Final Project: Basket Ball

NUMA01: Computational Programming with Python

Malin Christersson, Robert Klöfkorn

This assignment has 5 tasks.

The goal of the final project is to experience programming in a group. Proceed in the following way

- discuss the mathematical part of the project in the group first until you fully understand the problem
- divide the problem in subproblems
- discuss how the different parts should be tested
- prepare a max 20 min presentation which should present the mathematical background, the organization of your work, your solutions and maybe alternative attempts, which you decided to reject.

Bring your Laptop or an USB stick for the presentation.

Background

We consider a basket ball player who wants to throw a ball into the basket. Which initial angle and velocity is required to reach the basket?
First some facts: A basket ball has a diameter of $d = 0.24\,\mathrm{m}$ and weighs $m = 0.6\,\mathrm{kg}$. Its air resistance coefficient is $c_w = 0.45$, the air density is assumed to be $\rho = 1.23\,\mathrm{kg/m^3}$ and the basket is mounted at a height $y_\mathrm{B} = 3.05$ m at a horizontal distance from the player $x_\mathrm{B} = 2.00$ m. Finally we need the gravitational constant $9.81\ \mathrm{m/s^2}$.
The motion of the ball is described by a differential equation of second order

$$
\begin{aligned}
m\ddot{x}(t) &= -F(s)\cos\alpha(t) && (1)\\
m\ddot{y}(t) &= -F(s)\sin\alpha(t) - mg, && (2)
\end{aligned}
$$

where s is the speed of the ball, $s^2 = \dot{x}^2 + \dot{y}^2$ and $F(v)$ is the air resistance force:

$$F(s) = \frac{1}{2}\rho c_w \frac{\pi}{4} d^2 s^2$$

and $\alpha = \arccos(\dot{x}/s)$.
It can be reformulated to a first order differential equation

$$\begin{aligned}
\dot{x} &= v_x && \text{(3a)}\\
\dot{y} &= v_y && \text{(3b)}\\
m\dot{v}_x &= -F(s)\cos\alpha && \text{(3c)}\\
m\dot{v}_y &= -F(s)\sin\alpha - mg. && \text{(3d)}
\end{aligned}$$

We assume initial conditions $x_0 = 0$ m, $y_0 = 1.75$ m and an initial speed $s_0 = 9$ m/s. The initial throwing angle α_0 with $\dot{x}_0 = s_0\cos\alpha_0$ and $\dot{y}_0 = s_0\sin\alpha_0$ is unknown. If this were given, the position of the ball at any time could be determined by solving the differential equation. Numerically this could be done by applying the explicit Euler method with a sufficiently small step size.
Here the final position $x_\mathrm{B}, y_\mathrm{B}$ is given instead of the initial velocity data. The time t_B when this position is reached is unknown.
Such a problem is called a free boundary value problem. By introducing scaled time $\tau = t/t_\mathrm{B}$ and introducing a new unknown $z = t_\mathrm{B}$ we can transform this into a classical boundary value problem

$$\begin{aligned}
\dot{x}(\tau) &= v_x(\tau)z(\tau) && \text{(4a)}\\
\dot{y}(\tau) &= v_y(\tau)z(\tau) && \text{(4b)}\\
\dot{z}(\tau) &= 0 && \text{(4c)}\\
m\dot{v}_x(\tau) &= -F(s(\tau))z(\tau)\cos(\alpha(\tau)) && \text{(4d)}\\
m\dot{v}_y(\tau) &= -z(\tau)(F(s(\tau))\sin(\alpha(\tau)) - mg) && \text{(4e)}
\end{aligned}$$

with the boundary conditions

$$x(0) = 0, y(0) = 1.75, \sqrt{\dot{x}_0^2 + \dot{y}_0^2} = s_0, x(1) = x_\mathrm{B}, y(1) = y_\mathrm{B}.$$

The idea (not the best) how to solve such a problem is quite simple: We just guess an initial value for z, z_0, and an initial angle α_0 and solve this problem as if it were an initial value problem, using the explicit Euler method. The result we obtain clearly depends on these guesses. Let's denote the result by

$$x(1; z_0, \alpha_0) \quad y(1; z_0, \alpha_0).$$

We require now that the initial conditions fulfill the conditions

$$\begin{aligned}
x(1; z_0, \alpha_0) - x_\mathrm{B} &= 0\\
y(1; z_0, \alpha_0) - y_\mathrm{B} &= 0.
\end{aligned}$$

This defines a nonlinear equation system

$$G(z_0, \alpha_0) = \begin{pmatrix} x(1; z_0, \alpha_0) - x_\mathrm{B} \\ y(1; z_0, \alpha_0) - y_\mathrm{B} \end{pmatrix} = 0, \tag{5}$$

which can be solved by Newton's method.
A good initial guess may be obtained graphically by doing some experiments with different initial values and ODE solvers in Scipy.
The following tasks are guidelines to what you need to do. Feel free to do more, or modify them, as long as the spirit of the project remains the same.

Task 1

The explicit Euler method for solving $y'(t) = f(t, y(t))$ over $t \in [0, T]$ using N steps introduces a grid $t_i = i\frac{T}{N} = ih$ with corresponding approximations $u_i \approx y(t_i)$. The approximations are related through

$$\begin{aligned} &u_i = u_{i-1} + hf(t_{i-1}, u_{i-1}), \quad i = 1, \dots, N, \\ &u_0 = y(0). \end{aligned}$$

The reasoning behind this is just approximating the derivative by a small difference quotient: $y'(t_i) \approx \frac{y(t_{i+1}) - y(t_i)}{t_{i+1} - t_i}$.

Implement the explicit Euler method in a function which has f, T and N as input. It should return arrays of both the t_i and the u_i.

Task 2

Implement a function or class for solving the boundary value problem (4) as an initial value problem, ignoring the boundary condition at $\tau = 1$. The function needs to take z_0 and α_0 as input (at least). It is enough if your code uses the explicit Euler method, but you can experiment with other solvers if you want, for example the Adams-Bashforth method from Exercise 1.

Task 3

Implement a function that plots the basket ball trajectory.

Task 4

Implement a function that uses Newton's method to solve a nonlinear equation system.

Task 5

Apply the last function to the system given in (5) to find the optimal angle α_0 and the corresponding t_B. Plot the final result as well as the intermediary trajectories.