5CCP211C — Introduction to Numerical Modelling
Physics Department — King's College London

KING'S College LONDON

Simulating planetary motion using molecular dynamics

Written by M. Haider and Z.J. Cheung

Group 1 Pair 2

Conducted on November 20, 2018

1 What is molecular dynamics?

Molecular dynamics is a numerical method used to investigate the physical properties that arise when assembling groups of atoms or molecules in terms of their structure and microscopic interactions. The interactions of these particles are assumed from a given potential energy function. These functions allow the calculation of the force being experienced by any atom by considering the positions of all other atoms. Newton's equations of motion can then be used to determine how these forces will affect the motion of the atoms. The interaction is carried out for a given unit of time, allowing the dynamic evolution of the system to be investigated.

The time must be divided into discretised time steps which are suitably small compared to the total simulated time. At each time step the force acting on each particle is calculated by using the specified potential energy function and the system of particles is dynamically evolved by updating the position and velocities of each particle, utilising Newtons equations of motion.

Here are the key equations which are used in the algorithm.

Newtons second law:

$$\mathbf{F} = m\mathbf{a} \tag{1}$$

Where $\mathbf{F}$ is the force on a particle, m is the mass and $\mathbf{a}$ is it's acceleration.

Relation between force and potential energy:

$$F(\mathbf{x}) = -\nabla U(\mathbf{x}) \tag{2}$$

Where $\mathbf{U}$ represents the given potential energy function and $\mathbf{x}$ represent the x component.

The equations of position and velocity may now be calculated:

$$\frac{d\mathbf{x}}{dt} = \mathbf{v} \tag{3}$$

Using the fact that $\mathbf{a} = \frac{d\mathbf{v}}{dt}$,

$$\frac{d\mathbf{v}}{dt} = \frac{\mathbf{F}(\mathbf{x})}{m} \tag{4}$$

The inherent properties of MD simulation ensure that the system of particles does not reach an energy minimum; the particles are in constant motion just like real life. If the simulation time scales are suitably large, the distribution of particle energies will resemble the Boltzmann distribution. This method of simulation also ensures that energy (the sum of potential and kinetic energies) are conserved throughout the simulation. This ensures that particles with lower potential energies will have greater kinetic energies - moving at greater speeds. However due to the numerical errors arising when rounding in calculations, the total energy of the system grows through the simulation.

The main constraint of simulations utilising MD methods are the timescales which are used to progress each step simulation. These need to be very short to ensure numerical stability - since the algorithm relies on the differential equations with respect to time. Physical changes in a system such as the structural change in proteins may take as little time as nanoseconds (10^{-9} s) and depending on the simulated time and numbers of particles being investigated, may take million to trillions of steps to achieve. This requires a substantial amount of computing power and time to simulate a system where these processes can be seen. This can be improved by limiting the simulation to only important interactions, implementing faster algorithms, reducing number of simulated particles or by reducing

the physical time which is to be simulated. Alternative methods such as Monte Carlo, which have been previously investigated, can be used to approximate interactions by using a stochastic approach. The Metropolis model also ensures that the ensemble average obeys the Bolztmann distribution.

2 Investigation

This simulation aims to simulate the orbits of planets around a Sun, which lies in the middle of the system. In this case, many body interactions are neglected due to their complexity, resulting in two body interaction between each planet and the Sun. The integration of the equations of motion are based on three different methods; the Euler method, Verlet method and Velocity Verlet method. The potential energy is a function that accounts for all the positions of the particles within a system. Due to this complexity, there are no analytical solutions to the differential equations being used. Taylor's expansion is used to approximate the positions, velocities and accelerations. Boundary conditions are implemented in the form of isolated boundary condition (IBC) [**?**]. This is where the particles within the system are isolated such that they only interact amongst themselves. The interaction between the bodies are limited to that between the Sun and each planet - as mentioned before, this satisfies the condition needed for IBC's.

2.1 Euler Method

The Euler method assumes the time interval is sufficiently short, such that an approximation can be made,

$$dt \approx \Delta t$$

The forward finite difference formula can be used to replace the derivative.

$$v(t) = \frac{dx}{dt} \approx \frac{x(t+\Delta t) - x(t)}{\Delta t} \tag{5}$$

$$a(t) = \frac{dv}{dt} \approx \frac{v(t+\Delta t) - v(t)}{\Delta t} \tag{6}$$

Solving for the new values of displacement and velocity,

$$x(t+\Delta t) = x(t) + v(t)\Delta t \tag{7}$$

$$v(t+\Delta t) = v(t) + a(t)\Delta t \tag{8}$$

Equation 7 and 8 can also be derived by taking the first two terms of the Taylor expansion of the left hand side, ignoring higher order terms.

The potential energy which is being investigated is constrained to the gravitational interaction between the masses of the two bodies. The force between the bodies is given by:

$$F_i = G\frac{M_i M_{Sun}}{r_{i,Sun}^2} \tag{9}$$

Where **i** represents the body being investigated, **M** is the mass of the body and $r_{i,Sun}$ is it's distance from the Sun.
The potential energy is given by (in accordance to equation 2):

$$U_i = -G\frac{M_i M_{Sun}}{r_{i,Sun}} \tag{10}$$

The acceleration is found using Newton's second law and Newton's law of universal gravitation,

$$\mathbf{a(t)} = -G\frac{M_i M_{Sun}}{r^3_{i,Sun}}\mathbf{x(t)} \tag{11}$$

where $x(t)$ is the x component of the vector describing displacement of the planet from the Sun. The y component for acceleration can be found using the same equation.

2.1.1 Implementation in code

The first procedure of the Euler method in code is moving the position of the planet by a time interval, Δt (expressed as dt in code). The acceleration for the x and y components at the position after movement is calculated. This value can be used in Equation 8 to find the new velocity.

This is performed for each planet in each time step, thus allowing the system to evolve and update its position, velocity and acceleration at each interval.

2.2 Verlet method

The next position is determined by the current position and the position at time $t - \Delta t$. The Taylor expansion is limited to the first three terms:

$$x(t + \Delta t) = 2x(t) - x(t - \Delta t) + a(t)\Delta t^2 \tag{12}$$

$x(t + \Delta t)$ represents the position when time progresses from $t \to \Delta t$ where t is the current time and Δt is the discretised time interval and a is the acceleration.

Disadvantage: Not accurate - lack of explicit velocity terms

2.2.1 Implementation in code

The Verlet method relies on the current position and previous position to calculate the next position for the planet. Therefore, a list property is introduced for each planet, which can be referenced as a class attribute.

The first step uses the Euler method to move the planet by a single time increment. At this stage, each planet has a current position and previous position, which is its original position.

Given this information, the next position can be calculated using the Verlet method by referencing the previous position via indexing the second last item in the list with negative indexing.

The acceleration is calculated for the current position each time.

2.3 Velocity Verlet method

The next position in time is given by:

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \frac{a(t)}{2}\Delta t^2 \tag{13}$$

and the next velocity by:

$$v(t + \Delta t) = v(t) + \frac{a(t) + a(t + \Delta t)}{2}\Delta t \tag{14}$$

Advantage: Velocities calculated explicitly, simultaneously with position.

2.3.1 Implementation in code

The procedure of carrying out the Velocity Verlet method is done by first moving the position of the planet. The velocity variable is given since it is initialized as the initial velocity of the planet, and the acceleration can be found as a function of the distance from the Sun.

This is done to allow allow calculations for $v(t + \Delta t)$ and $a(t + \Delta t)$ to be made. After moving the position, the velocity and acceleration at position $t + \Delta t$ can be found.

2.4 General execution of code

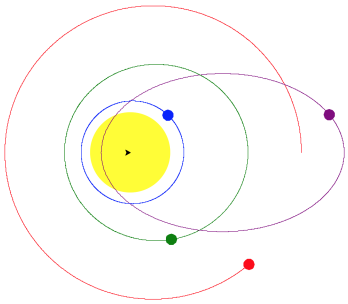
Using three different classes and their own class methods, a system can be initialized with the required objects. These classes are the Planet, Sun and milkyWay class. After initialization, the orbits can be ran.

The number of iterations of time increments is set by a variable $timeFrame$. Note that this is not the actual time; and actual time can be found by dividing the iteration by the time increment dt. The code can simulate planetary motion graphically using Turtle graphics and via a method of choice (either Euler, Verlet or Velocity Verlet).

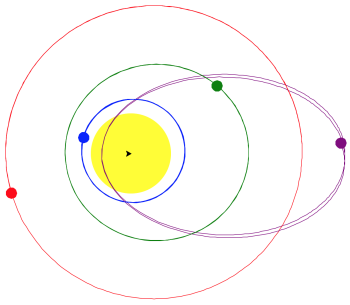
The properties of each Planet object is updated within the iteration, in the class method. Throughout the iteration, some important variables are collected. These include all previous positions of the planet and the gravitational potential energy of the planet as a result of a two body interaction with the Sun.

After completing the simulation, the orbital period is found for each planet, if a full revolution has been completed. This value cannot be found is the planet has not completed a revolution. A plot of the potential energy against time is also shown at the end of the simulation (the simulation must be finished for these to be printed/shown).

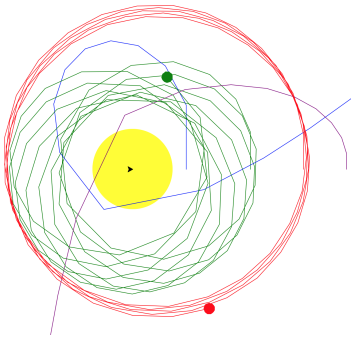
3 Results



$G = 1$, $\Delta t = 0.1$



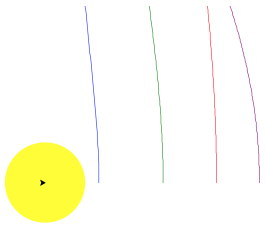
$G = 1$, $\Delta t = 1$



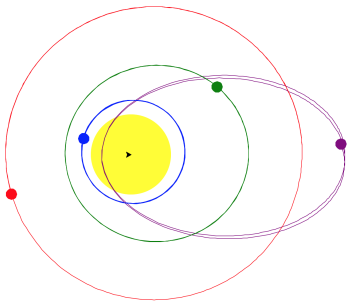
$G = 1$, $\Delta t = 10$

Figure 1: Different arrangements obtained by keeping the Gravitational constant, G at 1 and changing the discretised time, Δt to 0.1, 1 and 10 respectively.

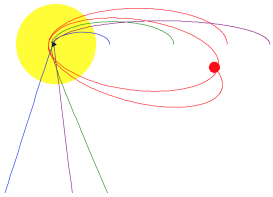
The value of the gravitational constant, G, is appears in the calculation of force and acceleration. It acts as a linear multiplier within both equations. The greater the value, the greater the force/ acceleration. The value of $G = 1$ results in orbits which are stable and return to their original positions. In the figure above it can be seen that the arrangement corresponding to the shortest time period ($\Delta t = 0.1$) has the smallest number of completed orbits, whilst the largest time period ($\Delta t - 10$) has the most. This is because the calculations are able to be completed faster for a larger time interval for the same number of steps. This however means that the trajectories of the orbitals will not be as resolute as using smaller time intervals - as can be seen by the changing orbitals of the four planets in the last sub-figure. As the time interval approaches zero, the properties of the planets reach their most accurate values.



$\Delta t = 1$, $G = 0.1$



$\Delta t = 1$, $G = 1$



$\Delta t = 1$, $G = 10$

Figure 2: Different arrangements obtained by keeping the discretised time, Δt, at 1 and changing the Gravitational constant, G, to 0.1, 1 and 10 respectively.

As the value of the Gravitational constant, G, is increased the gravitational force and acceleration is also increased. This can be seen by the contrast between the values $G = 0.1$ and $G = 10$. As $G = 0.1$ corresponds to the smallest force/acceleration, the mass of the Sun has little attraction to the trajectories of the planets and they are seen to exit the view with little perturbation in their paths. Increasing the value of G increased the gravitational attraction between the two bodies. This can be seen by the change in planet orbitals when $G = 10$. The planets are attracted so greatly to the Sun that many are slingshotted out of view. The radius of orbit for the red planet is greatly reduced too- with the eccentricity of orbit increasing too.

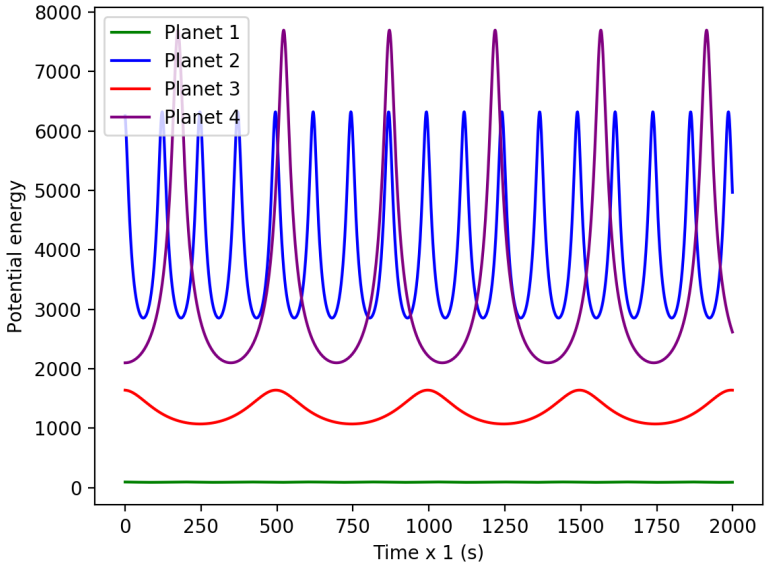


Figure 3: Potential energies plotted against time for the Velocity-Verlet method. Δt is set a 1, G as 0.7. This is run for 2000 MD steps.

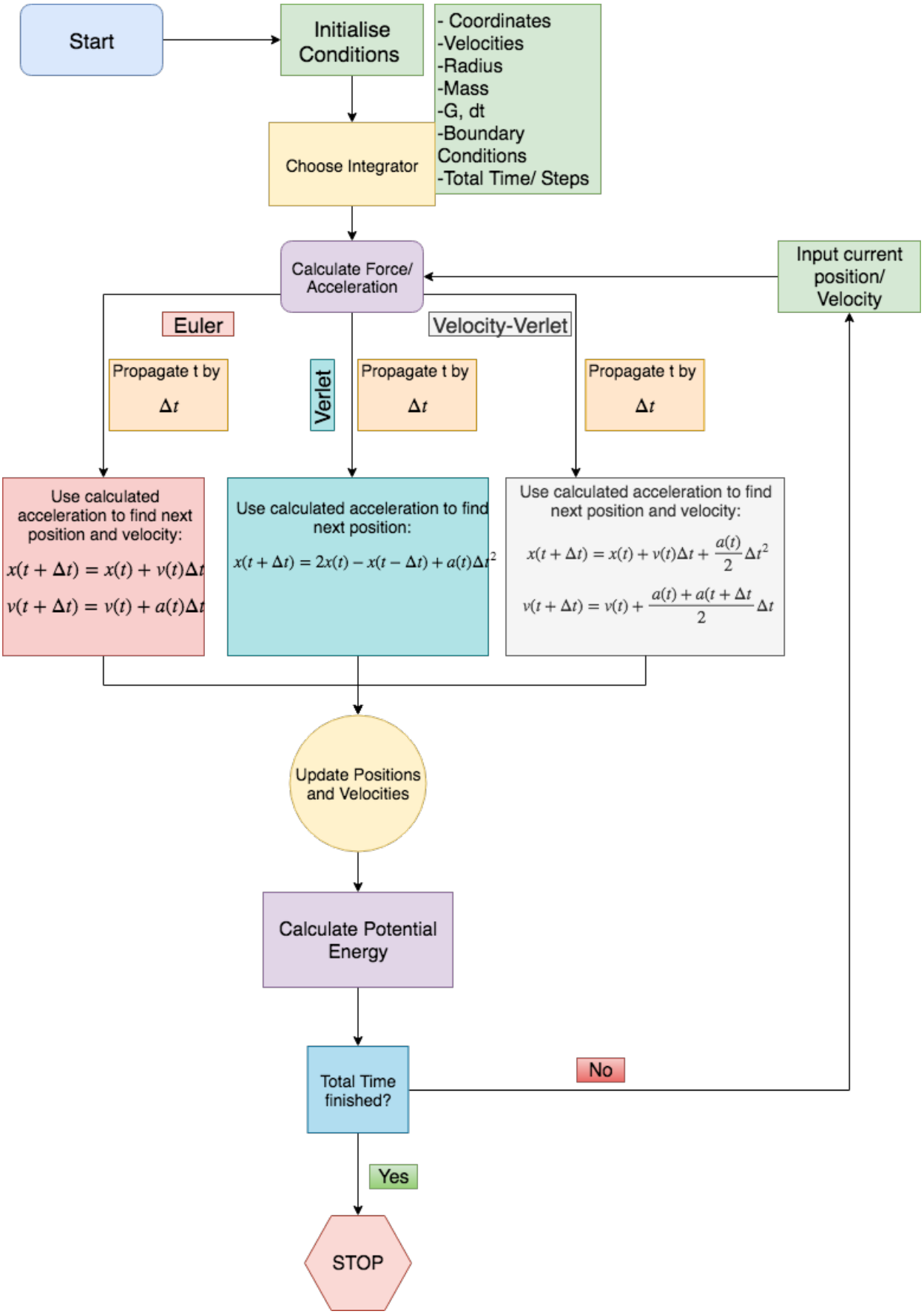


Figure 4: Basic Structure of cMD code