Project 1: Solving ordinary differential equations numerically using the Euler method

PHYS 311, Spring 2018

Submission deadline: Thursday, February 08, 2018

Hint/Tip: Read the entire project first before starting to code/solve/answer the problems. Make a mental note of the various things that have been asked. While there are several problems and sub-problems for this project, it is often the case that you may be able to write one code that answers/addresses more than one problem. That is, you may not need to write a new code for every single problem. **Our class has people with a *very wide* range of `python` expertise. Please discuss with each other (and DM, of course) to *deepen* and *widen* your expertise!**

Q1. Write a python code called `ode_fname_lname.py` (replace fname by your first name and lname by your last name) to solve the differential equation $\ddot{x} = g$, where $g = 9.8\ m/s^2$ is a constant, with the following initial conditions: $x(t = 0) = 0\ m$, and $\dot{x}(t = 0) = 0\ m/s$. The code should stop when you get to $t = 10$ seconds.

Q2. Since you know that the analytic solution for velocity in this equation is $\dot{x}(t) = v(t) = gt$, calculate the percentage velocity error for each time step in Q1. The percentage error is defined as

$$err(t) = 100 \times \frac{v_{\text{numerical}}(t) - v_{\text{analytical}}(t)}{v_{\text{analytical}}(t)}$$

Here $v_{\text{analytical}}(t)$ is the velocity at time t calculated using the analytic formula $v(t) = gt$, and $v_{\text{numerical}}(t)$ is the velocity at that same time t predicted numerically by your code.

Make a plot showing how the percentage error varies with time. I strongly suggest using python's built-in `matplotlib` library (`https://matplotlib.org/2.0.2/users/pyplot_tutorial.html`) so that you can do Q1, Q2, Q3 all in one single code. While plotting, make sure that you

- label the X- and Y-axes correctly, and with proper units (where applicable), and
- the value of time increment (h) that you used to obtain this plot should also be indicated on the top-right of the plot.
- save this plot as a PNG image with the name `ode_fname_lname.png`.

Q3. What is the largest value of the time increment (h) in your code such that the percentage error in the velocity is always less than 1%?

Q4. Write a second python code called `ode2_fname_lname.py` (replace fname by your first name and lname by your last name) to solve the differential equation $\ddot{x} = g - k\dot{x}$, where $g = 9.8\ m/s^2$ is a constant, and the drag coefficient $k = 0.5\ s^{-1}$ is also a constant. Assume the following initial conditions: $x(t = 0) = 0\ m$, and $\dot{x}(t = 0) = 0\ m/s$.

Q5. Make a plot showing how the velocity changes with time. Based on this plot comment on how long does it take for the object to attain terminal velocity. While plotting make sure that you

- label the X- and Y-axes correctly, and with proper units (where applicable), and
- the value of time increment (h) that you used to obtain this plot should also be indicated on the top-right of the plot.
- save this plot as a PNG image with the name `terminalvelocity_fname_lname.png`.

Q6. Since you know the analytic solution for velocity in this equation with drag (we derived this equation in class), calculate the percentage velocity error for each time step in Q4.

Make a plot showing how the percentage error varies with time. While plotting, make sure that you

- label the X- and Y-axes correctly, and with proper units (where applicable), and
- the value of time increment (h) that you used to obtain this plot should also be indicated on the top-right of the plot.
- save this plot as a PNG image with the name `ode2_fname_lname.png`.

Q7. What is the largest value of the time increment (h) in your code such that the percentage error in the velocity is always less than 1%?

Submission guidelines: On the onCourse submission site for this project please upload the following: (1) your codes `ode_fname_lname.py` and `ode2_fname_lname.py`, and (2) your plots `ode_fname_lname.png`, `terminalvelocity_fname_lname.png` and `ode2_fname_lname.png`.