

Computer Simulation Methods

University of Cambridge Part II Natural Sciences Tripos

Yue Wu

*Yusuf Hamied Department of Chemistry
Lensfield Road,
Cambridge, CB2 1EW*

yw628@cam.ac.uk

Acknowledgements

Nothing in these lecture notes is original. They are largely based on the notes by Prof. Rosana Colleparado, who lectured this course in 2025. Moreover, they are nowhere near accurate representations of what was actually lectured, and in particular, all errors are almost surely mine.

Contents

1	Molecular Dynamics	4
1.1	Integrating the Equations of Motion	4
	Appendices	12
A	Noether's Theorem	12

1 Molecular Dynamics

1.1 Integrating the Equations of Motion

A famous conclusion in classical mechanics is that the motion of three bodies under gravitational interaction has no general analytic solution in closed form. The problem will only get worse if we introduce more bodies into our systems, or we consider more complex form of the interactions. It is not uncommon for a system in physics and chemistry to have more than billions of interacting particles, so we usually have no choice but to treat their interactions numerically on a computer. The aim of *Molecular Dynamics* (MD) is to study a system by recreating it on the computer as close to nature as possible.

1.1.1 Newton's Equations of Motion

We start from the most fundamental law in classical mechanics — the Newton's equation. If we have a system of N particles, and their positions are given by $\{\mathbf{r}_i\}_{i=1}^N$ which we collectively denote as \mathbf{r}^N , then the interactions between the particles will be completely determined by the positions of the particles, specified by the potential

$$V = V(\mathbf{r}_1, \dots, \mathbf{r}_N) \equiv V(\mathbf{r}^N). \quad (1.1)$$

The *force* \mathbf{f}_i acted on particle i is the negative gradient of V , which is again a function of the configuration \mathbf{r}^N :

$$\mathbf{f}_i(\mathbf{r}^N) = -\nabla_i V(\mathbf{r}^N) = -\frac{\partial V(\mathbf{r}^N)}{\partial \mathbf{r}_i}. \quad (1.2)$$

Then the future evolution of the system is given by the Newton's second law

$$m_i \ddot{\mathbf{r}}_i = \mathbf{f}_i(\mathbf{r}^N), \quad (1.3)$$

where m_i is the mass of particle m_i . This is system of N coupled second order differential equation. Additionally, it is often useful to introduce the *momentum* of a particle. In Cartesian coordinates, the momentum of a particle is given by

$$\mathbf{p}_i = m_i \dot{\mathbf{r}}_i. \quad (1.4)$$

When modelling a system composed of atoms and molecules, interatomic interactions, as opposed to the forces related to chemical bonds keeping molecules together, are relatively weak. A good, and extremely common approximation for intermolecular interactions is that they are *pairwise additive*. Moreover, for atoms, these pair potentials can be assumed to be central so that they depend only on the interatomic distances. Then the total potential V can be resolved into a sum of pair potentials v_{ij} between all pairs of atoms i and j , where v_{ij} is only a function of the interatomic distance $r_{ij} \equiv \|\mathbf{r}_{ij}\| := \|\mathbf{r}_i - \mathbf{r}_j\|$

$$V(\mathbf{r}^N) = \sum_{i=1}^N \sum_{j>i}^N v_{ij}(r_{ij}). \quad (1.5)$$

The sum is taken over $j > i$ so that each pair of atoms i and j is summed exactly once. This restriction can be lifted by noting v_{ij} and v_{ji} both describes the potential between particle i and j so they should be the same, and $r_{ij} = r_{ji}$, so

$$V(\mathbf{r}^N) = \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i}^N v_{ij}(r_{ij}). \quad (1.6)$$

The factor of a half cancels with the problem of double counting, and self interactions with $i = j$ is excluded.

If we identify the force acted on particle i by particle j is

$$\mathbf{f}_{ij} := -\frac{\partial v_{ij}}{\partial \mathbf{r}_i}, \quad (1.7)$$

then it is easy to show that the total force acted on particle i is the sum of the forces acted by all other particles using the form of potential given by (1.6)

$$\begin{aligned} \mathbf{f}_i &= -\frac{\partial V}{\partial \mathbf{r}_i} \\ &= -\frac{1}{2} \sum_{j \neq i}^N \left(\frac{\partial v_{ij}(r_{ij})}{\partial \mathbf{r}_i} + \frac{\partial v_{ji}(r_{ji})}{\partial \mathbf{r}_i} \right) \\ &= -\sum_{j \neq i}^N \frac{\partial v_{ij}(r_{ij})}{\partial \mathbf{r}_i} = \sum_{j \neq i}^N \mathbf{f}_{ij}. \end{aligned} \quad (1.8)$$

It is also not difficult to should that the pair forces satisfy the Newton's third law

$$\mathbf{f}_{ij} = -\mathbf{f}_{ji}. \quad (1.9)$$

1.1.2 Properties of Classical Dynamics

Energy Conservation

A fundamental property of mechanical systems, for pair or many-body interactions, provided that they can be derived from a potential invariant in time, is that the total energy is conserved during the motion. The total energy is sum of the potential energy V and the kinetic energy K defined by

$$K = \sum_{i=1}^N \frac{1}{2} m_i \dot{\mathbf{r}}_i^2. \quad (1.10)$$

We define the *Hamiltonian* of a system to be

$$H(\mathbf{r}^N, \dot{\mathbf{r}}^N, t) := K + V = \sum_{i=1}^N \frac{1}{2} m_i \dot{r}_i^2 + V(\mathbf{r}^N, t), \quad (1.11)$$

For our purposes, it is just another way of saying the total energy of the system.¹ In our cases, the potential V and hence the whole Hamiltonian has no explicit time dependence, meaning t does not appear explicitly in the expression of both K and V , although both \mathbf{r} and $\dot{\mathbf{r}}$ evolve in time. If this is the case, then the Hamiltonian (energy) does not change in time. This is the conservation of energy.

Theorem 1.1 (Energy Conservation). If $H(\mathbf{r}_i, \dot{\mathbf{r}}_i, t)$ has no explicit time dependence,

$$\frac{\partial H}{\partial t} = 0, \quad (1.13)$$

then H is a constant of motion,

$$\frac{dH}{dt} = 0. \quad (1.14)$$

¹Formally in classical mechanics, the Hamiltonian is defined as the Legendre transform of the Lagrangian, by

$$H = \sum_{i=1}^n p_i \dot{q}_i - L(q_i, \dot{q}_i, t). \quad (1.12)$$

p_i is shown as the *conjugated momentum* of the *generalised coordinate* q_i . Since the generalised coordinate is not necessarily the Cartesian coordinates (e.g. spherical, or even non-orthogonal), H may not be the energy.

Proof. By chain rule,

$$\frac{dH}{dt} = \frac{\partial H}{\partial t} + \sum_{i=1}^N \frac{\partial H}{\partial \mathbf{r}_i} \frac{\partial \mathbf{r}_i}{\partial t} + \sum_{i=1}^N \frac{\partial H}{\partial \dot{\mathbf{r}}_i} \frac{\partial \dot{\mathbf{r}}_i}{\partial t} \quad (1.15)$$

$$= 0 + \sum_{i=1}^N \frac{\partial V}{\partial \mathbf{r}_i} \cdot \dot{\mathbf{r}}_i + \sum_{i=1}^N m_i \dot{\mathbf{r}}_i \cdot \ddot{\mathbf{r}}_i \quad (1.16)$$

$$= \sum_{i=1}^N \dot{\mathbf{r}}_i \cdot (\mathbf{f}_i - m_i \ddot{\mathbf{r}}_i) = 0 \quad (1.17)$$

from Newton's second law. \square

Since the Hamiltonian is not explicitly dependent on time, the system has time-translational symmetry, meaning that we start the system at time t and at time $t + \delta t$, the system will evolve in the same way. This result of a continuous symmetry of the system leading to a conserved quantity is an example of the Noether's theorem, which is arguably the most beautiful and profound result in physics. We will briefly introduce this result in section A.

The conservation of energy is fundamental in the derivation of the equilibrium ensembles in statistical mechanics. It can also be applied as a very powerful test of the stability of a numerical scheme for the integration of the equations of motion. We will repeatedly return to this point.

Time Reversal Symmetry

Another feature of Newtonian dynamics that plays a role both in the theory of statistical mechanics and in the practice of the development of molecular dynamics algorithms is *time reversal symmetry*. This states that if we reverse all velocities at time t while keeping the positions the same, then the system will retrace its trajectory back into the past. We introduce the notation

$$\mathbf{r}^N(t \mid \mathbf{r}_0^N, \mathbf{p}_0^N) \quad (1.18)$$

to be \mathbf{r}^N under the initial condition $\mathbf{r}^N(0) = \mathbf{r}_0^N$ and $\mathbf{p}^N(0) = \mathbf{p}_0^N$. Then time reversal symmetry implies the relations

$$\mathbf{r}^N(t \mid \mathbf{r}^N(0), -\mathbf{p}^N(0)) = \mathbf{r}^N(-t \mid \mathbf{r}^N(0), \mathbf{p}^N(0)), \quad (1.19)$$

$$\mathbf{p}^N(t \mid \mathbf{r}^N(0), -\mathbf{p}^N(0)) = -\mathbf{p}^N(-t \mid \mathbf{r}^N(0), \mathbf{p}^N(0)). \quad (1.20)$$

1.1.3 The Euler's Algorithm

Molecular dynamics methods are iterative numerical schemes for solving the equations of motion of the system. The first step is to discretise the time into small intervals, and we assume each interval has equal length δt . Then the evolution of the system is then described by the series of coordinates and velocities

$$\mathbf{r}^N(t_0) \equiv \mathbf{r}^N(0), \dots, \mathbf{r}^N(t_{m-1}) \equiv \mathbf{r}^N(t_m - \delta t), \mathbf{r}^N(t_m), \mathbf{r}^N(t_{m+1}) \equiv \mathbf{r}^N(t_m + \delta t), \dots \quad (1.21)$$

$$\dot{\mathbf{r}}^N(t_0) \equiv \dot{\mathbf{r}}^N(0), \dots, \dot{\mathbf{r}}^N(t_{m-1}) \equiv \dot{\mathbf{r}}^N(t_m - \delta t), \dot{\mathbf{r}}^N(t_m), \dot{\mathbf{r}}^N(t_{m+1}) \equiv \dot{\mathbf{r}}^N(t_m + \delta t), \dots \quad (1.22)$$

The Schemes we will discuss all use the Cartesian coordinates.

The most fundamental integrator in molecular dynamics is the *Euler's algorithm*. It approximates

the position of the molecule at time $t + \delta t$ by a Taylor series about t truncated at $O(\delta t^2)$

$$\begin{aligned}\mathbf{r}_i(t + \delta t) &= \mathbf{r}_i(t) + \dot{\mathbf{r}}(t)\delta t + \frac{1}{2}\ddot{\mathbf{r}}(t)\delta t^2 + O(\delta t^3) \\ &= \mathbf{r}_i(t) + \delta t\mathbf{v}_i(t) + \frac{\delta t^2}{2m_i}\mathbf{f}_i(t) + O(\delta t^3).\end{aligned}\tag{1.23}$$

Similarly one can obtain an expansion for $\mathbf{v}_i(t + \delta t)$

$$\mathbf{v}_i(t + \delta t) = \mathbf{v}_i(t) + \frac{\delta t}{m_i}\mathbf{f}_i(t) + O(\delta t^2).\tag{1.24}$$

Having an expression of \mathbf{f}_i , which can be worked out from the potential V , we can perform this process iteratively.

Algorithm 1.2 (Euler's Algorithm).

$$\mathbf{r}_i(t + \delta t) = \mathbf{r}_i(t) + \delta t\mathbf{v}_i(t) + \frac{\delta t^2}{2m_i}\mathbf{f}_i(t)\tag{1.25}$$

$$\mathbf{v}_i(t + \delta t) = \mathbf{v}_i(t) + \frac{\delta t}{m_i}\mathbf{f}_i(t).\tag{1.26}$$

The error is $O(\delta t^3)$.

The Euler's algorithm is simple, but it has a huge drawback — it is simply not accurate enough.

To illustrate this, we calculated the equation of motion of a standard Harmonic oscillator, with Hamiltonian

$$H = \frac{1}{2}v^2 + \frac{1}{2}x^2\tag{1.27}$$

and initial conditions $x_0 = 0$ and $v_0 = 1$. The results are plotted in figure 1. The exact solution is a sine wave

$$x(t) = \sin t.\tag{1.28}$$

We can see that the Euler's solution quickly diverges from the exact solution as amplitude of oscillation quickly grows larger and larger. The Euler's algorithm has several problems, making it very limited in current practical use:

- The solution is not time reversible.
- *Liouville theorem* states that the volume of a set in the phase space should conserve as it evolve with time. In Euler's algorithm, the volume in the phase space is not conserved.
- The system is susceptible to energy drift. As we can see in the plot, the energy drifts exponentially fast.

This suggests that we need a more accurate algorithm.

1.1.4 Verlet Algorithm

An obvious thing to do to improve the Euler's algorithm is to include more terms in the Taylor expansions. If we truncate the expansion of $\mathbf{x}(t + \delta t)$ at $O(\delta t^3)$, we get

$$\mathbf{r}_i(t + \delta t) = \mathbf{r}_i(t) + \delta t\mathbf{v}_i(t) + \frac{\delta t^2}{2m_i}\mathbf{f}_i(t) + \frac{\delta t^3}{6}\mathbf{b}_i(t) + O(\delta t^4),\tag{1.29}$$

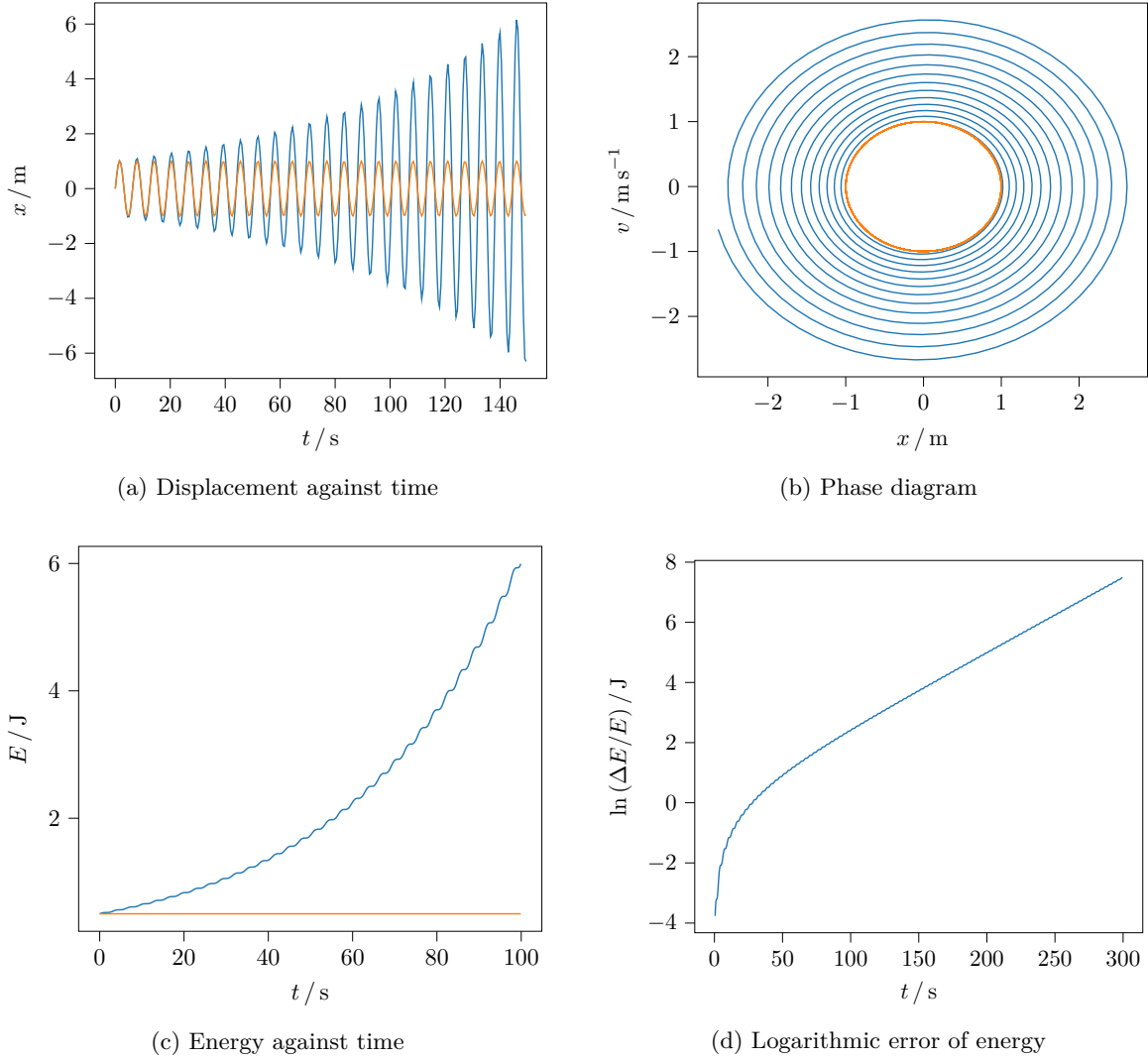


Figure 1: The motion of a harmonic oscillator solved by Euler's method (blue) and the exact solution (orange). The step length in Euler's method is set to be $\delta t = 0.05s$.

where $\mathbf{b}_i := \ddot{\mathbf{r}}_i$ is the third derivative of position. But this leads to a problem — it is not easy to evaluate this third derivative. But the trick is we don't have to evaluate it. If we expand the position backward in time, we get

$$\mathbf{r}_i(t - \delta t) = \mathbf{r}_i(t) - \delta t \mathbf{v}_i(t) + \frac{\delta t^2}{2m_i} \mathbf{f}_i(t) - \frac{\delta t^3}{6} \mathbf{b}_i(t) + O(\delta t^4). \quad (1.30)$$

Now if we add these two expressions together, the third derivatives nicely cancel out and we are left with

$$\mathbf{r}_i(t + \delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t - \delta t) + \frac{\delta t^2}{m_i} \mathbf{f}_i(t) + O(\delta t^4). \quad (1.31)$$

Now this expression is accurate to $O(\delta t^3)$, and we don't even need to evaluate the velocities $\mathbf{v}(t)$ if we are only interested in time.

However, in most cases, the velocities still tell us valuable informations. To calculate it, we can subtract the two expansions forward and backward in time and get

$$\mathbf{v}_i(t) = \frac{1}{2\delta t} [\mathbf{r}_i(t + \delta t) - \mathbf{r}_i(t - \delta t)] + O(\delta t^3). \quad (1.32)$$

Algorithm 1.3 (Verlet Algorithm).

$$\mathbf{r}_i(t + \delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t - \delta t) + \frac{\delta t^2}{m_i} \mathbf{f}_i(t) \quad (1.33)$$

$$\mathbf{v}_i(t) = \frac{1}{2\delta t} [\mathbf{r}_i(t + \delta t) - \mathbf{r}_i(t - \delta t)]. \quad (1.34)$$

The error of displacement is $O(\delta t^4)$ and that of velocity is $O(\delta t^3)$.

However, to calculate $\mathbf{v}(t)$, you need the knowledge of position at $t + \delta t$, i.e. the velocity update in the Verlet algorithm is one step behind the position update. This is not a problem for propagating position because, assuming that the forces are not dependent on velocity, information on $\mathbf{v}_i(t)$ is not needed in (equation (1.33)). Still, this may be inconvenient for the determination of velocity-dependent quantities or for the algorithms which manipulate the velocity during dynamics. The position and velocity update can be brought in the same step by a reformulation of the Verlet scheme, called *velocity Verlet*. The prediction for position is now simply obtained from the Taylor expansion, again keeping to the second order

$$\mathbf{r}_i(t + \delta t) = \mathbf{r}_i(t) + \delta t \mathbf{v}_i(t) + \frac{\delta t^2}{2m_i} \mathbf{f}_i(t) + O(\delta t^3). \quad (1.35)$$

For the advanced position obtained this way we compute the force at time $t + \delta t$

$$\mathbf{f}_i(t + \delta t) = \mathbf{f}_i(\{\mathbf{r}_j(t + \delta t)\}) = \mathbf{f}_i\left(\left\{\mathbf{r}_i(t) + \delta t \mathbf{v}_i(t) + \frac{\delta t^2}{2m_i} \mathbf{f}_i(t)\right\}\right), \quad (1.36)$$

in which all particles have proceeded to their positions at $t + \delta t$. Substituting this expression into the Taylor expansion of $\mathbf{r}_i(t)$ about $t + \delta t$ backward in time, we obtain

$$\mathbf{r}_i(t) = \mathbf{r}_i(t + \delta t) - \delta t \mathbf{v}_i(t + \delta t) + \frac{\delta t^2}{2m_i} \mathbf{f}_i(t + \delta t) + O(\delta t^3). \quad (1.37)$$

Adding this to the forward expansion (1.35) gives the prediction of velocity

$$\mathbf{v}_i(t + \delta t) = \mathbf{v}_i(t) + \frac{\delta t}{2m_i} [\mathbf{f}_i(t) + \mathbf{f}_i(t + \delta t)] + O(\delta t^3). \quad (1.38)$$

Algorithm 1.4 (Velocity Verlet Algorithm).

$$\mathbf{r}_i(t + \delta t) = \mathbf{r}_i(t) + \delta t \mathbf{v}_i(t) + \frac{\delta t^2}{2m_i} \mathbf{f}_i(t) \quad (1.39)$$

$$\mathbf{v}_i(t + \delta t) = \mathbf{v}_i(t) + \frac{\delta t}{2m_i} [\mathbf{f}_i(t) + \mathbf{f}_i(t + \delta t)]. \quad (1.40)$$

The velocity Verlet looks rather different than the Verlet algorithm, especially the $O(\delta t^3)$ error terms when we derive it look concerning. However, we can show that these two algorithms are equivalent. We can show this by rewriting the velocity Verlet prediction of the position.

Proposition 1.5. The velocity Verlet algorithm is equivalent to the Verlet algorithm.

Proof. If we subtract the $t - \delta t \rightarrow t$ prediction for the position from the $t \rightarrow t + \delta t$ prediction in the velocity Verlet algorithm, we find

$$\mathbf{r}_i(t + \delta t) - \mathbf{r}_i(t) = \mathbf{r}_i(t) - \mathbf{r}_i(t - \delta t) + \delta t [\mathbf{v}_i(t) - \mathbf{v}_i(t - \delta t)] + \frac{\delta t^2}{2m_i} [\mathbf{f}_i(t) - \mathbf{f}_i(t - \delta t)]. \quad (1.41)$$

The $t - \delta t \rightarrow t$ update for the velocity is

$$\mathbf{v}_i(t) = \mathbf{v}_i(t - \delta t) + \frac{\delta t}{2m_i} [\mathbf{f}_i(t - \delta t) + \mathbf{f}_i(t)]. \quad (1.42)$$



Figure 2: Schemes of Verlet and velocity Verlet algorithms.

If we substitute this into (1.41), we get

$$\mathbf{r}_i(t + \delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t - \delta t) + \frac{\delta t^2}{m_i} \mathbf{f}_i(t). \quad (1.43)$$

The velocity Verlet algorithm gives the same prediction with the Verlet algorithm. \square

Now let's examine the accuracy of Verlet's algorithm using harmonic oscillator.

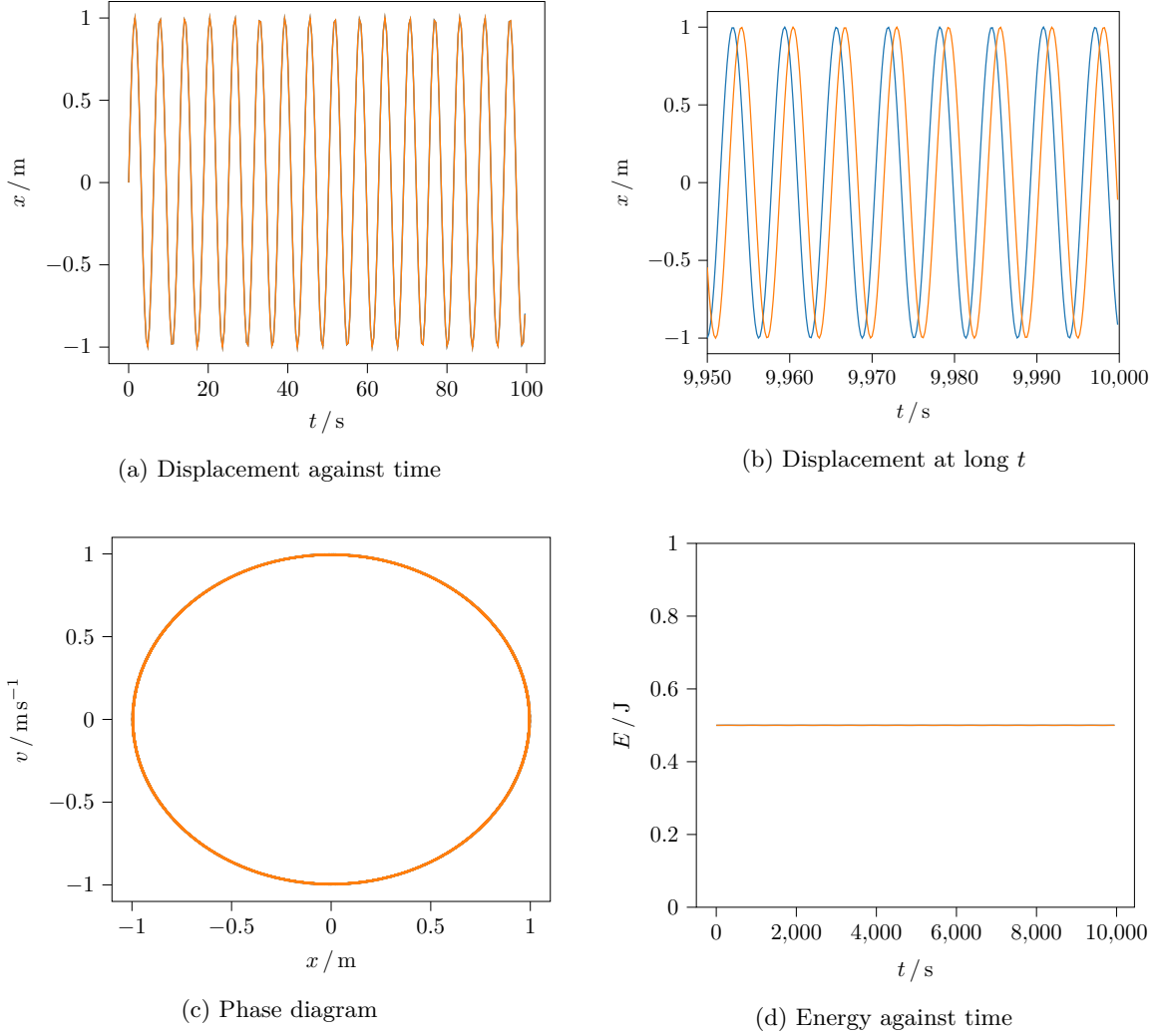


Figure 3: The motion of a harmonic oscillator solved by Verlet algorithm (blue) and the exact solution (orange). The step length in Verlet's method is also $\delta t = 0.05$ s.

We can see that the Verlet algorithm is very accurate. Although the displacements start to deviate from the exact displacement and a very long time, the energy and the volume in the phase space is always conserved.

1.1.5 Why use the Verlet Algorithm?

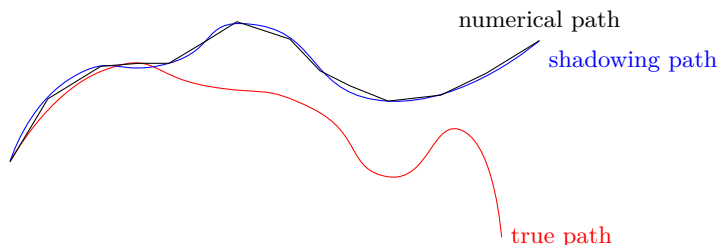
While there are algorithms with better short time accuracy than the Verlet algorithm, the overwhelming majority of condensed matter molecular dynamics simulations is based on just the Verlet algorithm. There are a number of reasons for its popularity.

- (i) The Verlet algorithm is simple and only depends on forces. No higher derivatives of the energy is needed. This is important because the force evaluation is the most CPU time consuming in MD simulations of interacting many-particle systems. Computation of higher derivatives of energy will increase computational costs substantially. Although the algorithms using force derivatives are more accurate, this gain is actually relatively minor. Because of the chaotic nature of the motion in many-particle systems, the particles rapidly deviate from the “true” trajectories. This is known as *Lyapunov instability*: trajectories that differ slightly in initial conditions will diverge exponentially in time. If we denote $\mathbf{r}(t) = \mathbf{r}(t \mid \mathbf{r}_0, \mathbf{p}_0)$ and $\mathbf{r}'(t) = \mathbf{r}(t \mid \mathbf{r}_0, \mathbf{p}_0 + \epsilon)$, then

$$|\mathbf{r}(t) - \mathbf{r}'(t)| \sim \epsilon \exp(\lambda t), \quad (1.44)$$

where λ is the *Lyapunov exponent*. This seems like a huge problem, but it is actually not as big a problem as it seems!

Theorem 1.6 (Shadowing theorem). Every numerical trajectory will be uniformly close to some true trajectory with slightly altered initial position. In other words, a numerical trajectory is “shadowed” by a true one.



This strong molecular chaos is ultimately the justification of methods of statistical mechanics.

- (ii) Even though only using forces, the Verlet algorithm is correct up to and including $O(\delta t^3)$.
- (iii) The Verlet algorithm is explicitly time reversible and, even though the trajectory relatively quickly diverges substantially from the true trajectory, the energy is conserved over an extremely long period of time. Moreover, the Verlet algorithm rigorously conserves the normalisation of an ensemble probability distribution of points in phase space. In more advanced language, the Verlet integrator is said to be *symplectic*. These formal properties contribute to the superior long time stability of the Verlet algorithm.

For example, as we will discuss later, energy is the defining quantity for the microcanonical ensemble, since, for chaotic systems, there are no constraints on the regions trajectories can reach in phase space other than that they are confined to the hypersurface of constant energy. Energy conservation, together with norm conservation are therefore necessary conditions for thermodynamic stability, and ultimately for a proper definition of temperature. Long time stability is particularly important for the simulation of liquids which are stabilised by finite temperature dynamical fluctuations.

Appendices

A Noether's Theorem

The simplest way of deriving the Noether's theorem is to use the Lagrangian mechanics, which is another way of formulating classical mechanics. First let's be clear of our notations. For a system of N particles in d dimensions, we will rewrite the coordinates \mathbf{r}_i as x^A , where $A = 1, \dots, dN$. The Newton's equations are

$$\dot{p}_A = -\frac{\partial V}{\partial x^A}, \quad (\text{A.1})$$

where $p_A = m_A \dot{x}^A$. To reduce the cluttering in notations, when we write x^A in the argument of a function, we mean that it is a function of all x^A .

Lagrangian mechanics starts from defining the Lagrangian of a system.

Definition A.1. The *Lagrangian* for a system is defined by

$$L(x^A, \dot{x}^A) = T(\dot{x}^A) - V(x^A), \quad (\text{A.2})$$

where $T = \frac{1}{2} \sum_A m_A (\dot{x}^A)^2$ is the kinetic energy and $V(x^A)$ is the potential energy.

Note the weird minus sign between the kinetic and the potential energy. Despite this strange definition of the Lagrangian, it works really elegantly.

If we know that at $t = t_0$, the particles are at $x^A(t_0) = x_0^A$, and at $t = t_1$, the particles are at $x^A(t_1) = x_1^A$, there are infinite ways the systems can evolve with times between these two end points. How do we find the true paths $x^A(t)$ taken by the particles?

Theorem A.2 (Principle of Least Action). The actual path taken by the system is an extremum of the *action*, defined by

$$S[x^A(t)] = \int_{t_0}^{t_1} dt L(x^A(t), \dot{x}^A(t)). \quad (\text{A.3})$$

The S is an example of a *functional*. It maps functions to a number.

Proof. Consider varying a given path slightly, so

$$x^A(t) \longrightarrow x^A(t) + \delta x^A(t), \quad (\text{A.4})$$

where we fix the end points of the path by demanding $\delta x^A(t_0) = \delta x^A(t_1) = 0$. Then this results in a change in the action

$$\delta S = \delta \left[\int_{t_0}^{t_1} dt L \right] \quad (\text{A.5})$$

$$= \int_{t_0}^{t_1} dt \delta L \quad (\text{A.6})$$

$$= \int_{t_0}^{t_1} dt \sum_A \frac{\partial L}{\partial x^A} \delta x^A + \frac{\partial L}{\partial \dot{x}^A} \delta \dot{x}^A. \quad (\text{A.7})$$

We integrate the second term by parts to get

$$\delta S = \int_{t_0}^{t_1} dt \sum_A \left[\frac{\partial L}{\partial x^A} - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}^A} \right) \right] \delta x^A + \left[\frac{\partial L}{\partial \dot{x}^A} \delta x^A \right]_{t_0}^{t_1}. \quad (\text{A.8})$$

The boundary term vanishes since we required $\delta x^A(t_0) = \delta x^A(t_1) = 0$. At an extremum of the action S , $\delta S = 0$ for all changes in the path $\delta x^A(t)$. This holds if and only if

$$\frac{\partial L}{\partial x^A} - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}^A} \right) = 0. \quad (\text{A.9})$$

for all A . These are known as the Euler–Lagrange equations. To finish the proof, we only need to show that Euler–Lagrange equations are equivalent to Newton’s equations. From the definition of the Lagrangian, we have

$$\frac{\partial L}{\partial x^A} = -\frac{\partial V}{\partial x^A}, \quad (\text{A.10})$$

while

$$\frac{\partial L}{\partial \dot{x}^A} = p_A. \quad (\text{A.11})$$

Then it’s easy to see that Newton’s equations (A.1) are indeed equivalent to Euler–Lagrange equations (A.9). \square

In fact Lagrangian mechanics is much more powerful than that. It turns out we can use any generalised coordinate we want (e.g. spherical, hyperbolic, or just some arbitrary parameters that uniquely defines the configuration of the system), and we may add constraints to the coordinates, making it much more powerful than Newton’s formulation of classical mechanics. Unfortunately, we can’t go into too much of detail here. If you are interested, see e.g. Prof. David Tong’s notes on Classical Dynamics. But the important conclusion is that for any Lagrangian written in generalised coordinates $L(q_i, \dot{q}_i, t)$, the Euler–Lagrange equations still hold:

$$\frac{\partial L}{\partial q_i} - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) = 0. \quad (\text{A.12})$$

Definition A.3. Consider a one-parameter transformation of maps

$$q_i(t) \longrightarrow Q_i(s, t) \quad (\text{A.13})$$

for $s \in \mathbb{R}$ such that $Q_i(0, t) = q_i(t)$. Then this transformation is said to be a *continuous symmetry* of the Lagrangian L if

$$\frac{\partial}{\partial s} L(Q_i(s, t), \dot{Q}_i(s, t), t) = 0. \quad (\text{A.14})$$

Theorem A.4 (Noether’s theorem). For each continuous symmetry, there is a conserved quantity.

Proof.

$$\frac{\partial L}{\partial s} = \sum_i \frac{\partial L}{\partial Q_i} \frac{\partial Q_i}{\partial s} + \frac{\partial L}{\partial \dot{Q}_i} \frac{\partial \dot{Q}_i}{\partial s}, \quad (\text{A.15})$$

so we have

$$\begin{aligned} 0 &= \frac{\partial L}{\partial s} \Big|_{s=0} = \sum_i \frac{\partial L}{\partial Q_i} \frac{\partial Q_i}{\partial s} \Big|_{s=0} + \frac{\partial L}{\partial \dot{Q}_i} \frac{\partial \dot{Q}_i}{\partial s} \Big|_{s=0} \\ &= \sum_i \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{Q}_i} \right) \frac{\partial Q_i}{\partial s} \Big|_{s=0} + \frac{\partial L}{\partial \dot{Q}_i} \frac{\partial \dot{Q}_i}{\partial s} \Big|_{s=0} \\ &= \frac{d}{dt} \left(\sum_i \frac{\partial L}{\partial \dot{Q}_i} \frac{\partial Q_i}{\partial s} \Big|_{s=0} \right). \end{aligned} \quad (\text{A.16})$$

The quantity

$$\sum_i \frac{\partial L}{\partial \dot{Q}_i} \frac{\partial Q_i}{\partial s} \Big|_{s=0} \quad (\text{A.17})$$

is constant for all time. \square

Let's find some examples.

Example. Homogeneity of space.

Consider a system of N particles with Lagrangian

$$L = \frac{1}{2} \sum_i m_i \dot{\mathbf{r}}_i^2 - V(r_{ij}), \quad (\text{A.18})$$

where $V(r_{ij})$ means that the potential is only dependent on the relative distances $r_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$ between particles, not on their absolute positions. Then this Lagrangian has symmetry of translation: $\mathbf{r}_i \rightarrow \mathbf{r}_i + s\mathbf{n}$ for any vector \mathbf{n} and real number s .

$$L(\mathbf{r}_i, \dot{\mathbf{r}}_i, t) = L(\mathbf{r}_i + s\mathbf{n}, \dot{\mathbf{r}}_i, t). \quad (\text{A.19})$$

Then by Noether's theorem, the constant that holds in constant is

$$\sum_n \frac{\partial L}{\partial \dot{\mathbf{r}}_i} \cdot \mathbf{n} = \sum_i \mathbf{p}_i \cdot \mathbf{n}. \quad (\text{A.20})$$

The component of linear momentum in any direction is conserved, as so

$$\sum_i \mathbf{p}_i \quad (\text{A.21})$$

is also conserved.

Homogeneity in space \implies translational invariance of $L \implies$ conservation of total linear momentum.

Example. Isotropy of Space.

The isotropy of space means that a closed system is invariant under rotations around an axis $\hat{\mathbf{n}}$, so all $\mathbf{r}_i \rightarrow \mathbf{r}'_i$ are rotated by the same amount. To work out the corresponding conserved quantity it suffices to work with the infinitesimal form of the rotations

$$\mathbf{r}_i \longrightarrow \mathbf{r}_i + \delta \mathbf{r}_i = \mathbf{r}_i + \alpha \hat{\mathbf{n}} \times \mathbf{r}_i, \quad (\text{A.22})$$

where α is infinitesimal. To see that this is indeed a rotation, you can calculate the length of the vector and notice it is preserved to linear order in α . Then we have

$$L(\mathbf{r}_i, \dot{\mathbf{r}}_i) = L(\mathbf{r}_i + \alpha \hat{\mathbf{n}} \times \mathbf{r}_i, \dot{\mathbf{r}}_i + \alpha \hat{\mathbf{n}} \times \dot{\mathbf{r}}_i), \quad (\text{A.23})$$

giving us the conserved quantity

$$\sum_i \frac{\partial L}{\partial \dot{\mathbf{r}}_i} \cdot (\hat{\mathbf{n}} \times \mathbf{r}_i) = \sum_i \hat{\mathbf{n}} \cdot (\mathbf{r}_i \times \mathbf{p}_i) = \hat{\mathbf{n}} \cdot \mathbf{K}. \quad (\text{A.24})$$

This is the component of the total angular momentum in the direction $\hat{\mathbf{n}}$. Since $\hat{\mathbf{n}}$ is arbitrary, \mathbf{L} is conserved.

Isotropy of space \implies rotational invariance of $L \implies$ conservation of total angular momentum.