

Computational Statistical Physics

Part II: Interacting Particles and Molecular Dynamics

Oded Zilberberg

March 30, 2020

Contents

<i>2.1</i>	<i>Molecular Dynamics</i>	<i>2-3</i>
<i>2.1.1</i>	<i>Introduction</i>	<i>2-3</i>
<i>2.1.2</i>	<i>Equations of Motion</i>	<i>2-5</i>
<i>2.1.3</i>	<i>Contact Time</i>	<i>2-6</i>
<i>2.1.4</i>	<i>Verlet Method</i>	<i>2-7</i>
<i>2.1.5</i>	<i>Leapfrog Method</i>	<i>2-8</i>
<i>2.2</i>	<i>Optimization</i>	<i>2-9</i>
<i>2.2.1</i>	<i>Verlet Tables</i>	<i>2-10</i>
<i>2.2.2</i>	<i>Linked-Cell Method</i>	<i>2-11</i>
<i>2.3</i>	<i>Dynamics of Composed Particles</i>	<i>2-11</i>
<i>2.3.1</i>	<i>Lagrange Multipliers</i>	<i>2-12</i>
<i>2.3.2</i>	<i>Rigid Bodies</i>	<i>2-13</i>
<i>2.4</i>	<i>Long-range Potentials</i>	<i>2-18</i>
<i>2.4.1</i>	<i>Ewald Summation</i>	<i>2-18</i>
<i>2.4.2</i>	<i>Particle-Mesh Method</i>	<i>2-20</i>
<i>2.4.3</i>	<i>Reaction Field Method</i>	<i>2-23</i>
<i>2.5</i>	<i>Canonical Ensemble</i>	<i>2-24</i>
<i>2.5.1</i>	<i>Velocity Rescaling</i>	<i>2-25</i>
<i>2.5.2</i>	<i>Constraint Method</i>	<i>2-25</i>
<i>2.5.3</i>	<i>Nosé-Hoover Thermostat</i>	<i>2-26</i>
<i>2.5.4</i>	<i>Stochastic Method</i>	<i>2-29</i>
<i>2.5.5</i>	<i>Constant Pressure</i>	<i>2-29</i>
<i>2.5.6</i>	<i>Parrinello-Rahman Barostat</i>	<i>2-31</i>

2.6	<i>Event-driven Molecular Dynamics</i>	2-31
2.6.1	<i>Elastic Collisions</i>	2-32
2.6.2	<i>Lubachevsky Method</i>	2-33
2.6.3	<i>Collision with Perfect Slip</i>	2-34
2.6.4	<i>Collision with Rotation</i>	2-35
2.6.5	<i>Inelastic Collisions</i>	2-37
2.6.6	<i>Inelastic Collapse</i>	2-39
2.7	<i>Inelastic Collisions in Molecular Dynamics</i>	2-40
2.7.1	<i>Damped Oscillator</i>	2-40
2.7.2	<i>Plastic Deformation</i>	2-41
2.7.3	<i>Coulomb Friction and Discrete Element Method</i>	2-42
2.8	<i>Arbitrary Shapes</i>	2-43
2.8.1	<i>Ellipsoidal Particles</i>	2-43
2.8.2	<i>Polygons</i>	2-46
2.8.3	<i>Spheropolygons</i>	2-46
2.9	<i>Contact Dynamics</i>	2-48
2.9.1	<i>One-dimensional Contact</i>	2-49
2.9.2	<i>Generalization to N Particles</i>	2-52
2.10	<i>Particles in Fluids</i>	2-54
2.10.1	<i>Continuum Methods</i>	2-55
2.10.2	<i>Stokesian Dynamics</i>	2-56
2.10.3	<i>Lattice Boltzmann Method</i>	2-57
2.10.4	<i>Stochastic Rotation Dynamics</i>	2-57
2.10.5	<i>Direct Simulation Monte Carlo</i>	2-58
2.10.6	<i>Dissipative Particle Dynamics</i>	2-59
2.10.7	<i>Smoothed Particle Hydrodynamics</i>	2-59
2.11	<i>Quantum Mechanical Approaches</i>	2-60
2.11.1	<i>Introduction</i>	2-60
2.11.2	<i>Implementation of Wave Functions</i>	2-63
2.11.3	<i>Born-Oppenheimer Approximation</i>	2-63
2.11.4	<i>Hohenberg-Kohn Theorems</i>	2-63
2.11.5	<i>Kohn-Sham Approximation</i>	2-65
2.11.6	<i>Hellmann-Feynman Theorem</i>	2-67
2.11.7	<i>Car-Parrinello Method</i>	2-68

2.1 Molecular Dynamics

In the first part of this lecture, we introduced methods from statistical physics to describe the properties of a large number of interacting units, e.g., particles, spins or other entities. Instead of a statistical description, we now focus on modeling the microscopic particle interactions by solving the corresponding equations of motion. This technique is also known as *molecular dynamics* (henceforth MD). We start with some straightforward and intuitive methods such as the so-called Verlet and Leapfrog schemes for solving Newton's equations of motion. In what follows, we discuss Lagrange multipliers and long-range potential methods as tools to simulate composed and long-range interacting particle systems. We conclude this section by introducing event-driven MD simulations. For further reading, we refer to Ref. [1] which covers most of the topics treated in this section in a more detailed manner.

2.1.1 Introduction

In this section, we discuss methods to simulate the interactions of particles. We begin our discussion with the description of classical systems what implies that we have to solve Newton's equations of motion. The mathematical framework of classical mechanics dates back to the work of Isaac Newton in the 17th century [2]. But only with the rise of computers in the second half of the 20th century, it became possible to perform first MD simulations¹. Many of the techniques contained in this chapter form the basis of modern commercial softwares that are frequently applied to many engineering and industrial problems.

One of the pioneers of this field is Bernie Alder [3]. He was one of the first who developed MD methods to computationally study the interactions of particles.

To model interacting particle systems, we use generalized coordinates

$$\mathbf{q}_i = (q_i^1, \dots, q_i^d) \quad \text{and} \quad \mathbf{p}_i = (p_i^1, \dots, p_i^d) \quad (2.1)$$

in a system where each particle has d degrees of freedom. We then describe the system consisting of N particles by

$$Q = (\mathbf{q}_1, \dots, \mathbf{q}_N) \quad \text{and} \quad P = (\mathbf{p}_1, \dots, \mathbf{p}_N), \quad (2.2)$$

and the Hamiltonian

$$\mathcal{H}(P, Q) = K(P) + V(Q) \quad (2.3)$$

with $K(P) = \sum_{i,k} \frac{(p_i^k)^2}{2m_i}$ being the kinetic energy, m_i the mass of the i^{th} particle and $V(Q)$ the potential energy. The sum over $k \in \{1, \dots, d\}$

¹ Numerical methods are almost as old as mathematics itself. Simple calculations such as linear interpolation and square root approximations were developed a few thousands of years ago (e.g., the Babylonian clay tablet). More refined methods in differential analysis started to become of great interest with the rise of physical sciences in the 17th century. As an example, Newton's or Euler's method was described by Euler in 1768 and has been certainly known for long time. Even *Runge-Kutta* methods, that are still used today for very precise calculations, e.g., in celestial mechanics, were developed around 1900.

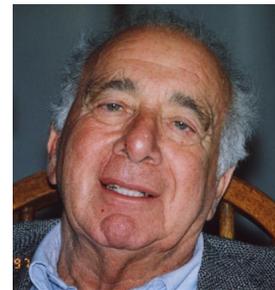


Figure 2.1: Bernie Alder

accounts for the d degrees of freedom.

The potential (e.g., an attractive or repulsive electromagnetic potential) determines the interactions of all particles among each other, and therefore their dynamics. An expansion of the potential energy yields

$$V(Q) = \sum_i v_1(q_i) + \sum_i \sum_{j>i} v_2(q_i, q_j) + \sum_i \sum_{j>i} \sum_{k>j} v_3(q_i, q_j, q_k) + \dots \quad (2.4)$$

The used summation ($j > i, k > j > i$) avoids to sum over any pair twice. The first term $v_1(\cdot)$ describes interactions of particles with the environment (e.g., boundary conditions for particle-wall interactions). The second term $v_2(\cdot)$ accounts for pairwise interactions.

Typically three or more body interactions are neglected and their effect is considered in an effective two body interaction described by

$$v_2^{\text{eff}}(q_i, q_j) = v^{\text{attr}}(r) + v^{\text{rep}}(r) \quad \text{with} \quad r = |\mathbf{q}_i - \mathbf{q}_j|, \quad (2.5)$$

where $v^{\text{attr}}(r)$ and $v^{\text{rep}}(r)$ represent attractive and repulsive parts of the potential, respectively. Examples of potentials involving three particles are the Stillinger–Weber potential [4] and the Axilrod–Teller potential [5].

For now, we only consider potentials that depend on distance, and not on particle orientation. Analytically, the simplest potential is the hard sphere interaction potential

$$v^{\text{rep}}(r) = \begin{cases} \infty & \text{if } r < \sigma, \\ 0 & \text{if } r \geq \sigma. \end{cases} \quad (2.6)$$

An illustration of the hard sphere potential is shown in Fig. 2.2. This potential is not well-suited for numerical simulations due to the fact that the force is infinite at $r = \sigma$. A smoother variant of the potential should be used in numerical simulations. One possibility is the use of a potential which describes the spring-like repulsion according to

$$v^{\text{rep}}(r) = \begin{cases} \frac{k}{2}(R-r)^2 & \text{if } r < R \\ 0 & \text{if } r > R \end{cases} \quad \text{with} \quad R = R_1 + R_2, \quad (2.7)$$

where k is the elastic spring constant, and $R = R_1 + R_2$ the sum of the radii of the individual particles. Such a soft sphere interaction may lead to unrealistic simulation outcomes (overlapping particles). More realistic choices are based on combinations of the two mentioned and additional potentials. Depending on the underlying potential, the dynamics will be different, and the computation might be easier or more cumbersome. For example, it is possible to define a cutoff for certain short-range potentials what minimizes the computational effort. This is, however, not applicable to long-range potentials which are therefore more challenging to simulate.

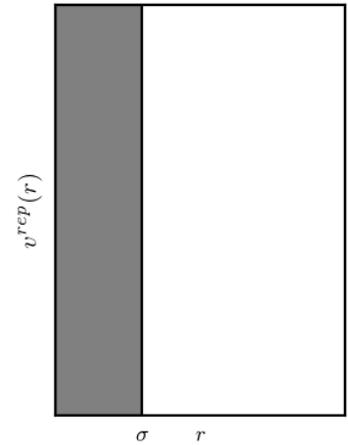


Figure 2.2: An example of a hard sphere potential.

2.1.2 Equations of Motion

Another important potential for modeling molecular interactions is the *Lennard-Jones* potential

$$v^{\text{LJ}}(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right], \quad (2.8)$$

where ϵ is the attractive energy and σ the interaction range. It is a mathematically simple model that approximates the spherically symmetric interaction between a pair of neutral atoms or molecules. The first term accounts for Pauli repulsion at short ranges and the second term describes attractive van der Waals forces. An example of a Lennard-Jones potential is shown in Fig. 2.3.

Once the interaction potential has been defined, we can easily derive the equations of motion using the Hamilton equations

$$\dot{q}_i^k = \frac{\partial \mathcal{H}}{\partial p_i^k}, \quad \dot{p}_i^k = -\frac{\partial \mathcal{H}}{\partial q_i^k}, \quad (2.9)$$

where $k \in \{1, \dots, d\}$ and $i \in \{1, \dots, N\}$.

For every particle, we identify \mathbf{q}_i with the position vector \mathbf{x}_i and $\dot{\mathbf{q}}_i = \dot{\mathbf{x}}_i$ with the velocity vector \mathbf{v}_i . Due to $\dot{\mathbf{x}}_i = \mathbf{v}_i = \mathbf{p}_i/m_i$ and $\dot{\mathbf{p}}_i = -\nabla_i V(Q) = \mathbf{f}_i$, the equations of motion are

$$m_i \ddot{\mathbf{x}}_i = \dot{\mathbf{p}}_i = \mathbf{f}_i = \sum_j \mathbf{f}_{ij}, \quad (2.10)$$

where \mathbf{f}_{ij} is the force exerted on particle i by particle j . Simulating Hamiltonian dynamics implies that our molecular system may exhibit certain conservation laws. First, the total energy is conserved if our considered Hamiltonian $\mathcal{H}(P, Q)$ is not explicitly time dependent (i.e., if $\partial_t \mathcal{H} = 0$). Second, if the system is translational invariant in a certain direction, then the corresponding momentum is conserved. And third, if the simulated system is rotational invariant about a certain axis, the corresponding angular momentum component is conserved. For example, a cubic box with periodic boundary conditions leads to a conserved total momentum

$$\mathbf{P} = \sum_i \mathbf{p}_i. \quad (2.11)$$

For a spherical symmetric box, the total angular momentum

$$\mathbf{L} = \sum_i \mathbf{x}_i \wedge \mathbf{p}_i \quad (2.12)$$

about the center of symmetry is conserved.

We now have to solve the last equations of motion defined by Eq. (2.10) with the help of appropriate numerical integration methods. When numerically computing the motion of particles, it is very important to use

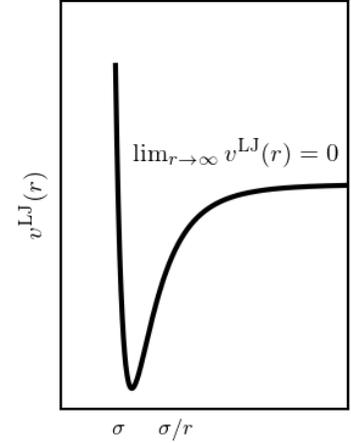


Figure 2.3: An example of a Lennard-Jones potential, cf. <http://www.atomsinmotion.com/>.

a sufficiently small time step Δt . If our chosen time step is too large, we may encounter unrealistic overlaps and particles may pass through the interaction range of other particles without interacting with them. We therefore need a measure that estimates the necessary time step and the corresponding integration error. Such a measure is the so-called *contact time*.

2.1.3 Contact Time

Due to the fact that we consider an interaction force which only depends on distance, we first analyze our particle dynamics in a one-dimensional setting. The interaction of the particle with the potential

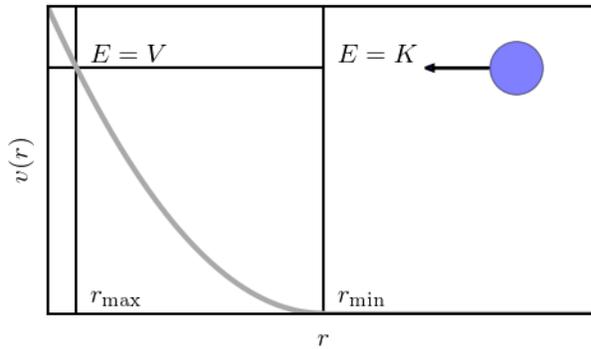


Figure 2.4: Derivation of the contact time.

is illustrated in Fig. 2.4. Using the equations for energy

$$E = \frac{1}{2}m\dot{r}^2 + V(r) = \text{const.} \quad (2.13)$$

and radial velocity

$$\frac{dr}{dt} = \left[\frac{2}{m} (E - V(r)) \right]^{\frac{1}{2}}, \quad (2.14)$$

we find for the contact time

$$t_c = 2 \int_0^{\frac{1}{2}t_c} dt = 2 \int_{r_{\min}}^{r_{\max}} \frac{dt}{dr} dr = 2 \int_{r_{\min}}^{r_{\max}} \left[\frac{2}{m} (E - V(r)) \right]^{-\frac{1}{2}} dr, \quad (2.15)$$

where r_{\min} and r_{\max} are the range of the potential and the turning point of a colliding particle, respectively.

The contact time is therefore an estimate for the appropriate time step of a MD simulation.

We expect reasonable results only if the time step is not larger than the smallest contact time. The time integration of the equations of motion is then possible using an integration method such as

- Euler’s method,
- Runge-Kutta methods,
- Predictor-corrector methods,
- Verlet methods,
- Leap-frog methods.

In the subsequent sections, we only focus on the last two methods which have been developed specifically for solving Newton’s equations.

2.1.4 Verlet Method

This integration method was developed by Loup Verlet to solve Newton’s equations of motion [6]. The procedure is simple and related to forward Euler integration. We begin with a Taylor expansion of $x(t + \Delta t)$ for sufficiently small time steps Δt so that

$$\begin{aligned} \mathbf{x}(t + \Delta t) &= \mathbf{x}(t) + \Delta t \mathbf{v}(t) + \frac{1}{2} \Delta t^2 \dot{\mathbf{v}} + \mathcal{O}(\Delta t^3), \\ \mathbf{x}(t - \Delta t) &= \mathbf{x}(t) - \Delta t \mathbf{v}(t) + \frac{1}{2} \Delta t^2 \dot{\mathbf{v}} - \mathcal{O}(\Delta t^3). \end{aligned} \quad (2.16)$$

Adding the last two expressions yields

$$\mathbf{x}(t + \Delta t) = 2\mathbf{x}(t) - \mathbf{x}(t - \Delta t) + \Delta t^2 \ddot{\mathbf{x}}(t) + \mathcal{O}(\Delta t^4). \quad (2.17)$$

Newton’s second law enables us to express $\ddot{\mathbf{x}}(t)$ for the i^{th} particle as

$$\ddot{\mathbf{x}}_i(t) = \frac{1}{m_i} \sum_j \mathbf{f}_{ij}(t) \quad \text{with} \quad \mathbf{f}_{ij}(t) = -\nabla V(r_{ij}(t)). \quad (2.18)$$

The particle trajectories are then computed by plugging in the last results in Eq. (2.17). Typically, we use a time step of approximately $\Delta t \approx t_c/20$.

Some general remarks about the Verlet method:

- Two time steps need to be stored (t and $t - \Delta t$).
- Velocities can be computed with $\mathbf{v}(t) = \frac{\mathbf{x}(t+\Delta t) - \mathbf{x}(t-\Delta t)}{2\Delta t}$.
- The local numerical error is of order $\mathcal{O}(\Delta t^4)$, i.e., it is globally a third order algorithm.
- The numbers which are added are of order $\mathcal{O}(\Delta t^0)$ and $\mathcal{O}(\Delta t^2)$.
- Improvable by systematical inclusion of higher orders (very inefficient).

- The method is time reversible, which allows to estimate the error accumulation by reversing the process and comparing it to the initial conditions.

2.1.5 Leapfrog Method

For the derivation of the Leapfrog method, we consider velocities at intermediate steps

$$\begin{aligned} \mathbf{v}\left(t + \frac{1}{2}\Delta t\right) &= \mathbf{v}(t) + \frac{1}{2}\Delta t \dot{\mathbf{v}}(t) + \mathcal{O}\left(\Delta t^2\right), \\ \mathbf{v}\left(t - \frac{1}{2}\Delta t\right) &= \mathbf{v}(t) - \frac{1}{2}\Delta t \dot{\mathbf{v}}(t) + \mathcal{O}\left(\Delta t^2\right). \end{aligned} \quad (2.19)$$

Taking the difference of the last two equations leads to

$$\mathbf{v}\left(t + \frac{1}{2}\Delta t\right) - \mathbf{v}\left(t - \frac{1}{2}\Delta t\right) = \Delta t \dot{\mathbf{x}}(t) + \mathcal{O}\left(\Delta t^3\right) \quad (2.20)$$

and we then update the positions according to

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \mathbf{v}\left(t + \frac{1}{2}\Delta t\right) + \mathcal{O}\left(\Delta t^4\right). \quad (2.21)$$

The leapfrog method has the same order of accuracy as the Verlet method. However, both methods differ in the way in which the variables are integrated.

The analogies and differences between the Leapfrog method

$$\begin{aligned} \dot{\mathbf{v}}(t + \Delta t) &= \frac{f(\mathbf{x}(t))}{m}, \\ \mathbf{v}(t + \Delta t) &= \mathbf{v}(t) + \Delta t \dot{\mathbf{v}}(t + \Delta t), \\ \mathbf{x}(t + \Delta t) &= \mathbf{x}(t) + \Delta t \mathbf{v}(t + \Delta t) \end{aligned} \quad (2.22)$$

and the forward Euler integration

$$\begin{aligned} \dot{\mathbf{v}}(t + \Delta t) &= \frac{f(\mathbf{x}(t))}{m}, \\ \mathbf{x}(t + \Delta t) &= \mathbf{x}(t) + \Delta t \mathbf{v}(t), \\ \mathbf{v}(t + \Delta t) &= \mathbf{v}(t) + \Delta t \dot{\mathbf{v}}(t + \Delta t) \end{aligned} \quad (2.23)$$

are the following: Both methods rely on explicit forward integration. The update of the variables is then done in a different order. In the case of the Leapfrog method, the position is not updated using the previous velocity, as it is done in the usual Euler method.

Verlet and Leapfrog schemes are microcanonical, i.e., they are update schemes which conserve energy. Therefore, for sufficiently small time steps Δt , energy is usually conserved on average during a simulation and fluctuations are due to round-off errors. Large fluctuations

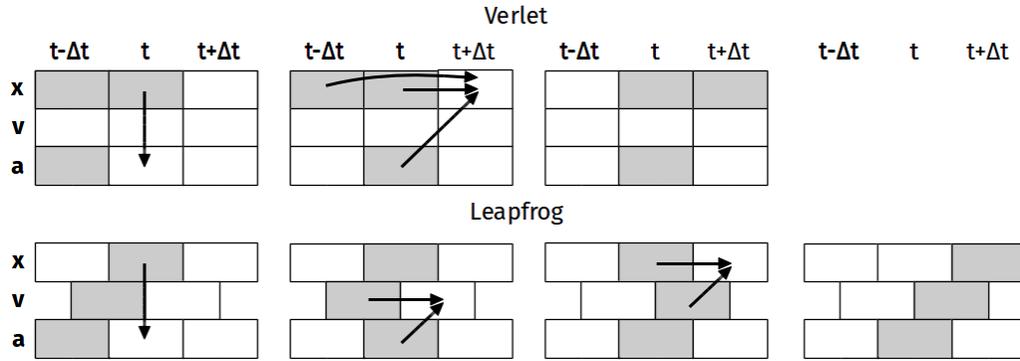


Figure 2.5: A comparison between Verlet and Leapfrog update schemes.

in energy are usually a hint for either the time step being too large or an erroneous code. It is also possible to estimate the accuracy of the method by analyzing the underlying energy fluctuations.

As in the case of the Verlet integration scheme, this method is completely time reversible and we can assess the error also by looking at the change in the initial configuration after having simulated forward and then backward in time. The difference between the two configurations can be taken as a measure for the error of the method.

Another approach consists in taking two initial configurations that differ only slightly in the velocity or the position of a particle, and to observe the difference in the time evolution of the two systems. Initially, both systems should behave similarly, whereas the difference increases with time. For instance, let ΔE be the energy difference between both systems. The slope of $\log \Delta E$ is an indicator of the precision of the method. The slope is the so-called *Lyapunov exponent*, and describes the divergence of the simulated trajectory from the true one. To circumvent this sensitive dependence on initial conditions, other integration methods (e.g., splitting methods) are used to simulate the long-term evolution of such Hamiltonian systems.

2.2 Optimization

It is often the case that we are interested in performing MD simulation for a large number of particles. This makes it necessary to compute the possible pair interactions for all N particles—an operation of complexity $\mathcal{O}(N^2)$. However, there exist different ways of optimizing our MD simulations.

For example, let our potential be a function $v(r) \propto r^{-2n}$ with $n \geq 1$. As a consequence of

$$\mathbf{f} = -\nabla r^{-2n} \propto r^{2(n-1)} \mathbf{r}, \quad (2.24)$$

it is possible to omit the computation of the square root in

$$r_{ij} = \sqrt{\sum_{k=1}^d (x_i^k - x_j^k)^2}. \quad (2.25)$$

If the potential has a complicated form which makes it computationally expensive to evaluate, we may discretize the potential and storing its values in a lookup table. For short range potentials, we define a cutoff r_c and divide the interval $(0, r_c^2)$ into K pieces with subinterval lengths

$$l_j = \frac{j}{K} r_c^2, \quad (2.26)$$

where $j \in \{1, \dots, K\}$. The force values stored in a lookup table are $f_j = f(\sqrt{l_j})$, and the corresponding index j is given by

$$j = \left\lfloor S \sum_{k=1}^d (x_i^k - x_j^k)^2 \right\rfloor + 1, \quad (2.27)$$

where $\lfloor \cdot \rfloor$ denotes the floor function and $S = K/r_c^2$. Interpolation methods are useful to obtain intermediate values.

The definition of a cutoff makes it necessary that we introduce a cutoff potential $\tilde{v}(r)$ according to

$$\tilde{v}(r) = \begin{cases} v(r) - v(r_c) - \frac{\partial v}{\partial r} \Big|_{r=r_c} (r - r_c) & \text{if } r \leq r_c, \\ 0 & \text{if } r > r_c, \end{cases} \quad (2.28)$$

where $v(r_c)$ is the value of the original potential at r_c . Without adding the derivative term to the potential $\tilde{v}(r)$, there would be a discontinuity in the corresponding force. In the case of the Lennard-Jones potential, a common cutoff value is $r_c = 2.5\sigma$. Care must be taken for potentials decaying with r^{-1} , because the forces at large distances are not negligible.

2.2.1 Verlet Tables

To reduce the amount of necessary computations, we should ignore particle-particle interactions whenever their force contributions are negligible. Therefore, we only consider particles in a certain range $r_l > r_c$. An illustration of this concept is shown in Fig. 2.6.

For every particle, we store the coordinates of the neighboring particles in a list which is referred to as *Verlet table*. As the particles move over time, the table has to be updated after

$$n = \frac{r_l - r_c}{\Delta t v_{\max}} \quad (2.29)$$

time steps with v_{\max} being the maximal velocity. Updating the whole list is still an operation of complexity $\mathcal{O}(N^2)$.

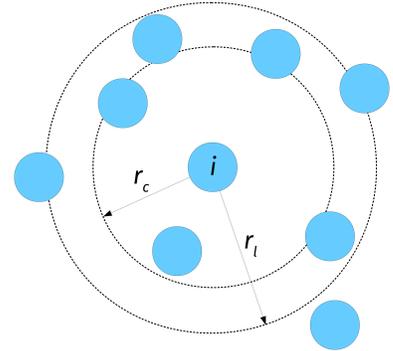


Figure 2.6: An illustration of the Verlet table method. Only particles within a distance of $r_l > r_c$ from particle i are considered.

2.2.2 Linked-Cell Method

Another possibility of optimizing our MD simulations is the *linked-cell method* [7]. In this method, the domain is discretized using a regular grid with a grid spacing of M satisfying $\frac{r_c}{2} < M < r_c$ where r_c is the range of the potential (or the cutoff range). Each particle is located within a certain grid cell. Due to our choice of M , we only have to consider the particle interactions in certain cell neighborhoods. Particles located in more distant cells do not contribute to the force term.

In d dimensions there are 3^d neighboring cells of interest. On average, we thus have to compute the interactions of $N3^dN/N_M$ particles where N_M is the number of grid cells. To keep track of the locations of all particles, we define a vector **FIRST** of length N_M to store the index of a particle located in cell j in **FIRST** [j]. If cell j is empty, then **FIRST** [j] = 0. In a second vector **LIST** of length N , the indices of the remaining particles located in the same cell are stored. If the particle i is the last one in a cell, then **LIST** [i] = 0. The following code shows an example of how to extract the particles located in cell $i = 2$.

```

i = 2;
M[1] = FIRST [ i ];
j = 1;
while ( LIST [ M [ j ] ] != 0 )
{
    M [ j + 1 ] = LIST [ M [ j ] ];
    j = j + 1;
}
    
```

When a particle changes its cell, **FIRST** and **LIST** are updated locally to avoid loops over all particles. The algorithm is thus of order $\mathcal{O}(N)$. In addition, this method is well suited for parallelization (domain-decomposition).

2.3 Dynamics of Composed Particles

Until now, we just considered single particle interactions with potentials that depend on the distance between particles. In nature, however, there exist many examples of systems in which the interactions also depend on size and shape of the considered particles. Examples include crystals and molecules such as the water molecule shown in Fig. 2.8. To describe the time evolution of such composed systems, we have to consider their shapes and constituents. We therefore begin our treatment of composed particles with the model of rigid bodies to then slightly relax the condition of rigidity. It is important to bear in mind that we should only simulate molecular systems at energies

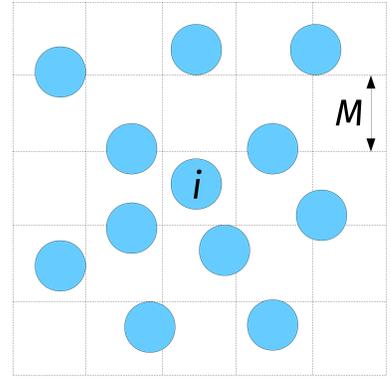


Figure 2.7: An illustration of the linked-cell method. A grid with grid spacing M ($\frac{r_c}{2} < M < r_c$) is placed on top of the MD simulation geometry. Only interactions between particles in a certain cell neighborhood have to be considered.

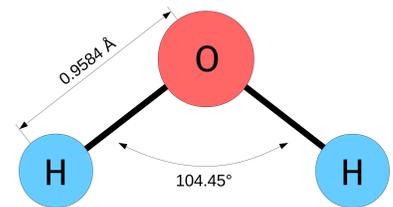


Figure 2.8: A water molecule as a composed particle system consisting of two hydrogen and one oxygen atom.

that do not affect the stability of the inter-particle bonds. Under the assumption that the bonds are stable in the simulated energy regime, there is a wide range of situations in which these methods are very useful. As an example, distance and angles between the atoms of a water molecule are constant for temperatures in many practical applications. In the subsequent sections, we discuss Lagrange multipliers and rigid body approximations to describe composed particle systems.

2.3.1 Lagrange Multipliers

One possibility to model composed particle systems is to introduce additional constraint forces as suggested in Ref. [8]. Such constraint forces are used to establish rigid bonds between individual particles. The idea is to rewrite the equation of motion for each particle as

$$m_i \ddot{\mathbf{x}}_i = \underbrace{\mathbf{f}_i}_{\text{external interaction}} + \underbrace{\mathbf{g}_i}_{\text{internal constraints}}, \quad (2.30)$$

where the first term accounts for interactions between different composed particles and the second one describes the constraint forces.

We now impose the constraints forces to account for the geometric structure of the molecules, e.g., certain distances d_{12} and d_{23} between atoms. Therefore, we define a potential such that the constraint forces \mathbf{g}_i are proportional to the difference of the actual and the desired distance of the particles. Considering a water molecule consisting of three particles, the two distance measures

$$\chi_{12} = r_{12}^2 - d_{12}^2, \quad (2.31)$$

$$\chi_{23} = r_{23}^2 - d_{23}^2, \quad (2.32)$$

are zero if the particles have the desired distance.

With $r_{ij} = \|\mathbf{r}_{ij}\|$ and $\mathbf{r}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ we obtain

$$\mathbf{g}_k = \frac{\lambda_{12}}{2} \nabla_{\mathbf{x}_k} \chi_{12} + \frac{\lambda_{23}}{2} \nabla_{\mathbf{x}_k} \chi_{23}, \quad (2.33)$$

for $k \in \{1, 2, 3\}$.

The yet undetermined Lagrange multipliers are defined by λ_{12} and λ_{23} . We compute these multipliers by imposing the constraints. According to Eq. (2.33), the constraint forces are

$$\mathbf{g}_1 = \lambda_{12} \mathbf{r}_{12}, \quad \mathbf{g}_2 = \lambda_{23} \mathbf{r}_{23} - \lambda_{12} \mathbf{r}_{12}, \quad \mathbf{g}_3 = -\lambda_{23} \mathbf{r}_{23}. \quad (2.34)$$

The last equations describe nothing but a linear spring with a yet to be determined spring constant $\lambda_{(\cdot)}$. To obtain the values of the Lagrange multipliers $\lambda_{(\cdot)}$, the Verlet algorithm is executed in two steps.

We first compute the Verlet update without constraint to obtain

$$\tilde{\mathbf{x}}_i(t + \Delta t) = 2\mathbf{x}_i - \mathbf{x}_i(t - \Delta t) + \Delta t^2 \frac{\mathbf{f}_i}{m_i}. \quad (2.35)$$

Then we correct the value using the constraints according to

$$\mathbf{x}_i(t + \Delta t) = \tilde{\mathbf{x}}_i(t + \Delta t) + \Delta t^2 \frac{\mathbf{g}_i}{m_i}. \quad (2.36)$$

By combining Eqs. (2.36) and (2.33), the updated positions are given by

$$\mathbf{x}_1(t + \Delta t) = \tilde{\mathbf{x}}_1(t + \Delta t) + \Delta t^2 \frac{\lambda_{12}}{m_1} \mathbf{r}_{12}(t), \quad (2.37)$$

$$\mathbf{x}_2(t + \Delta t) = \tilde{\mathbf{x}}_2(t + \Delta t) + \Delta t^2 \frac{\lambda_{23}}{m_2} \mathbf{r}_{23}(t) - \Delta t^2 \frac{\lambda_{12}}{m_2} \mathbf{r}_{12}(t), \quad (2.38)$$

$$\mathbf{x}_3(t + \Delta t) = \tilde{\mathbf{x}}_3(t + \Delta t) - \Delta t^2 \frac{\lambda_{23}}{m_3} \mathbf{r}_{23}(t). \quad (2.39)$$

With these expressions, we now obtain λ_{12} and λ_{23} by inserting (2.37), (2.38) and (2.39) into the constraint condition, i.e.,

$$\begin{aligned} |\mathbf{x}_1(t + \Delta t) - \mathbf{x}_2(t + \Delta t)|^2 &= d_{12}^2, \\ |\mathbf{x}_2(t + \Delta t) - \mathbf{x}_3(t + \Delta t)|^2 &= d_{23}^2, \end{aligned} \quad (2.40)$$

and finally

$$\begin{aligned} \left| \tilde{\mathbf{r}}_{12}(t + \Delta t) + \Delta t^2 \lambda_{12} \left(\frac{1}{m_1} + \frac{1}{m_2} \right) \mathbf{r}_{12}(t) - \Delta t^2 \frac{\lambda_{23}}{m_2} \mathbf{r}_{23}(t) \right|^2 &= d_{12}^2, \\ \left| \tilde{\mathbf{r}}_{23}(t + \Delta t) + \Delta t^2 \lambda_{23} \left(\frac{1}{m_2} + \frac{1}{m_3} \right) \mathbf{r}_{23}(t) - \Delta t^2 \frac{\lambda_{12}}{m_2} \mathbf{r}_{12}(t) \right|^2 &= d_{23}^2, \end{aligned} \quad (2.41)$$

where $\tilde{\mathbf{r}}_{ij} = \tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j$. The last expressions are then solved for λ_{12} and λ_{23} to compute the next position $\mathbf{x}_i(t + \Delta t)$. Depending on the precision needed one might ignore the higher order terms of Δt .

2.3.2 Rigid Bodies

Systems whose n constituents of mass m_i are located at fixed positions \mathbf{x}_i are referred to as rigid bodies. The motion of such objects is described by translations of the center of mass and rotations around it. The center of mass is defined as

$$\mathbf{x}_{\text{cm}} = \frac{1}{M} \sum_{i=1}^n \mathbf{x}_i m_i \quad \text{with} \quad M = \sum_{i=1}^n m_i. \quad (2.42)$$

The equation of motion of the center of mass and the corresponding torque are given by

$$M\ddot{\mathbf{x}}_{\text{cm}} = \sum_{i=1}^n \mathbf{f}_i = \mathbf{f}_{\text{cm}} \quad \text{and} \quad \mathbf{M} = \sum_{i=1}^n \mathbf{d}_i \wedge \mathbf{f}_i, \quad (2.43)$$

where $\mathbf{d}_i = \mathbf{x}_i - \mathbf{x}_{\text{cm}}$. In two dimensions, the rotation axis always points in the direction of the normal vector of the plane. Therefore, there exist only three degrees of freedom: two translational and one rotational. In three dimensions, there are six degrees of freedom: three translational and three rotational. We first discuss the treatment of rigid bodies in two dimensions and then generalize it to the three-dimensional case.

Two dimensions

In two dimensions, the moment of inertia and the torque are given by

$$I = \int \int_A r^2 \rho(r) dA \quad \text{and} \quad M = \int \int_A r f_t(r) dA, \quad (2.44)$$

where $\rho(r)$ is the mass density and f_t the tangential force. In general, the mass density may be constant or depending on the actual position and not only on the radius r . The equation of motion is given by

$$I\dot{\omega} = M. \quad (2.45)$$

We now apply the Verlet algorithm to \mathbf{x} and the rotation angle ϕ to compute the corresponding time evolutions according to

$$\begin{aligned} \phi(t + \Delta t) &= 2\phi(t) - \phi(t - \Delta t) + \Delta t^2 \frac{M(t)}{I}, \\ \mathbf{x}(t + \Delta t) &= 2\mathbf{x}(t) - \mathbf{x}(t - \Delta t) + \Delta t^2 M^{-1} \sum_{j \in A} f_j(t), \end{aligned} \quad (2.46)$$

where the total torque is the sum over all the torques acting on the rigid body, i.e.,

$$M(t) = \sum_{j \in A} \left[f_j^y(t) d_j^x(t) - f_j^x(t) d_j^y(t) \right]. \quad (2.47)$$

Three dimensions

To describe the motion of rigid bodies in three dimensions, we consider a lab-fixed and a body-fixed coordinate system x and y , respectively. The transformation between both systems is given by

$$\mathbf{x} = R(t)\mathbf{y}, \quad (2.48)$$

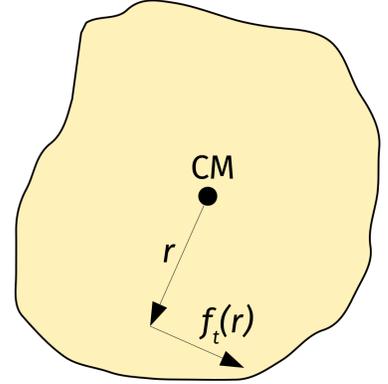


Figure 2.9: An example of a rigid body in two dimensions. The black dot show the center of mass (CM), and $f_t(r)$ represents the tangential force component.

where $R(t) \in \text{SO}(3)$ denotes a rotation matrix².

Furthermore, we define $\Omega = R^T \dot{R}$ and find with $R^T R = 1$ that

$$R^T \dot{R} + \dot{R}^T R = \Omega + \Omega^T = 0. \quad (2.49)$$

The last equation implies that Ω is skew-symmetric and thus of the form

$$\Omega = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \quad \text{and} \quad \Omega \mathbf{y} = \boldsymbol{\omega} \wedge \mathbf{y}, \quad (2.50)$$

where $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$.

The angular momentum is then given by

$$\mathbf{L} = \sum_{i=1}^n m_i \mathbf{x}_i \wedge \dot{\mathbf{x}}_i = \sum_{i=1}^n m_i R \mathbf{y}_i \wedge \dot{R} \mathbf{y}_i. \quad (2.51)$$

Combing Eqs. (2.51) and (2.50) yields

$$\mathbf{L} = R \sum_{i=1}^n m_i \mathbf{y}_i \wedge (\boldsymbol{\omega} \wedge \mathbf{y}_i) = R \sum_{i=1}^n m_i [\boldsymbol{\omega} (\mathbf{y}_i \cdot \mathbf{y}_i) - \mathbf{y}_i (\boldsymbol{\omega} \cdot \mathbf{y}_i)]. \quad (2.52)$$

The components of the inertia tensor are defined as

$$I_{jk} = \sum_{i=1}^n m_i [(\mathbf{y}_i \cdot \mathbf{y}_i) \delta_{jk} - \mathbf{y}_i^j \mathbf{y}_i^k] \quad (2.53)$$

and thus

$$\mathbf{L} = \mathbf{R} \mathbf{S} \quad \text{with} \quad S_j = \sum_{k=1}^3 I_{jk} \omega_k. \quad (2.54)$$

where I is the inertia tensor.

Considering a coordinate system whose axes are parallel to the principal axes of inertia of the body, the inertia tensor takes the form

$$I = \begin{pmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{pmatrix} \quad \text{and} \quad S_j = I_j \omega_j. \quad (2.55)$$

With Eq. (2.54), the equations of motion are determined by

$$\dot{\mathbf{L}} = \dot{R} \mathbf{S} + R \dot{\mathbf{S}} = \widetilde{\mathbf{M}}, \quad (2.56)$$

where $\widetilde{\mathbf{M}} = \mathbf{R} \mathbf{M}$ represents the torque in the lab-fixed coordinate system. By multiplying the last equation with R^T , we find the *Euler equations* in the principal axes coordinate system, i.e.,

$$\begin{aligned} \dot{\omega}_1 &= \frac{M_1}{I_1} + \left(\frac{I_2 - I_3}{I_1} \right) \omega_2 \omega_3, \\ \dot{\omega}_2 &= \frac{M_2}{I_2} + \left(\frac{I_3 - I_1}{I_2} \right) \omega_3 \omega_1, \\ \dot{\omega}_3 &= \frac{M_3}{I_3} + \left(\frac{I_1 - I_2}{I_3} \right) \omega_1 \omega_2. \end{aligned} \quad (2.57)$$

² The group $\text{SO}(3)$ is the so-called three dimensional rotation group, or special orthogonal group. All rotation matrices $R \in \text{SO}(3)$ fulfill $R^T R = R R^T = 1$.

The angular velocities are then integrated according to

$$\begin{aligned}
 \omega_1(t + \Delta t) &= \omega_1(t) + \Delta t \frac{M_1(t)}{I_1} + \Delta t \left(\frac{I_2 - I_3}{I_1} \right) \omega_2 \omega_3, \\
 \omega_2(t + \Delta t) &= \omega_2(t) + \Delta t \frac{M_2(t)}{I_2} + \Delta t \left(\frac{I_3 - I_1}{I_2} \right) \omega_3 \omega_1, \\
 \omega_3(t + \Delta t) &= \omega_3(t) + \Delta t \frac{M_3(t)}{I_3} + \Delta t \left(\frac{I_1 - I_2}{I_3} \right) \omega_1 \omega_2.
 \end{aligned} \tag{2.58}$$

From these expressions, we obtain the angular velocity in the laboratory frame

$$\tilde{\omega}(t + \Delta t) = R\omega(t + \Delta t). \tag{2.59}$$

Since the particles are moving all the time, the rotation matrix is not constant. We therefore have to find an efficient way to determine and update R at every step in our simulation. In the following, we therefore discuss Euler angles and quaternions.

Euler Angles

One possible parameterization of the rotation matrix R is denoted by

$$\begin{aligned}
 R &= R(\phi, \theta, \psi) \\
 &= \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}.
 \end{aligned} \tag{2.60}$$

An illustration of the Euler angle parameterization is presented in Fig. 2.10. As a consequence of the occurrence of products of multiple trigonometric functions for arbitrary rotations, this parameterization is not well-suited for efficient computations. We have to keep in mind that this operation has to be performed for every particle and every time step, leading to high computational effort. For the computation of angular velocities, derivatives of Eq. (2.60) have to be considered. Specifically, we would have to compute

$$\begin{aligned}
 \dot{\phi} &= -\tilde{\omega}_x \frac{\sin \phi \cos \theta}{\sin \theta} + \tilde{\omega}_y \frac{\cos \phi \cos \theta}{\sin \theta} + \tilde{\omega}_z, \\
 \dot{\theta} &= \tilde{\omega}_x \cos \theta + \tilde{\omega}_y \sin \theta, \\
 \dot{\psi} &= \tilde{\omega}_x \frac{\sin \phi}{\sin \theta} - \tilde{\omega}_y \frac{\cos \phi}{\sin \theta}.
 \end{aligned} \tag{2.61}$$

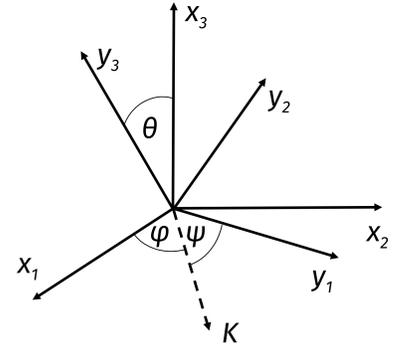


Figure 2.10: Euler angle parameterization of a rotation matrix.

Quaternions

Denis J. Evans, a professor in Canberra, Australia, came up with a trick to optimize the computation of rotational velocities [9, 10].

Quaternions are a generalization of complex numbers, where four basis vectors span a four-dimensional space. We define

$$q_0 = \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi + \psi}{2}\right), \quad (2.62)$$

$$q_1 = \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi - \psi}{2}\right), \quad (2.63)$$

$$q_2 = \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi - \psi}{2}\right), \quad (2.64)$$

$$q_3 = \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi + \psi}{2}\right), \quad (2.65)$$

with $0 < q_i < 1$ and $\sum_i q_i = 1$ for $i \in \{1, \dots, 4\}$, to represent the angles in dependence of a set of quaternion parameters q_i . The euclidean norm of \mathbf{q} equals unity and thus there exist only three independent parameters.

The rotation matrix as defined in Eq. (2.60) has a quaternion representation

$$R = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}. \quad (2.66)$$

We now found a more efficient way of computing rotations without the necessity of computing lengthy products of sine and cosine functions. This approach is much faster than the one of Eq. (2.60). The angular velocities are then computed according to

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \begin{pmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (2.67)$$

Since quaternion and euclidean representations are connected by a diffeomorphism, there is always the possibility of calculating the values of the Euler angles if needed

$$\phi = \arctan \left[\frac{2(q_0q_1 + q_2q_3)}{1 - 2(q_1^2 + q_2^2)} \right] \quad (2.68)$$

$$\theta = \arcsin [2(q_0q_2 - q_1q_3)] \quad (2.69)$$

$$\psi = \arctan \left[\frac{2(q_0q_3 + q_1q_2)}{1 - 2(q_2^2 + q_3^2)} \right] \quad (2.70)$$

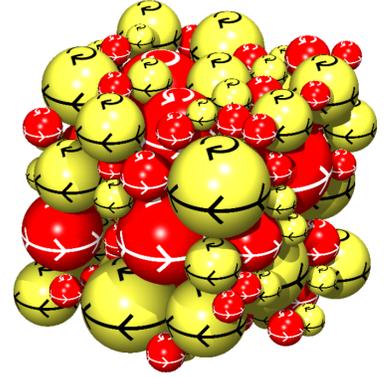


Figure 2.11: Rotating spheres in a sphere assembly as an example of rigid body dynamics [11].

There is no need of calculating the Euler angles at each integration step. We now simulate our rigid body dynamics in quaternion representation according to the following strategy:

- Compute the torque in the body frame $\mathbf{M} = R^T \widetilde{\mathbf{M}}$,
- Obtain $\omega(t + \Delta t)$ according to Eq. (2.58),
- Update the rotation matrix as defined in Eq. (2.66) by computing $\mathbf{q}(t + \Delta t)$ according to Eq. (2.67).

An example of a composed particle system whose dynamics has been simulated using a quaternion approach is shown in Fig. 2.11.

2.4 Long-range Potentials

In Sec. 2.2, we discussed different optimization techniques to speed up our MD simulations. If the potential decays sufficiently fast, it is possible to define a cutoff to neglect the vanishing force contributions at large distances. However, if the potential of a system in d dimensions decays as $V(r) \propto r^{-d}$ or even slower, it is not possible to define a cutoff anymore. The reason is that there is a non-negligible energy contribution even at large distances. Examples of such potentials occur in electrostatic, gravitational, and dipole interactions. To simulate such systems, we discuss the following methods:

- Ewald summation,
- Particle-Mesh methods (PM, PPPM & APPPM),
- Reaction field method.

2.4.1 Ewald Summation

Paul Ewald³ developed a method to compute long-range interactions in periodic systems. It has been originally developed to study ionic crystals. Our goal is to now apply this method to MD simulations.

Until now, we used periodic boundary conditions for systems of finite size. This was only possible, because the interaction correlations were decaying with increasing distance.

In the case of long-range potentials, the particles are no longer uncorrelated since they are able to interact even at large distances. What we can do is to periodically repeat our system by attaching its own image at its borders. This is illustrated in Fig. 2.12. We consider those repeated systems as really existing, and compute the interaction of the

³ Paul Ewald was professor at the University of Stuttgart, Germany, also active during the period before the second world war. He was elected rector in 1932. However, due to increasing difficulties with the *Dozentenbund* (the professors' association), which was affiliated to the National Socialist party in Germany, he had to resign his position in the spring of 1933. He continued his activities, until Wilhelm Stortz, the new rector, asked Ewald to leave the university. For further reading, see Ref. [12]

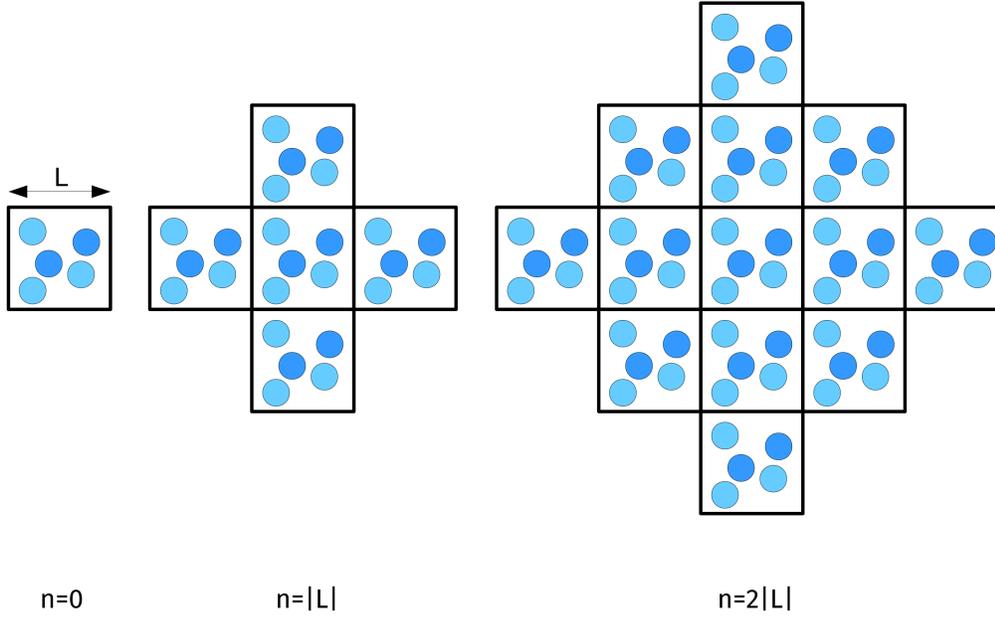


Figure 2.12: An example of a Ewald summation procedure. From left to right, the system is periodically attached at the boundaries.

particles in our field of interest with all the other particles in the system. Every cell is characterized by a vector that goes from the origin of the central system of interest to the origin of the outer cell.

The resulting potential is thus the sum over particle interactions that repeat over several boxes, i.e.,

$$V = \frac{1}{2} \sum_{\mathbf{n}}' \sum_{i,j} z_i z_j |\mathbf{r}_{ij} + \mathbf{n}|^{-1}, \quad (2.71)$$

where z_i and z_j represent the corresponding charges or masses. For the sake of brevity, we are omitting $4\pi\epsilon_0$ or $4\pi G$ factors. The sum over \mathbf{n} denotes the summation over all lattice points

$$\mathbf{n} = (n_x L, n_y L, n_z L)$$

with integers n_x, n_y, n_z and sublattice size L . The prime indicates that we are excluding $\mathbf{n} = 0$ for $i = j$.

The first sum comprises the central box at $\mathbf{n} = 0$ and the following terms are the ones for which $|\mathbf{n}| = L$, i.e., $\mathbf{n} = (\pm L, 0, 0)$, $\mathbf{n} = (0, \pm L, 0)$ and $\mathbf{n} = (0, 0, \pm L)$. The Ewald sum is only conditionally convergent, i.e., the convergence depends on the order of the summation and the convergence is very slow. Since Ewald never used a computer, he intended to sum over an infinite number of cells. What one has to do in reality is to truncate the sum at some point and try to estimate the error.

Because of the nature of the algorithm, it is only used for systems consisting of a few particles. Several approaches are possible to im-

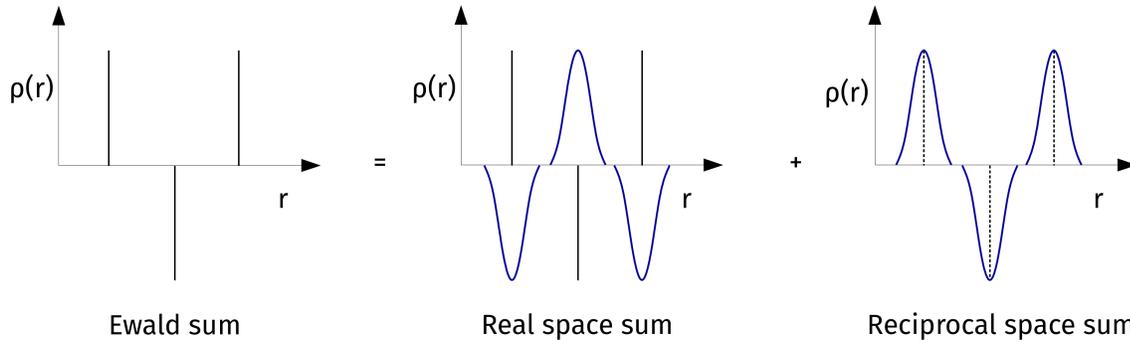


Figure 2.13: The Ewald sum consists of a sum in real and another one in reciprocal space. The charges are screened with Gaussian charge distribution. Other distributions are also possible.

prove the convergence [13, 14]. One possible technique consists of screening the charges by a Gaussian distribution

$$\rho_i(r) = \frac{z_i \kappa^3}{\pi^{3/2}} \exp(-\kappa^2 r^2) \quad (2.72)$$

with some arbitrary screening factor κ .

After having introduced the screening charges one has to cancel their total effect using charges of opposite sign. The basic idea behind this summation is shown in Fig. 2.13. This way, the result is a sum in real and reciprocal space which converges faster

$$\begin{aligned} V = & \frac{1}{2} \sum_{ij} \left[\sum_{\mathbf{n}} z_i z_j \frac{\text{erfc}(\kappa |\mathbf{r}_{ij} + \mathbf{n}|)}{|\mathbf{r}_{ij} + \mathbf{n}|} \right. \\ & \left. + \frac{1}{\pi L^3} \sum_{\mathbf{k} \neq 0} z_i z_j \frac{4\pi^2}{k^2} \exp\left(\frac{-k^2}{4\kappa^2}\right) \cos(\mathbf{k} \mathbf{r}_{ij}) \right] \quad (2.73) \\ & - \frac{\kappa}{\sqrt{\pi}} \sum_i z_i^2 + \frac{2\pi}{3L^3} \left| \sum_i z_i \mathbf{r}_i \right|^2 \end{aligned}$$

with the *error function* $\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty \exp(-t^2) dt$. The last term is necessary if the boxes are surrounded by vacuum. These formulas are mentioned here only to show that the Ewald sum is a conceptually straightforward idea with many implementation difficulties. It is only applicable under certain circumstances and we therefore focus on other approaches in the next sections.

2.4.2 Particle-Mesh Method

Another possibility of simulating long-range particle interactions is the so-called Particle-Mesh (PM) method. This method was invented by Eastwood and Hockney, and is not very well suited for inhomogeneous distributions of particles [15, 16]. It is based on the following steps

- Put a fine mesh on top of the system.

- Distribute the charges onto the mesh points.
- Calculate electrostatic potential by solving the Poisson equation on the mesh using FFT.
- Calculate force on each particle by numerically differentiating the potential and interpolating back from the mesh to the particle position.

The mesh concept is similar to the one shown in Fig. 2.7. Once the potential is known, we compute the force exerted on the particles by interpolating the potential at the vertices.

The accuracy of the results depends on the following criteria:

- Errors should vanish for large distances between the particles.
- Momentum should always be conserved (from $\mathbf{F}_{ij} = -\mathbf{F}_{ji}$).
- Charges or masses on the mesh and the interpolated forces should vary smoothly.

This method is not very efficient and time consuming for

- Inhomogeneous distributions of charges or masses.
- Strong correlations, like bound states in molecules.
- Deviation from the point-like object approximation (e.g., tidal effects).
- Complex geometries of the system.

In these cases one might consider using the *Particle-Particle-Particle-Mesh* (P³M) or the *Adaptive Particle-Particle-Particle-Mesh* (AP³M) algorithms. These methods are presented later in the subsequent sections. The PM method is well-suited for gravitational interactions, because such systems are characterized by overall low mass densities, i.e., smooth variations in the potential, which are relatively homogeneously distributed.

Implementation

Once we discretized our d dimensional system, there exists more than one possibility to assign charge and force values to vertices:

- Nearest grid point: Put particle on nearest grid point and also evaluate its force at the nearest grid point.
- Cloud in cell: Assign the charge to the 2^d nearest grid points and also interpolate from these 2^d grid points.

In two dimensions, a possible cloud in cell implementation of a charge q located at (x, y) is given by the charge distribution

$$\begin{aligned}\rho_{ij} &= \frac{q}{2\Delta l^2}(x_{i+1} - x)(y_{i+1} - y), \\ \rho_{i+1j} &= \frac{q}{2\Delta l^2}(x - x_i)(y_{i+1} - y), \\ \rho_{ij+1} &= \frac{q}{2\Delta l^2}(x_{i+1} - x)(y - y_i), \\ \rho_{i+1j+1} &= \frac{q}{2\Delta l^2}(x - x_i)(y - y_i),\end{aligned}$$

where Δl is the grid spacing. The potential at a certain point \mathbf{r} is given by the convolution of $\rho(\mathbf{r})$ with the Green's function, i.e.,

$$\phi(\mathbf{r}) = \int \rho(\mathbf{r}') G(\mathbf{r}, \mathbf{r}') d\mathbf{r}'. \quad (2.74)$$

In the case of an electrostatic or a gravitational interaction, the Green's function is given by $G(\mathbf{r}, \mathbf{r}') \propto |\mathbf{r} - \mathbf{r}'|^{-1}$.

In Fourier space this corresponds to a multiplication with $\hat{G}(\mathbf{k}) \propto \mathbf{k}^{-2}$, i.e., $\hat{\phi}(\mathbf{k}) = \hat{\rho}(\mathbf{k}) \hat{G}(\mathbf{k})$ (convolution theorem). After applying the inverse Fourier transform, the forces are then given by

$$\mathbf{F}(\mathbf{r}_{ij}) = -\nabla\phi(\mathbf{r}_{ij}). \quad (2.75)$$

The FFT determines the computational complexity $\mathcal{O}(N \log N)$ of the method. More information about further properties are contained in Ref. [17].

Knowing the field at every point, we compute the force at every vertex of the mesh and then interpolate the forces on the corners of the mesh to find the force that is exerted by the field on a particle. As already mentioned, the PM algorithm is not very well suited for inhomogeneous distributions of particles, strong correlations like bound states or complex geometries. In these cases, the P³M and AP³M methods, tree codes or multiple expansions constitute other possible simulation techniques. One important example is the very heterogeneous distribution of stars and galaxies in astrophysics. For further reading, we recommend Ref. [18].

P³M (Particle-Particle-Particle-Mesh)

For particles with close neighbors, we decompose the force acting on it into two terms, i.e.,

$$\mathbf{F} = \mathbf{F}_s + \mathbf{F}_l, \quad (2.76)$$

where the first term accounts for short-range interactions and second term describes interactions over large distances. The short-range interactions are simulated by solving Newton's equation of motion

whereas the PM method is applied to the long-range forces. This approach has the drawback that the long-range potential field has to be computed for each particle and we still have not solved the problem of heterogeneity. Most of the grid points are empty making this algorithm not very efficient.

AP³M (Adaptive Particle-Particle-Particle-Mesh)

A possibility to improve the particle mesh simulations is to adapt the mesh to the particles density, i.e., using an adaptive mesh with higher resolution where the particle density is higher. One implementation could be to refine a cell as long as there are more particles than a certain accepted quantity. Tree codes are then used to organize the structures emerging in the continuous refinement process [18]. The advances in the simulation of many-body systems often depend on the development of such bookkeeping techniques. As an example, present simulation frameworks allow to simulate the interaction of up to 10^{18} particles [19].

2.4.3 *Reaction Field Method*

In the case of more complex interactions such as composed molecules or non-point-like particles, a good solution is to ignore the complexity of distant particles and only take into account their mean effect while calculating explicitly the interaction with close particles. The concept finds its root in the work of Onsager on the dielectric constant but it was introduced as an explicit computational technique in the 1970s [20, 21, 22].

This method is mostly used for the simulation of dipole-dipole interactions. We consider a sphere N_i of radius r_c . The dipole moments $\boldsymbol{\mu}_j$ within the sphere lead to an electric field (reaction field)

$$\mathbf{E}_i = \frac{2(\epsilon_s - 1)}{2\epsilon_s + 1} \frac{1}{r_c^3} \sum_{j \in N_i} \boldsymbol{\mu}_j, \quad (2.77)$$

with ϵ_s being an effective parameter to model the dielectric continuum outside the sphere. The resulting total force exerted on particle i is

$$\mathbf{F}_i = \sum_{j \in N_i} \mathbf{F}_{ij} + \boldsymbol{\mu}_i \wedge \mathbf{E}_i. \quad (2.78)$$

As the particles are moving, the number of particles inside the cavity is not constant. This causes force discontinuities because of instantaneous force changes. To avoid this effect, it is possible to introduce a

distance depending weighting factor

$$g(r_j) = \begin{cases} 1, & \text{for } r_j < r_t \\ \frac{r_c - r_j}{r_c - r_t}, & \text{for } r_t \leq r_j \leq r_c \\ 0, & \text{for } r_c < r_j \end{cases} \quad (2.79)$$

2.5 Canonical Ensemble

After discussing various techniques to simulate the interaction of particles and rigid bodies, we now again focus on their statistical description. Often we are not interested in the properties of single particles, but rather in macroscopic quantities such as temperature, pressure, and density. Until now, we have not introduced any method to control such macroscopic properties in our molecular dynamics simulations. In this section, some important methods are presented. Particular attention is paid to the Nosé-Hoover thermostat and the Parrinello-Rahman barostat. For further readings, Ref. [1] is recommended.

Experiments are often conducted at constant temperature and not at constant energy. This is a common situation, because systems are usually able to exchange energy with their environment. We therefore first couple our system to a heat bath to realize this situation. There are various options to do this

- Rescaling of velocities,
- Introducing constraints (Hoover),
- Nosé-Hoover thermostat,
- Stochastic method (Anderson).

However, before focusing on the discussion of the last methods, we shall define the concept of temperature used in the subsequent sections. We start from the equipartition theorem

$$\left\langle q_\mu \frac{\partial \mathcal{H}}{\partial q_\nu} \right\rangle = \left\langle p_\mu \frac{\partial \mathcal{H}}{\partial p_\nu} \right\rangle = \delta_{\mu\nu} k_B T \quad (2.80)$$

for a Hamiltonian \mathcal{H} with the generalized coordinates \mathbf{p} and \mathbf{q} .

We consider a classical system whose Hamiltonian is given by

$$\mathcal{H} = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i} + V(\mathbf{x}_1, \dots, \mathbf{x}_N) \quad (2.81)$$

and we define the instantaneous temperature

$$\mathcal{T} = \frac{2}{3k_B(N-1)} \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i}. \quad (2.82)$$

2.5.1 Velocity Rescaling

We now discuss a straightforward yet physically wrong method to simulate a canonical ensemble of particles. The equipartition theorem as defined by Eq. (2.80) tells us that the kinetic energy of the particles corresponds to a certain temperature. Intuitively, we should be able to adjust the instantaneous temperature of the system by rescaling the velocities of the particles according to

$$\mathbf{v}_i \rightarrow \alpha \mathbf{v}_i. \quad (2.83)$$

The measured temperature is proportional to the squared velocities and thus

$$\mathcal{T} \rightarrow \alpha^2 \mathcal{T}. \quad (2.84)$$

Therefore, we have to set

$$\alpha = \sqrt{\frac{T}{\mathcal{T}}} \quad (2.85)$$

to stay at a certain desired temperature T .

This method is very easy to implement. However, the problem is that the resulting velocity distribution deviates from the canonical one. A modification of this method makes use of an additional parameter τ_T which describes the coupling to heat bath. The scaling factor is then (Berendsen thermostat)

$$\alpha = \sqrt{1 + \frac{\Delta t}{\tau_T} \left(\frac{T}{\mathcal{T}} - 1 \right)}. \quad (2.86)$$

Still, we do not recover the canonical velocity distribution (Maxwell-Boltzmann). Velocity rescaling should be only applied to initialize a MD simulation at a given temperature.

2.5.2 Constraint Method

Another possibility to adjust the temperature of the system is to add a friction term to the equations of motion, i.e.,

$$\dot{\mathbf{p}}_i = \mathbf{f}_i - \zeta \mathbf{p}_i, \quad (2.87)$$

where $\mathbf{p}_i = m_i \dot{\mathbf{x}}_i$. Various definitions of the friction coefficient ζ are possible. Hoover's original proposal is based on the following constant temperature condition:

$$\dot{\mathcal{T}} \propto \frac{d}{dt} \left(\sum_{i=1}^N \mathbf{p}_i^2 \right) \propto \sum_{i=1}^N \dot{\mathbf{p}}_i \mathbf{p}_i = 0. \quad (2.88)$$

By combining Eqs. (2.87) and (2.88), we find

$$\zeta = \frac{\sum_{i=1}^N \mathbf{f}_i \mathbf{p}_i}{\sum_{i=1}^N |\mathbf{p}_i|^2}. \quad (2.89)$$

This method makes it necessary to already start at the desired temperature. Another possibility is to determine the friction coefficient according to (Berendsen)

$$\xi = \gamma \left(1 - \frac{T}{\mathcal{T}} \right), \quad (2.90)$$

or (Hoover)

$$\dot{\xi} = \frac{fk_B}{Q} (\mathcal{T} - T). \quad (2.91)$$

The parameters γ and Q determine the temperature adaption rate, and f is the number of degrees of freedom. Still, all these have the drawback that they either do not recover the Maxwell–Boltzmann velocity distribution, or that they assume an artificial relaxation time scale defined by Q . In addition, these methods are not time reversible.

2.5.3 Nosé-Hoover Thermostat

In order to overcome the problem of the wrong velocity distribution, we are now going to discuss the Nosé-Hoover thermostat as the correct method to simulate heat bath particle dynamics. Shuichi Nosé introduced a new degree of freedom s that describes the heath bath [23, 24]. The corresponding potential and kinetic energy are

$$\begin{aligned} \mathcal{V}(s) &= (3N + 1) k_B T \ln s, \\ K(s) &= \frac{1}{2} Q \dot{s}^2. \end{aligned} \quad (2.92)$$

The new degree of freedom s rescales the time step dt and momenta \mathbf{p}_i according to

$$dt' = s dt \quad \text{and} \quad \mathbf{p}'_i = s \mathbf{p}_i. \quad (2.93)$$

The velocities are also rescaled:

$$\mathbf{v}'_i = \frac{d\mathbf{x}_i}{dt'} = \frac{d\mathbf{x}_i}{dt} \frac{dt}{dt'} = \frac{\mathbf{v}_i}{s}. \quad (2.94)$$

The Hamiltonian is thus

$$\mathcal{H} = \sum_{i=1}^N \frac{\mathbf{p}'_i{}^2}{2m_i s^2} + \frac{1}{2} Q \dot{s}^2 + V(\mathbf{x}_1, \dots, \mathbf{x}_N) + \mathcal{V}(s), \quad (2.95)$$

with $p_s = Q\dot{s}$ being the inertia term and p_s the momentum corresponding to s . The velocities are

$$\begin{aligned} \frac{d\mathbf{x}_i}{dt'} &= \nabla_{\mathbf{p}'_i} \mathcal{H} = \frac{\mathbf{p}'_i}{m_i s^2}, \\ \frac{ds}{dt'} &= \frac{\partial \mathcal{H}}{\partial p_s} = \frac{p_s}{Q}. \end{aligned} \quad (2.96)$$

With $\mathbf{p}'_i = m_i s^2 \dot{\mathbf{x}}_i$ we find

$$\mathbf{f}_i = \frac{d\mathbf{p}'_i}{dt'} = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}_i} = -\nabla_{\mathbf{x}_i} V(\mathbf{x}_1, \dots, \mathbf{x}_N) = 2m_i s \dot{s} \dot{\mathbf{x}}_i + m_i s^2 \ddot{\mathbf{x}}_i \quad (2.97)$$

and

$$\frac{d\mathbf{p}_s}{dt'} = -\frac{\partial \mathcal{H}}{\partial \mathbf{s}} = \frac{1}{s} \left[\sum_{i=1}^N \frac{\mathbf{p}'_i{}^2}{m_i s^2} - (3N+1) k_B T \right]. \quad (2.98)$$

Based on the last Hamilton equations, we find for the equations of motion in virtual time t'

$$m_i s^2 \ddot{\mathbf{x}}_i = \mathbf{f}_i - 2m_i \dot{s} \dot{\mathbf{x}}_i \quad \text{with } i \in \{1, \dots, N\} \quad (2.99)$$

and

$$Q \dot{s} = \sum_{i=1}^N m_i s \dot{\mathbf{x}}_i^2 - \frac{1}{s} (3N+1) k_B T. \quad (2.100)$$

Note that Eqs. (2.99) and (2.100) are coupled. This reflects the fact that the two systems are not isolated and exchange energy in form of heat.

In order to obtain the equations of motion in real time, we have to remind ourselves that $dt = dt'/s$ and $\mathbf{p}'_i = s\mathbf{p}_i$. Thus, we find for the velocities

$$\begin{aligned} \frac{d\mathbf{x}_i}{dt} &= s \frac{d\mathbf{x}_i}{dt'} = \frac{\mathbf{p}'_i}{m_i s} = \frac{\mathbf{p}_i}{m_i} \\ \frac{ds}{dt} &= s \frac{ds}{dt'} = s \frac{\mathbf{p}_s}{Q} \end{aligned} \quad (2.101)$$

and for the forces

$$\begin{aligned} \frac{d\mathbf{p}_i}{dt} &= s \frac{d}{dt'} \left(\frac{\mathbf{p}'_i}{s} \right) = \frac{d\mathbf{p}'_i}{dt'} - \frac{1}{s} \frac{ds}{dt'} \mathbf{p}'_i = \mathbf{f}_i - \frac{1}{s} \frac{ds}{dt'} \mathbf{p}_i \\ \frac{d\mathbf{p}_s}{dt} &= s \frac{d\mathbf{p}_s}{dt'} = \sum_{i=1}^N \frac{\mathbf{p}'_i{}^2}{m_i} - (3N+1) k_B T. \end{aligned} \quad (2.102)$$

With $\zeta = \frac{d \ln(s)}{dt} = \frac{\dot{s}}{s}$ representing a friction term, the equations of motions (2.99) and (2.100) are given in real time by

$$\ddot{\mathbf{x}}_i = \frac{\mathbf{f}_i}{m_i} - \zeta \dot{\mathbf{x}}_i \quad (2.103)$$

and

$$\frac{1}{2} Q \dot{\zeta} = \frac{1}{2} \sum_{i=1}^N m_i \dot{\mathbf{x}}_i^2 - \frac{1}{2} (3N+1) k_B T. \quad (2.104)$$

The first term in Eq. (2.104) denotes the measured kinetic energy whereas the second one corresponds to the desired kinetic energy. The quantity Q represents the coupling to the heat bath. The higher its value, the stronger the system reacts to temperature fluctuations. For $Q \rightarrow \infty$, we recover microcanonical MD, and for small values of Q we

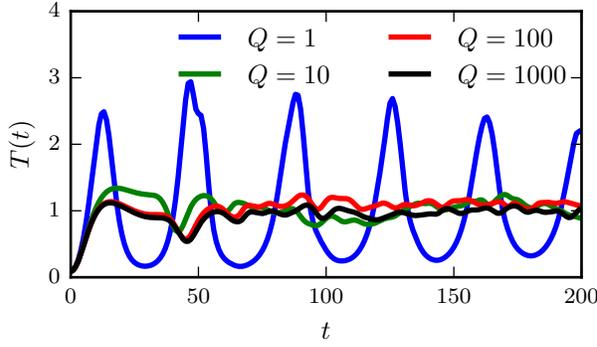


Figure 2.14: Temperature fluctuations of the Nosé-Hoover thermostat for 64 particles and different values of the thermal inertia Q .

find spurious temperature oscillations. Such oscillations are shown in Fig. 2.14.

A reasonable value of Q is characterized by the fact that normal temperature fluctuations are observed, i.e.,

$$\overline{\Delta T} = \sqrt{\frac{2}{Nd} \overline{T}}, \quad (2.105)$$

where d is the system's dimension and N the number of particles.

We now show that the Nosé-Hoover thermostat recovers the canonical partition function. Therefore, we start from microcanonical MD and the corresponding partition function

$$Z = \int \delta(\mathcal{H} - E) ds dp_s d^3 x' d^3 p', \quad (2.106)$$

where the x and p integration has to be taken over a three dimensional space with N particles. With $\mathcal{H} = \mathcal{H}_1 + (3N + 1) k_B T \ln(s)$ and in real time, we find

$$\begin{aligned} Z &= \int \delta[(\mathcal{H}_1 - E) + (3N + 1) k_B T \ln(s)] s^{3N} ds dp_s d^3 x d^3 p \\ &= \int \delta \left[s - e^{-\frac{\mathcal{H}_1 - E}{(3N+1)k_B T}} \right] \frac{s^{3N+1}}{(3N+1)k_B T} ds dp_s d^3 x' d^3 p', \end{aligned} \quad (2.107)$$

where we used the identity $\delta[f(s)] = \delta(s - s_0) / f'(s)$ with $f(s_0) = 0$ in the second step. Integrating Eq. (2.107) over s yields

$$\begin{aligned} Z &= \int \frac{1}{(3N+1)k_B T} e^{-\frac{\mathcal{H}_1 - E}{k_B T}} dp_s d^3 x d^3 p \\ &= \int e^{-\frac{\mathcal{H}_0}{k_B T}} d^3 x d^3 p \int \frac{1}{(3N+1)k_B T} e^{\frac{E - p_s^2}{2Q}} dp_s, \end{aligned} \quad (2.108)$$

with $\mathcal{H}_1 = \mathcal{H}_0 + \frac{p_s^2}{2Q}$. The first term of the last equation is the canonical partition function and the last term a constant prefactor. In 1985,

Hoover proved that the Nosé–Hoover thermostat is the only method with a single friction parameter that gives the correct canonical distribution. Hoover also showed that this thermostat satisfies the Liouville equation, i.e., that the density of states is conserved in the phase space.

2.5.4 Stochastic Method

Another method has been proposed by Andersen [25]. The so-called *stochastic method* is a combination of molecular dynamics and a Monte Carlo algorithm.

At temperature T , the velocities are distributed according to the Maxwell-Boltzmann distribution

$$P(\mathbf{p}) = \frac{1}{(\pi k_B T)^{\frac{3}{2}}} e^{-\frac{\mathbf{p}^2}{k_B T}}. \quad (2.109)$$

Every n time steps, a particle is then selected uniformly at random and given a new momentum according to Eq. (2.109). If n is too small one has pure Monte Carlo and loses the real time scale, e.g., the long time tail of the velocity correlation. If n is too large the coupling to the heat bath is too weak, equilibration is slow and one will essentially work microcanonically.

2.5.5 Constant Pressure

Another important situation is the one of constant pressure. We again consider the equipartition theorem and a Hamiltonian as defined by Eqs. (2.80) and (2.81). We find

$$\frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{x}_i \cdot [\nabla_{\mathbf{x}_i} V(\mathbf{x})] \right\rangle = N k_B T. \quad (2.110)$$

We now distinguish between particle-particle and particle-wall interactions \mathbf{f}^{part} and \mathbf{f}^{ext} , respectively. This leads to

$$\begin{aligned} \frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{x}_i \cdot [\nabla_{\mathbf{x}_i} V(\mathbf{x})] \right\rangle &= -\frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{x}_i \cdot (\mathbf{f}_i^{\text{part}} + \mathbf{f}_i^{\text{ext}}) \right\rangle \\ &= -\frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{x}_i \cdot \mathbf{f}_i^{\text{part}} \right\rangle - \frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{x}_i \cdot \mathbf{f}_i^{\text{ext}} \right\rangle. \end{aligned} \quad (2.111)$$

We define

$$w = \frac{1}{3} \sum_{i=1}^N \mathbf{x}_i \cdot \mathbf{f}_i^{\text{part}} \quad (2.112)$$

as the virial. Based on

$$\frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{x}_i \cdot \mathbf{f}_i^{\text{ext}} \right\rangle = -\frac{1}{3} \int_{\Gamma} p \mathbf{x} d\mathbf{A} = -\frac{1}{3} p \int_V (\nabla \cdot \mathbf{x}) dV = -pV, \quad (2.113)$$

we define the instantaneous pressure \mathcal{P} by

$$\mathcal{P}V = Nk_B T + \langle w \rangle. \quad (2.114)$$

Similarly to the Nosé-Hoover thermostat, we introduce a parameter W which adjusts the pressure of the system. The volume change is given by

$$V = 1 - \alpha_T \frac{\Delta t}{\tau_p} (p - \mathcal{P}) \quad (2.115)$$

where α_T is the isothermal compressibility and τ_p is a relaxation time for the pressure. The characteristic lengths of the systems change according to Eq. (2.115) and $\mathbf{x} \rightarrow V^{\frac{1}{3}}\mathbf{x}$. This rescaling is only valid in isotropic systems where a change in length is the same in every directions. In this case, we just rescale the derivatives as in Sec. 2.5.3. The rescaled Hamiltonian is then given by⁴

$$\mathcal{H} = \sum_{i=1}^N \frac{1}{2} m_i \dot{\mathbf{x}}_i^2 + \frac{1}{2} W \dot{V}^2 + V(\mathbf{x}_1, \dots, \mathbf{x}_N) + pV \quad (2.116)$$

where the new variable V is a volume change controlled by a piston of mass W , that also defines the canonical momentum

$$p_V = W\dot{V}. \quad (2.117)$$

We again derive the equations of motion from the Hamiltonian. The velocities are

$$\begin{aligned} \frac{d\mathbf{x}_i}{dt} &= \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} = \frac{\mathbf{p}_i}{m_i V^{\frac{1}{3}}}, \\ \frac{dV}{dt} &= \frac{\partial \mathcal{H}}{\partial p_V} = \frac{p_V}{W} \end{aligned} \quad (2.118)$$

and the corresponding forces

$$\begin{aligned} \frac{d\mathbf{p}_i}{dt} &= -\frac{\partial \mathcal{H}}{\partial \mathbf{x}_i} = -\nabla_{\mathbf{x}_i} V(V^{\frac{1}{3}}\mathbf{x}_i) = \mathbf{f}_i, \\ \frac{dp_V}{dt} &= -\frac{\partial \mathcal{H}}{\partial V} = -\frac{1}{3V} \sum_{i=1}^N [\mathbf{x}_i \cdot \nabla_{\mathbf{x}_i} V(V^{\frac{1}{3}}\mathbf{x}_i)] - p. \end{aligned} \quad (2.119)$$

The equations of motion of the Berendsen barostat are then

$$\begin{aligned} \ddot{\mathbf{x}}_i &= \frac{\mathbf{f}_i}{m_i} - \frac{\dot{V}}{3V} \mathbf{x}_i, \\ W\ddot{V} &= \frac{1}{3V} \sum_{i=1}^N m_i \dot{\mathbf{x}}_i^2 + \frac{1}{3V} \sum_{i=1}^N \mathbf{f}_i \cdot \mathbf{x}_i - p. \end{aligned} \quad (2.120)$$

⁴ The choice of the letters representing volume and the potential might be confusing since we refer to the volume as V and to the potential as $V(\cdot)$.

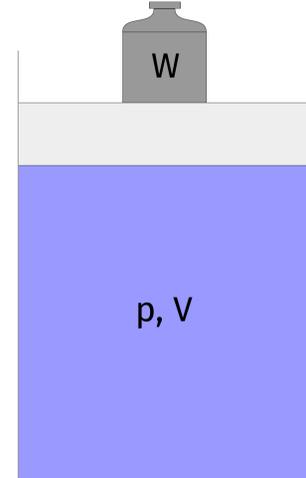


Figure 2.15: A weight of mass W exerts a pressure p on the system with volume V .

2.5.6 Parrinello-Rahman Barostat

In the case of the Berendsen barostat, an isotropic rescaling of space and an isotropic medium are assumed. However, this is generally not the case since, for instance, the volume of certain crystals cannot be rescaled equally in every direction. The first and simplest generalization of the Berendsen barostat is an orthogonal scaling of a box described by three vectors, \mathbf{a} , \mathbf{b} and \mathbf{c} of volume

$$V = \mathbf{a} \cdot (\mathbf{b} \wedge \mathbf{c}) = \det(H) \quad \text{with} \quad H = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}. \quad (2.121)$$

The position of a particle i in the box is described by

$$\mathbf{r}_i = H\mathbf{s}_i = x_i\mathbf{a} + y_i\mathbf{b} + z_i\mathbf{c} \quad \text{with} \quad 0 < x_i, y_i, z_i < 1. \quad (2.122)$$

The distance between two particles i and j is then given by

$$\mathbf{r}_{ij}^2 = \mathbf{s}_{ij}^T G \mathbf{s}_{ij} \quad \text{with} \quad G = H^T H. \quad (2.123)$$

The Hamiltonian as defined by Eq. (2.116) is transformed to

$$\mathcal{H} = \frac{1}{2} \sum_i m_i \dot{\mathbf{s}}_i^T G \dot{\mathbf{s}}_i + \sum_{ij} V(\mathbf{r}_{ij}) + \frac{1}{2} W \text{Tr}(\dot{H}^T \dot{H}) + pV. \quad (2.124)$$

The corresponding equations of motion are

$$\begin{aligned} m_i \ddot{\mathbf{s}}_i &= H^{-1} \mathbf{f}_i - m_i G^{-1} (\dot{G} \dot{\mathbf{s}}_i), \\ W \dot{H} &= pV (H^{-1})^T. \end{aligned} \quad (2.125)$$

The Parrinello-Rahman barostat is very important when simulating crystals, e.g., in solid state physics or in material science. Our degree of freedom is not a simple scalar like in the Hoover thermostat but a matrix H given by the geometry of the system.

2.6 Event-driven Molecular Dynamics

So far, we only considered integration schemes such as Verlet or Leap Frog methods as described in Secs. 2.1.4 and 2.1.5 to solve MD problems. These approaches are, however, not suited to handle hard-core potentials because of the occurrence of infinite forces. In this case, event-driven methods are applicable to simulate the elastic particle collisions by considering the collision events and not the complete time evolution of the systems. In addition, it is also possible to model inelastic collisions with event-driven methods. As opposed to “batch” programming the flow of an event-driven program is not determined by loops but by events, and therefore has branching points and conditional logic. In the next sections, we discuss different examples of event-driven dynamics for the modeling of elastic and inelastic collisions.

2.6.1 Elastic Collisions

As a first example of the application of event-driven methods to MD, we consider a work of Alder [3]. In this work, Alder studied rigid bodies of finite volume such as billiard balls. A hard-core potential is assumed to model the elastic collisions between these particles not taking into account friction. Our standard MD integration schemes fail to describe such systems and we therefore now focus on an event-driven formulation of particle interactions. We therefore regard particle collisions as instantaneous events and between them particles do not interact.

In this method only the exchange of the particles' momenta is taken into account and no forces are calculated. Furthermore, only binary collisions are considered and interactions between three or more particles are neglected. Between two collision events, the particles follow ballistic trajectories. To perform an event-driven MD simulation, we need to determine the time t_c between two collisions to then obtain the velocities of the two particles after the collision from the velocities of the particles before the collision using a look-up table. To

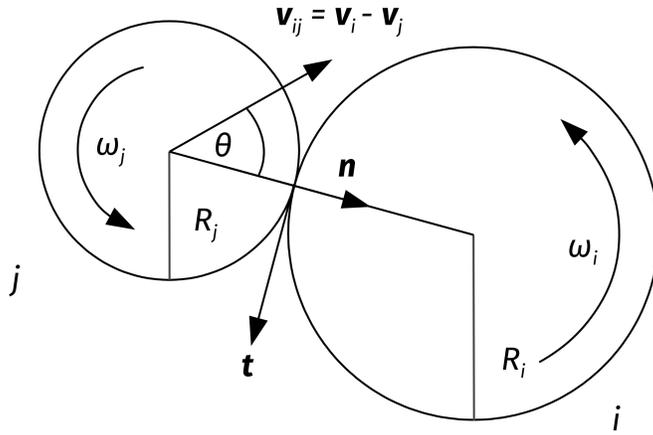


Figure 2.16: Two particles collide elastically.

determine t_c , we have to identify the next collision event. We therefore consider the collision of two disks i and j with radii R_i and R_j , respectively. We show this collision in Fig. 2.16. The *collision angle* θ is the angle between the connecting vector $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and the relative velocity $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$. For the moment, we are not taking into account the influence of friction and thus neglect the exchange of angular momentum. We compute the times t_{ij} at which the next collision between particles i and j occur. At time t_{ij} , the distance between the two particles is

$$|\mathbf{r}_{ij}(t_{ij})| = |R_i + R_j|. \quad (2.126)$$

Given a relative velocity v_{ij} at time t_0 , the contact time t_{ij} can be obtained from

$$v_{ij}^2 t_{ij}^2 + 2 [\mathbf{r}_{ij}(t_0) \cdot \mathbf{v}_{ij}] t_{ij} + [r_{ij}(t_0)]^2 - (R_i + R_j)^2 = 0. \quad (2.127)$$

We should bear in mind that the solutions of Eq. (2.127) are only meaningful if the trajectories of particles i and j cross. The next collision occurs at time

$$t_c = \min_{ij} (t_{ij}). \quad (2.128)$$

Thus, in the time interval $[t_0, t_c]$, the particles' positions and angular orientations evolve according to

$$\mathbf{r}_i(t_0 + t_c) = \mathbf{r}_i(t_0) + \mathbf{v}_i(t_0) t_c \quad \text{and} \quad \phi_i(t_0 + t_c) = \phi_i(t_0) + \omega_i(t_0) t_c. \quad (2.129)$$

It is also possible to add gravitational or electrostatic force terms. The main bottleneck is the computation of the contact times t_{ij} since it is of order $\mathcal{O}(N^2)$. For each particle i , we have to identify particle j which first collides with particle i to then compute t_{ij} . After performing this procedure for all particles, we have to find the minimum according to Eq. (2.128). For high-density particle systems, it is not necessary to determine t_{ij} for distant particles. Still, the distances between all particles need to be considered and thus the algorithm may still be very inefficient. Instead of looking at distances between particles, it is also possible to divide the system into sectors and treat those separately. The crossing of particles through the boundaries of a sector is then considered as a "collision". Another possibility to speed-up this algorithm is given by the *Lubachevsky method* [26].

2.6.2 Lubachevsky Method

Unfortunately, the loop to calculate t_c is of order $\mathcal{O}(N^2)$ when simply checking all pairs. Checking all particles is, however, not very efficient because (i) some collisions are scheduled but only occur after $\mathcal{O}(N)$ steps and (ii) some particles are not participating in any collisions but their positions are always updated. This can be improved by only considering and updating the particles participating in the collision event.

Tricks due to Lubachevsky allow to reduce the order to $\mathcal{O}(N \log N)$ [26]. The improved algorithm is based on lists of events and binary stacks. Specifically, in addition to the particle position and velocity, we store the last event and the next event for each particle. In this way, we are keeping track of the time of the event and the partner particle involved in the event.

In practice, this can be implemented in six arrays (*event times*, *new partners*, *positions* and *velocities*) of dimension N (number of particles

in the system). Alternatively, one creates a list of pointers pointing to a data structure for each particle consisting of six variables.

Storing the last event is needed as particles are only updated after being involved in an event. For each particle i , the time $t^{(i)}$ is the minimal time of all possible collisions involving this particle, i.e.,

$$t^{(i)} = \min_j (t_{ij}). \quad (2.130)$$

Comparing particle i with $N - 1$ others can be improved by dividing the systems in sectors such that only neighboring sectors have to be considered in this step.

These sector boundaries have to be treated similar to obstacles such that when particles cross sector boundaries a collision event happens. For each particle i , this step would then be of order $\mathcal{O}(1)$ instead of $\mathcal{O}(N)$. The next collision occurs at time

$$t_c = \min_i (t^{(i)}). \quad (2.131)$$

We store $t^{(i)}$ in increasing order in a stack:

- The vector `part[m]` points to particle i which is at position m in the stack. (Sometimes also a vector `pos[i]` is used to store position m of particle i in the stack.)
- This constitutes an implicit ordering of the collision times $t^{(i)}$, where $m = 1$ points to the smallest time.
- `part[1]` is the particle with minimal collision time: $t_c = t^{(\text{part}[1])}$.
- After the event for both particles all 6 entries (event times, new partners, positions and velocities) have to be updated. Additionally, the vector `part[m]` has to be re-ordered.

Re-ordering the times $t^{(i)}$ after each event is of order $\mathcal{O}(\log N)$ when using, e.g., binary trees for sorting. The advantages of this method are that it is not necessary to minimize all the collision times of all the pairs at every step, and that it is unnecessary to update the positions of particles that do not collide. Only the position and velocity of the particle involved in the collision event are updated.

2.6.3 Collision with Perfect Slip

After having identified the next collision event, we have to update the particles' position, velocities and angular orientation. With Eq. (2.129), we only considered the time evolution until the collision occurs, and

we therefore now have to consider the particle dynamics after the collision.

In a first approximation, we are assuming perfect slip and neglect any tangential exchange of momentum. Only linear momentum and no angular momentum is exchanged. The conservation of momentum leads to

$$\mathbf{v}_i^{\text{after}} = \mathbf{v}_i^{\text{before}} + \frac{\Delta \mathbf{p}}{m_i}, \quad (2.132)$$

$$\mathbf{v}_j^{\text{after}} = \mathbf{v}_j^{\text{before}} - \frac{\Delta \mathbf{p}}{m_j}, \quad (2.133)$$

and energy conservation to

$$\frac{1}{2}m_i \left(\mathbf{v}_i^{\text{before}}\right)^2 + \frac{1}{2}m_j \left(\mathbf{v}_j^{\text{before}}\right)^2 = \frac{1}{2}m_i \left(\mathbf{v}_i^{\text{after}}\right)^2 + \frac{1}{2}m_j \left(\mathbf{v}_j^{\text{after}}\right)^2. \quad (2.134)$$

The exchanged momentum is

$$\Delta \mathbf{p} = -2m_{\text{eff}} \left[\left(\mathbf{v}_i^{\text{before}} - \mathbf{v}_j^{\text{before}} \right) \cdot \mathbf{n} \right] \mathbf{n} \quad (2.135)$$

with $m_{\text{eff}} = \frac{m_i m_j}{m_i + m_j}$ being the *effective mass* and $\mathbf{n} = \mathbf{r}_{ij} / |\mathbf{r}_{ij}|$. If $m_i = m_j$, the velocity updates are

$$\mathbf{v}_i^{\text{after}} = \mathbf{v}_i^{\text{before}} - v_{ij}^n \cdot \mathbf{n}, \quad (2.136)$$

$$\mathbf{v}_j^{\text{after}} = \mathbf{v}_j^{\text{before}} + v_{ij}^n \cdot \mathbf{n}, \quad (2.137)$$

with $v_{ij}^n = \left(\mathbf{v}_i^{\text{before}} - \mathbf{v}_j^{\text{before}} \right) \cdot \mathbf{n}$. The values can be stored once in a look-up table such that there is no need of calculating the correction to the velocities at every collision.

2.6.4 Collision with Rotation

We now consider two spheres i and j of the same radius R and mass m . Due to friction, angular momentum is exchanged if particles collide with nonzero tangential velocity. The equations of motion for rotation are

$$I \frac{d\boldsymbol{\omega}_i}{dt} = \mathbf{r} \wedge \mathbf{f}_i, \quad (2.138)$$

where I denotes the moment of inertia and \mathbf{f}_i the forces exerted on particle i . For more details on rigid body dynamics, we refer to Sec. 2.3.2.

In the case of two colliding disks of radius R , moment of inertia I and mass m , the exchange of angular momentum is

$$\begin{aligned} I \left(\boldsymbol{\omega}'_i - \boldsymbol{\omega}_i \right) &= -mR\mathbf{n} \wedge \left(\mathbf{v}'_i - \mathbf{v}_i \right), \\ I \left(\boldsymbol{\omega}'_j - \boldsymbol{\omega}_j \right) &= mR\mathbf{n} \wedge \left(\mathbf{v}'_j - \mathbf{v}_j \right), \end{aligned} \quad (2.139)$$

with the primed velocities representing the ones after the collision. Together with the conservation of momentum

$$\mathbf{v}'_i + \mathbf{v}'_j = \mathbf{v}_i + \mathbf{v}_j, \quad (2.140)$$

we obtain the angular velocities after the collision according to

$$\boldsymbol{\omega}'_i - \boldsymbol{\omega}_i = \boldsymbol{\omega}'_j - \boldsymbol{\omega}_j = -\frac{Rm}{I} \mathbf{n} \wedge (\mathbf{v}'_i - \mathbf{v}_i). \quad (2.141)$$

The relative velocity between particles i and j is

$$\mathbf{u}_{ij} = \mathbf{v}_i - \mathbf{v}_j - R(\boldsymbol{\omega}_i + \boldsymbol{\omega}_j) \wedge \mathbf{n}, \quad (2.142)$$

with \mathbf{n} being the unit vector pointing from particle i to particle j . We decompose the relative velocity into their normal and tangential components \mathbf{u}_{ij}^n and \mathbf{u}_{ij}^t , respectively. It is important to keep in mind that we are at this point not interested in the relative velocities of the centers of mass of the particles. For the angular momentum exchange, we have to consider the relative velocity of the particle surfaces at the contact point. The normal and tangential velocities are given by

$$\begin{aligned} \mathbf{u}_{ij}^n &= (\mathbf{u}_{ij} \mathbf{n}) \mathbf{n}, \\ \mathbf{u}_{ij}^t &= \mathbf{u}_{ij} - \mathbf{u}_{ij}^n = -\mathbf{n} \wedge (\mathbf{n} \wedge \mathbf{u}_{ij}) \\ &= \mathbf{v}_i^t - \mathbf{v}_j^t - R(\boldsymbol{\omega}_i + \boldsymbol{\omega}_j) \wedge \mathbf{n}, \end{aligned} \quad (2.143)$$

General slips are described by

$$\mathbf{u}_{ij}^{t'} = e_t \mathbf{u}_{ij}^t, \quad (2.144)$$

where the *tangential restitution coefficient* e_t accounts for different slip types. The perfect slip collision is recovered for $e_t = 1$ which implies that no rotation energy is transferred from one particle to the other. No slip at all corresponds to $e_t = 0$. Energy conservation only holds if $e_t = 1$. Energy is dissipated if $e_t < 1$.

If we compute the difference of the relative tangential velocities before and after the slip event, we obtain

$$\begin{aligned} (1 - e_t) \mathbf{u}_{ij}^t &= \mathbf{u}_{ij}^t - \mathbf{u}_{ij}^{t'} \\ &= -\left[(\mathbf{v}_i^{t'} - \mathbf{v}_i^t - \mathbf{v}_j^{t'} + \mathbf{v}_j^t) - R(\boldsymbol{\omega}'_i - \boldsymbol{\omega}_i + \boldsymbol{\omega}'_j - \boldsymbol{\omega}_j) \wedge \mathbf{n} \right]. \end{aligned} \quad (2.145)$$

Combining the last equation with Eq. (2.141), we obtain an expression without angular velocities

$$\begin{aligned} (1 - e_t) \mathbf{u}_{ij}^t &= \mathbf{u}_{ij}^t - \mathbf{u}_{ij}^{t'} \\ &= -\left[2(\mathbf{v}_i^{t'} - \mathbf{v}_i^t) + 2q(\mathbf{v}_i^{t'} - \mathbf{v}_i^t) \right] \end{aligned} \quad (2.146)$$

and finally

$$\mathbf{v}_i^{t'} = \mathbf{v}_i^t - \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)} \quad \text{with} \quad q = \frac{mR^2}{I}. \quad (2.147)$$

Analogously, we find for the remaining quantities

$$\begin{aligned} \mathbf{v}_j^{t'} &= \mathbf{v}_j^t + \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}, \\ \boldsymbol{\omega}_i' &= \boldsymbol{\omega}_i - \frac{(1 - e_t) \mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R(1 + q^{-1})}, \\ \boldsymbol{\omega}_j' &= \boldsymbol{\omega}_j - \frac{(1 - e_t) \mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R(1 + q^{-1})}. \end{aligned} \quad (2.148)$$

And the updated velocities are

$$\begin{aligned} \mathbf{v}_i' &= \mathbf{v}_i - \mathbf{u}_{ij}^n - \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}, \\ \mathbf{v}_j' &= \mathbf{v}_j + \mathbf{u}_{ij}^n + \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}. \end{aligned} \quad (2.149)$$

2.6.5 Inelastic Collisions

The kinetic energy of interacting and colliding particles is not constant due to friction, plastic deformation or thermal dissipation. As an example, we could think of a rubber ball dropped from a certain height. Due to energy dissipation, the ball will not reach the same height as before after bouncing back from the ground. We account for energy dissipation effects in an effective manner by introducing the *restitution coefficient* r . The restitution coefficient is defined as

$$r = \frac{E^{\text{after}}}{E^{\text{before}}} = \left(\frac{v^{\text{after}}}{v^{\text{before}}} \right)^2, \quad (2.150)$$

where E^{after} and E^{before} are the energies before and after the interaction event, and v^{after} and v^{before} are the corresponding velocities. Elastic collisions correspond to $r = 1$ whereas perfect plasticity is described by $r = 0$. Complementary to our previous discussion of collisions with rotations, we distinguish between normal and tangential energy transfer and define the corresponding coefficients

$$e_n = \sqrt{r_n} = \frac{v_n^{\text{after}}}{v_n^{\text{before}}}, \quad (2.151)$$

$$e_t = \sqrt{r_t} = \frac{v_t^{\text{after}}}{v_t^{\text{before}}}. \quad (2.152)$$

In the case of a bouncing ball, the restitution coefficient accounts for effects such as air friction, deformations and thermal dissipation. These coefficients strongly depend on the material, the shape of the particles, the involved energies, the angle of impact and other factors. Usually, they are determined experimentally.

The relative velocity of the particles at their contact point is

$$\mathbf{u}_{ij}^n = (\mathbf{u}_{ij}\mathbf{n}) \mathbf{n} = [(\mathbf{v}_i - \mathbf{v}_j) \mathbf{n}] \mathbf{n}. \quad (2.153)$$

The normal velocity components are affected by inelasticity. In the case of an inelastic collision, dissipation effects lead to reduced normal velocities

$$\mathbf{u}_{ij}^{n'} = e_n \mathbf{u}_{ij}^n. \quad (2.154)$$

For $e_n = 1$, there is no dissipation whereas dissipation effects occur for $e_n < 1$. We follow the derivations of Sec. 2.6.4.

The velocities after the collision are given by

$$\begin{aligned} \mathbf{v}_i' &= \mathbf{v}_i - \frac{(1 + e_n)}{2} \mathbf{u}_{ij}^n, \\ \mathbf{v}_j' &= \mathbf{v}_j + \frac{(1 + e_n)}{2} \mathbf{u}_{ij}^n. \end{aligned} \quad (2.155)$$

In the case of perfect slip, the momentum exchange is

$$\Delta \mathbf{p}_n = -m_{\text{eff}}(1 + e_n) [(\mathbf{v}_i - \mathbf{v}_j) \mathbf{n}] \mathbf{n}. \quad (2.156)$$

With $q = \frac{mR^2}{I}$, the generalized set of Eqs. (2.147) and (2.148) is given by

$$\begin{aligned} \mathbf{v}_i' &= \mathbf{v}_i - \frac{(1 + e_n)}{2} \mathbf{u}_{ij}^n - \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}, \\ \mathbf{v}_j' &= \mathbf{v}_j + \frac{(1 + e_n)}{2} \mathbf{u}_{ij}^n + \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}, \\ \boldsymbol{\omega}_i' &= \boldsymbol{\omega}_i - \frac{(1 - e_t) \mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R(1 + q^{-1})}, \\ \boldsymbol{\omega}_j' &= \boldsymbol{\omega}_j - \frac{(1 - e_t) \mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R(1 + q^{-1})} \end{aligned} \quad (2.157)$$

and describes inelastic collisions of rotating particles. For almost all physical particle simulations, it is important to incorporate energy dissipation effects by accounting for inelastic collisions. A prominent example is the so-called *inelastic collapse*. This effect occurs in regions of high particle densities, because the larger number of interactions leads to larger dissipation effects and allows particles to form clusters of locally higher densities. This effect is important to simulate the dynamics of galaxies. Without this effect, stars would not be clustered in the way they are in the universe.

2.6.6 Inelastic Collapse

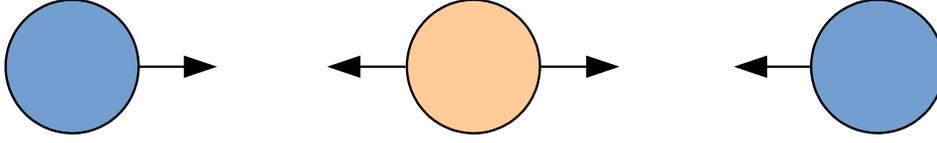


Figure 2.17: The orange particle at the center is bouncing between the blue particles which approach each other.

When simulating inelastic collisions, we may encounter finite time singularities as described by McNamara [27, 28]. This effect is particularly important for the simulation of high density particle systems. One example is a particle bouncing between two other particles which approach each other. This situation is illustrated in Fig. 2.17. To understand the effect with an even simpler model, we consider a ball bouncing vertically on a hard surface. Every time the ball hits the surface its kinetic energy is lowered according to Eq. (2.150). As a consequence, the ball will not reach the initial height anymore and the time between two contacts with the surface approaches zero. After a finite time, the ball comes to a rest, but the simulation takes infinite time to run. In an event-driven simulation, the ball never stops its motion and the number of events per time step increases. A similar problem is the famous *Zenon Paradox*⁵. The total time needed for the bouncing ball to come to rest is the sum over an infinite number of times between two surface contacts t_i .

Since the height is directly proportional to the energy, it also scales with the restitution coefficient at every surface contact. Consequently, after the i^{th} surface contact, the damping of the height is proportional to r^i . The total time is given by

$$\begin{aligned} t_{\text{tot}} &= \sum_{i=1}^{\infty} t_i \\ &= 2\sqrt{\frac{2h^{\text{initial}}}{g}} \sum_{i=1}^{\infty} \sqrt{r^i} \\ &= 2\sqrt{\frac{2h^{\text{initial}}}{g}} \left(\frac{1}{1-\sqrt{r}} - 1 \right), \end{aligned} \quad (2.158)$$

where h^{initial} is the initial height and g the gravitational acceleration.

The problem for event-driven particle simulations lies in the assumption that interactions are instantaneous. However, real collisions have a certain duration. Luding and McNamara introduced a restitution coefficient that depends on the time elapsed since the last event [29]. It is assumed that a collision takes a finite time t_{contact} . If the time since the last collision of one of the interacting particles $t^{(i)}$ or $t^{(j)}$ is less than t_{contact} , the coefficient is set to unity and otherwise to r . We thus obtain

⁵ https://en.wikipedia.org/wiki/Zeno's_paradoxes

$$r^{(i,j)} = \begin{cases} r, & \text{for } t^{(i)} > t_{\text{contact}} \quad \text{or} \quad t^{(j)} > t_{\text{contact}} \\ 1, & \text{otherwise.} \end{cases} \quad (2.159)$$

With this new re-definition of the restitution coefficient, the collision type changes from inelastic to elastic if too many collisions occur during t_{contact} . Depending on the material properties of the colliding particles, it is also possible to use more complex functional dependencies of the restitution coefficient. In the case of very dense and viscous materials, zero instead of unity may be a better choice. In this case, the particles form clusters and stick together.

We now understand the main differences between the hard sphere event-driven algorithm and the soft potential approaches. Binary and instantaneous collisions determine the dynamics in an event-driven simulation whereas multiple interactions might occur at the same time in the other case.

2.7 Inelastic Collisions in Molecular Dynamics

In many cases, inelasticity is not just a minor correction to a well-behaved and converging mathematical method. Many systems exhibit no convergent behavior if dissipation is not taken into account. From landslides to turbulent flow in aircraft engines, energy loss and dissipation are often necessary for the simulations to be stable and useful in practice. In this section, we briefly return to non event-driven MD techniques to slightly modify and adapt these methods to incorporate inelastic effects. We first focus on simple models such as damped oscillators and plastic deformations.

2.7.1 Damped Oscillator

In MD simulations that are not based on event-driven dynamics, we incorporate inelasticity with the help of a damped oscillator. For the corresponding equations of motion, we define the radial distance between two particles as $r = R_i + R_j - |\mathbf{x}_i - \mathbf{x}_j|$ and the effective mass as $m_{\text{eff}} = \frac{m_1 m_2}{m_1 + m_2}$. The equation of motion is then

$$m_{\text{eff}} \ddot{r} = -kr - \gamma \dot{r}, \quad (2.160)$$

where k and γ are the spring constant and a viscous damping coefficient, respectively. The solution is given by

$$r(t) = \frac{v^{\text{before}}}{\omega} \sin(\omega t) \exp(-\delta t), \quad (2.161)$$

where $\omega = \sqrt{\omega_0^2 - \delta^2}$, $\omega_0 = \sqrt{\frac{k}{m_{\text{eff}}}}$ and $\delta = \frac{\gamma}{2m_{\text{eff}}}$ are the oscillator frequency, the ground frequency of the undamped oscillator and the

damping factor, respectively.

A damping of the oscillator changes its frequency and the period time T generally increases with the damping.

We regard collisions as a single damped oscillator cycle, for which some energy is dissipated due to the damping. The period time of the oscillator is defined by attractive and repulsive motion of interacting particles. The collision lasts half the period of an oscillation, and when the particle bounces back its kinetic energy is reduced. We therefore compute the restitution coefficient according to

$$e_n = \frac{\dot{r}(t+T)}{\dot{r}(t)} = \exp(-\delta T) = \exp\left(-\frac{\gamma\pi}{\sqrt{4m_{\text{eff}}k - \gamma^2}}\right). \quad (2.162)$$

This relation implies that a restitution coefficient uniquely determines a viscous damping coefficient according to

$$\gamma = 2 \ln(e_n) \sqrt{\frac{m_{\text{eff}}k}{\ln(e_n)^2 + \pi^2}}. \quad (2.163)$$

An important assumption that has been made is that the restitution coefficient is constant throughout the complete interaction. This assumption is violated for a Hertzian Contact for which sphere interactions lead to altered shapes. In every collision, the kinetic energy is reduced due to plasticity. The dependence between the energy loss and the interaction overlap is non trivial. Our approach has then to be corrected since the restitution coefficient is not constant.

2.7.2 Plastic Deformation

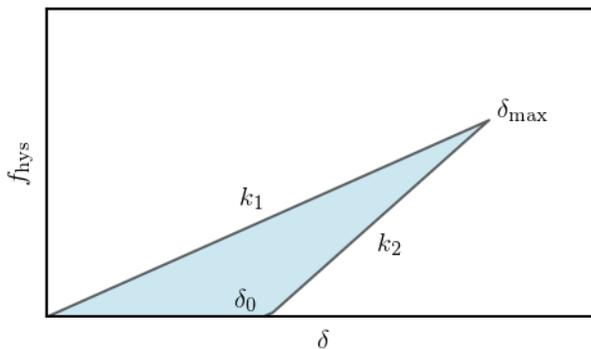


Figure 2.18: An example of a plastic deformation.

In many engineering and industrial applications, collisions of two deformable objects are characterized by an irreversible change of their

shapes, a so-called *plastic deformation*. An example is shown in Fig. 2.18.

One approach to model plasticity has been introduced by Walton and Braun [30]. The attraction and repulsion events are modeled by springs with two different spring constants k_1 and k_2 . This results in hysteretic and non-linear behavior. In particular, the interaction is dissipative and non-elastic. We show the interaction process in Fig. 2.18.

The force is

$$f_{\text{hys}} = \begin{cases} k_1 \delta, & \text{loading,} \\ k_2 (\delta - \delta_0), & \text{unloading.} \end{cases} \quad (2.164)$$

First, the objects approach each other according to Hook's law and overlap up to δ_{max} so that $k_1 = f_{\text{max}}/\delta_{\text{max}}$. During this time the objects are deformed and therefore repulsion occurs with a different spring constant $k_2 = f_{\text{max}}/(\delta_{\text{max}} - \delta_0)$ such that the repulsive forces vanish before the overlap parameter does.

The collision time is not symmetric since loading lasts longer than unloading, i.e.,

$$t_c = \frac{\pi}{2} \left(\sqrt{\frac{m_{ij}}{k_1}} + \sqrt{\frac{m_{ij}}{k_2}} \right). \quad (2.165)$$

The dissipated energy corresponds to the area enclosed by the two curves. From this observation, we obtain the restitution coefficient

$$r = \frac{E^{\text{after}}}{E^{\text{before}}} = \frac{k_1}{k_2}. \quad (2.166)$$

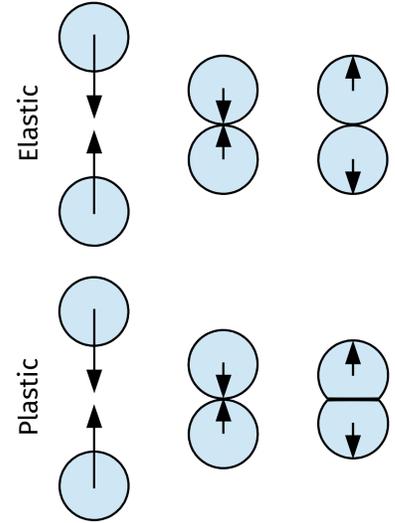


Figure 2.19: Two particles undergoing elastic and plastic interactions.

2.7.3 Coulomb Friction and Discrete Element Method

The energy dissipation due to Coulomb friction is proportional to the normal force. Two friction coefficients are used to take such dissipation effects into account. The static friction coefficient describes friction effects of non-moving objects and the dynamic coefficient accounts for friction of moving particles. The classic example of Coulomb friction is an object located on an inclined plane. At a certain inclination angle, the *friction angle*, the object begins to slide down the plane. The tangent of the angle gives the static friction coefficient. The motion of the object is independent of the contact area.

Numerically, the problem is very difficult to handle, because the friction coefficient μ is described by a discontinuous function. For non-zero tangential velocities $v_t > 0$ the dynamic friction coefficient μ_d is different from the static one μ_s when $v_t = 0$.

One approach is to distinguish between a small shear velocity v_s below which static friction is implemented and above which dynamic

friction is used [31]. The tangential momentum transfer is then given by

$$\begin{aligned}\Delta p_t &= \mu_s m_{\text{eff}} (e_t + 1) \left(\mathbf{v}_i^{\text{before}} - \mathbf{v}_j^{\text{before}} \right) \mathbf{t} \quad \text{for } v_s \ll 1, \\ \Delta p_t &= \mu_d m_{\text{eff}} (e_t + 1) \left(\mathbf{v}_i^{\text{before}} - \mathbf{v}_j^{\text{before}} \right) \mathbf{t} \quad \text{for } v_s \gg 1,\end{aligned}\tag{2.167}$$

where \mathbf{t} is the tangential unit vector. For multiple particles interacting via friction such as sand or stones, a method proposed by Cundall is nowadays a standard [32]. This forward integration technique is widely used in engineering and industrial applications. The company Itasca has been founded by Cundall himself and is the leading provider of software which is based on this method.

Similarly to the method mentioned before, we introduce two terms for the strength, one for the static and one for the dynamic interaction, i.e.,

$$f_t = -\min[\gamma v_t, \mu_d f_n] \text{sign}(v_t)\tag{2.168}$$

$$f_t = -\min[|k_s \zeta|, \mu_s f_n] \text{sign}(v_t)\tag{2.169}$$

The difference from the method used before is the behaviour of the particles if their velocities are very small. If the velocities are smaller than a certain value, then a spring is applied by introducing an elastic force with constant k_s . If $|k_s \zeta| > \mu_s f_n$, then the spring is removed and the particles are free to move with the static or dynamic friction coefficient depending on their velocity.

2.8 Arbitrary Shapes

So far, we only considered spherical particles in our discussion of event-driven MD methods. However, it would be also desirable to simulate interactions between arbitrarily shaped particles. Such methods are relevant for the study of numerous systems which consist of particles of irregular shapes. For some cases such as ellipsoidal particles, it is possible to obtain analytical solutions of the interaction dynamics. For the more general treatment of arbitrarily shaped particles, we also discuss polygon and spheropolygon methods.

2.8.1 Ellipsoidal Particles

To simulate the interaction of ellipsoidal particles, we consider two ellipses whose parameterization is given by

$$\left(\frac{x - x_a}{a_1} \right)^2 + \left(\frac{y - y_a}{a_2} \right)^2 = 1,\tag{2.170}$$

$$\left(\frac{x - x_b}{b_1} \right)^2 + \left(\frac{y - y_b}{b_2} \right)^2 = 1.\tag{2.171}$$

The two ellipses are illustrated in Fig. 2.20. As suggested by Perram

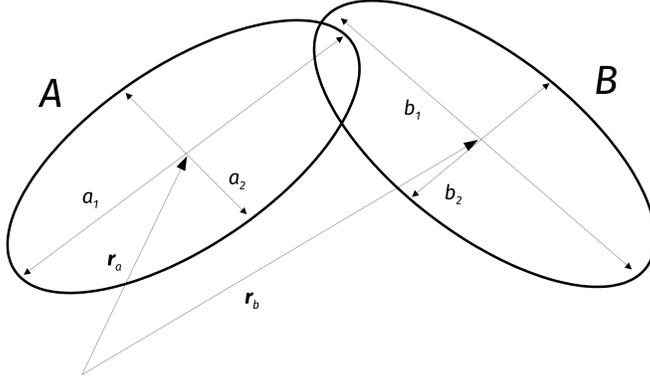


Figure 2.20: Two overlapping ellipses.

and Wertheim, we take the overlap of the two ellipses as a measure for the interaction [33]. To calculate the overlap, we transform the ellipses into circles using an appropriate metric. Let $\mathbf{u}_1, \mathbf{u}_2$ and $\mathbf{v}_1, \mathbf{v}_2$ be orthonormal vectors along the major axes of the ellipses A and B, respectively. The rotated ellipses are defined by the matrices

$$A = \sum_k a_k^{-2} \mathbf{u}_k \otimes \mathbf{u}_k^T \quad \text{and} \quad B = \sum_k b_k^{-2} \mathbf{v}_k \otimes \mathbf{v}_k^T. \quad (2.172)$$

Ellipse A is then described by the functional

$$G_A(\mathbf{r}) = (\mathbf{r} - \mathbf{r}_a)^T A (\mathbf{r} - \mathbf{r}_a) \quad (2.173)$$

which is smaller than one inside the ellipse, larger than one outside and exactly one on the ellipse, i.e.,

$$G_A(\mathbf{r}) = \begin{cases} < 1, & \text{if } \mathbf{r} \text{ is inside the ellipse,} \\ 1, & \text{if } \mathbf{r} \text{ is on the ellipse,} \\ > 1, & \text{if } \mathbf{r} \text{ is outside the ellipse.} \end{cases} \quad (2.174)$$

The same formalism is applicable to ellipse B. We now define another functional

$$G(\mathbf{r}, \lambda) = \lambda G_A(\mathbf{r}) + (1 - \lambda) G_B(\mathbf{r}), \quad (2.175)$$

which describes the two ellipses with a parameter $\lambda \in [0, 1]$ that interpolates between the two centers of the ellipses defined by Eq. (2.173). With the help of $G(\mathbf{r}, \lambda)$, we want to compute the contact point. Therefore, we have to minimize the functional according to

$$\nabla_{\mathbf{r}} G(\mathbf{r}, \lambda)|_{\mathbf{r}=\mathbf{r}_m} = 0. \quad (2.176)$$

This yields

$$\mathbf{r}_m(\lambda) = [\lambda A + (1 - \lambda) B]^{-1} [\lambda A \mathbf{r}_a + (1 - \lambda) B \mathbf{r}_b]. \quad (2.177)$$

If we start from $\lambda = 0$ and arrive at $\lambda = 1$, we obtain a path from the center of the first ellipse to the center of the second one. We rewrite the path in terms of $\mathbf{r}_{ab} = \mathbf{r}_b - \mathbf{r}_a$ and find

$$\begin{aligned} \mathbf{r}_m(\lambda) &= \mathbf{r}_a + (1 - \lambda) A^{-1} \left[(1 - \lambda) A^{-1} + \lambda B^{-1} \right]^{-1} \mathbf{r}_{ab}, \\ \mathbf{r}_m(\lambda) &= \mathbf{r}_b - \lambda B^{-1} \left[(1 - \lambda) A^{-1} + \lambda B^{-1} \right]^{-1} \mathbf{r}_{ab}. \end{aligned} \quad (2.178)$$

If the value of the functional along these paths between the centers is always smaller than unity, we know that we never left the interior of the ellipses, and hence they overlap. We define the *overlap function*

$$S(\lambda) = G(\mathbf{r}_m(\lambda), \lambda) \quad (2.179)$$

and by inserting Eq. (2.178) we obtain

$$S(\lambda) = \lambda(1 - \lambda) \mathbf{r}_{ab}^T \left[(1 - \lambda) A^{-1} + \lambda B^{-1} \right]^{-1} \mathbf{r}_{ab}. \quad (2.180)$$

This corresponds to the length of the minimal path that connects the two centers. As already mentioned, we are interested if the path ever exhibits values larger than unity. If we maximize S , we will be able to tell if the ellipses are overlapping or not:

$$S(\lambda_{\max}) = \begin{cases} < 1, & \text{if the ellipses overlap,} \\ 1, & \text{if the ellipses touch,} \\ < 1, & \text{if the ellipses are separated.} \end{cases} \quad (2.181)$$

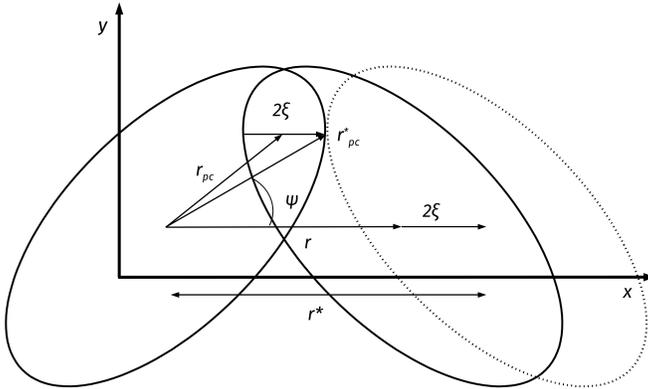


Figure 2.21: Overlap vector and contact point for two ellipses.

With this knowledge, we compute the contact point for stiff ellipses where overlaps in the simulations should be avoided. We set $S(\lambda_{\max})$ to unity and find the contact point with

$$\mathbf{r}^* = \left\{ \lambda_{\max} (1 - \lambda_{\max}) \mathbf{e}_r^T \left[(1 - \lambda_{\max}) A^{-1} + \lambda_{\max} B^{-1} \right]^{-1} \mathbf{e}_r \right\}^{-\frac{1}{2}}, \quad (2.182)$$

where $\mathbf{r}_{ab} = r\mathbf{e}_r$. We define the overlap vector as

$$\boldsymbol{\zeta} = \frac{\mathbf{r}^* \mathbf{e}_r - \mathbf{r}_{ab}}{2} \quad (2.183)$$

and the corresponding contact point as

$$\mathbf{r}_{pc} = \mathbf{r}_{pc}^* + \boldsymbol{\zeta}. \quad (2.184)$$

This is illustrated in Fig. 2.21.

Moreover, there exist generalized versions of ellipses, so-called *superellipsoids*, and their macroscopic properties exhibit interesting features such as high packing densities [34, 35]. An example of a superellipsoid packing is shown in Fig. 2.22. Industrial engineering, biological system design (e.g., blood cells) and many other fields often rely on simulations of macroscopic particles, often ellipsoidal, and this is why these techniques found such a big resonance today.

We saw that even for the simplest generalization of spheres, it was already necessary to develop complex analytical methods. This substantially increases the computational complexity of the needed simulation algorithms. For this reason, it is necessary to develop new approaches to handle objects of arbitrary shapes.

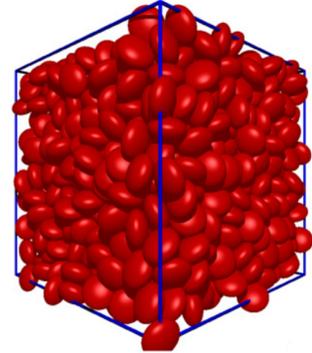


Figure 2.22: An example of a superellipsoid packing. The figure is taken from Ref. [35].

2.8.2 Polygons

A particular class of macroscopic particles are those described by polygons such as rocks and sand grains. In this case, a better measure for the repulsive forces is the overlap area. The advantage of simple polygons is that it is possible to compute the overlap area using simple geometries (e.g., dividing the area into triangles). However, between polygons there can be many different types of contacts, and the classification and the identification of all the types of contacts is very cumbersome as shown in Fig. 2.23.

Additional complexities arise when the overlap area does not represent the actual overlap of the polygons or when discontinuities appear while particles move into another. Furthermore, rotation of particles require an additional treatment, as the torque strongly depends on the shape of the particles. Due to the mentioned reasons, the simulation of objects composed of polygons may be cumbersome.

2.8.3 Spheropolygons

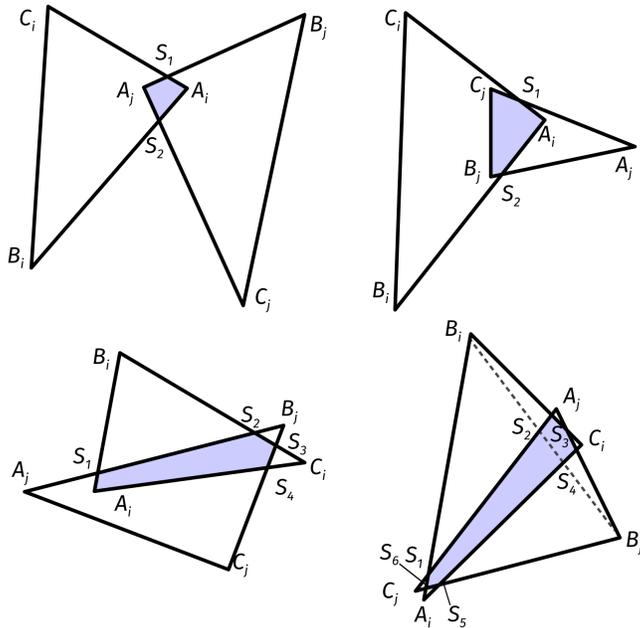


Figure 2.23: Possible overlaps of triangles.

Another class of methods that is more efficient than the mere division into polygons relies on *spheropolygons* [36]. This technique was invented by Fernando Alonso-Marroquin, who uses the *Minkowski cow* for demonstrative purposes. An example of a Minkowski cow is shown in Fig. 2.24. This particular shape is obtained by sweeping a disk around a polygon of multiple and arbitrary edges that resemble the shape of a cow.

The important feature of the spheropolygon method is that it is possible to simulate arbitrary shapes. Once the shape is decomposed into spheres, we only have to compute the contact between all the pairs of spheres that define the edges and vertices of the shape which is easier than considering arbitrary shaped polygons. There are of course a number of constraints when approximating the shapes in such a way. For example, too large spheres would smear out the original shape and would even lead to wrong results. Imagine that the spheres at the edges were larger than the distance between the hooves or the distance between the tale and the rest of the cow. Substantial features of the shape would be lost.

Many other techniques have been developed to describe arbitrary shaped objects. Many attempts to create more effective implementations are developed by engineers, physicists and mathematicians. An important example is the field of *mathematical morphology* developed by Jean Serra in the 1960s [37]. The techniques of Marroquin are related to the so-called *dilation techniques* but there are also other methods like

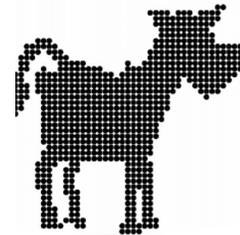


Figure 2.24: The Minkowski cow as an example of an irregular shape composed of spheropolygons. The figure is taken from Ref. [36].

the *erosion technique*. There exist specialized courses which only cover some of these methods and techniques.

2.9 Contact Dynamics

We started our discussion of MD simulations by introducing integration methods to directly solve Newton's equations of motion for a certain number of interacting particles. Such direct integration methods are able to capture the dynamics of particles which interact via soft potentials. To overcome the problem of infinite forces in the case of hard-sphere potentials, we introduced event-driven MD techniques in Sec. 2.6. The advantage of event-driven MD is that it is not necessary to compute forces, because we are only considering binary collision events and accounting for the corresponding momentum and angular momentum transfer. However, there also exist some drawbacks such as the inapplicability of this method for studying long-lasting contacts. Such long-lasting contacts may occur in granulates, rocks, and machine parts. One example is shown in Fig. 2.25. Until now, we have not introduced a method which is able to simulate such a woodpecker toy. Simulating such and related systems requires to also resolve the observed contact interactions. This is done in the context of *contact dynamics*. The basic idea is that particle interactions are fully determined by constraints. Specifically, to prevent the penetration of particles, we impose constraint forces at their contacts.

Per Lötstedt is one of the founders of contact dynamics but just with contributions of Jean-Jacques Moreau the field started flourishing. Moreau was working in Montpellier and he mainly focused on topics in the fields of numerical analysis and elliptic equations. Even after his retirement, he was actively engaged in scientific research, and worked almost until the end of his life, aged 90 years [39]. The field of contact dynamics is strongly related to the study of problems in non-smooth mechanics where the time evolution of the particles' positions and momenta are not assumed to be described by smooth functions anymore. An important problem in this field is the *ambiguous boundary condition problem* or *Signorini problem* which has been originally formulated by Alberto Signorini. The problem is to find the shape of an elastic body located on a rigid surface. We could, for example, think of a rubber placed on the surface of a table. Perfectly rigid bodies with perfect volume exclusion are described by the *Signorini graph*. If particles are not in contact, there are also no forces between them. Forces only occur if the distance between both particles is zero. As shown in Fig. 2.28, this causes force discontinuities which are difficult to handle numerically. Such force discontinuities may also occur due to friction between particles. We therefore consider two particles that are in contact. If their

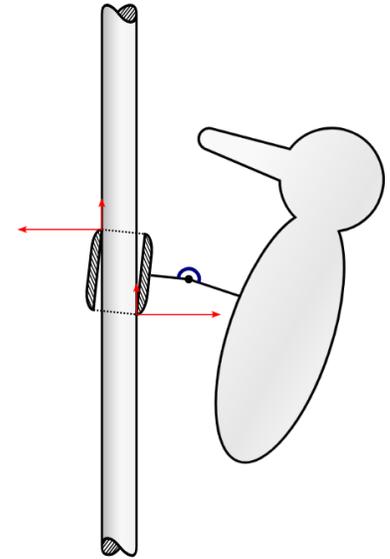


Figure 2.25: A woodpecker as a benchmark example for contact dynamics problems [38]. An animation can be found on <https://github.com/gabyx/Woodpecker>.



Figure 2.26: Per Lötstedt



Figure 2.27: Jean-Jacques Moreau (1923-2014)

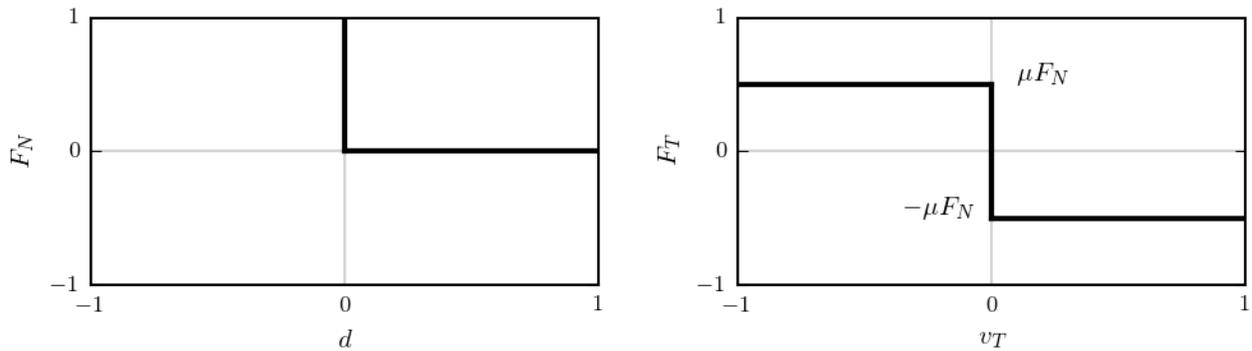


Figure 2.28: Signorini (left) and Coulomb (right) graphs.

relative velocity is zero, there is no friction. However, for non-zero relative velocities, we have to take into account friction between the two particles. The resulting discontinuous behavior is shown in the *Coulomb graph* in Fig. 2.28.

2.9.1 One-dimensional Contact

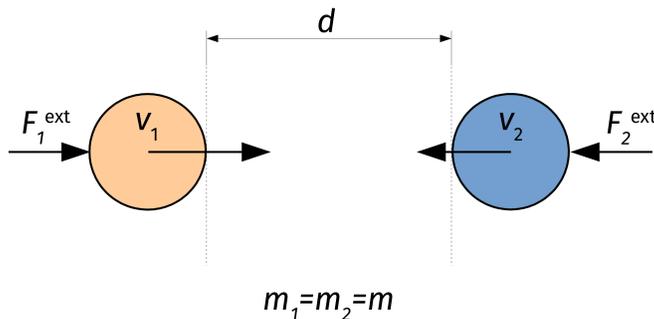


Figure 2.29: Example of a one-dimensional contact.

To familiarize with the concepts of contact dynamics, we first focus on the example of a one-dimensional contact as shown in Fig. 2.29. In this example, we neglect friction and only consider normal forces at the contact point of the two particles. Both particles have mass m and velocities v_1 and v_2 , respectively. We have to make sure that these two particles do not overlap. Therefore, we impose constraint forces in such a way that they compensate all other forces which would lead to overlaps. Such constraint forces should be defined in such a way that they have no influence on the particle dynamics before and after the contact. The time evolution of the particles' positions and velocities is

described by an implicit Euler scheme which is given by

$$\begin{aligned}\mathbf{v}_i(t + \Delta t) &= \mathbf{v}_i(t) + \Delta t \frac{1}{m_i} \mathbf{F}_i(t + \Delta t), \\ \mathbf{r}_i(t + \Delta t) &= \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t + \Delta t).\end{aligned}\quad (2.185)$$

The forces $\mathbf{F}_i(t) = \mathbf{F}_i^{\text{ext}}(t) + \mathbf{R}_i(t)$ consist of an external $\mathbf{F}_i^{\text{ext}}(t)$ and a contact term $\mathbf{R}_i(t)$. So far, we only considered forces that act on the center of mass. However, contact forces act locally on the contact point and not on the center of mass. We therefore introduce a matrix H which transforms local contact forces into particle forces, and the corresponding transpose H^T transforms particle velocities into relative velocities. We consider the example of Fig. 2.29.

This leads to

$$v_n^{\text{loc}} = v_2 - v_1 = \begin{pmatrix} -1 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = H^T \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}\quad (2.186)$$

and

$$\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} -R_n^{\text{loc}} \\ R_n^{\text{loc}} \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} R_n^{\text{loc}} = H R_n^{\text{loc}}.\quad (2.187)$$

The equations of motion for both particles are

$$\frac{d}{dt} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \frac{1}{m} \left[\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} + \begin{pmatrix} F_1^{\text{ext}} \\ F_2^{\text{ext}} \end{pmatrix} \right].\quad (2.188)$$

Combining the last equation with the transformation of Eq. (2.187), we obtain

$$\begin{aligned}\frac{dv_n^{\text{loc}}}{dt} &= \begin{pmatrix} -1 & 1 \end{pmatrix} \frac{1}{m} \left[\begin{pmatrix} -1 \\ 1 \end{pmatrix} R_n^{\text{loc}} + \begin{pmatrix} F_1^{\text{ext}} \\ F_2^{\text{ext}} \end{pmatrix} \right] \\ &= \frac{1}{m_{\text{eff}}} R_n^{\text{loc}} + \frac{1}{m} (F_2^{\text{ext}} - F_1^{\text{ext}}),\end{aligned}\quad (2.189)$$

where $m_{\text{eff}} = m/2$ is the effective mass and $\frac{1}{m} (F_2^{\text{ext}} - F_1^{\text{ext}})$ the acceleration without contact forces.

We integrate the last equation with an implicit Euler method as defined by Eq. (2.185) and find

$$\frac{v_n^{\text{loc}}(t + \Delta t) - v_n^{\text{loc}}(t)}{\Delta t} = \frac{1}{m_{\text{eff}}} R_n^{\text{loc}}(t + \Delta t) + \frac{1}{m} (F_2^{\text{ext}} - F_1^{\text{ext}}).\quad (2.190)$$

The unknown quantities in this equation are v_n^{loc} and R_n^{loc} . To find a solution, we make use of the Signorini constraint and compute

$$R_n^{\text{loc}}(t + \Delta t) = m_{\text{eff}} \frac{v_n^{\text{loc}}(t + \Delta t) - v_n^{\text{loc, free}}(t + \Delta t)}{\Delta t}\quad (2.191)$$

with

$$v_n^{\text{loc, free}}(t + \Delta t) = v_n^{\text{loc}}(t) + \Delta t \frac{1}{m} (F_2^{\text{ext}} - F_1^{\text{ext}}). \quad (2.192)$$

No overlap (e.g., no contact or closing contact) at time $t + \Delta t$ corresponds to

$$d(t + \Delta t) = d(t) + v_n^{\text{loc}}(t + \Delta t)\Delta t \geq 0 \quad (2.193)$$

and $R_n^{\text{loc}}(t + \Delta t) = 0$. Otherwise a contact persists or forms during the time step, i.e., the gap d closes, and we impose

$$v_n^{\text{loc}}(t + \Delta t) = -\frac{d(t)}{\Delta t}. \quad (2.194)$$

The contact force is then given by

$$R_n^{\text{loc}}(t + \Delta t) = m_{\text{eff}} \frac{-d(t)/\Delta t - v_n^{\text{loc, free}}(t + \Delta t)}{\Delta t}. \quad (2.195)$$

The Signorini condition can be reformulated in terms of velocities to then find the intersection with Eq. (2.191). For particles that are not in contact, the contact force vanishes and the relative velocity is different from zero. This case corresponds to the *open solution*. In the case of a *persisting contact*, the contact force is different from zero and the relative velocity vanishes.

We distinguish between the possible cases:

- Particles are not in contact,
- Particles are in closing contact,
- Particles are in persisting contact and,
- Particles are in opening contact.

It is not necessary to consider the influence of contact forces, if the particles are not in contact. However, for particles that are approaching each other, we have to take into account the possibility of overlaps in every time step. In the case of an overlap, constraint forces have to be applied such that the overlap vanishes. In addition to the normal contact forces, we may also consider tangential contact forces contributions. The reason is that collisions between grains are usually inelastic, i.e., energy is dissipated through vibrations (sound) and eventually also small plastic deformations and heat production. Therefore, we also have to consider tangential contact forces in the case of friction. For simplicity, we may assume that the static and dynamic friction coefficients are equal ($\mu_s = \mu_d$). Similar to the solution of Eq. (2.191), the solution for the tangential contact force can be obtained with the help of the Coulomb graph of Fig. 2.28.

We now extend the described contact dynamics approach to three dimensions. In particular, we consider particle interactions without

friction. Thus, we do not need to take into account angular velocities and torques. In three dimensions, velocities and forces are given by

$$\mathbf{v}_{12} = \begin{pmatrix} v_{12}^x \\ v_{12}^y \\ v_{12}^z \end{pmatrix} \quad \mathbf{R}_{12} = \begin{pmatrix} R_{12}^x \\ R_{12}^y \\ R_{12}^z \end{pmatrix} \quad \mathbf{F}_{12}^{\text{ext}} = \begin{pmatrix} F_{12}^{x,\text{ext}} \\ F_{12}^{y,\text{ext}} \\ F_{12}^{z,\text{ext}} \end{pmatrix}. \quad (2.196)$$

Only normal components v_n^{loc} and R_n^{loc} have to be considered during particle contact. We therefore project all necessary variables onto the normal vector

$$\mathbf{n} = \begin{pmatrix} n^x \\ n^y \\ n^z \end{pmatrix} \quad (2.197)$$

and obtain

$$v_n^{\text{loc}} = \mathbf{n} \cdot (\mathbf{v}_2 - \mathbf{v}_1) \quad \mathbf{R}_1 = -\mathbf{n}R_n^{\text{loc}} \quad \mathbf{R}_2 = \mathbf{n}R_n^{\text{loc}}. \quad (2.198)$$

From the projection, we obtain the matrix H for the coordinate transformation

$$v_n^{\text{loc}} = H^T \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{pmatrix} = HR_n^{\text{loc}}, \quad (2.199)$$

with

$$H^T = (-n_x, -n_y, -n_z, n_x, n_y, n_z). \quad (2.200)$$

Friction can be included by considering angular velocities and torques.

2.9.2 Generalization to N Particles

To describe the interactions of N particles using contact dynamics methods, we use the generalized coordinates

$$\dot{q} = \begin{pmatrix} \mathbf{v}_1 \\ \boldsymbol{\omega}_1 \\ \vdots \\ \mathbf{v}_N \\ \boldsymbol{\omega}_N \end{pmatrix}, \quad R = \begin{pmatrix} \mathbf{R}_1 \\ \mathbf{T}_1 \\ \vdots \\ \mathbf{R}_N \\ \mathbf{T}_N \end{pmatrix} \quad \text{and} \quad F^{\text{ext}} = \begin{pmatrix} \mathbf{F}_1^{\text{ext}} \\ 0 \\ \vdots \\ \mathbf{F}_N^{\text{ext}} \\ 0 \end{pmatrix}. \quad (2.201)$$

The number of components depends on the dimension of the simulated system. In two dimensions we have $3N$ components (2 translational and 1 rotational per particle) whereas in three dimensions there are $6N$ components (3 translational and 3 rotational per particle).

Let c be the number of contacts. In two dimensions, there exist $2c$ contact variable components (1 normal and 1 tangential), while in

three dimensions the number of contact variable components is $3c$ (1 normal and 2 tangential). We neglect the influence of torques at the contact points and find

$$\mathbf{u} = \begin{pmatrix} \mathbf{v}_1^{\text{loc}} \\ \vdots \\ \mathbf{v}_c^{\text{loc}} \end{pmatrix} \quad \text{and} \quad \mathbf{R}^{\text{loc}} = \begin{pmatrix} \mathbf{R}_1^{\text{loc}} \\ \vdots \\ \mathbf{R}_c^{\text{loc}} \end{pmatrix}. \quad (2.202)$$

The corresponding transformation into contact variables is given by

$$u = H^T \dot{q} \quad \text{and} \quad R = HR^{\text{loc}}. \quad (2.203)$$

The matrix H is not constant as in the case of the one-dimensional interactions of two particles. At every time step, the particles involved in a certain contact and the kind of contact change. The dimensions of the matrix are $2c \times 3N$ in two dimensions and $3c \times 6N$ in three dimensions.

In two dimensions, we define the diagonal mass matrix M as

$$M = \begin{pmatrix} \xi_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \xi_N \end{pmatrix} \quad \text{with} \quad \xi_i = \begin{pmatrix} m_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & I_i \end{pmatrix}. \quad (2.204)$$

With the prior definitions we obtain

$$M\ddot{q}(t) = R(t) + F^{\text{ext}}. \quad (2.205)$$

Based on the relation

$$\dot{\mathbf{u}} = H^T \ddot{\mathbf{q}}, \quad (2.206)$$

we find

$$\begin{aligned} \dot{\mathbf{u}} &= H^T \left(M^{-1}R(t) + M^{-1}F^{\text{ext}} \right) \\ &= H^T M^{-1}R(t) + H^T M^{-1}F^{\text{ext}}. \end{aligned} \quad (2.207)$$

Using contact variables yields

$$\dot{u} = H^T M^{-1}HR^{\text{loc}} + H^T M^{-1}F^{\text{ext}} \quad (2.208)$$

and we define the effective inverse mass matrix

$$M_{\text{eff}}^{-1} = H^T M^{-1}H. \quad (2.209)$$

We solve the resulting system of equations with an implicit Euler scheme as in the case of the one-dimensional contact according to

$$R^{\text{loc}}(t + \Delta t) = M_{\text{eff}} \frac{u(t + \Delta t) - u^{\text{free}}(t + \Delta t)}{\Delta t}. \quad (2.210)$$

The dimensions of the vectors change over time since the number of particles in contact is not constant. To solve Eq. (2.210), we perform a direct inversion of $M_{\text{eff}}^{-1} = H^T M^{-1} H$. To determine $u(t + \Delta t)$, the constraints are added by checking if the contacts close.

Friction, rolling friction, and cohesion can be simulated by modifying the constraint forces. Up to now, we only spherical particles. As described in Sec. 2.8.3, non-spherical objects can be composed of spherical objects, e.g., by additional constraints between particles. For non spherical rigid objects of finite volume (e.g., polyhedra) this can be done approximately.

2.10 Particles in Fluids

In our event-driven and contact dynamics models of particle interactions, we considered friction and energy dissipation effects in terms of the interaction of rigid bodies. However, inelastic collisions are also relevant if we think of fluids surrounding particles. Simulations of particle dynamics in fluids is highly relevant for optimizing certain structures in the sense of minimizing friction and turbulence effects.

We therefore consider an incompressible fluid of density ρ and dynamic viscosity μ . It is described by the *incompressible Navier-Stokes equations*

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}(\nabla \mathbf{u}) = -\frac{1}{\rho} \nabla p + \mu \Delta \mathbf{u}. \quad (2.211)$$

The velocity and pressure fields are denoted by $\mathbf{u}(\mathbf{x})$ and $p(\mathbf{x})$, respectively. Since the density ρ is constant, the continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \mathbf{u}) = 0 \quad (2.212)$$

leads to $\nabla \mathbf{u} = 0$.

We classify the fluid flow according to the Reynold's number

$$\text{Re} = \frac{uh}{\mu} = \begin{cases} \ll 1 \text{ Stokes limit,} \\ \gg 1 \text{ Turbulent limit,} \end{cases} \quad (2.213)$$

where u and h represent a characteristic velocity and length scale, respectively.

There are two possibilities of modeling particle-fluid interactions. First, in a *continuum approach* the fluid is described by differential equations such as Eqs. (2.211) and (2.212). Second, it is possible to use *particle-based* models of fluids. Different methods are applicable to solve such problems. Some examples include

- Penalty method with MAC,
- Finite volume method (FLUENT),

- $k - \epsilon$ model or spectral methods for the turbulent case,
- Lattice-Boltzmann method,
- Discrete methods.

In this section we briefly describe some continuum approaches, and give an overview on particle-based methods.

2.10.1 Continuum Methods

Fluid-particle interactions occur on the exposed surface. To describe particle dynamics in a fluid, we have to solve the Navier-Stokes equations by considering the boundaries defined by the moving particles in the fluid. Consequently, the boundaries of the fluid are also moving. Thus, the motion of the fluid has to be determined for each time step.

Based on the fluid motion, we are able to extract the forces exerted on the particles which enables us to solve their equations of motion. The total drag force is obtained by integrating the stress tensor Θ of the fluid over the particles' surfaces

$$\mathbf{F}_D = \int_{\Gamma} \Theta d\mathbf{A}, \quad (2.214)$$

with

$$\Theta_{ij} = -p\delta_{ij} + \eta \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad (2.215)$$

where $\eta = \rho\mu$ is the static viscosity and p the hydrostatic pressure. In the Stokes limit for $\text{Re} \ll 1$, the drag law is given by

$$F_D = 6\pi\eta R u, \quad (2.216)$$

where R is the radius of the particle moving in the fluid. The Stokes law is exact for $\text{Re} = 0$. In the case of turbulent flow for $\text{Re} \gg 1$, the drag force is (Newton's law)

$$F_D = 0.22\pi\rho R^2 u^2. \quad (2.217)$$

The general drag law is

$$F_D = \frac{\pi\eta^2}{8\rho} C_D \text{Re}^2, \quad (2.218)$$

where C_D denotes the drag coefficient. It depends on the velocity of the particle in the fluid, and on the density and the viscosity of the fluid. These laws are based on the assumption of spherical particles and other simplifications, and we may encounter substantial deviations in experiments. In certain cases, it is important to also consider

the influence of pressure or velocity gradients which lead to lift forces

$$F_L = \frac{1}{2} C_L \rho A u^2, \quad (2.219)$$

where C_L is the lift coefficient. In addition to drag and lift forces, rotating particles experience a torque

$$\mathbf{T} = \int_{\Gamma} \mathbf{r}_{\text{cm}} \wedge \Theta d\mathbf{A}. \quad (2.220)$$

For cylinder of radius R and angular velocity ω , the corresponding Magnus force is

$$F_M = 2\pi R^2 \rho u \omega. \quad (2.221)$$

There exist empirical relation for drag coefficient in certain Reynold's numbers regimes. For example, Dimitri Gidaspow suggested the following drag coefficient dependence:

$$C_D = \begin{cases} \frac{24}{\text{Re}_\rho} \left(1 + 0.15 \text{Re}_\rho^{0.687}\right) & \text{Re}_\rho < 1000, \\ 0.44 & \text{Re}_\rho \geq 1000, \end{cases} \quad (2.222)$$

where $\text{Re}_\rho = \frac{\rho_f |\mathbf{v} - \mathbf{u}| D_s}{\nu}$. Here D_s is the particle diameter and $|\mathbf{v} - \mathbf{u}|$ is the absolute value of the particle velocities compared to the fluid.

2.10.2 Stokesian Dynamics

Brady and Bossis introduced a method to study Stokesian dynamics ($\text{Re} \ll 1$) [40]. The Stokes equation is

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho} \nabla p + \mu \Delta \mathbf{u}. \quad (2.223)$$

The Green's function of the Stokes equation is the Stokeslet

$$G_{\alpha\beta}^S(\mathbf{r}) = \frac{1}{8\pi\eta} \left(\frac{\delta_{\alpha\beta}}{r} + \frac{r_\alpha r_\beta}{r^3} \right). \quad (2.224)$$

A general solution for the velocities fields of N particles is then

$$\mathbf{u}(\mathbf{x}) = -\sum_{i=1}^N \int_{\Gamma_i} G^S \Theta \mathbf{n} d\Gamma_i. \quad (2.225)$$

The drag force on a surface element ijk is determined according to

$$\mathbf{f}_{ijk} = \Theta^{ijk} \mathbf{n}. \quad (2.226)$$

2.10.3 Lattice Boltzmann Method

Based on the Chapman–Enskog theory, it is possible to derive the Navier–Stokes equations from the Boltzmann equation. This connection between fluid dynamics and Boltzmann transport theory allows us to simulate the motion of fluids by solving the corresponding Boltzmann equation on a lattice. The basic idea is that we define on each site x of a lattice on each outgoing bond i a velocity distribution function $f(x, v_i, t)$ whose updates are given by

$$f(x + v_i, v_i, t + 1) - f(x, v_i, t) + F(v_i) = \frac{1}{\tau} [f^{\text{eq}} - f(x, v_i, t)], \quad (2.227)$$

the equilibrium distribution is

$$f_i^{\text{eq}} = n w_i \left[1 + \frac{3}{c^2} \mathbf{u} \mathbf{v}_i + \frac{9}{2c^4} (\mathbf{u} \mathbf{v}_i)^2 - \frac{3}{2c^2} \mathbf{u} \mathbf{u} \right]. \quad (2.228)$$

One possible choice of the weights in two dimensions is

$$w_i = \begin{cases} 4/9 & i = 0, \\ 1/9 & i = 1, 2, 3, 4, \\ 1/36 & i = 5, 6, 7, 8. \end{cases} \quad (2.229)$$

The positions of the indices are illustrated in Fig. 2.30.

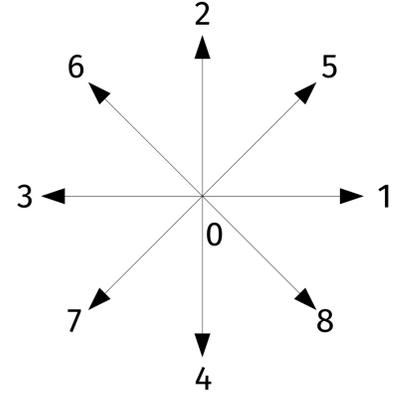


Figure 2.30: Lattice Boltzmann weights in 2D (D2Q9).

2.10.4 Stochastic Rotation Dynamics

Stochastic Rotation Dynamics (SRD) is a particle-based fluid modeling approach [41]. This technique is also known as Multi-particle Collision Dynamics (MPC). In this method, we discretize the space into cells and model the fluid as a system composed of N particles with mass m and coordinates \mathbf{x}_i and \mathbf{v}_i . The particle positions and velocities are updated according to

$$\begin{aligned} \mathbf{x}'_i &= \mathbf{x}_i + \Delta t \mathbf{v}_i, \\ \mathbf{v}'_i &= \mathbf{u} + \Omega (\mathbf{v}_i - \mathbf{u}) + \mathbf{g}, \end{aligned} \quad (2.230)$$

where $\mathbf{u} = \langle \mathbf{v} \rangle$ is the mean velocity of particles in the respective cell and Ω is the rotation matrix. It is given by

$$\Omega = \begin{pmatrix} \cos(\alpha) & \pm \sin(\alpha) & 0 \\ \mp \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (2.231)$$

The collective fluid particle interaction is modeled by rotations of local particle velocities. In this model, Brownian motion is intrinsic. These very simple dynamics recovers hydrodynamics correctly.

2.10.5 Direct Simulation Monte Carlo

Direct Simulation Monte Carlo (DSMC) is a particle-based simulation technique which is appropriate to model particle systems at large Knudsen numbers [42, 43]

$$\text{Kn} = \frac{\lambda}{L}, \quad (2.232)$$

where λ is the mean free path and L a characteristic system length scale. It is very popular in aerospace modeling, because the atmosphere is very thinned out at high altitudes and the corresponding Knudsen numbers are large.

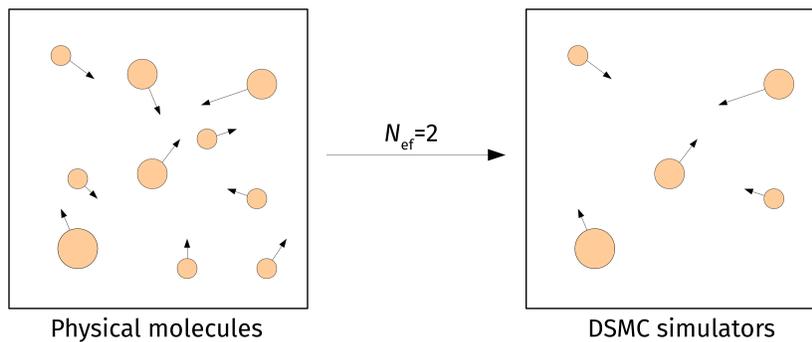


Figure 2.31: In DSMC N_{ef} simulators represent one physical particle.

In this method, we divide the system into cells and generate particles in each cell according to desired density, fluid velocity, and temperature. For fluid velocities and temperatures, we assign a velocity to each particle which is distributed according to the Maxwell-Boltzmann distribution. In DSMC the number of simulation particles (*simulators*) is typically a small fraction of the number of physical molecules. Each simulator represents N_{ef} physical molecules. This is shown in Fig. 2.31. The accuracy of DSMC scales as $1/N$; for traditional DSMC about 20 particles per collision cell is the rule-of-thumb.

Collision are modeled by sorting particles into spatial collision cells. We then iterate over all cells and

1. compute the collision frequency in each cell,
2. randomly select collision partners within cell,
3. process each collision.

We note that collision pairs with large relative velocity are more likely to collide but they do not have to be on a collision trajectory.

The material surface may be treated with a thermal wall, which

resets the velocity of a particle as a biased-Maxwellian distribution

$$\begin{aligned}
 P_{v_x}(v_x) &= \pm \frac{m}{k_B T_W} v_x e^{-\frac{mv_x^2}{2k_B T_W}}, \\
 P_{v_y}(v_y) &= \sqrt{\frac{m}{2\pi k_B T_W}} e^{-\frac{m(v_y - u_W)^2}{2k_B T_W}}, \\
 P_{v_z}(v_z) &= \sqrt{\frac{m}{2\pi k_B T_W}} e^{-\frac{mv_z^2}{2k_B T_W}}.
 \end{aligned} \tag{2.233}$$

2.10.6 Dissipative Particle Dynamics

Another particle-based fluid simulation approach is the so-called *Dissipative Particle Dynamics* (DPD) [44]. The particle interactions are described by

$$\mathbf{F}_i = \sum_{i \neq j} \left(\mathbf{f}_{ij}^C + \mathbf{f}_{ij}^R + \mathbf{f}_{ij}^D \right), \tag{2.234}$$

where \mathbf{f}_{ij}^C denotes the conservative forces, \mathbf{f}_{ij}^R the random force and \mathbf{f}_{ij}^D the dissipative forces. The dissipative forces are proportional to the particle velocities. The weights of the random and dissipative forces must be chosen such that thermal equilibrium is reached [45].

2.10.7 Smoothed Particle Hydrodynamics

Another important technique in the field of computational fluid dynamics is *Smoothed Particle Hydrodynamics* (SPH). This method uses *smooth kernel functions* W to represent properties of particles in a weighted sense [46, 47]. Instead of localized positions and velocities, the particle characteristics are smoothed over a smoothing length h . An arbitrary quantity A is then given by

$$A(r) = \int_{\Omega} W(|r - r'|, h) A(r') dr' \approx \sum_j \frac{m_j}{\rho_j} W(|r - r_j|, h) A_j. \tag{2.235}$$

In this method, no spatial discretization is necessary and even complex geometries can be interpolated and simulated with SPH. This makes this method broadly applicable in many different fields where fluids interact with complex structures.

Example of a kernel functions include Gaussians or quadratic functions

$$W(r, h) = \frac{3}{2\pi h^2} \left(\frac{1}{4} q^2 - q + 1 \right) \tag{2.236}$$

with $q = \frac{r}{h}$ and $r = |r_a - r_b|$. Another advantage of this method is that kernels may be changed without much effort for a given simulation framework [48].

2.11 Quantum Mechanical Approaches

Simulating the interaction of macroscopic objects such as billiard balls, grains and rigid bodies is based on solving Newton's equations of motion. To a certain extent, it is also possible to consider classical approaches for the simulation of molecules if the bond length can be assumed to be constant given certain environmental conditions, cf. Sec. 2.3.1. However, for the ab-initio (i.e., from first principles) simulation of molecular and atomic interactions we have to take quantum mechanical effects into account. In the subsequent sections, we therefore briefly discuss the main quantum-mechanical methods which are necessary for ab-initio MD simulations. In particular, we introduce the concept of wave functions, the Born-Oppenheimer and the Kohn-Sham approximations. We conclude the section with introducing the Car-Parrinello method.

2.11.1 Introduction

In quantum mechanics the wave function $\psi(\mathbf{r}, t)$ at position \mathbf{r} is represented by a vector $|\psi\rangle$ of the Hilbert space $L^2(\mathbb{R}^3)$. The corresponding scalar product is

$$\langle\psi|\phi\rangle = \int_{\mathbb{R}^3} \psi^*(\mathbf{r}) \phi(\mathbf{r}) \, d\mathbf{r}. \quad (2.237)$$

In general, a quantum state is a vector $|\psi\rangle$ of a Hilbert space over \mathbb{C} with scalar product $\langle\psi|\phi\rangle$. All states fulfill the normalization condition

$$\langle\psi|\psi\rangle = \int_{\mathbb{R}^3} \psi^*(\mathbf{r}) \psi(\mathbf{r}) \, d\mathbf{r} = 1 \quad (2.238)$$

and are uniquely defined up to a phase $e^{i\alpha}$ with $\alpha \in \mathbb{R}$, i.e.,

$$|\psi\rangle \propto e^{i\alpha} |\psi\rangle. \quad (2.239)$$

According to the statistical interpretation of the wave function (Born rule), the probability density $|\psi(\mathbf{r})|^2$ denotes the probability of finding a particle described by ψ at position \mathbf{r} . Observables are represented by self-adjoint operators A with the expectation value $\langle\psi|A|\psi\rangle$.

Given a *Hamilton operator*

$$H = \frac{p^2}{2m} + V(\mathbf{r}) \quad \text{with} \quad p = -i\hbar\nabla, \quad (2.240)$$

the time evolution of the function $t \rightarrow |\psi_t\rangle$ is described by the time-dependent Schrödinger equation

$$i\hbar \frac{d}{dt} |\psi_t\rangle = H |\psi_t\rangle. \quad (2.241)$$

Applying the method of separation of variables with $\psi_t(\mathbf{r}) = \psi(\mathbf{r})\phi(t)$ reveals that the time evolution is given by

$$\phi(t) = e^{-iEt/\hbar} \quad (2.242)$$

with E being the expectation value of the Hamilton operator. The time-independent Schrödinger equation is thus

$$-\frac{\hbar^2}{2m}\nabla^2|\psi\rangle + V(\mathbf{r})|\psi\rangle = E|\psi\rangle. \quad (2.243)$$

For the simulations of quantum-mechanical particle interactions, we have to consider the many-body wave function of N particles

$$\psi(\mathbf{r}_1, s_1, \mathbf{r}_2, s_2, \dots, \mathbf{r}_N, s_N), \quad (2.244)$$

where $\{s_i\}_{i \in \{1, 2, \dots, N\}}$. Similarly, many-body operators A are defined by considering their action on all individual particles.

In the following, we use $\psi(1, 2, \dots, N)$ to abbreviate the wave function of Eq. (2.244) and $A(1, 2, \dots, N)$ as an abbreviation of a corresponding operator. We now define the exchange operator P_{ij} ($1 \leq i, j \leq N$) according to

$$\begin{aligned} P_{ij}\psi(1, 2, \dots, i, \dots, j, \dots, N) &= \psi(1, 2, \dots, j, \dots, i, \dots, N), \\ P_{ij}A(1, 2, \dots, i, \dots, j, \dots, N) &= A(1, 2, \dots, j, \dots, i, \dots, N). \end{aligned} \quad (2.245)$$

The Hamiltonian commutes with the exchange operation since it is invariant under particle exchange, i.e.,

$$[H, P_{ij}] = 0. \quad (2.246)$$

We note that $(P_{ij})^{-1} = P_{ij}$ and $(P_{ij})^2 = 1$. The two possible eigenvalues are thus $+1$ and -1 which are both realized in nature with the corresponding wave functions

$$\psi(1, 2, \dots, i, \dots, j, \dots, N) = \begin{cases} +\psi(1, 2, \dots, j, \dots, i, \dots, N) & \text{Bosons,} \\ -\psi(1, 2, \dots, j, \dots, i, \dots, N) & \text{Fermions.} \end{cases} \quad (2.247)$$

The wave functions of bosons are completely symmetric under particle exchange whereas fermions exhibit completely antisymmetric wave functions.

In the case of fermions, this implies that two particles cannot be in the same state due to

$$\psi(1, 2, \dots, i, \dots, i, \dots, N) = -\psi(1, 2, \dots, i, \dots, i, \dots, N) = 0. \quad (2.248)$$

This corresponds to the Pauli exclusion principle. A quantum-mechanical description of N particles makes it necessary to construct many-body wave functions by considering all possible particle permutations since all particles (unlike classical ones) are indistinguishable. We consider N particles which are not interacting with each other. The corresponding Hamilton operator is

$$H = \sum_{i=1}^N H_i \quad \text{with} \quad H_i = \frac{p_i^2}{2m} + V(\mathbf{r}_i). \quad (2.249)$$

The states of each particle ψ_ν satisfy

$$H_i \psi_\nu(\mathbf{r}_i, s_i) = \epsilon_\nu \psi_\nu(\mathbf{r}_i, s_i). \quad (2.250)$$

For bosons, the many-body wave function is

$$\begin{aligned} \langle 1, 2, \dots, N | \psi_B \rangle &= \psi_B(1, 2, \dots, N) \\ &= \sum_{P \in S_N} P \psi_{\nu_1}(\mathbf{r}_1, s_1) \psi_{\nu_2}(\mathbf{r}_2, s_2) \cdots \psi_{\nu_N}(\mathbf{r}_N, s_N) \end{aligned} \quad (2.251)$$

and for fermions we find

$$\begin{aligned} \langle 1, 2, \dots, N | \psi_F \rangle &= \psi_F(1, 2, \dots, N) \\ &= \sum_{P \in S_N} \text{sgn}(P) P \psi_{\nu_1}(\mathbf{r}_1, s_1) \psi_{\nu_2}(\mathbf{r}_2, s_2) \cdots \psi_{\nu_N}(\mathbf{r}_N, s_N), \end{aligned} \quad (2.252)$$

where P is the exchange operator for the states ν_i and $\text{sgn}(P)$ is the sign of the permutation P which is $+1$ (-1) if P is composed of an even (odd) number of transpositions.

For fermions, it is possible to rewrite Eq. (2.252) in terms of the so-called *Slater determinant*

$$\psi_F(1, \dots, N) = \left| \begin{pmatrix} \psi_{\nu_1}(1) & \cdots & \psi_{\nu_N}(1) \\ \vdots & \ddots & \vdots \\ \psi_{\nu_1}(N) & \cdots & \psi_{\nu_N}(N) \end{pmatrix} \right|. \quad (2.253)$$

We still have to normalize the many-body wave functions according to

$$\langle \psi_B | \psi_B \rangle = N! n_{\nu_1}! n_{\nu_2}! \cdots n_{\nu_N}! \quad \text{and} \quad \langle \psi_F | \psi_F \rangle = N!. \quad (2.254)$$

In the case of Bosons, the index ν may appear multiple times in Eq. (2.251). Therefore, we have to account for the additional normalization factors n_{ν_1} which denote the number of particles in the stationary single particle state ν_1 . For fermions, the occupation numbers n_ν are either zero or one. The calculation of Eq. (2.253) is computationally very expensive and time consuming.

2.11.2 Implementation of Wave Functions

For the implementation of wave functions, the single particle wave functions are often expanded in an orthonormal basis system with some cutoff in the following sense

$$\psi_j = \sum_k c_{jk} \chi_k. \quad (2.255)$$

In solid state physics a plane wave basis $\chi_k = \exp(ikx)$ is often used (plane wave functions are not localized, as the electrons in crystals).

We typically use as many plane waves as necessary until the energy converges. For 16 water molecules, a cutoff of about 70 Ry is needed. This corresponds in this case to about 15000 plain waves per electron.

Using localized basis sets can help to decrease the number of basis functions for chemical problems. Wave functions are then dependent on the ion positions. One possible choice are Gaussian-type orbitals (GTOs)

$$\chi_l(r) = c_l r^l \exp(-\alpha r^2). \quad (2.256)$$

For localized basis sets so called *Pulay forces* appear. These forces are due to numerical artifacts originating from a finite basis. They may be up to an order of magnitude bigger than the physical forces and have to be corrected.

2.11.3 Born-Oppenheimer Approximation

For the simulation of atomic and molecular interactions, we have to describe the dynamics of the individual constituents. Instead of a full quantum-mechanical description of both the nuclei and the electrons, we make use of the fact that their masses differ by three orders of magnitude. In this way, we determine the electron distribution for fixed ion positions. This is the so-called *Born-Oppenheimer approximation*.

The Born-Oppenheimer Approximation implies that the time step for the motion of ions has to be small enough, such that when ions move, the electrons do not skip any energy level transition from energy ϵ_i to ϵ_{i+1} according to

$$|\epsilon_i(\mathbf{r} + \Delta\mathbf{r}) - \epsilon_i(\mathbf{r})| \ll |\epsilon_{i+1}(\mathbf{r}) - \epsilon_i(\mathbf{r})|, \quad (2.257)$$

where \mathbf{r} is the ion position before the update and $\Delta\mathbf{r} = \mathbf{u}\Delta t$ with \mathbf{u} being the ion velocity and Δt the corresponding time step.

2.11.4 Hohenberg-Kohn Theorems

If the ground state is not degenerated, all information on this ground state is contained in its density distribution $n(\mathbf{r})$. It is possible to obtain all observables based on $n(\mathbf{r})$. The knowledge of the wave

functions is not needed. To formalize this approach, we consider N electrons described by a wave function Ψ with normalization $\langle \Psi | \Psi \rangle = N$. These electrons move around a nucleus with potential

$$V(\mathbf{r}) = - \sum_i \frac{Ze^2}{|\mathbf{r} - \mathbf{r}_i|}, \quad (2.258)$$

where Z is the nuclear charge number and e the elementary charge.

The corresponding Hamiltonian operator is given by

$$H = \sum_i \frac{p_i^2}{2m} + \frac{1}{2} \sum_i \sum_{j \neq i} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_i V(\mathbf{r}_i). \quad (2.259)$$

We abbreviate the Hamiltonian with $H = F + V$ where F represents the kinetic and electron-interaction energy contributions. All electrons experience the same contribution of F . The ground state $|\Psi_0\rangle$ is thus completely determined by N and $V(\mathbf{r})$.

We define the ground state particle density as

$$n_0(\mathbf{r}) = \langle \Psi_0 | n(\mathbf{r}) | \Psi_0 \rangle = N \int |\Psi_0(\mathbf{r}, \mathbf{r}_2, \dots, \mathbf{r}_N)|^2 d\mathbf{r}_2 d\mathbf{r}_3 \cdots d\mathbf{r}_N, \quad (2.260)$$

where $n(\mathbf{r}) = \sum_i \delta(\mathbf{r} - \mathbf{r}_i)$ is the particle density operator. Hohenberg and Kohn (1964) have shown that (i) the external potential V_{ext} is uniquely determined by the ground-state electron density, and that (ii) the ground state energy can be obtained by minimizing an energy functional [49]. It is thus not necessary anymore to consider the wave functions for non-degenerate ground states. Instead we use a density functional approach.

Specifically, we proof claim (i) by contradiction and assume that there exist two external potentials $V(\mathbf{r})$ and $V'(\mathbf{r})$ which give rise to the same particle density $n(\mathbf{r})$. The corresponding Hamiltonians are $H = F + V$ and $H' = F + V'$ and let

$$E_0 = \langle \Psi_0 | H | \Psi_0 \rangle \quad \text{and} \quad E'_0 = \langle \Psi'_0 | H' | \Psi'_0 \rangle. \quad (2.261)$$

With these definitions, we obtain

$$\begin{aligned} E_0 < \langle \Psi'_0 | H | \Psi'_0 \rangle &= \langle \Psi'_0 | H' | \Psi'_0 \rangle + \langle \Psi'_0 | (H - H') | \Psi'_0 \rangle \\ &= E'_0 + \int n_0(\mathbf{r}) [V(\mathbf{r}) - V'(\mathbf{r})] d\mathbf{r} \end{aligned} \quad (2.262)$$

and

$$\begin{aligned} E'_0 < \langle \Psi_0 | H' | \Psi_0 \rangle &= \langle \Psi_0 | H | \Psi_0 \rangle + \langle \Psi_0 | (H' - H) | \Psi_0 \rangle \\ &= E_0 + \int n_0(\mathbf{r}) [V'(\mathbf{r}) - V(\mathbf{r})] d\mathbf{r}. \end{aligned} \quad (2.263)$$

Adding Eqs. (2.262) and (2.263) leads to the contradiction

$$E_0 + E'_0 < E_0 + E'_0. \quad (2.264)$$

Thus the potential V is (up to a constant) uniquely determined by $n(\mathbf{r})$. We are now able to recast the problem of solving the Schrödinger equation for our N particle system in variational form by defining the energy functional

$$E_V[n] = \int n(\mathbf{r}) V(\mathbf{r}) d\mathbf{r} + F[n]. \quad (2.265)$$

The particle density $n(\mathbf{r})$ determines the potential $V(\mathbf{r})$ and the ground-state $|\Psi\rangle$. For another potential $\tilde{V}(\mathbf{r})$, we find

$$\begin{aligned} \langle \Psi | \tilde{H} | \Psi \rangle &= \langle \Psi | F | \Psi \rangle + \langle \Psi | \tilde{V} | \Psi \rangle \\ &= F[n] + \int n(\mathbf{r}) \tilde{V}(\mathbf{r}) d\mathbf{r} = E_{\tilde{V}}[n] \geq E_0 \end{aligned} \quad (2.266)$$

according to the variational principle. The functional $E_{\tilde{V}}[n]$ equals E_0 if $|\Psi\rangle$ is the ground-state for $\tilde{V}(\mathbf{r})$.

2.11.5 Kohn-Sham Approximation

Based on the Hohenberg-Kohn theorems, we are able to describe a non-degenerate ground-state with density functionals. This was the beginning of density functional theory (DFT). Earlier works by Thomas and Fermi were too inaccurate in most applications. Further developments by Kohn and Sham made it possible to approximate the electron-electron interactions in terms of non-interacting single particles moving in a potential which also only depends on the density distribution. Additional correction terms account for many-particle effects.

In the Kohn-Sham approximation, electrons are treated as single particles interacting with the effective Kohn-Sham potential V_{KS} . The corresponding wave functions $\psi_i(\mathbf{r})$ with $i \in \{1, \dots, N\}$ and energies ϵ_i obey

$$\left[-\frac{\hbar^2}{2m} \nabla^2 + V_{\text{KS}}(\mathbf{r}) \right] \psi_i(\mathbf{r}) = \epsilon_i \psi_i(\mathbf{r}). \quad (2.267)$$

The Kohn-Sham potential is given by

$$V_{\text{KS}}(\mathbf{r}) = V(\mathbf{r}) + e^2 \int \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' + V_{\text{xc}}(\mathbf{r}), \quad (2.268)$$

where $V(\mathbf{r})$ is an external potential (e.g., nucleus), $V_{\text{xc}}(\mathbf{r}) = \delta E_{\text{xc}}[n] / \delta n(\mathbf{r})$ a potential term accounting for exchange and correlation effects, and $E_{\text{xc}}(\mathbf{r})$ the exchange-correlation energy.

We start with an initial guess for $n(\mathbf{r})$ and then solve Eq. (2.267) to obtain the wave functions $\psi_i(\mathbf{r})$. This allows us to determine the

particle density

$$n(\mathbf{r}) = \sum_{i=1}^N |\psi_i(\mathbf{r})|^2 \quad (2.269)$$

and the kinetic energy

$$T_s[n] = - \sum_{i=1}^N \int \psi_i^*(\mathbf{r}) \left(-\frac{\hbar^2}{2m} \nabla^2 \right) \psi_i(\mathbf{r}) \, d\mathbf{r}. \quad (2.270)$$

The orbital energies ϵ_i have to sum up to the total energy

$$\sum_{i=1}^N \epsilon_i = T_s[n] + \int n(\mathbf{r}) V_{\text{KS}}(\mathbf{r}) \, d\mathbf{r}. \quad (2.271)$$

Unlike for real many-particle orbitals the eigenvalues of the single particle orbitals have no physical meaning. The sum of them is not the total energy. The only physical meaning of the single particle wave functions is that they yield the correct density.

Until here, density functional theory is describing a many-body quantum system in an exact way. However, for the computation of the exchange-correlation term $E_{\text{xc}}[n] = E_x[n] + E_c[n]$ approximations have to be made. Under the assumption of a homogeneous electron gas the exchange energy in the local-density approximation (LDA) is

$$E_x^{\text{LDA}}[n] = -\frac{3}{4} \left(\frac{3}{\pi} \right)^{1/3} \int n(\mathbf{r})^{4/3} \, d\mathbf{r}. \quad (2.272)$$

In the case of the correlation energy

$$E_c^{\text{LDA}}[n] = \int \epsilon_c[n(\mathbf{r})] \, d\mathbf{r} \quad (2.273)$$

no analytical expression is known for $\epsilon_c[n(\mathbf{r})]$. LDA has been applied for calculations of band structures and the calculation of the total energy in solid-state physics. In quantum chemistry it has been much less popular since chemical bonds need more accurate results.

It is possible to improve the LDA by adding a dependence on the gradient of the density. This approach is called General Gradient Approximation (GGA). The physical intuition for this correction is that quantum mechanical effects are very strong when there is a change in the slope of the wave function and in particular when two identical fermions come closer. Due to the Pauli exclusion principle, there are no classical effects which are stronger if the particles are close. The GGA approximation of the exchange energy is

$$E_x^{\text{GGA}}[n] = \int \epsilon_x[n(\mathbf{r}), |\nabla n(\mathbf{r})|] \, d\mathbf{r}. \quad (2.274)$$

One possibility of a GGA approximation of the exchange energy is

$$E_x^{\text{GGA}}[n] = E_x^{\text{LDA}}[n] - \beta \int \frac{(\nabla n)^2}{n^{4/3}} \mathbf{d}\mathbf{r} \quad (2.275)$$

where β is a parameter. In addition, Axel Becke suggested another approach [50]

$$E_x^{\text{Becke}}[n] = E_x^{\text{LDA}}[n] - \beta \int n^{4/3} \frac{x^2}{1 + 6\beta x \sinh^{-1}(x)} \mathbf{d}\mathbf{r} \quad (2.276)$$

with $x = |\nabla n| / n^{4/3}$ and $\beta = 0.0042$.

The corresponding paper is one of the most cited works in modern science. It shows that DFT is a very important topic in different subject and with many possible applications. The approximation by Axel Becke is empirically obtained and not derived analytically. GGA gives much better results than LDA. They can be used to calculate all kinds of chemical bonds like covalent, ionic, metallic, and hydrogen bridges. However DFT using GGA fails to describe van-der-Waals interactions properly. A possible improvement has been suggested by Grimme [51]. This leads to an additional energy term

$$E_{\text{disp}} = -s_6 \sum_{i=1}^{N-1} \sum_{j=1}^N \frac{C_6^{ij}}{R_{ij}} f_{\text{dmp}}(R_{ij}) \quad \text{with} \quad f_{\text{dmp}}(R_{ij}) = \frac{1}{1 + e^{-d(R_{ij}/R_r - 1)}} \quad (2.277)$$

where s_6 , C_6^{ij} , R_r and d are parameters and R_{ij} is the distance between two ions.

2.11.6 Hellmann-Feynman Theorem

With the previous methods we are able to describe the motion of electrons in the potential of the (slower) moving nucleus. After a sufficiently small time step, we also have to update its position. Given a Hamiltonian operator H of the system, the forces acting on the ions are (*Hellmann-Feynman Theorem*)

$$m_\alpha \frac{d^2 \mathbf{R}_\alpha}{dt^2} = - \left\langle \psi \left| \frac{\partial H}{\partial \mathbf{R}_\alpha} \right| \psi \right\rangle, \quad (2.278)$$

where m_α and \mathbf{R}_α are the mass and position of ion α .

Combining DFT approaches with the Hellmann-Feynman theorem yields an iterative method for solving the motion of molecules and other quantum-mechanical systems according to the following steps:

- Solve the electron dynamics in the ion potential.
- Solve the motion of the ions using classical MD techniques in combination with the Hellmann-Feynman theorem.

The problem of this method is that the first step is computationally very expensive. For this reason, the contribution of Car and Parrinello has been very important.

2.11.7 Car-Parrinello Method

Car and Parrinello reformulated the ab-initio MD problem in terms of the Hamiltonian [52]

$$\begin{aligned}
 H_{CP} [\{\mathbf{R}_I\}, \{\dot{\mathbf{R}}_I\}, \{\psi_i\}, \{\dot{\psi}_i\}] &= \sum_I \frac{M_I}{2} \dot{\mathbf{R}}_I^2 + \sum_i \frac{\mu}{2} \langle \dot{\psi}_i | \dot{\psi}_i \rangle \\
 &+ E_{KS}(\{\mathbf{R}_I\}, \{\psi_i\}) + E_{\text{ion}}(\{\mathbf{R}_I\}),
 \end{aligned}
 \tag{2.279}$$

where M_I , R_I , ψ are the masses and positions of the nuclei, and the orbital wave functions, respectively. The parameter μ is a fictitious mass which is used to limit the energy transfer from nuclei to electrons. The remaining terms are the Kohn-Sham energy E_{KS} and the ion-ion interaction energy E_{ion} . The Kohn-Sham energy is the sum of the kinetic energy of Eq. (2.270) and the potential energy of Eq. (2.268).

The corresponding equations of motion are

$$\begin{aligned}
 M_I \ddot{\mathbf{R}}_I &= -\nabla_{\mathbf{R}_I} E_{KS} + \sum_{ij} \Lambda_{ij} \nabla_{\mathbf{R}_I} \langle \psi_i | \psi_j \rangle, \\
 \mu \ddot{\psi}_i(\mathbf{r}, t) &= -\frac{\delta E_{KS}}{\delta \psi_i^*(\mathbf{r}, t)} - \frac{\delta E_{\text{ion}}}{\delta \psi_i^*(\mathbf{r}, t)} + \sum_j \Lambda_{ij} \psi_j(\mathbf{r}, t).
 \end{aligned}
 \tag{2.280}$$

The additional Lagrange multiplier Λ_{ij} is used to ensure orthonormality of wave functions.

Similar to the Nosé-Hoover approach, we regard ions and electrons as two different systems coupled by the Hamiltonian as defined in Eq. (2.279). The coupling is realized with a fictitious mass μ which should not be confused with the electron mass. The fictitious mass is a tunable parameter of the method describing how stiff the electron motion is coupled to the nuclei. If $\mu \rightarrow 0$, the electronic response is very rapid and the electrons remain to a sufficiently high degree in the ground state. But numerically we want to avoid too large accelerations. So electron dynamics is made artificially slower than reality. Since the electron configuration is now following an equation of motion, the integration step has to be small enough to resolve the electronic motion. Since μ is usually smaller than the mass of the ions, the CP time steps are usually smaller than the time steps of the direct Born-Oppenheimer approach. Often time steps of the order of a tenth of a femtosecond are used.

Car-Parrinello equations need much less computer time per time step. But they need smaller time steps. There is still speed gain in real

time. In addition, the energy-fluctuations are smaller since we always consider the same bindings. However, it is difficult to simulate light ions and large values of μ .

References

- [1] Allen, M. & Tildesley, D. *Computer Simulation of Liquids* (Clarendon Press, Oxford, 1987).
- [2] Newton, I. *Philosophiae naturalis principia mathematica*, vol. 1 (G. Brookman, 1833).
- [3] Alder, B. & Wainwright, T. Phase transition for a hard sphere system. *J. Chem. Phys.* **27**, 1208–1209 (1957).
- [4] Vink, R., Barkema, G., Van der Weg, W. & Mousseau, N. Fitting the stillinger–weber potential to amorphous silicon. *Journal of non-crystalline solids* **282**, 248–255 (2001).
- [5] Axilrod, B. & Teller, E. Interaction of the van der waals type between three atoms. *J. Chem. Phys.* **11**, 299–300 (1943).
- [6] Verlet, L. Computer experiments on classical fluids. *Phys. Rev.* **159** (1967).
- [7] Knuth, D. *The Art of Computer Programming* (Addison-Wesley, 1968).
- [8] Ryckaert, J., Ciccotti, G. & Berendsen, H. Numerical integration of the cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes. *J. Comp. Phys.* **239**, 327–341 (1977).
- [9] Evans, D. On the representatation of orientation space. *Mol. Phys.* **34**, 317–325 (1977).
- [10] Evans, D. Singularity free algorithm for molecular dynamics simulation of rigid polyatomics. *Mol. Phys.* **34**, 327–331 (1977).
- [11] Stäger, D. V., Araújo, N. & Herrmann, H. J. Prediction and control of slip-free rotation states in sphere assemblies. *Phys. Rev. Lett.* **116**, 254301 (2016).
- [12] Weiner, C. *Interview with Paul Peter Ewald* (1968). URL http://www.aip.org/history/ohilist/4596_1.html.
- [13] Sangster, M. & M.Dixon. *Adv. Phys.* **25**, 247 (1976).

- [14] Wolf, D., Keblinski, P., Phillpot, S. & Eggebrecht, J. Exact method for the simulation of coulombic systems by spherically truncated, pairwise r^{-1} summation. *J. Phys. Chem.* **110**, 8254–8282 (1999).
- [15] Eastwood, J., Hockney, R. & Lawrence, D. *Comput. Phys. Commun.* (1980).
- [16] Hockney, R. & Eastwood, J. *Computer Simulations Using Particles* (McGraw-Hill, New York, 1981).
- [17] Knott, G. *Fourier Transforms* (2012). URL <http://www.civilized.com/files/newfourier.pdf>.
- [18] Pfalzner, S. & Gibbon, P. *Many-Body Tree Methods in Physics* (Cambridge University Press, 2005).
- [19] Potter, D., Stadel, J. & Teyssier, R. Pkdgrav3: beyond trillion particle cosmological simulations for the next era of galaxy surveys. *Comp. Astrop. Cosm.* **4**, 2 (2017).
- [20] Onsager, L. Electric moments of molecules in liquids. *J. Am. Chem. Soc.* 1486–1493 (1936). URL <http://pubs.acs.org/doi/abs/10.1021/ja01299a050>.
- [21] Yoshida, H. Construction of higher order symplectic integrators. *Phys. Lett. A* **150** (1990). URL <http://cacs.usc.edu/education/phys516/yoshida-symplectic-pla00.pdf>.
- [22] Watts, R. Monte carlo studies of liquid water. *Mol. Phys.* **26**, 1069–1083 (1974). URL http://www.tandfonline.com/doi/abs/10.1080/00268977400102381#.U_Wxd9bPHVM.
- [23] Nosé, S. A molecular dynamics method for simulations in the canonical ensemble. *Mol. Phys.* **52** (1984). URL <http://www.tandfonline.com/doi/abs/10.1080/00268978400101201#.U3Ic-3LI9yA>.
- [24] Nosé, S. A unified formulation of the constant temperature molecular dynamics methods. *J. Chem. Phys.* **81** (1984). URL <http://scitation.aip.org/content/aip/journal/jcp/81/1/10.1063/1.447334>.
- [25] Andersen, H. Molecular dynamics simulations at constant pressure and/or temperature. *J. Chem. Phys.* **72** (1980). URL <http://scitation.aip.org/content/aip/journal/jcp/72/4/10.1063/1.439486>.
- [26] Lubachevsky, B. D. How to simulate billiards and similar systems. *J. Comp. Phys.* **94**, 255–283 (1991).

- [27] McNamara, S. & Young, W. Inelastic collapse and dumping in a one-dimensional granular medium. *Phys. Rev.* (1991). URL <http://www-pord.ucsd.edu/~wryoung/reprintPDFs/InelasticCollapse.pdf>.
- [28] McNamara, S. & Young, W. Inelastic collapse in two dimensions. *Phys. Rev.* (1993). URL <http://www.uvm.edu/~pdodds/files/papers/others/1994/mcnamara1994a.pdf>.
- [29] Luding, S. & McNamara, S. How to handle inelastic collapse of a dissipative hard-sphere gas with the tc model. *Granul. Matter* (1998). URL <http://arxiv.org/abs/cond-mat/9810009>.
- [30] Walton, O. & Braun, R. Viscosity, granular-temperature and stress calculations for shearing assemblies of inelastic frictional disks. *J. Rheol.* 949–980 (1986). URL <http://arxiv.org/abs/cond-mat/9810009>.
- [31] Luding, S. Particle-particle interactions and models (discrete element method) (2010). URL http://www2.msm.ctw.utwente.nl/sluding/TEACHING/ParticleTechnology/Luding_PForcesModels.pdf.
- [32] Cundall, P. & Strack, O. A discrete numerical model for granular assemblies (1979). URL <http://www.icevirtuallibrary.com/content/article/10.1680/geot.1979.29.1.47>.
- [33] Perram, J. W. & Wertheim, M. S. Statistical mechanics of hard ellipsoids. overlap algorithm and the contact function. *J. Comput. Phys.* 409–416 (1985).
- [34] Donev, A. *et al.* Improving the density of jammed disordered packings using ellipsoids. *Science* 303, 990–993 (2004). URL <http://www.sciencemag.org/content/303/5660/990.full>.
- [35] Delaney, G. W. & Cleary, P. W. The packing properties of superellipsoids. *Europhys. Lett.* 89, 34002 (2010).
- [36] Alonso-Marroquin, F. Spheropolygons: A new method to simulate conservative and dissipative interactions between 2d complex-shaped rigid bodies. *Europhys. Lett.* 83, 14001 (2008). URL <http://iopscience.iop.org/0295-5075/83/1/14001>.
- [37] Matheron, G. & Serra, J. *The Birth of Mathematical Morphology* (2000). URL http://cmm.enscm.fr/~serra/pdf/birth_of_mm.pdf.
- [38] Leine, R. I., Glocker, C. & Van Campen, D. H. Nonlinear dynamics of the woodpecker toy. In *Proceedings of DETC 2001 ASME*

- Symposium on Nonlinear Dynamics and Control In Engineering Systems*, CDROM-9 (2001).
- [39] Maisonneuve, O. *Commemoration of Jean-Jacques Moreau* (2014). URL <http://www.isimmforum.tu-darmstadt.de/fileadmin/home/users/186/ISIMM/JJMoreau-ISIMM-forum.pdf>.
- [40] Brady, J. F. & Bossis, G. The rheology of concentrated suspensions of spheres in simple shear flow by numerical simulation. *J. Fluid Mech.* **155**, 105–129 (1985).
- [41] Malevanets, A. & Kapral, R. Mesoscopic model for solvent dynamics. *J. Chem. Phys.* **110**, 8605–8613 (1999).
- [42] Tsein, H. Superaerodynamics, mechanics of rarefied gases. *J. Aerospace Sci.* 342 (1946).
- [43] Bird, G. Approach to translational equilibrium in a rigid sphere gas. *Phys. Fluids* **6**, 1518 (1963).
- [44] Hoogerbrugge, P. & Koelman, J. Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. *Europhys. Lett.* (1992). URL <http://iopscience.iop.org/0295-5075/19/3/001>.
- [45] Español, P. & Warren, P. Statistical mechanics of dissipative particle dynamics. *Europhys. Lett.* (1994). URL <http://iopscience.iop.org/0295-5075/30/4/001>.
- [46] Gingold, R. & Monaghan, J. Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Mon. Not. R Astron. Soc.* **181**, 375–389 (1977). URL <http://adsabs.harvard.edu/abs/1977MNRAS.181..375G>.
- [47] Lucy, L. Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Astron. J.* **82**, 1013–1024 (1977). URL <http://adsabs.harvard.edu/abs/1977AJ.....82.1013L>.
- [48] Harada, T., Koshizuka, S. & Kawaguchi, Y. Smoothed particle hydrodynamics on gpus (2007). URL http://inf.ufrgs.br/cgi2007/cd_cgi/papers/harada.pdf.
- [49] Hohenberg, P. & Kohn, W. Inhomogeneous electron gas. *Phys. Rev.* **136**, B864 (1964).
- [50] Becke, A. Density-functional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev.* 3098 (1988). URL <http://journals.aps.org/pr/abstract/10.1103/PhysRevA.38.3098>.

- [51] Grimme, S. Semiempirical gga-type density functional constructed with a long-range dispersion correction. *J. Comp. Chem.* **27**, 1787–1799 (2006). URL <http://onlinelibrary.wiley.com/doi/10.1002/jcc.20495/abstract>.
- [52] Car, R. & Parrinello, M. Unified approach for molecular dynamics and density-functional theory. *Phys. Rev. Lett.* **2471** (1985). URL <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.55.2471>.