FYS4150 - Computational Physics
Project 3

Alex Ho

October 14, 2016

3a)

3b) + 3c)

Implementing the Euler and Verlet method is quite easy. We have the Euler method as

$$x_{i+1} = x_i + v_i dt$$
$$v_{i+1} = v_i + a_i dt$$

One needs to calculate the acceleration a_i using the discretized equations that we have previously shown. One can also implement a more precise method, the Euler Cromer's method, which is given as

$$v_{i+1} = v_i + a_i dt$$
$$x_{i+1} = x_i + v_{i+1} dt$$

Which is almost the same as the Euler method, however, the position now depends on the new velocity v_{i+1}. Unlike the Euler method, we now need to calculate the new velocity before calculating the new position. In the Euler method, we did not have to take this into account. In the C++ program, the ordering of the position and velocity calculation are swapped for these two methods, and we will see later that this will give two different results.

The next method we will look at is the Verlet method. It is given as

$$x_{i+1} = x_i + v_i dt + \frac{dt^2}{2} a_i$$
$$v_{i+1} = v_i + \frac{dt}{2}\left(a_{i+1} + a_i\right)$$

Implementing this may be a little tricky. Calculating the position is straight forward, but the velocity now depends on a_{i+1} and a_o. To bypass this problem, we will have to save a_i so its own variable and then calculate the new acceleration a_{i+1} using the new calculated position x_{i+1}.