THE UNIVERSITY of EDINBURGH
School of Physics and Astronomy

Exercise 3 – Marking Sheet. Marker: TD

Group Members

1. Marina Ruiz Sanchez-Oro
2. Cara Lynch
3.

Marks

Particle3D Class [13]	Constructors and properties [3] • Properties position, velocity, mass, label • __init__() method correct	**2/3**
	Methods are correct [5] • __str__ () instance method, correct format • Method for kinetic energy of particle • Method for first-order velocity update • Method for first-order position update • Method for second-order position update • Method for particle separation vector	**4/5**
	Code layout, naming conventions and comments [5]	**4/5**
Particle simulations [12]	Symplectic Euler and velocity Verlet implementations [3]	**3/3**
	Force and energy law implementations [2]	**2/2**
	Trajectory data are present and correct [2]	**2/2**
	Code layout, naming conventions and comments [5]	**5/5**
Report [5]	Plots of trajectories [1]	**1/1**
	Timestep estimates [1]	**1/1**
	Discussion of results and conclusions [3]	**1/3**
Total [30]		**25/30**

Feedback

Code

1. Constructors should take vectors, not individual floats.
2. Constructor does not use a label (string).
3. toString method does not use required format.
4. Good use of methods for force and potential energy.

Report

1. "The first method is symplectic Euler, which only takes into account a first order approximation of the position." – All methods use actual position. Euler uses "dr" to first

order. Absolute position and change in position are two different things. This same comment also applies to description of velocity Verlet.

2. "It was shown that for larger timesteps, the velocity Verlet method provides more accurate results, whereas for small time steps, it is computationally more efficient to use the symplectic Euler approach." – Verlet is always more accurate and Euler is always more computationally efficient independent of time step size, by virtue higher order terms being considered and less calculations per time step, respectively. Accuracy and computational efficiency should never be considered in isolation since both are important.

3. "When $dt \rightarrow 0$ the symplectic Euler and velocity Verlet method will converge to the same model (see Figure 1 -right-), which means that in this case the Euler method should be used, since it uses less computational power." – this does not follow logically and it is not clear what you mean by "this case". Extremely small time steps (dt->0) means extremely long simulation time. **The goal is always to achieve your required degree of accuracy in the shortest computational time possible.** Sometimes this can mean Euler can be used instead of the more accurate Verlet.

4. "Since the timestep is very large, the trajectory shows a behaviour which is equal to the initial behaviour at all times for the velocity Verlet time integration method." – this is true for both methods, the reason being that after the first step (with too large a dt) the particle travels beyond significant influence of the central force and continues linearly as if the central mass was not there. Only before the first time step does the force provide significant effect, resulting in the diagonal trajectory.

5. "The particle veers towards the left as it is initially attracted by the central mass and the velocity is in the y direction. In contrast, the symplectic Euler method shows a linear trajectory in the +y direction, consistent with the initial velocity therefore as if the central mass was exerting no force on the particle." - Both methods show diagonal motion, neither are purely vertical. This statement is confusing.

6. The time units in your simulation are not seconds, therefore (e.g. dt = 0.4111 seconds has incorrect units. Similarly, energy units are not Joules.

7. "The symplectic Euler method is preferable for timesteps of the order of 0.01s and smaller whereas the velocity Verlet method should be used for larger timesteps." - See previous comment, **the goal is always to achieve your required degree of accuracy in the shortest computational time possible.** Algorithm preference is based on preferred accuracy, not preferred time step. The appropriate time step is then a consequence of this reasoning.