MCS 320 Project Three: Problems of Pursuit

The goal of this project is to use MATLAB or Octave to study problems of pursuit via solving Ordinary Differential Equations (ODEs). We start with a classical problem.

1. The Tractrix Problem

The independent variable in our problem is time $t$. Given are coordinate functions $(x_1(t), x_2(t))$ that define the path of a tractor. The tractor is connected to a trailer by a rigid bar of length $L$. Given $(x_1(t), x_2(t))$ and $L$, we want to compute the path of the trailer, that is: to find values for the unknown coordinate functions $(y_1(t), y_2(t))$.

At all times does the bar between tractor and trailer stay rigid and

$$(y_1 - x_1)^2 + (y_2 - x_2)^2 = L^2$$

is an identity for all values of time $t$. The trailer moves always in the direction of the rigid bar, or equivalently, its velocity vector $(\dot{y}_1, \dot{y}_2)$ is parallel to the direction of the bar, expressed as

$$\left( \begin{array}{c} \dot{y}_1 \\ \dot{y}_2 \end{array} \right) = \lambda \left( \begin{array}{c} y_1 - x_1 \\ y_2 - x_2 \end{array} \right), \quad \lambda > 0.$$

To obtain a system of first-order differential equations, we need to find an expression of the velocity vector of the trailer. The velocity vector of the trailer is the projection of the velocity vector of the tractor onto the direction of the rigid bar. For

$$u = \frac{(y_1 - x_1, y_2 - x_2)}{||(y_1 - x_1, y_2 - x_2)||} \quad \text{and} \quad v = (\dot{x}_1, \dot{x}_2),$$

we compute the projection of the velocity vector as $(v^T u)u$. Then the system of first-order equation that defines the path of the trailer is given by

$$\left( \begin{array}{c} \dot{y}_1 \\ \dot{y}_2 \end{array} \right) = (v^T u)u,$$

with $u$ the normalized vector of the direction of the bar and $v$ the velocity vector of the tractor.

**Assignment One.** Consider a circling tractor with its path defined by $(\cos(t), \sin(t))$, starting at $(1, 0)$. Assume at $t = 0$ the trailer is at $(2, 0)$, so $L = 1$. Set up the differential equation and define .m files to calculate the path of the trailer for $t$ ranging from 0 to 100.

For $t \in [0, 100]$, make a plot of the tractor curve (in red), the path of the trailer (in green) and at regular time intervals the rigid bar between tractor and trailer (in blue).

**Assignment Two.** Consider $(t, 5 \sin(t))$ as path for the tractor and let the trailer start at $(0, 10)$. Compute the path of the trailer for $t \in [0, 20]$. Make a plot as in assignment one.

**Assignment Three.** Suppose the tractor follows a path defined by $(5 \cos(t), 5 \sin(t))$ and the trailer starts at $(0, 10)$. Compute the path of the trailer for $t \in [0, 100]$ and make a plot as in assignment one.
2. Solving ODEs in MATLAB and Octave

A first important difference between MATLAB and Octave is in the setup:

**MATLAB:** \( \frac{dy}{dx} = f(x,y), \quad y(x_0) = y_0, \)

**Octave:** \( \frac{dy}{dx} = f(y,x), \quad y(x_0) = y_0. \)

In MATLAB, the independent variable \( x \) is the first argument of \( f \), whereas in Octave, the dependent variable \( y \) is the first argument of \( f \). Our test problem is \( f = y, \quad y(0) = 1, \) for \( x \in [0,1] \).

In MATLAB, we use the command **ode45**:

```
matlab >> xspan = [0 1]; yzero = 1; % x in interval [0,1], y(0) = 1
matlab >> f = inline('y','x','y'); % define f(x,y) = y, solve dy/dx = f
matlab >> [x y] = ode45(f,xspan,yzero); % solve initial value problem
matlab >> y(end) - exp(1)
```

The output of the last command shows the accuracy of the computed solution, as the exact solution for the test problem is the exponential function \( e^x \). To plot, do `plot(x,y)`.

In Octave, the command to solve ODEs is **lsode**:

```
octave >> f = inline('y','y','x'); % dependent variable comes first!
octave >> xspan = [0 1]; yzero = 1; % range and initial condition
octave >> y = lsode(f,yzero,xspan); % initial condition before range
octave >> y(end) - exp(1)
```

Consistent with swapping the order of dependent and independent variables in the definition of \( f \), in **lsode**, the initial condition comes before the range of \( x \). In the output of **lsode** (in contrast to the output of **ode45**), the range of \( y \) is always the same as the range of \( xspan \). If we would like to plot the solution for \( x \) in the range 0:0.01:1, then we must define \( xspan \) also as 0:0.01:1.

A simple model of a pendulum is given by the following second-order equation:

\[
\frac{d^2\theta(t)}{dt^2} + \sin(\theta(t)) = 0 \quad \text{equivalent to} \quad \begin{cases} 
\frac{dy_1(t)}{dt} = y_2(t) \\
\frac{dy_2(t)}{dt} = -\sin(y_1(t)) 
\end{cases}
\]

where \( y_1 = \theta \).

The equivalent system at the right is the input to MATLAB or Octave.

Solving this problem in MATLAB for initial conditions \( y_1(0) = 1 \) and \( y_2(0) = 0 \) goes as:

```
matlab >> f = inline(['[y(2); -sin(y(1))]',',','t',',:)']); % define right-hand side f
matlab >> yzero = [1 0]; tsdp = [0 10]; % initials and range
matlab >> [t y] = ode45(f,tsdp,yzero); % solve the ODE
matlab >> plot(t,y(:,1)) % plot the values for theta
```

In Octave, the corresponding command sequence is:

```
octave >> f = inline(['[y(2); -sin(y(1))]',',','y',',:)']); % y comes before t
octave >> yzero = [1 0]; tsdp = [0 10]; % initials and range
octave >> y = lsode(f,yzero,tsdp) % solve the ODE
```

If the function \( f \) is defined in a MATLAB or Octave script, then we must put quotes around \( f \), that is: `'f'` is the first argument of both **ode45** and **lsode**.
3. Modeling Predator-Prey Pursuits

Instead of tractor and a trailer remaining at a fixed distance from each other, we now consider the tractor as a prey fleeing from a predator.

The path of the prey is given via two coordinate functions, denoted by \((x_1(t), x_2(t))\). The coordinates of the predator are \((y_1(t), y_2(t))\). The predator again moves in the direction of the prey, as before we compute the projection of the velocity vector of the prey onto the direction of pursuit.

The normalized direction vector of the predator is

\[
w = \frac{(x_1 - y_1, x_2 - y_2)}{||(x_1 - y_1, x_2 - y_2)||}.
\]

Modeling the predator moving towards the prey is then done via

\[
\begin{pmatrix}
\dot{y}_1 \\
\dot{y}_2
\end{pmatrix} = s(t)w
\]

where \(s(t)\) is a function in \(t\) that regulates the speed of the predator.

**Assignment Four.** Consider a prey moving at constant speed as \((8t, 0)\). At \(t = 0\), the predator starts at \((20, 20)\), and with constant speed \(s(t) = 10\). Compute the trajectory of the predator.

Find the right time interval till the predator meets the prey. Plot the path of the prey in red and that of the predator in green. Adjust the axis of the plot window appropriately.

**Assignment Five.** The functions \((10 + 20 \cos(t), 20 + 15 \sin(t))\) define an elliptic path for the prey. For a predator as in the previous assignment, starting at \((20, 20)\) at with constant speed \(s(t) = 10\), will the predator ever catch the prey?

Compute trajectories and make plots to answer the question.

**Assignment Six.** For the same path of the prey as in the previous assignment and same initial position for the predator. Approximate the value of the constant speed of the predator that catches the prey. Find suitable ranges for the time interval and make an appropriate plot to illustrate your answer.

4. The deadline is Friday 30 April 2010 at 10AM

Bring *your* solution to the project to class. The *your* is emphasized to stress that your solution is the result of an *individual* effort. Collaborations are *not* permitted.

Please to not mix MATLAB and Octave scripts. You are free to choose either MATLAB or Octave, but you may not solve one assignment with MATLAB and another with Octave. Every script must contain a line documenting whether it runs under MATLAB or Octave.

The solution to this project consists in two parts:
1. A print out of all the .m files that you bring to class.
   In addition, make plots and a .txt file created with diary. Include a summary of your experiments, description of your observation and explanation of the results.
2. Email your scripts as an attachment to me so I can verify your runs.

If you have questions or difficulties, feel free to come to my office for help.