Numerical Solution of ODEs

Due at 4:00pm on Friday October 25, 2019

What you need to get

- **https://www.overleaf.com/read/vztpkwrpmtcn**: (optional) Overleaf template
- `YOU_a3-q3.ipynb`: Jupyter notebook for Q3
- `YOU_a3-q4.ipynb`: Jupyter notebook for Q4
- `YOU_a3-q5.ipynb`: Jupyter notebook for Q5

What you need to know

This assignment uses a very simple class called `MyODESolution`. Its definition is in the notebook for question 3.

What to do

1. **[2 marks]** Prove that

$$x(t) = 3e^t - 2e^{2t}$$
$$y(t) = 3e^t - 4e^{2t}$$

   is a solution to the IVP

$$x'(t) = y(t)$$
$$y'(t) = -2x(t) + 3y(t)$$
$$x(0) = 1\,, \quad y(0) = -1$$

   Show your work.

2. **[6 marks]** Compute **two** Euler steps of size $h = 0.1$ for the initial value problem

$$\frac{dx}{dt} = xy - t + 2$$
$$\frac{dy}{dt} = \frac{3x}{y} + 5\,t\,x$$
$$x(1) = 2\,, \quad y(1) = -3$$

3. **[10 marks]** Complete the Python function `MyODE`,

```
sol = MyODE(de, tspan, y0, rtol=0.05, event=[])
```

   See the function's documentation for a description of the inputs. Your function should have the following features:

   (a) **Modified Euler:** Implement Modified Euler's method.

(b) **Adaptive Time-Stepping:** Do adaptive time-stepping so that your estimate of the local relative error is less than `rtol` for each step. If the error is less than or equal to `rtol`, increase the time step by 20% for the next step. If the error is greater than `rtol`, then reduce your step size by 50% and repeat the step.

(c) **Events Function:** Stop when an event function returns a negative value.

(d) **Interpolate Terminal Event:** If the integration ends because an event is triggered, your function should linearly interpolate between the last two points to find a more accurate estimate for the time of the event. Then it should interpolate the state at that new, interpolated end time. You may assume that the state and the event function both change linearly between the last two time steps.
(*Hint: Think similar triangles.*)

4. **[8 marks]** A fox is chasing a rabbit on a field with an electric fence along one side. Suppose that $\mathbf{f}(t) = (f_x(t), f_y(t))$ is the location of the fox at time t (in metres), and $(r_x(t), r_y(t))$ is the location of the rabbit (also a function of time). Suppose the fox always *accelerates* toward the rabbit. Let the vector pointing from the fox to the rabbit be $\mathbf{a}(t, f) = (a_x(t, f), a_y(t, f))$, as shown in the diagram.

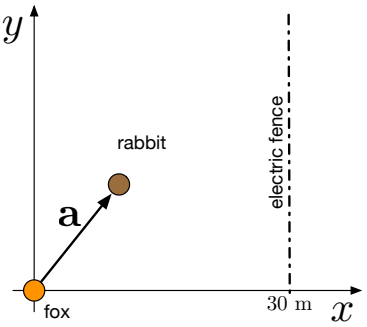
Then, the initial value problem governing the fox's movements is,

$$f_x'' = A\frac{a_x(t,\mathbf{f})}{\|\mathbf{a}(t,\mathbf{f})\|} - \left(\|(f_x', f_y')\| - S\right) f_x' \qquad (1)$$

$$f_y'' = A\frac{a_y(t,\mathbf{f})}{\|\mathbf{a}(t,\mathbf{f})\|} - \left(\|(f_x', f_y')\| - S\right) f_y' \qquad (2)$$

$$f_x(0) = 0\,, \quad f_y(0) = 0 \qquad (3)$$

$$f_x'(0) = 0\,, \quad f_y'(0) = 0 \qquad (4)$$



where A and S are both positive scalar constants. Generally, A corresponds to the rate of acceleration of the fox, and S limits the fox's maximum speed. The electric fence follows the line $x = 30$ (metres).

(a) Convert the system of differential equations to a system of first-order differential equations. Show your work, and clearly state the new system. Place your solution in a markdown box in the notebook using LaTeX.

(b) Complete the function `fox_dynamics` in the notebook so that it computes the dynamics function for your system of differential equations.

(c) The notebook includes the function `rabbit(t)`, which gives the rabbit's position as a function of time. Complete the event functions `capture` and `fence` to implement the following two events:

i. `capture`: The first event is the fox catching the rabbit. You may assume that the fox catches the rabbit if it comes within 10 cm of the rabbit.

ii. `fence`: The second event is the fox hitting the electric fence.

(d) Run some simulations of your system using parameter values in the range $S \in [2, 2.5]$, and $A \in [4, 7]$. You can either use your implementation of `MyODE` with a tolerance of 0.01, or you can use `solve_ivp`. If you decide to use your version of `MyODE`, just copy the function (as well as the definition of the class `MyODESolution`) into the notebook.

Find parameter values that result in each of the following outcomes:

i. The fox hits the fence.

ii. The rabbit escapes.

iii. The fox catches the rabbit.

For each case, plot the rabbit's trajectory and the fox's trajectory on the same plot, along with the fence.

5. **[4 marks]** Consider the IVP,

$$\frac{dy}{dt} = \frac{4}{1+t^2}$$
$$y(0) = 0$$

(a) Prove that $y(t) = 4 \arctan t$ is a solution to the IVP. (Don't forget about the initial state.) Place your solution in a markdown block in the notebook.

(b) Note that $\arctan 1 = \frac{\pi}{4}$. Thus, $y(1) = \pi$. Add code to your notebook to compute an estimate of π by numerically solving the IVP. You may use your own `MyODE` function (by copying it), or SciPy's `solve_ivp` function.

(c) Calculate and display the relative error of your estimate to show that it is less than 10^{-8}.

What to submit

You must submit a series of PDF documents to Crowdmark. For each coding question, export your jupyter notebook to PDF. When a proof or manual calculation is requested, you can typeset your solution (eg. using LaTeX or Word), or write your solution by hand. In either case, your solution should also be submitted as a PDF. **If you submit photos of handwritten work, it is your responsibility to ensure that they are legible.**