Numerical Analysis and Computational Mathematics

Fall Semester 2025 – CSE Section

Prof. Laura Grigori

Assistant: Israa Fakih

Session 12 – December 3, 2025

Ordinary differential equations

Exercise I (MATLAB)

Consider the Cauchy problem

$$\text{find } y \ : I \subset \mathbb{R} \to \mathbb{R} \qquad : \qquad \begin{cases} y'(t) = f(t, y(t)) & \text{for all } t \in I, \\ y(t_0) = y_0, \end{cases} \tag{1}$$

where $I = (t_0, t_f)$ is the integration interval, $f \ : I \times \mathbb{R} \to \mathbb{R}$ is a given continuous function, and $y_0 \in \mathbb{R}$ is the initial datum.
As a particular case of (1), the following model problem can be defined by setting $f(t, y) = \lambda\, y$ for some $\lambda \in \mathbb{R}$:

$$\text{find } y \ : I \subset \mathbb{R} \to \mathbb{R} \qquad : \qquad \begin{cases} y'(t) = \lambda\, y(t) & \text{for all } t \in I, \\ y(t_0) = y_0. \end{cases} \tag{2}$$

The exact solution of the model problem is $y(t) = y_0\, e^{\lambda\,(t - t_0)}$, for all $t \in (t_0, +\infty)$.

a) Write the MATLAB functions `forward_euler.m` and `heun.m` that implement the *forward Euler* and *Heun* methods for the solution of (1). The functions should output the vector of discrete times $\{t_n\}_{n=0}^{N_h}$ ($t_n = t_0 + n\, h$ for $n = 0, \ldots, N_h$) and the numerical solution $\{u_n\}_{n=0}^{N_h}$. Use the function `forward_euler_template.m` as template.

```matlab
function [ tv, uv ] = forward_euler( fun, y0, t0, tf, Nh )
% FORWARD_EULER Forward Euler method for the scalar ODE in the form
% y'(t) = f(t,y(t)),  t \in (t0,tf)
% y(0) = y_0
%
%  [ tv, uv ] = forward_euler( fun, y0, t0, tf, Nh )
%  Inputs: fun    = function handle for f(t,y), fun = @(t,y) ...
%          y0     = initial value
%          t0     = initial time
%          tf     = final time
%          Nh     = number of time subintervals
%  Output: tv     = vector of time steps (1 x (Nh+1))
```

```
%          uv     = vector of approximate solution at times tv
%

return
```

b) Use the functions `forward_euler.m` and `heun.m` to solve (1) for $f(t,y) = 1 - y^2$, $t_0 = 0$, $t_f = 5$, and $y_0 = \frac{e-1}{e+1}$. Set $N_h = 10$ and compare the numerical solutions with the exact solution $y(t) = \frac{e^{2t+1}-1}{e^{2t+1}+1}$.

c) Repeat point b) for the model problem (2), with $\lambda = -0.5$, $t_0 = 0$, $t_f = 15$, $y_0 = 1$, and $N_h = 10$.

d) Write the functions `backward_euler_modelproblem.m` and `crank_nicolson_modelproblem.m` that implement the *backward Euler* and *Crank-Nicolson* methods for the solution of (2). Use the function `backward_euler_modelproblem_template.m` as template.

```
function [ tv, uv ] = backward_euler_modelproblem( lambda, y0, t0, tf, Nh )
% BACKWARD_EULER_MODELPROBLEM Backward Euler method for the model problem
% ODE in the form
% y'(t) = lambda y(t),   t \in (t0,tf)
% y(0) = y_0
%
%  [ tv, uv ] = backward_euler_modelproblem( lambda, y0, t0, tf, Nh )
%  Inputs: lambda = real parameter (negative)
%          y0     = initial value
%          t0     = initial time
%          tf     = final time
%          Nh     = number of time subintervals
%  Output: tv     = vector of time steps (1 x (Nh+1))
%          uv     = vector of approximate solution at times tv
%

return
```

e) Repeat point c) using `backward_euler_modelproblem.m` and `crank_nicolson_modelproblem.m`.

f) Consider the setting of points c) and e). Compute the errors $e_n^{FE} = |y(t_n) - u_n^{FE}|$, $e_n^{BE} = |y(t_n) - u_n^{BE}|$, $e_n^{H} = |y(t_n) - u_n^{H}|$, and $e_n^{CN} = |y(t_n) - u_n^{CN}|$ corresponding to the forward Euler, backward Euler, Heun, and Crank-Nicolson solutions. Select n such that the computed errors correspond to the time $\bar{t} = 10$ for increasing values of the number of subintervals $N_h = 15, 30, 60, 120, 240, 480$. Plot the computed errors vs the size h of the subintervals. Deduce the convergence orders of the methods.

g) Repeat points c) and e) by setting $t_f = 40$ and $N_h = 9, 10,$ and 11. Discuss the results in terms of absolute stability of the numerical methods.

Exercise II (MATLAB)

Consider the Cauchy problem (1) introduced in Exercise 1. We assume that $\frac{\partial f}{\partial y}(t,y)$ exists for all $t \in I$ and for all y.

a) Write the MATLAB function `backward_euler.m` that implements the *backward Euler* method for the solution of (1). At each iteration, use the *Newton method* to solve for the next time step. To this aim, use the function `newton.m` from Series 3, with tolerance $tol = 10^{-10}$ and maximum number of iterations equal to 20. Use the function `backward_euler_template.m` as template.

```
function [ tv, uv ] = backward_euler( fun, dfun_y, y0, t0, tf, Nh )
% BACKWARD_EULER Backward Euler method for the scalar ODE in the form
% y'(t) = f(t,y(t)),  t \in (t0,tf)
% y(0) = y_0
%
% The Newton method is used to solve the nonlinear equation at each time
% step. The function newton.m is used.
%
%  [ tv, uv ] = backward_euler( fun, dfun_y, y0, t0, tf, Nh )
%  Inputs: fun    = function handle for f(t,y), fun = @(t,y) ...
%          dfun_y = derivative of f(t,y) w.r.t. y, dfun_y = @(t,y) ...
%          y0     = initial value
%          t0     = initial time
%          tf     = final time
%          Nh     = number of time subintervals
%  Output: tv     = vector of time steps (1 x (Nh+1))
%          uv     = vector of approximate solution at times tv
%

return
```

b) Set $f(t, y) = \alpha y \left(1 - \frac{y}{\beta}\right)$, with $\alpha, \beta > 0$ and $0 < y_0 < \beta$. The corresponding exact solution is $y(t) = \beta \frac{e^{\alpha(t-t_0)+\gamma}}{1+e^{\alpha(t-t_0)+\gamma}}$, with $\gamma := \log\left(\frac{y_0}{\beta - y_0}\right)$. Set $\alpha = \frac{\pi}{2}$, $\beta = \frac{\pi}{3}$, $y_0 = 0.4$, $t_0 = 0$, $t_f = 20$, and $N_h = 20$. Apply `forward_euler.m` and `backward_euler.m` to approximate the solution of the Cauchy problem. Compare the results with the exact solution $y(t)$.

c) Consider the Cauchy problem defined at point b). Recall the *heuristic* stability condition given in Remark 8.14 of the lecture notes for the forward Euler method: if f is smooth enough, $\frac{\partial f}{\partial y}(t, y(t)) < 0$ for all $t > t_0$, and

$$0 < h < \frac{2}{\sup_{t>t_0} \left|\frac{\partial f}{\partial y}(t, y(t))\right|},$$

then we can expect the forward Euler method to be absolutely stable. (This is *not* a rigorous implication.) Using this criterion, find the maximum size h_{max} of the subintervals that ensures, for $h < h_{max}$, absolute stability of the forward Euler method. Calculate the corresponding number of subintervals $N_{h_{max}}$. (*Hint*: note that the solution $y(t)$ satisfies $y(t) \in [y_0, \beta)$ for all $t \geq t_0$.)

d) Solve the Cauchy problem from point b) by means of the forward and backward Euler methods for different values of h around h_{max}. Compare the numerical solutions with the exact one.