

Exercises for Day 33

Exercise 1. Implement the “More Complicated Euler Example” discussed in class in MATLAB. Loop from $i = 1$ to 20. Use a time step of 0.1 seconds. Plot y versus t on Figure 1, and V versus t on Figure 2. Recall that the magnitude of the velocity of the falling ball should increase linearly, but its position should be quadratic.

Exercise 2. Modify your program from Exercise 1 to loop while y is greater than y_f , rather than for a specified number of steps. Use $y_f = -4$ meters (note that this must be negative!) and a time step of 0.1 seconds. Plot y versus t on Figure 1, and V versus t on Figure 2. It’s ok if you have one step that comes just after y gets to y_f .

Exercise 3. Modify your program from Exercise 2 by putting the Euler integration into a function, while still doing the plotting in the main routine.

For the function, use the function call

```
function [y,v,t] = Euler1(dt,yi,yf,Vi,g)
```

where

- dt is the time step for the Euler integration
- y_i and V_i are the initial values of y and V , respectively
- y_f is the final value of y (note that this must be negative!)
- g is the acceleration due to gravity
- y , v , and t are vectors containing the solution

Do not name your function Euler — name it Euler1 or something else because Euler is already a MATLAB function.

For the main routine,

- Call the function with $dt = 0.1$ s, $y_i = 0$, $V_i = 0$, $y_f = -4$ meters, and $g = 9.8$ m/s²
- Plot the solution for y on Figure 1 and the solution for V on Figure 2
- Give good labels and titles to the plots

Exercise 4. Modify your main routine (only) from Exercise 3 to call the function 4 times using time step sizes 0.2, 0.1, 0.05, and 0.01 sec. Plot your y results (position versus time) on Figure 1, with the title “Convergence of the Position Solution.” Plot your V results (velocity versus time) on Figure 2, with the title “Convergence of the Velocity Solution.” Use good line types and add a legend to indicate which curve is which.

(over)

Exercise 5. Now we would like to plot the *exact* solution to the equations on our curve. As you recall, the exact answer for a free fall under the influence of gravity and without drag is

$$V = -gt + V(0)$$

$$y = -\frac{gt^2}{2} + V(0)t + y(0)$$

We want the y values to go from 0 to y_f , and we will need to pick enough points so that the curve is smooth.

(Notice that we will *NOT* use Euler's method on these equations because these are not differential equations!)

Create a function `Exact` that will find the exact solution for y values from a given initial value until the ball falls below a given final value:

- The function statement should be

```
function [y,v,t] = Exact(delta,yi,yf,Vi,g)
```

- The inputs and outputs are similar to before, except that `delta` is the time increment used to generate the time vector. We aren't doing Euler's method on these equations, but we still need to know what times to use in the time vector.

Modify your main routine to call the function and plot the results:

- Call the function with `delta = 0.01 s`, `yi = 0`, `Vi = 0`, `yf = -4 meters`, and `g = 9.8 m/s2`
- Add the exact solution for y on Figure 1 from Exercise 4, and change the legend appropriately
- Add the exact solution for V on Figure 2 from Exercise 4, and change the legend appropriately

What to turn in: Turn in the just the main routine, two functions, and two plots from Exercise 5.