Workshop 4

Topics: Simulation of differential equations.

In this laboratory we will simulate a physical system and display our results of using videos. This type of simulators have a great pedagogical value because they allow anyone (even without knowledge in physics or mathematics) to observe the behavior of physical systems and how this behavior is influenced by the variation of some parameters (e.g., angle of throw in parabolic movement, gravity in free fall, etc.).

Problem 1

In Matlab we can create videos from figures. A simple way to do it is shown in the following link: `https://www.youtube.com/watch?v=mvXJh_TDKG8&ab_channel=CodingLikeMad`.

Create a video where a point moves from x=0 to x=1. To make the dot look bigger you can use `plot(x,y,'Marker','o','MarkerSize',10)`, where x and y are the coordinates of the point.
Note: To play the video, type in the command window `implay('video name')`

Problem 2

Many models of physical systems result in differential equations, so it is important to know numerical methods for solving differential equations using Matlab. To solve an ordinary differential equation numerically you can use the ODE solvers available in Matlab. Using these solvers is easy. To illustrate how they work consider the following example:
Find the solution of the following differential equation from t_0 to t_f,

$$\frac{dx}{dt} = cos(x) + t, \quad x(0) = x_0.$$

The first thing we need to do is to create a function (it can be in a separate script) where we specify the differential equation to be solved. This function must have the following structure:

```
function xp = ODEx (t,x)
xp = cos(x) + t
end
```

The name of the function, in this case ODEx, is arbitrary. You must save the function in the same folder as the mainscript. Note that the first argument of the function is the independent variable (time) and the second is the dependent variable (x). Also, the result of the function, xp, is the value of the derivative of the dependent state variable with respect to the independent variable, which is given by the differential equation above, the one we want to solve numerically. This being the case, note that, in principle, this

structure only allows solving first order differential equations.

Later, in the main script, use the following command line to solve the equation.

```
t0 = 0;% initial time
tf = 10;% final time
x0 = 1;% initial condition
[t,x] = ode45(@ODEx,[t0,tf],x0);
plot(t,x)
```

The numerical solution is the function that relates x to t. In this case, we have invoked the ode45 solver, which uses the fourth-order Runge-Kutta method to solve the differential equation. However, there are other ODE solvers available, see:
`https://la.mathworks.com/help/matlab/math/choose-an-ode-solver.html`
Implement the described steps in Matlab and upload to e.aulas the graph that shows the solution of the proposed differential equation

Problem 3

Hypatia is a little girl, very curious and restless. Every morning she brings lemonade to her school in a big bottle. Tired of getting up so early, she wants to optimize the filling of lemonade in her bottle and for this she wants to know what to use: whether to invest in a funnel (cone) or simply use a plastic cup (cylinder) with a hole in the bottom. With her savings, she has hired you to help her decide on the best option. However, Hypatia is incredulous, if you give her a solution, she will not be convinced of it until she sees with her own eyes what is the most convenient. Since Hypatia is still small, to show her a graph of the liquid level against time in each one of the vessels would be unsuccessful. So the best option is to make a video that shows the two types of containers (cone and cylinder) being emptied, in this way, it will be evident which corresponds to the best decision.

Make a video of a conical and a cylindrical tank of the same volume being emptied through a small hole (of the same radius) that is located at the bottom of the containers (of course, you do not have to recreate this situation really but in a simulation). The following hints can help you with this task:

Hint 1: Propose a model using differential equations for the emptying of each of the tanks. For your model, consider the following: emptying/filling a tank containing a volume of liquid V, can be modeled by the following differential equation:

$$\dot{V} = f_{in} - f_{out},$$

where $\dot{V}$ is the rate of change of the liquid volume (the derivative of the volume with respect to time); f_{in} is the liquid inlet flow to the tank; and f_{out} is the outflow of liquid from the tank. If the tank has a hole where liquid comes out, then $f_{out} = av$, where a is the area of the hole and v is the speed at which the liquid comes out. This velocity obeys

Torricelli's Law.

According to Torricelli's law, the velocity of the liquid leaving a hole at the bottom of a tank is the same speed reached by a particle dropped freely (in a vacuum) from the level of the liquid to the site where the hole is.

Hint 2: To determine the height of the water at each instant of time you can use a Matlab ODE solver.

Hint 3: You can use Matlab's cylinder function to graph both the cylinder and the cone. To give an idea of how this function works, it is recommended to look at the Matlab help center (`mathworks`). Here we present two examples that may also help you:
Cylinder example

```
r = 1;% radius
h = 2;% height
x = 1;% x coordinate of the center of the base of the cylinder
y = 1;% y coordinate of the center of the base of the cylinder
[X,Y,Z] = cylinder (r);
surf(X+x, Y+y, Z*h, 'facecolor', 'b','FaceAlpha',0.5,'Edgecolor','none')
```

facecolor is the color of the cylinder, facealpha the transparency, edgecolor the color of the lines.

Cone example

```
r = 2;% radius
h = 3;% height
x = 2;% x coordinate of the center of the base of the cone
y = 4;% y coordinate of the center of the base of the cone
[X,Y,Z] = cylinder ([0,r]);
surf(X+x, Y+y, Z*h, 'facecolor', 'b','FaceAlpha',0.5,'Edgecolor','none')
```

Hint 4: Before plotting your figures, it is convenient to fix the axes and the view. If this is not done, in each graph Matlab will optimize the view and you will see sharp transitions in the video. This can be done with the following lines of code:

```
figure()
xlim([-2,2])% fixes the x-axis of the figure
ylim([-2,2])% sets the y-axis of the figure
zlim([0,10])% fixes the z-axis of the figure
view(60,30)% sets the perspective of the figure
surf(X+x, Y+y, Z*h, 'facecolor', 'b','FaceAlpha',0.5,'Edgecolor','none')
```

Hint 5: To delete a figure without deleting the axes, use the command: `cla`
Hint 6: Calculate the height of the water in time intervals of equal length, so your simulation will run in "real time".