THE UNIVERSITY of EDINBURGH
School of Physics and Astronomy

Exercise 3 – Marking Sheet. Marker: AH

Group Members

1. J. Maroulis
2. S. Sirur
3. S. van de Bund

Marks

Particle3D Class [13]	Constructors and properties [3] • Private properties position, velocity, mass, label • Constructors: default, explicit, from Scanner object	**2**
	Methods are correct [5] • Getters for all properties • Setters for all properties • toString() 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	**4**
	Code layout, naming conventions and comments [5]	**5**
Particle simulation [12]	Symplectic Euler and velocity Verlet implementations [3]	**3**
	Force and energy law implementations [2]	**2**
	Trajectory data are correct [2]	**1**
	Code layout, naming conventions and comments [5]	**4**
Report [5]	Plots of trajectories [1]	**1**
	Timestep estimates [1]	**1**
	Discussion of results and conclusions [3]	**2**
Total [30]		**25**

Feedback

- Particle3D: Default ordering in a class is usually Constructors – Setters/Getters – Instance – Class methods. Remove references to Particle1D.

- toString() method is not using the requested format (which means you have to rewrite it later to interface to VMD).
- 2nd order position update written somewhat disjointedly. Easier to read would be to calculate dx as a whole, then add it to the current position. But it's good to see you're recycling methods you already have.

- Scanner method is going beyond what we asked of it: it should have taken a Scanner object as input, read out content, and return a particle. You have included the entire file I/O in the method, which arguably makes it less flexible to use – imagine you wanted to read in only every second particle listed in some input file. Plus, it gives me a compilation error (on a Mac, but not on SL7).
- Comments are very good, and bar for a couple of return values give very comprehensive web documentation.
- Particle3DEuler and Particle3DVerlet: very well written and documented, very good to keep the Javadoc going (almost complete, bar for the last method), good use of external methods for computations.
- A Particle3DVerlet issue: you do two force evaluations at every time step, where only one is necessary (the 'old' forces are taken from the previous time step). In the long run this will slow your calculation by a factor of two (forces are the most expensive object to calculate).
- Trajectory data are missing but can be re-created.
- Report: trajectory plots are fine, but could benefit from similar x-y data ranges (both are circles but one does not look like one). Characteristic time is indeed the (shortest) orbit in the system, and velocity Verlet will not work if the time step is set to this number. Note you get decent accuracy when sampling an orbit with only ten steps! Energy fluctuation plot has meaningless y-axis labels (not enough digits), so can't judge the fluctuation scale. Maximum time step estimate for velocity Verlet is correct, but missing for symplectic Euler (it's ~0.14, so not 20 times smaller when the maximum allowed error is 10^{-6}). Conclusions are correct: v-Verlet is better than s-Euler.