Laboratory 2 - 17/10/16 - Solution

Exercise 1

(a) In the file `FE.m`, we report the implementation of the forward Euler method for the solution of the reference problem with $\lambda \in \mathbb{R}^-$, whose exact solution is given by $y(t) = y_0 e^{\lambda(t-t_0)}$. The scheme is obtained by applying the forward Euler method to the reference problem:

$$u^{n+1} = u^n + \Delta t\, \lambda\, u^n = (1 + \lambda \Delta t)\, u^n.$$

The input parameters of the function `FE` are: the initial time `t0`, the final time `tf`, the initial condition `u0`, the time step `dt` and the coefficient `lambda`.

(b) We report in the file `Lab2Es1b.m` the Matlab instructions to solve the reference problem with $\lambda = -2$ in the interval $\mathcal{I} = (0, 12]$ with $\Delta t = 0.05$ and $\Delta t = 1.2$.

In Figure 1a we report the comparison between the exact solution and the numerical one computed with $\Delta t = 0.05$, whereas in Figure 1b we report the one obtained with $\Delta t = 1.2$. We notice that the solution computed with $\Delta t = 1.2$ exhibits numerical oscillations due to the fact that the method is not absolutely stable for the chosen value of Δt. Indeed, the absolute stability for the forward Euler method is guaranteed for

$$\Delta t < \frac{2}{|\lambda|}, \tag{1}$$

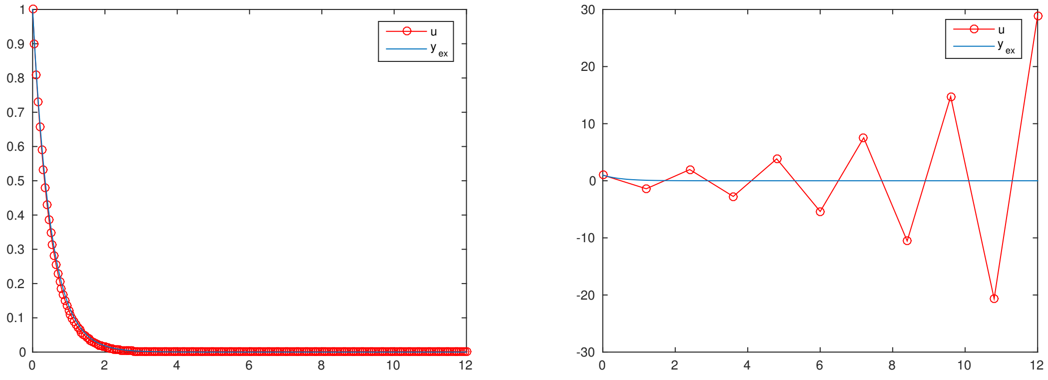
so that in our case we have to satisfy $\Delta t < 1$ in order to obtain a stable solution.

(c) Both the forward and backward Euler method have order of accuracy equal to 1, i.e. $|u^n - y(t^n)| = \mathcal{O}(\Delta t)$. To numerically prove such result we consider the solution of the reference problem for halved values of Δt and we assume that the method is of order p. Thus, in order to estimate the error reduction, we can use the following formulas

$$E_1(\Delta t) = \mathcal{O}\left(\Delta t^p\right),$$
$$E_2\left(\frac{\Delta t}{2}\right) = \mathcal{O}\left(\left(\frac{\Delta t}{2}\right)^p\right),$$

from which we obtain

$$\frac{E_1}{E_2} = \frac{\mathcal{O}\left(\Delta t^p\right)}{\mathcal{O}\left(\left(\frac{\Delta t}{2}\right)^p\right)} \approx 2^p.$$



(a) $\Delta t = 0.05$ (b) $\Delta t = 1.2$

Figure 1: Forward Euler method.

Hence, the order of accuracy can be estimated by $\overline{p} = \log_2(E_1/E_2)$.

We report in the file `accuracyOrder.m` a possible implementation for the computation of the order of accuracy of the forward Euler method applied to the reference problem. The function requires as input parameter the number `M` of successive halved time steps that we want to use for the estimation of the order `p` (besides the parameters required by the function `FE`).

Using for example $M = 8$ and the initial time step $\Delta t = 0.05$, we obtain (see the file `Lab2Es1c.m`)

```
p =

     1.0314    1.0154    1.0076    1.0038    1.0019    1.0009    1.0005.
```

These results confirm that the order of accuracy of the forward Euler method is 1.

Exercise 2

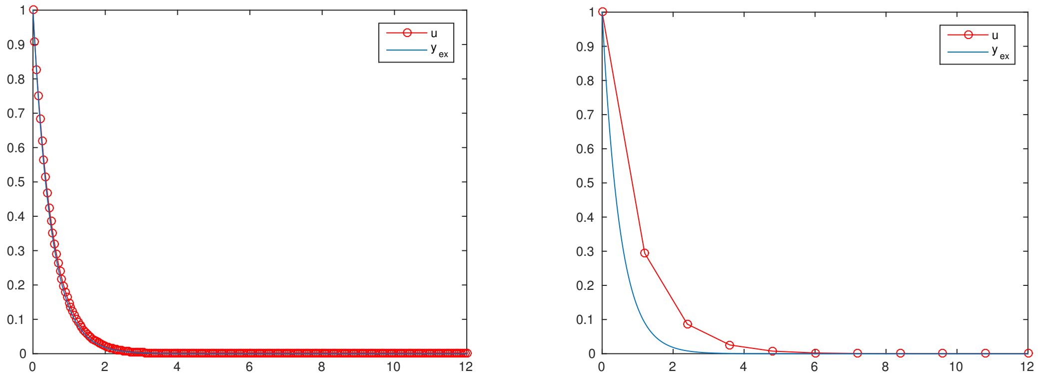
(a) In the file `BE.m`, we report the implementation of the backward Euler method for the solution of the reference problem with $\lambda \in \mathbb{R}^-$. The scheme is obtained by applying the backward Euler method to the reference problem:

$$u^{n+1} = u^n + \Delta t\, \lambda\, u^{n+1},$$

from which

$$u^{n+1} = \frac{1}{1 - \Delta t\, \lambda} u^n.$$

The input parameters of the function `BE` are the same of the one defined for the function `FE`.



(a) $\Delta t = 0.05$ (b) $\Delta t = 1.2$

Figure 2: Backward Euler method.

Notice that, even if the backward Euler method is in general an implicit method (i.e. it requires the solution of a non-linear equation, for example with the Newton method, thus requiring a greater computational cost with respect to the explicit method), in this case it is possible to obtain an explicit expression of u^{n+1}.

(b) We report in the file `Lab2Es2b.m` the Matlab instructions to solve the reference problem with $\lambda = -2$ in the interval $\mathcal{I} = (0, 12]$ with $\Delta t = 0.05$ and $\Delta t = 1.2$.

In Figure 2a we report the comparison between the exact solution and the numerical one computed with $\Delta t = 0.05$, whereas in Figure 2b we report the one obtained with $\Delta t = 1.2$. We notice that, differently from the previous exercise, the solution computed with the bakward Euler method is stable even for $\Delta t = 1.2$. Indeed, the backward Euler method is unconditionally absolutely stable, i.e. $\lim_{n\to\infty} |u^n| = 0$ for any value of Δt. However, the choice of Δt obviously influences the accuracy of the solution, as we can observe in Figure 2b.

Exercise 3

(a) We report in the file `FEGeneric.m` the implementation of the forward Euler method for the solution of a generic Cauchy problem. The function $f(t, y)$ passed as input parameter can be defined using the command `inline`:

```
fun = inline('1./(1+t.^2) - 2*y.^2','t','y');
```

Consequently, we have to modify the formula for the computation of the numerical solution u^{n+1} by introducing the command `feval` for the evaluation of $f(t^n, u^n)$:

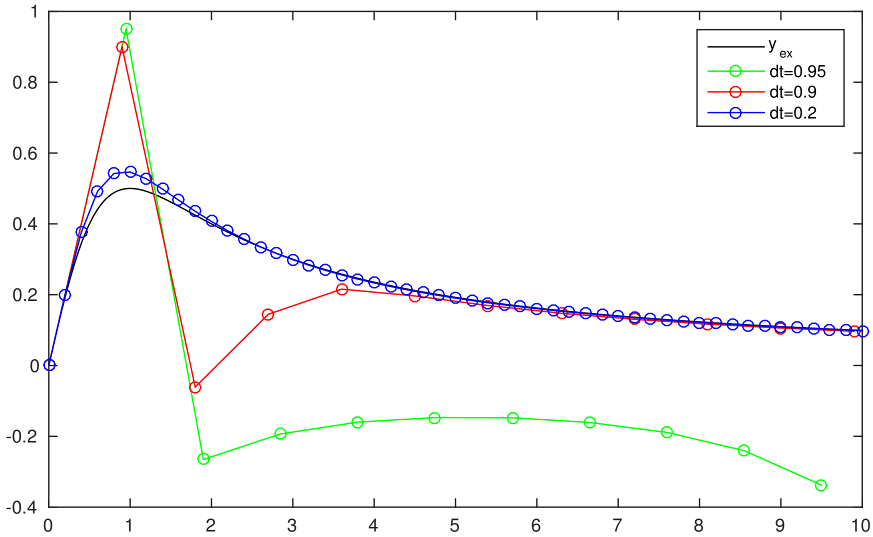


Figure 3: Evolution of the numerical solutions for different values of Δt.

```
for i=1:Nt-1
    u(i+1) = u(i)+ dt*feval(fun,t(i),u(i));
end
```

(b) We report in the file `Lab2Es3b.m` the Matlab instructions to solve the assigned problem in the interval $\mathcal{I} = (0, 10]$ with $\Delta t = 0.95$, $\Delta t = 0.9$ and $\Delta t = 0.2$.

The stability condition is

$$\Delta t < \frac{2}{\max_{t\in(0,10]} \left|\frac{\partial f}{\partial y}(t, y)\right|},$$

which gives $\Delta t < 1$ in our case. Indeed, in Figure 3 we notice that the accuracy of the numerical solution decreases for Δt approaching 1.