

## Theme: Rocket launches and ODEs Part II

---

Rules:

- Each student should hand in individually completed solution.
- Use the provided answer sheet and add printouts of pictures and source code as requested.
- You may discuss the problem among fellow students. If you receive considerable help from someone, say so in your solutions.
- Do not copy solutions or code from others. Do not lend your solution or your code to other students.
- A correct solution submitted at the latest **December 13 at 12.00 (noon)** is worth 2 bonus point on the final exam. An almost correct solution submitted in time will also entitle to bonus points, provided the solution is corrected before the final exam.

---

1. Implement your own version of Heun's method to solve the system of ODEs

$$\begin{aligned} \mathbf{y}' &= \mathbf{f}(t, \mathbf{y}) & t > 0, \\ \mathbf{y}(0) &= \mathbf{y}_0. \end{aligned} \tag{1}$$

Make the implementation in a separate MATLAB function, `heun.m`. The function header should look like as below,<sup>1</sup> and the function should follow the described behavior.

```
function [t,y] = heun(f,tspan,y0,h)
%HEUN solve differential equation with Heun's method
% [T, Y] = HEUN(F,TSPAN,Y0,H) with TSPAN = [T0, TFINAL] integrates
% the system of differential equations y' = f(t,y) from time T0 to
% TFINAL with initial conditions Y0 employing a constant time step
% H. F is a function handle. For a scalar T and a vector Y, F(T,Y)
% must return a column vector corresponding to f(t,y). Each row in
% the solution array Y corresponds to a time in the vector T.
```

The above function follows the same standard as Matlab's build in ODE solvers. Note the dimensions of the matrices! Assume that system (1) is of dimension  $n$  (that is, the length of vectors  $\mathbf{y}$  and  $\mathbf{f}$  is  $n$ ) and that `heun` has been solved for  $N$  time steps. The output matrix  $\mathbf{Y}$  should then have the shape

$$\mathbf{Y} = \begin{pmatrix} -\mathbf{y}_0- \\ -\mathbf{y}_1- \\ \vdots \\ -\mathbf{y}_N- \end{pmatrix},$$

---

<sup>1</sup>The comment rows just after the function declaration are displayed when you type the command `help heun`. The header can be downloaded from the course homepage.

where each row is an  $n$ -vector. The first row contains the initial condition, and each of the following rows the numerical solution at times  $t_1, t_2, \dots$ . However, the Matlab function  $f$  that implements the right-hand side  $\mathbf{f}(t, \mathbf{u})$  must return a *column* vector! It may seem unpractical to output the solution vectors row wise when the right-hand-side vector must be a column vector, but this is the standard chosen by Matlab. **Warning:** the prime operator  $'$  used as  $A'$  does not only transpose  $A$ , it also complex conjugates all elements. To only transpose  $A$ , without complex conjugation, use  $A.'$  or  $\text{transpose}(A)$ .

2. Test your implementation of Heun's method on the equation  $y' = \alpha y$ , where  $\alpha = -0.1 + 5i$ , from time  $T_0 = 0$  to time  $T = 30$  with initial condition  $y(0) = 1$ . Plot your numerical solution together with the exact solution.
3. Again, integrate the equation  $y' = \alpha y$ , where  $\alpha = -0.1 + 5i$ , from time  $T_0 = 0$  to time  $T = 30$  with initial condition  $y(0) = 1$  using your implementation of Heun's method. Try using different time steps  $h$  to
  - (a) experimentally find the stability limit,
  - (b) experimentally estimate the order of accuracy by computing the absolute value of the difference between your numerical solution and the exact solution at time  $T = 30$ .
4. Use your methods as well as MATLAB's build in function `ode45` to find the maximum height the rocket reaches with the strategy to burn off fuel as fast as possible, that is

$$\mu = \begin{cases} 10 \text{ kg/s} & \text{if } t \leq 90 \text{ s,} \\ 0 \text{ kg/s} & \text{if } t > 90 \text{ s.} \end{cases}$$

5. Try to find a better fuel burning strategy. Recall that there is only 900 kg of fuel at the launch! Describe your strategy, how high does your rocket fly?