

# Numerical Methods in Classical Physics

## PHYS3034 Assessment Questions

### PHYS3034 (Mainstream Students) – Exam Questions

#### 1 Instructions

Provide written answers, including explanations where required, for all questions. If a question involves writing or modifying Python code, include your code in a documented notebook alongside your submission. You are reminded that this work must be your own. For code, this means that all new parts required for a problem must be your own implementation. (A notebook containing code snippets to support your implementation can be found here: [Assessment3034.ipynb](#) on GitHub)

#### 2 A. Two-Body Problem

##### Question A1: Velocity-Verlet Simulation of Two-Body Orbits (10 points)

In this question, you will simulate two-body gravitational interaction in the centre-of-mass frame using non-dimensional units. Assume total mass is normalised to 1, and gravitational constant  $G = 1$ .

- i. (2 points) Write the equations of motion and convert them into dimensionless form. Implement a Python code that uses the **Velocity-Verlet algorithm** to solve the equations of motion for the two-body problem. Use the following initial conditions:
  - $\vec{r}_1 = [1, 0]$ ,  $\vec{v}_1 = [0, 2 - \sqrt{3}]$
  - Choose  $\vec{r}_2$  so the centre of mass is at the origin.
  - Choose  $\vec{v}_2$  so the total momentum is zero.
  - Mass ratio:  $a = 0.5$
- ii. (2 points) Use a time step  $\tau = 0.05$  and simulate until  $T = 25 \times 2\pi$  and produce:
  - a plot of the orbits
  - a plot of total energy versus time
  - plot of the phase space trajectory (momentum vs position)
- iii. (2 points) Repeat the simulation for  $a = 1/3$ . Adjust the time step if needed. Explain briefly how you verified the solution's accuracy and describe the new orbit and phase space area.
- iv. (2 points) Compare total energy conservation in both cases. Which one conserves energy better, and why?
- v. (2 points) Estimate the **orbital period** (convert back to the correct units) from your simulation and compare it with the theoretical value.

### Question A2: RK4 for Two-Body Problem (10 points)

- i. (2 points) Write the equations of motion in the form  $\frac{du}{dt} = f(u, t)$ , identifying the components of  $u$  and  $f$ .
- ii. (2 points) Implement a 4th-order RungeKutta (RK4) solver in Python. Explain how the parameter  $a$  is handled in your implementation.
- iii. (2 points) Simulate for  $a = 0.5$  using the same initial conditions and parameters as in A1. Plot the orbits and total energy and phase space trajectory.
- iv. (2 points) Repeat the simulation for  $a = 1/3$  and comment on any differences in the behaviour.
- v. (2 points) Compare the RK4 and VelocityVerlet methods in terms of accuracy and stability, plot and describe the errors as a function of time, and comment on how each method preserves key physical quantities such as energy and orbital shape.

## 3 B. Heat Diffusion

### Question B1: Two-Dimensional Diffusion Equation (10 points)

In lectures, we studied one-dimensional diffusion. This question extends the method to a two-dimensional spatial domain. We now compute the solution at spatial indices  $(i, j)$  and time index  $n$ , denoted as  $T_{ij}^n$ . The 2D diffusion equation is:

$$\frac{\partial T}{\partial t} = \kappa \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right), \quad \kappa > 0. \quad (1)$$

- i. (4 points) Derive the Forward Time Centered Space (FTCS) scheme for the 2D diffusion equation:

$$T_{jl}^{n+1} = T_{jl}^n + \frac{\kappa\tau}{h^2} (T_{j-1,l}^n + T_{j+1,l}^n + T_{j,l-1}^n + T_{j,l+1}^n - 4T_{jl}^n) \quad (2)$$

- ii. (6 points) Apply von Neumann analysis to the scheme above using the trial solution  $T_{jl}^n = A^n e^{i(k_x j h + k_y l h)}$  and derive the stability condition:

$$\frac{\kappa\tau}{h^2} \leq \frac{1}{4}. \quad (3)$$