h = 1$, $h = 2$.

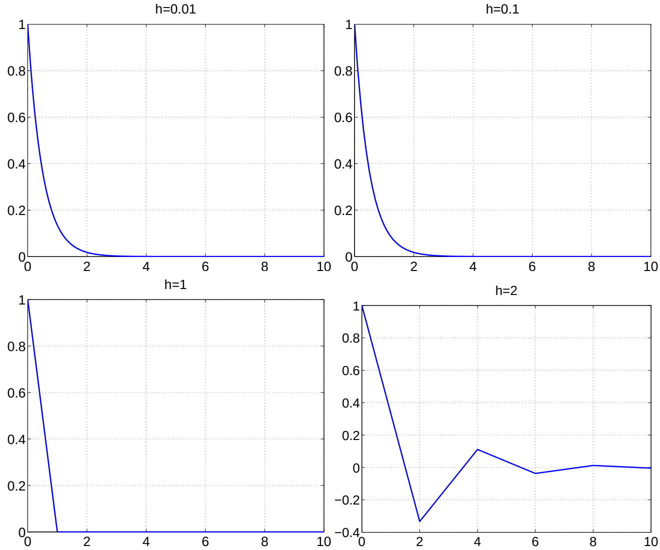
FE solution: absolutely stable $\forall h \in (0, 1)$



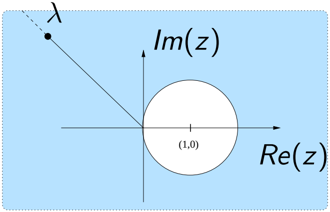
BE solution: absolutely stable $\forall h > 0$



CN solution: absolutely stable $\forall h > 0$

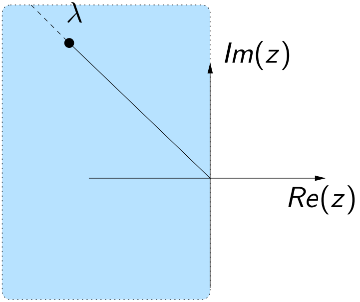


Absolute stability region for BE



$$\mathcal{A}_{BE} = \{z = h\lambda \in \mathbb{C} : \ |1 - z| > 1\}$$

Absolute stability region for CN



$$\mathcal{A}_{CN} = \{z = h\lambda \in \mathbb{C} : \ |2+z|/|2-z| < 1\}$$