Numerical Solution of ODEs

Due at 2:00pm on Friday **Nov 5**, 2021

What you need to get

- `a3q4_YOU.ipynb`: Jupyter notebook for Q4
- `a3q5_YOU.ipynb`: Jupyter notebook for Q5

What to do

1. **[8 marks]** Solve the initial value problem
$$\frac{dy(x)}{dx} = x + y, \quad y(0) = 2,$$
using Euler's method and modified Euler's method with step size $h = 0.5$. Show your work. Compare your results with the exact solution
$$y(x) = 3e^x - x - 1,$$
at $x = 0.5$ and 1. Compute the errors of the approximate solutions given by these methods. (You may use a calculator to perform the individual arithmetic steps, but be sure to show how you arrived at your result.)

2. **[5 marks]** Show that the local truncation error of the midpoint method is $O(h^3)$ where h is the time step size. You may find the following two formulas useful for your computation:
$$f(x+h, y+k) = f(x,y) + \frac{\partial f(x,y)}{\partial x}h + \frac{\partial f(x,y)}{\partial y}k + O(h^2) + O(k^2),$$
$$\frac{d^2y(t)}{dt^2} = \frac{df(t,y(t))}{dt} = \frac{\partial f(t,y(t))}{\partial t} + f(t,y(t))\frac{\partial f(t,y(t))}{\partial y}.$$

3. **[6 marks]** Consider solving a differential equation $y'(t) = f(t, y(t))$ using a second-order Runge-Kutta method given by
$$y_{n+1} = y_n + \frac{h}{4}\left(f(t_n, y_n) + 3f\left(t_n + \frac{2h}{3}, y_n + \frac{2h}{3}f(t_n, y_n)\right)\right). \tag{1}$$
Analyse the stability of this method with our usual test equation
$$f(t, y(t)) = -\lambda y(t); \qquad \lambda > 0. \tag{2}$$
Explain whether (and under what conditions) the method is stable.

4. **[14 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 the *second* time it hits the ground.

   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 0.5 for this task.

(a) Complete the Python function `MyOde`,

```
t, y = MyOde(f, 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 (`f`) described in the `MyOde` help.

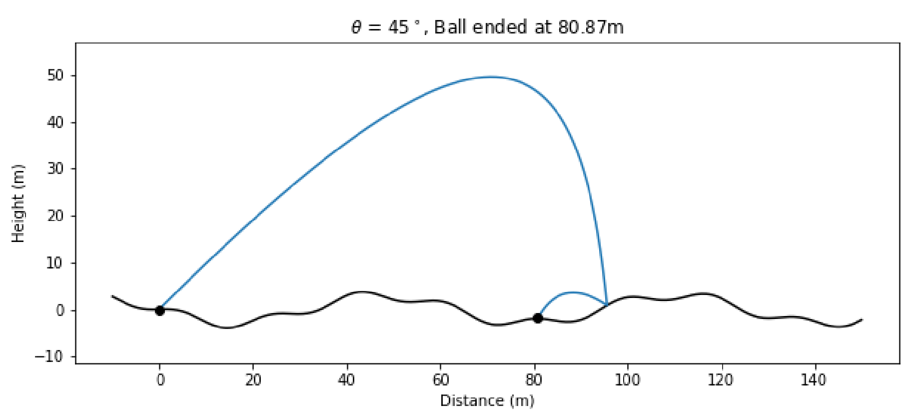
(c) The driving range has rolling hills, so that the level of the ground is the function,

$$\text{Ground}(d) = \sin\left(\frac{d}{3}\right) - 3\sin\left(\frac{d}{10}\right)$$

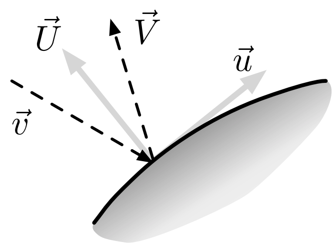
where d is the horizontal distance from the golf tee. The supplied function, `Ground`, evaluates this for you. Write a simple event function that changes sign as the golf ball hits the ground. Your event function should be continuous, not a step function. Call your event function `projectile_events`. 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 $70\,\mathrm{m/s}$ 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 hits the ground. The second flight starts at that first bounce, and ends when the ball strikes the ground for the second time. The figure below shows the ball's trajectory over 2 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 ($\vec{V}$), as well as the unit vectors 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 2 flights of the ball (as shown in the first figure). Note that each flight is simulated by a call to `MyOde`. For the second flight, the initial position and velocity is 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 2nd ground strike) the furthest horizontal distance from the golf tee. Plot the trajectory for your chosen θ-value.

5. **[8 marks]** The police arrived 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 21.5°C at 10:45am on October 7. 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 population growth rates depend on the body temperature, T, and are governed by the differential equations (in units $\frac{\text{u}}{\text{mm}^3}$)

$$\frac{dA}{dt} = \begin{cases} 0.0015(T-29)^2\left(1-e^{0.08(T-45)}\right)A(30-A) & \text{for } 29 \le T \le 45 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$\frac{dB}{dt} = \begin{cases} 0.002(T-13)^2\left(1-e^{0.05(T-26)}\right)B(20-B) & \text{for } 13 \le T \le 26 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

At the time of discovery, the concentration of bacteria A was $8.3\frac{\text{u}}{\text{mm}^3}$, and $19.2\frac{\text{u}}{\text{mm}^3}$ 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, T (in °C), follows the differential equation

$$\frac{dT}{dt} = -0.15\Big(T - T_a(t)\Big) + \frac{A+2B}{100} \tag{5}$$

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 6:00pm, and maintain 15°C from 6:00pm until 7:00am the next morning. It takes 30 minutes for the temperature to increase from 15°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 15°C at night (from 6:00pm until 8:00pm). 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 10:45am to see if you can generate the same body termperature/bacterial state. Display the final state (at 10:45am). Also, create a plot of time versus body temperature, starting with your estimated time of death, and ending at 10:45am on October 7.

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

Alibis for the prime suspects for October 6:

- **Dennis Rillerson**

9:00am-11:00am	Working in retail store (confirmed)
11:00am-11:30pm	Lunch in the food court (unconfirmed)
11:30pm-3:30pm	Working in retail store (confirmed)
4:00pm-6:00pm	Went running (unconfirmed)

- **James Carver**

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)
3:00pm-6:00pm	Golfing with a friend (confirmed)

- **Samantha Brundi**

9:00am-10:30am	Working from home (unconfirmed)
10:30am-1:00pm	Driving to out-of-town meeting (2.5 hour drive)
1:00pm-4:00pm	Meeting (confirmed)
4:00pm-6:00pm	Dinner with co-workers (confirmed)

What to submit

For handwritten or typeset questions, you can:

- typeset your solutions using a word-processing application, such as Microsoft Word, LaTeX, Google docs, etc., or
- write your solutions using a tablet computer, or
- write your solutions on paper and take photographs.

In either case, it is **your responsibility** to make sure that your solutions are sufficiently legible. Crowdmark will accept PDFs or image files (JPEG or PNG). You may submit multiple files for each question, if needed.

For programming questions, you must submit your code in two places: on Crowdmark, and on D2L.

Crowdmark: We suggest you rename your jupyter notebooks, replacing "`YOU`" with your WatIAM ID. For example, `a3q4_jorchard.ipynb`. Export each jupyter notebook as a PDF, and submit the PDF to Crowdmark.

D2L: Submit any `.ipynb` files to Desire2Learn in the designated dropbox. You do not need to zip these files.