Exercise 6

Date: 26.11.2021 & 29.11.2021

Task 1: The pendulum problem

The pendulum equations of motion defined as

$$\begin{aligned} \dot{\varphi} &= p \\ \dot{p} &= -\sin\varphi \end{aligned} \tag{1}$$

where φ is the angle between the pendulum and the vertical axis and p is the conjugate momentum. Eq. 1 are solved using the forward Euler method in the given `Week6_Exercise.py`. The equations are time normalised in the given formulation, such that the time unit is measured in the period of the pendulum swing.

- Compare the time evolution of the total energy E between $\Delta t = 0.01$ and $\Delta t = 0.1$. Answer the questions in the python code.

The computed values are obviously unsatisfactory. However, there are multiple ways to solve the equation of motion that can possibly give better results.

- Implement the Leapfrog algorithm.
- Now, implement the Verlet formulation of the Leapfrog algorithm. Note that the acceleration term of Eq. 1 is

$$\ddot{\varphi} = \dot{p} = -\sin\varphi.$$

- You can also solve the equations using the Trapezoidal method of integration. However, this is not straight forward due to the nonlinearity of the problem.
  - How would you numerically integrate Eqn. 1? Write down the numerical expression of the integration, given that the expression of the analytical integration is

$$\begin{aligned} \int \dot{\varphi}(t)\ \mathrm{d}t &= \int p(t)\ \mathrm{d}t \\ \int \dot{p}(t)\ \mathrm{d}t &= \int -\sin\left(\varphi(t)\right)\ \mathrm{d}t. \end{aligned}$$

    NOTE: Refer to 'Trapezoidal method' in slide 14 of Week 4 lecture.
  - In order to simplify the obtained expression (as well as the implementation), substitute $\varphi_* := \varphi_n + \frac{\Delta t}{2} p_n$ and $p_* := p_n - \frac{\Delta t}{2}\sin\varphi_n$ into the discretised equations.

    NOTE: Depending on your derivation, you will end up trying to solve for φ_{n+1} or p_{n+1}. Both of them can be solved in the same way since they are nonlinear either way. However, working with φ_{n+1} will be easier. You can compute the solution (either φ_{n+1} or p_{n+1}) using the Newton-Raphson method (similar to the prior exercise of solving the kepler equation).
- We'll now study the fourth order Runge-Kutta method by deriving the discretised expression of Eq. 1, since analytical expressions of p and φ is not given.

- Think of one method to obtain $p(t + \Delta t, \varphi + \Delta\varphi)$ and $\varphi(t + \Delta t, p + \Delta p)$.

  NOTE: This is useful since the method can be used to obtain the intermediate steps in RK4.

- Complete the derivation of RK4 implementation, showing the steps needed for $\varphi_{\mathrm{n}} \to \varphi_{\mathrm{n+1}}$. (Discretisation is also needed for the $\dot{p}$ equation, but the steps are similar to $\dot{\varphi}$ equation). Switch `ODE_method` to `'RK4'` and test the method.

Proceed to open `Week6_RK4.py` and compare your result with `solve_ivp` from `scipy.integrate`. Answer the questions within.