Lab and Exercises I

Exercise I.

Install Anaconda alongside with Python, Julia, and the Jupyter notebook.

Exercise II.

(i) Approximate the projectile ODE with Newtonian drag,

$$\begin{aligned} \partial_t x(t) &= v_x(t), & \partial_t v_x(t) &= -\frac{1}{2}\mu\|\vec{v}\|v_x \\ \partial_t y(t) &= v_y(t), & \partial_t v_y(t) &= -g - \frac{1}{2}\mu\|\vec{v}\|v_y \\ x(0) &= x_0, & v_x(0) &= v_{x,0}, \\ y(0) &= y_0, & v_y(0) &= v_{y,0}, \end{aligned}$$

with the forward Euler time-stepping scheme. Run your simulation for as long as $y(t^*) > 0$ and visualize the trajectories. Test your implementation by computing the trajectory for different initial conditions

$$x_0 = 0,\ y_0 = 0,\ v_{x,0} = s_0\,\cos(\alpha),\ v_{y,0} = s_0\,\sin(\alpha),$$

by varying angle α, initial speed s_0 and time step sizes τ.

(ii) Can you come up with an algorithm that finds the optimal angle α for given initial speed s_0, gravity constant g and drag coefficient μ (by maximizing the horizontal distance travelled)?

Exercise III.

Approximate the Lorenz system,

$$\partial_t x(t) = \sigma\big(y(t) - x(t)\big), \quad \partial_t y(t) = x\big(\rho - z(t)\big), \quad \partial_t z(t) = x(t)y(t) - \beta z(t),$$

with a time-stepping method of your choosing. Here, we have set $\sigma = 10$, $\beta = 8/3$, $\rho = 28$. For a chosen time step size τ first compute the approximation $\big\{x^n_\tau, y^n_\tau, z^n_\tau\big\}_0^N$ and then half the step size and compute the approximation $\big\{x^n_{\tau/2}, y^n_{\tau/2}, z^n_{\tau/2}\big\}_0^{2N}$. By comparing x^N_τ with $x^{2N}_{\tau/2}$ you can assess the quality of your approximation: What is the largest final time T that you can use where you still have a reasonable quality?