PHYS2160 Introductory Computational Physics
2021/22 Exercise 6

1. Write a Python program that solves the differential equation

$$\frac{dy}{dx} = 2x - 3y + 1$$

with the initial condition $y(1) = 5.0$ for $1 \leq x \leq 5$ using Euler's method with number of steps $N = 10, 20, 50, 100$ and then compares the numerical solutions with the analytical solution $y = 1/9 + 2x/3 + 38e^{-3(x-1)}/9$ by plotting them on the same graph.

2. Write a Python program that solves the differential equation

$$\frac{dy}{dx} = \frac{(x+1)y}{2x} - \frac{3y^3}{x}$$

with the initial condition $y(1) = 1.0$ for $1 \leq x \leq 5$ using the second order Runge-Kutta method with number of steps $N = 10, 20, 50, 100$ and then compares the numerical solutions with the analytical solution $y = \sqrt{x/(6 - 5e^{1-x})}$ by plotting them on the same graph.

3. Write a Python program that solves the differential equation

$$\frac{dy}{dx} = \frac{6y^2}{x^2} - \frac{3y}{x} - 2$$

with the initial condition $y(1) = -0.6$ for $1 \leq x \leq 5$ using the fourth order Runge-Kutta method with number of steps $N = 10, 20, 50, 100$ and then compares the numerical solutions with the analytical solution $y = (1 + 2x^8)x/(1 - 6x^8)$ by plotting them on the same graph.

4. The Lorenz model is a simple atmospheric model that simulates weather patterns by the Lorenz equations

$$\frac{dx}{dt} = \sigma(y - x),$$
$$\frac{dy}{dt} = x(\rho - z) - y,$$
$$\frac{dz}{dt} = xy - \beta z$$

where σ, ρ, β are positive parameters in the model. Write a Python program that solves these equations with the initial conditions $(x(0), y(0), z(0)) = (0, 1, 0)$ from $t = 0$ to $t = 50$ for $\sigma = 10, \rho = 28, \beta = 8/3$ using the fourth order Runge-Kutta method with step size $h = 0.001$ and then makes a plot of x, y, and z as a function of time t as well as a plot of z against x and y. From the latter plot, you should see a picture of the famous "strange attractor" of the Lorenz equations, a lop-sided butterfly-shaped plot that never repeats itself.

5. An object of mass m falling near the Earth's surface is retarded by air resistance proportional to the square of its speed. According to Newton's laws of motion, its motion is governed by the differential equation:

$$m\frac{d^2x}{dt^2} = mg - k\left(\frac{dx}{dt}\right)^2$$

where $x(t)$ is the displacement of the object at time t, $g = 9.81\,\mathrm{m/s^2}$ is the acceleration due to gravity, and $k > 0$ is a constant of proportionality. Write a Python program that solves this equation with the initial conditions $(x(0), x'(0)) = (0, 0)$ from $t = 0$ to $t = 5\,\mathrm{s}$ for $m = 2\,\mathrm{kg}$ and $k = 0.1\,\mathrm{kg/m}$ using the fourth order Runge-Kutta method with step size $h = 0.001\,\mathrm{s}$ and then plots x and x' as a function of time t on two separate graphs sharing the same horizontal axis.

6. A nonlinear pendulum like the one in Example 8.5 can be driven by exerting a small oscillating force horizontally on the mass. Then the equation of motion for the pendulum becomes

$$\frac{d^2\theta}{dt^2} = -\frac{g}{\ell}\sin\theta + C\cos\theta\cos(\Omega t)$$

where $\theta(t)$ is the angular displacement of the pendulum at time t, and C and Ω are positive constants. Write a Python program that solves this equation with the initial conditions $(\theta(0), \theta'(0)) = (0, 0)$ from $t = 0$ to $t = 100\,\mathrm{s}$ for $\ell = 0.1\,\mathrm{m}$, $C = 2\,\mathrm{s^{-2}}$, $\Omega = 5\,\mathrm{s^{-1}}$ using the fourth order Runge-Kutta method with step size $h = 0.001\,\mathrm{s}$ and then plots θ as a function of time t.

7. A damped oscillator driven by an external force undergoes motion governed by the differential equation:

$$\frac{d^2x}{dt^2} + \gamma\frac{dx}{dt} + \omega_0^2 x = F_0\cos(\omega t)$$

where $x(t)$ is the displacement of the oscillator at time t, and γ, ω_0, F_0, and ω are positive constants. Write a Python program that solves this equation with the initial conditions $(x(0), x'(0)) = (10\,\mathrm{m}, 0)$ from $t = 0$ to $t = 20\,\mathrm{s}$ for $\gamma = 1\,\mathrm{s^{-1}}$, $\omega_0 = 10\,\mathrm{rad/s}$, $F_0 = 50\,\mathrm{m/s^2}$, and $\omega = 12\,\mathrm{rad/s}$ using the fourth order Runge-Kutta method with target accuracy per unit time $\delta = 10^{-6}$ and then plots x as a function of time t.

8. A cannon shell is launched with an initial speed v_0 at an angle θ from the horizontal. Due to air resistance, the shell moves in the vertical plane with the magnitude of drag force given by $F_d = -kv^2$ where v is the speed of the shell and k is a positive constant. Applying Newton's second law, one can show that the motion of the shell is governed by the system of differential equations:

$$\begin{aligned} m\frac{d^2x}{dt^2} &= -kv\frac{dx}{dt} \\ m\frac{d^2y}{dt^2} &= mg - kv\frac{dy}{dt} \end{aligned}$$

where $x(t)$ and $y(t)$ are the x and y coordinates of the shell at time t, and $g = 9.81\,\mathrm{m/s^2}$ is the acceleration due to gravity. (Note that here x-axis points along the shell's initial direction of horizontal motion and y-axis points vertically upward.) Write a Python program that solves these equations until the shell reaches the ground for $\theta = 30°, 35°, 40°, 45°, 50°, 55°$ using the fourth order Runge-Kutta method with step size $h = 0.01\,\mathrm{s}$ and then plots all these trajectories (i. e. y versus x) on the same graph. In your calculation, assume $k/m = 4 \times 10^{-5}\,\mathrm{m^{-1}}$, $(x(0), y(0)) = (0, 0)$ and $v_0 = 700\,\mathrm{m/s}$.