Numerical Solution of ODEs

Due at 11:59pm on Sunday February 19, 2023

What you need to get

- **https://www.overleaf.com/read/vztpkwrpmtcn**: (optional) Overleaf template
- `YOU_a2-q2.ipynb`: Jupyter notebook for Q2
- `YOU_a2-q3.ipynb`: Jupyter notebook for Q3

What to do

1. [20 marks] Compute **two** Euler steps of size $h = 0.2$ for the initial value problem

$$\frac{dx(t)}{dt} = 4x(t) - 2y(t) + t$$
$$\frac{dy(t)}{dt} = 3x(t) + 5t$$
$$x(1) = 1 \,, \quad y(1) = 1$$

   Show your work.

2. [50 marks] You are taking part in a golf driving-range competition to see who can hit the ball farthest. The distance that counts is where the ball lands on its *third* bounce.

   In this task, you will write your own ODE solving suite that performs **Modified Euler** (also known as **Improved Euler**) time stepping, and use the suite to solve for the trajectory of a golf ball, accounting for air resistance. The system of differential equations that governs the motion of a projectile, like a golf ball, is

$$x''(t) = -\,K\,x'(t)$$
$$y''(t) = -\,g - K\,y'(t) \,,$$

   where g is $9.81\,\mathrm{m/s^2}$, and K is a coefficient of air resistance. You may assume that K always equals 0.3 for this task.

   (a) Complete the Python function `MyOde`,

```
t, y = MyOde(de, tspan, y0, h, event)
```

   that numerically solves the IVP using Modified Euler time stepping using a fixed time step of `h`. See the help documentation in the supplied code for an explanation of the input and output variables. Importantly, your function should ignore the event function for the first time step. Note that if `MyOde` detects the occurrence of an event, it 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.

   (b) Formulate the system of ODEs above into a first order system, and create a Python function called `projectile` that fits the prototype of the dynamics function described in the `MyOde` help.

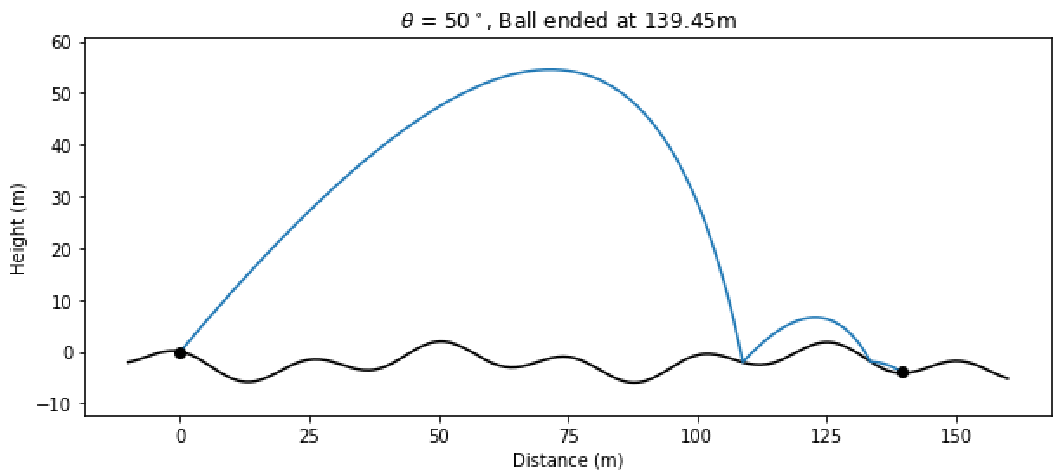
   (c) The driving range has rolling hills, so that the level of the ground is a cosine function,

$$\text{Ground}(d) = 2\left(\cos\left(\frac{d}{4}\right) - \sin\left(\frac{d}{11}\right) - 1\right)$$

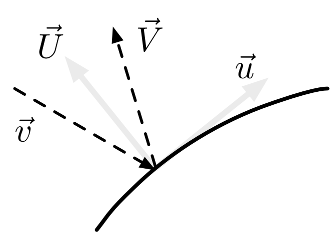
where d is the horizontal distance from the golf tee. The supplied function, `Ground`, evaluates this for you. Write a simple event function that changes sign as the golf ball hits the ground. Your event function should be continuous, not a step function. Call your event function `projectile_events`. This function should also fit the prototype described in the `MyOde` help.

The notebook calls your ODE suite to numerically solve the above golf-ball trajectory IVP and plots the resulting trajectory (including the level of the ground). The initial state is set so that the golf ball starts at the origin, and has an initial speed of $58\,\mathrm{m/s}$ at an angle of $\theta°$ from the ground.

A "flight" is the motion of the ball between contacts with the ground. For example, the first flight starts when the ball is initially hit, and ends when the ball next hits the ground. The second flight starts at the bounce from the first strike, and ends when the ball strikes the ground for the second time. The figure below shows the ball's trajectory over 3 flights.



When the ball bounces, its velocity vector is abruptly changed. The figure below shows the geometry of a bounce, with the ball's incident velocity vector ($\vec{v}$) and its velocity vector after the bounce ($\vec{V}$), based on unit vectors that are tangent to the ground ($\vec{u}$) and normal to the ground ($\vec{U}$). You can use the supplied function `GroundSlope` to get the slope of the ground, and use that to construct $\vec{u}$ and $\vec{U}$. Note that the bounce speed is only 85% of the incident speed.



$$\vec{V} = 0.85\left((\vec{v}\cdot\vec{u})\vec{u} - (\vec{v}\cdot\vec{U})\vec{U}\right)$$

(d) Add some code to the notebook to simulate the first 3 flights of the ball (as shown in the figure above). Note that each flight is simulated by a call to `MyOde`, with the initial position and velocity determined by the bounce from the previous flight.

(e) Re-run the code for different θ-values, and determine the angle (to the nearest degree) that has the ball end (on its 3rd ground strike) the furthest horizontal distance from the golf tee. Plot the trajectory for your chosen θ-value.

3. [30 marks] The police arrive at the home of Robert Durst, the trading network leader, to arrest him for questioning. However, Robert Durst is found dead in his home, strangled to death. Paramedics

record his core body temperature to be 25.0°C at 11:15pm on June 12. Normal body temperature is 37.5°C.

The forensic technicians also sample the concentration of two blood-borne bacterial populations, labelled A and B. In a living person, these populations are held constant at 1 unit per cubic millimetre of blood, denoted $1\frac{\text{u}}{\text{mm}^3}$. But after a person dies, these bacterial populations begin to flourish since the body's immune system is no longer functioning. The populations (in units $\frac{\text{u}}{\text{mm}^3}$) are governed by the differential equations

$$\frac{dA}{dt} = \begin{cases} 0.0008(T-29)^2\left(1-e^{0.08(T-45)}\right)A(30-A) & \text{for } 29 \leq T \leq 45 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$\frac{dB}{dt} = \begin{cases} 0.001(T-17)^2\left(1-e^{0.05(T-32)}\right)B(20-B) & \text{for } 17 \leq T \leq 32 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

At the time of discovery, the concentration of bacteria A was $6.6\frac{\text{u}}{\text{mm}^3}$, and $11.8\frac{\text{u}}{\text{mm}^3}$ for bacteria B. Note that the active metabolism of these bacterial populations heats the body slightly. Taking this metabolic heating into consideration, the body's core temperature follows the differential equation

$$\frac{dT}{dt} = -0.1\Big(T - T_a(t)\Big) + \frac{A+B}{100} \tag{3}$$

where T_a is the ambient air temperate (in °C), t is measured in hours, and A and B are the concentrations of bacteria A and B, respectively (in units of $\frac{\text{u}}{\text{mm}^3}$).

The ambient temperature in Mr. Durst's home is controlled by an automatic thermostat; it's set to maintain 22°C between the hours of 7:00am and 7:00pm, and maintain 16°C from 7:00pm until 7:00am the next morning. It takes 30 minutes for the temperature to increase from 16°C to 22°C in the morning (ie. from 7:00am until 7:30am), but takes about 2 hours to steadily drop from 22°C to 16°C at night (from 7:00pm until 9:00pm). You may assume these temperature changes are linear in time.

(a) Add code to the notebook that implements the dynamics function for the system of differential equations that govern T, A, and B, starting at the time of death. You may create and use helper functions.

(b) Use your model to try to estimate the time of death. Add lines to the notebook that start the simulation at the time you think the death occurred, and run it until 11:15pm to see if you can generate the same body/bacterial state. Display the final state (at 11:15pm). Also, create a plot of time versus body temperature, starting with your estimated time of death, and ending at 11:15pm on June 12.

(c) Considering the alibis below, who do you think killed Robert Durst?

Alibis for the prime suspects for June 12:

- **Dennis Reader**
  9:00am-11:00am Working in retail store (confirmed)
  11:00am-12:30pm Lunch in the food court (unconfirmed)
  12:30pm-4:00pm Working in retail store (confirmed)
- **Samantha Brundi**
  9:00am-11:00am Played squash with friend (confirmed)
  11:00am-1:00pm Lunch with family (confirmed)
  1:00pm-3:00pm At home watching TV (unconfirmed)

- **James Carver**
  9:00am-10:30am Went running (unconfirmed)
  10:30am-1:00pm Driving to out-of-town meeting (2.5 hour drive)
  1:00pm-3:00pm Meeting (confirmed)

What to submit

You must submit a series of PDF documents to Crowdmark. Each coding question should be in a PDF export of a jupyter notebook (sometimes several questions in a single notebook). When a proof or manual calculation is requested, you can typeset your solution using LaTeX or Word – **Photographs or scans of handwritten solutions should be legible, otherwise, the TAs might deduct marks.** . These solutions should also be submitted as PDFs.

Finally, upload your Python notebooks on Learn dropbox.