Optimization

For MD simulations of N particles – we have an operation of complexity $\mathcal{O}\left(N^2\right)$.

Alternatively, let our potential be a function $v(r) \sim r^{-2n}$ with $n \geq 1$. We can then omit the computation of the square root in

$$r_{ij} = \sqrt{\sum_{\alpha=1}^{d} \left(x_i^\alpha - x_j^\alpha\right)^2} \tag{23}$$

since for the chosen potential $\mathbf{f} = -\nabla r^{-2n} \propto r^{-2(n-1)}\mathbf{r}$ and $\mathbf{f}_i = f\left(r^{-2(n-1)}\right)\mathbf{r}_i$.

If the potential is not a simple function and its calculation would imply a lot of tedious calculations, discretizing the potential and storing its values in a lookup table might be helpful. For short range potentials, we define a cutoff r_c and discretize the interval $(0, r_c^2)$ in K pieces, i.e.,

$$l_k = \frac{k}{K} r_c^2. \tag{24}$$

The force values stored in a lookup table are $f_k = f\left(\sqrt{l_k}\right)$ and the corresponding index k is given by

$$k = \left\lfloor S \sum_{\alpha=1}^{d} \left(x_i^\alpha - x_j^\alpha\right)^2 \right\rfloor + 1, \tag{25}$$

where $\lfloor \cdot \rfloor$ denotes the floor function and $S = K/r_c^2$.

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

$$\tilde{v}\left(r\right) = \begin{cases} v(r) - v(r_c) - \left.\frac{\partial v}{\partial r}\right|_{r=r_c} (r - r_c) & \text{if } r \leq r_c, \\ 0 & \text{if } r > r_c, \end{cases} \tag{26}$$

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 value of $r_c = 2.5\sigma$ is typically used. Care must be taken for potentials decaying as r^{-1}, since forces at large distances are not negligible.

Linked-Cell Method

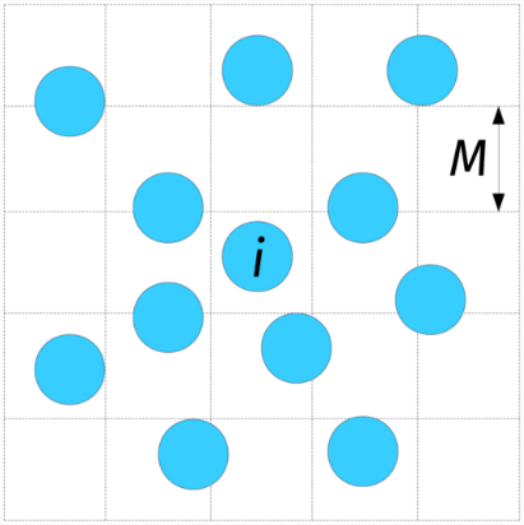


Figure 6: An illustration of the linked-cell method. A grid with grid spacing $M(\frac{r_c}{2} < M < r_c)$ is placed on top of the MD simulation geometry. Only interactions between particles in a certain cell neighborhood have to be considered [D. Knuth, The Art of Computer

In d dimensions there are 3^d cells of interest. On average, we thus have to compute the interactions of $N3^dN/M^d$ particles. To keep track of the locations of all particles, we define a vector FIRST of length $N_M = M^d$ to store the index of a particle located in cell j in $\mathrm{FIRST}[j]$. If cell j is empty, then $\mathrm{FIRST}[j] = 0$. 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 $\mathrm{LIST}[i] = 0$.

The following code shows an example of how to extract the particles located in cell $i = 2$.

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

When a particle changes the 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 composition).

```python
#celllist properties
l = int(L/rc)
l2 = l*l
n_cells = l2*l
cellsize = L/l

def update_celllist(r_current):

    celllist = [ [] for _ in range(n_cells) ]   #e

    for idx in range(N):
        r = r_current[idx,:]
        i_x = int(r[0]/cellsize)
        i_y = int(r[1]/cellsize)
        i_z = int(r[2]/cellsize)

        celllist[i_x*l2 + i_y*l + i_z].append(idx)

    return celllist

def get_cell_coord(cell_idx):
    cell_idx_yz = cell_idx % l2

    x = int(cell_idx / l2)
    y = int(cell_idx_yz / l)
    z = cell_idx_yz % l

    return x, y, z
```

```python
E_kin = 0.5*m*np.sum(np.square(v_current))
E_pot = 0.0

F = np.zeros((N,3))
for cell_idx in range(n_cells):
    main_cell = celllist[cell_idx]   #main cell

    cell_idx_x, cell_idx_y, cell_idx_z  = get_cell_coord(cell_idx)

    for idx_i in range(len(main_cell)):
        i = main_cell[idx_i]
        for idx_x in [(cell_idx_x-1+l)%l, cell_idx_x, (cell_idx_x+1)%l]:
            for idx_y in [(cell_idx_y-1+l)%l, cell_idx_y, (cell_idx_y+1)%l]:
                for idx_z in [(cell_idx_z-1+l)%l, cell_idx_z, (cell_idx_z+1)%l]:
                    other_cell = celllist[idx_x*l2 + idx_y*l + idx_z]
                    for idx_j in range(len(other_cell)):
                        j = other_cell[idx_j]

                        if i < j:
                            r_vec = r_rel_pbc(r_current[i,:], r_current[j,:])
                            r2 = r_vec.dot(r_vec)
                            if r2 < rc2:
                                force, potential = force_potential(r_vec, r2)

                                E_pot += potential - V_c
                                F[i,:] += force
                                F[j,:] -= force
```

Lagrange multipliers

One of the first description of composed particle systems based on an additional force term in the equations of motions has been suggested in [Ryckaert, Ciccotti and Berendsen, 1977]. The idea is to rewrite the equation of motion for each particle as

$$m_i \ddot{\mathbf{x}}_i = \underbrace{\mathbf{f}_i}_{\text{external interaction}} + \underbrace{\mathbf{g}_i}_{\text{internal constraints}}, \tag{28}$$

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

We now impose such constraints to enforce the geometric arrangement of the molecules, e.g., certain distances d_{12} and d_{23} between atoms. Therefore, we define a potential such that the constraint forces 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

$$\chi_{12} = r_{12}^2 - d_{12}^2, \tag{29}$$
$$\chi_{23} = r_{23}^2 - d_{23}^2, \tag{30}$$

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

$$\mathbf{g}_k = \frac{\lambda_{12}}{2} \nabla_{\mathbf{x}_k} \chi_{12} + \frac{\lambda_{23}}{2} \nabla_{\mathbf{x}_k} \chi_{23}, \tag{31}$$

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. (31), the constraint forces are

$$\mathbf{g}_1 = \lambda_{12} \mathbf{r}_{12}, \quad \mathbf{g}_2 = \lambda_{23} \mathbf{r}_{23} - \lambda_{12} \mathbf{r}_{12}, \quad \mathbf{g}_3 = -\lambda_{23} \mathbf{r}_{23}. \tag{32}$$

The previous 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

$$\tilde{\mathbf{x}}_i (t + \Delta t) = 2\mathbf{x}_i - \mathbf{x}_i (t - \Delta t) + \Delta t^2 \frac{\mathbf{f}_i}{m_i}. \tag{33}$$

Then we correct the value using the constraints according to

$$\mathbf{x}_i (t + \Delta t) = \tilde{\mathbf{x}}_i (t + \Delta t) + \Delta t^2 \frac{\mathbf{g}_i}{m_i}. \tag{34}$$

By combining Eqs. (34) and (31), the updated positions are given by

$$\mathbf{x}_1 (t + \Delta t) = \tilde{\mathbf{x}}_1 (t + \Delta t) + \Delta t^2 \frac{\lambda_{12}}{m_1} \mathbf{r}_{12} (t), \tag{35}$$

$$\mathbf{x}_2 (t + \Delta t) = \tilde{\mathbf{x}}_2 (t + \Delta t) + \Delta t^2 \frac{\lambda_{23}}{m_2} \mathbf{r}_{23} (t) - \Delta t^2 \frac{\lambda_{12}}{m_2} \mathbf{r}_{12} (t), \tag{36}$$

$$\mathbf{x}_3 (t + \Delta t) = \tilde{\mathbf{x}}_3 (t + \Delta t) - \Delta t^2 \frac{\lambda_{23}}{m_3} \mathbf{r}_{23} (t). \tag{37}$$

With these expressions, we now obtain λ_{12} and λ_{23} by inserting (35), (36) and (37) into the constraint condition, i.e.,

$$\begin{aligned} |\mathbf{x}_1 (t + \Delta t) - \mathbf{x}_2 (t + \Delta t)|^2 &= d_{12}^2, \\ |\mathbf{x}_2 (t + \Delta t) - \mathbf{x}_3 (t + \Delta t)|^2 &= d_{23}^2, \end{aligned} \tag{38}$$

and finally

$$\begin{aligned} \left| \tilde{\mathbf{r}}_{12} (t + \Delta t) + \Delta t^2 \lambda_{12} \left(\frac{1}{m_1} + \frac{1}{m_2} \right) \mathbf{r}_{12} (t) - \Delta t^2 \frac{\lambda_{23}}{m_2} \mathbf{r}_{23} (t) \right|^2 &= d_{12}^2, \\ \left| \tilde{\mathbf{r}}_{23} (t + \Delta t) + \Delta t^2 \lambda_{23} \left(\frac{1}{m_2} + \frac{1}{m_3} \right) \mathbf{r}_{23} (t) - \Delta t^2 \frac{\lambda_{12}}{m_2} \mathbf{r}_{12} (t) \right|^2 &= d_{23}^2, \end{aligned} \tag{39}$$

where $\tilde{\mathbf{r}}_{ij} = \tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j$. These coupled quadratic equations are solved in practice perturbatively by linearising in t.

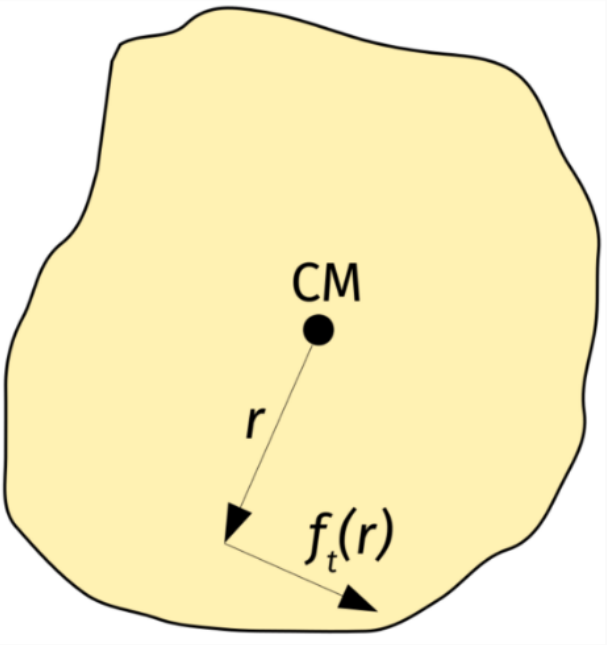


Figure 2: An example of a rigid body in two dimensions. The black dot show the center of mass (CM), and $f_t(r)$ represents the tangential force component.

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

$$I = \int\int_A r^2 \rho\,(r)\,\mathrm{dA} \quad \text{and} \quad \mathrm{M} = \int\int_A r f_t\,(r)\,\mathrm{dA}, \tag{3}$$

where f_t is 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

$$I\dot{\omega} = \mathrm{M}. \tag{4}$$

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

$$\begin{aligned} \phi\,(t+\Delta t) &= 2\phi\,(t) - \phi\,(t-\Delta t) + \Delta t^2 \frac{\mathrm{M}\,(t)}{I}, \\ \mathbf{x}\,(t+\Delta t) &= 2\mathbf{x}\,(t) - \mathbf{x}\,(t-\Delta t) + \Delta t^2 M^{-1}(t) \sum_{j\in A} f_j\,(t), \end{aligned} \tag{5}$$

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

$$\mathrm{M}\,(t) = \sum_{j\in A} \left[f_j^y\,(t)\, d_j^x\,(t) - f_j^x\,(t)\, d_j^y\,(t) \right]. \tag{6}$$

Rigid Bodies 3D

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

$$\mathbf{x} = R(t)\mathbf{y}, \tag{7}$$

where $R(t) \in \mathrm{SO}(3)$ denotes a rotation matrix[1].

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

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

$$R^T \dot{R} + \dot{R}^T R = \Omega + \Omega^T = 0. \tag{8}$$

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

$$\Omega = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \quad \text{and} \quad \Omega \mathbf{y} = \boldsymbol{\omega} \wedge \mathbf{y}, \tag{9}$$

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

The angular momentum is then given by

$$\mathbf{L} = \sum_{i=1}^{n} m_i \mathbf{x}_i \wedge \dot{\mathbf{x}}_i = \sum_{i=1}^{n} m_i R\mathbf{y}_i \wedge \dot{R}\mathbf{y}_i. \tag{10}$$

Combing Eqs. (10) and (9) yields

$$\mathbf{L} = R \sum_{i=1}^{n} m_i \mathbf{y}_i \wedge (\boldsymbol{\omega} \wedge \mathbf{y}_i) = R \sum_{i=1}^{n} m_i \left[\boldsymbol{\omega} (\mathbf{y}_i \cdot \mathbf{y}_i) - \mathbf{y}_i (\boldsymbol{\omega} \cdot \mathbf{y}_i) \right]. \tag{11}$$

The components of the inertia tensor are defined as

$$I_{jk} = \sum_{i=1}^{n} m_i \left[(\mathbf{y}_i \cdot \mathbf{y}_i) \delta_{jk} - \mathbf{y}_i^j \mathbf{y}_i^k \right] \tag{12}$$

and thus

$$\mathbf{L} = R\mathbf{S} \quad \text{with} \quad S_j = \sum_{k=1}^{3} I_{jk} \boldsymbol{\omega}_k. \tag{13}$$

where I is the inertia tensor.

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

$$I = \begin{pmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{pmatrix} \quad \text{and} \quad S_j = I_j \omega_j. \tag{14}$$

With Eq. (13), the equations of motion are determined by

$$\dot{\mathbf{L}} = \dot{R}\mathbf{S} + R\dot{\mathbf{S}} = \widetilde{\mathbf{M}}, \tag{15}$$

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

$$\dot{\omega}_1 = \frac{M_1}{I_1} + \left(\frac{I_2 - I_3}{I_1} \right) \omega_2 \omega_3, \tag{16}$$

$$\dot{\omega}_2 = \frac{M_2}{I_2} + \left(\frac{I_3 - I_1}{I_2} \right) \omega_3 \omega_1, \tag{17}$$

$$\dot{\omega}_3 = \frac{M_3}{I_3} + \left(\frac{I_1 - I_2}{I_3} \right) \omega_1 \omega_2. \tag{18}$$

The angular velocities are then integrated according to

$$\omega_1 (t + \Delta t) = \omega_1 (t) + \Delta t \frac{M_1 (t)}{I_1} + \Delta t \left(\frac{I_2 - I_3}{I_1} \right) \omega_2 \omega_3, \tag{19}$$

$$\omega_2 (t + \Delta t) = \omega_2 (t) + \Delta t \frac{M_2 (t)}{I_2} + \Delta t \left(\frac{I_3 - I_1}{I_2} \right) \omega_3 \omega_1, \tag{20}$$

$$\omega_3 (t + \Delta t) = \omega_3 (t) + \Delta t \frac{M_3 (t)}{I_3} + \Delta t \left(\frac{I_1 - I_2}{I_1} \right) \omega_1 \omega_2. \tag{21}$$

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

$$\widetilde{\boldsymbol{\omega}} (t + \Delta t) = R\boldsymbol{\omega} (t + \Delta t). \tag{22}$$

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.

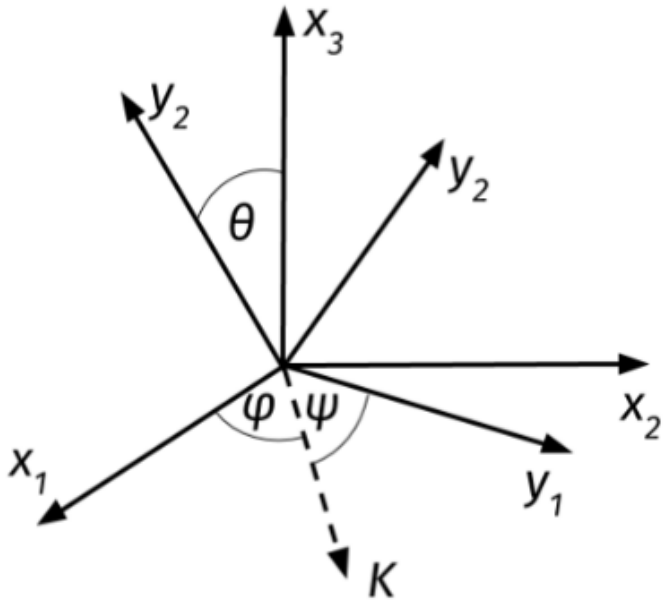


Figure 4: Euler angle parameterization of a rotation matrix.

One possible parameterization of the rotation matrix R is the following:

$$
\begin{aligned}
R &= R(\phi, \theta, \psi) \\
&= \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}
\end{aligned}
\tag{23}
$$

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, making this approach computationally too expensive. For the computation of angular velocities, derivatives of Eq. (23) have to be considered.

Quaternions

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

$$q_0 = \cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi+\psi}{2}\right), \tag{24}$$

$$q_1 = \sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi-\psi}{2}\right), \tag{25}$$

$$q_2 = \sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi-\psi}{2}\right), \tag{26}$$

$$q_3 = \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi+\psi}{2}\right), \tag{27}$$

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

The rotation matrix as defined in Eq. (23) has a quaternion representation, i.e.,

$$R = \begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix}. \tag{28}$$

We now found a more efficient way of computing rotations without the necessity of computing lengthy products of sine and cosine functions. This approach much faster than one of Eq. (23). The angular velocities are then computed according to

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix}\begin{pmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{29}$$

Since the world of quaternions and the normal Euclidean space are connected by a diffeomorphism, there is always the possibility of calculating the values of the Euler angles if needed

$$\phi = \arctan\left[\frac{2(q_0q_1+q_2q_3)}{1-2(q_1^2+q_2^2)}\right] \tag{30}$$

$$\theta = \arcsin\left[2(q_0q_2-q_1q_3)\right] \tag{31}$$

$$\psi = \arctan\left[\frac{2(q_0q_3+q_1q_2)}{1-2(q_2^2+q_3^2)}\right] \tag{32}$$

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 $M(t)$ in the body frame.
- Obtain $\omega(t+\Delta t)$ according to Eq. (21) (quaternion representation).
- Update the rotation matrix as defined in Eq. (28) by computing $q(t+\Delta t)$ according to Eq. (29).

Canonical Ensemble

Experiments are often conducted at constant temperature and not at constant energy. This is a common situation, since 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 latter methods, we shall define the concept of temperature used in the subsequent sections. We start from the equipartition theorem

$$\left\langle q_\mu \frac{\partial \mathcal{H}}{\partial q_\nu} \right\rangle = \left\langle p_\mu \frac{\partial \mathcal{H}}{\partial p_\nu} \right\rangle = \delta_{\mu\nu} kT \tag{1}$$

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

We consider a classical system whose Hamiltonian is given by

$$\mathcal{H} = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + V\left(\mathbf{x}_1, \ldots, \mathbf{x}_N\right) \tag{2}$$

and we define the instantaneous temperature

$$\mathcal{T} = \frac{2}{3k\left(N-1\right)} \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i}. \tag{3}$$

Velocity rescaling

Intuitively, we should be able to adjust the system's instantaneous temperature by rescaling the velocities of the particles according to

$$\mathbf{v}_i \to \alpha \mathbf{v}_i. \tag{4}$$

The measured tempereature is proportional to the squared velocities and thus

$$\mathcal{T} \to \alpha^2 \mathcal{T}. \tag{5}$$

Therefore, we have to set

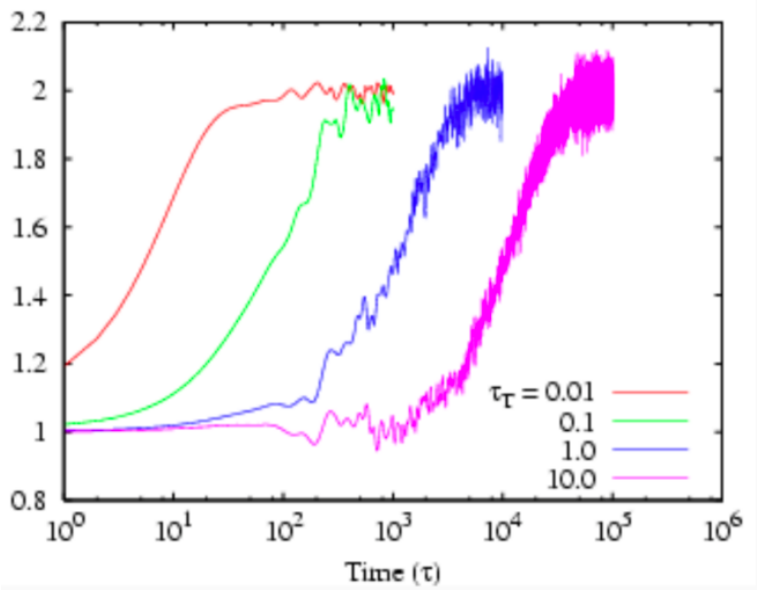
$$\alpha = \sqrt{\frac{T}{\mathcal{T}}} \tag{6}$$

to stay at a fixed desired temperature T.

This method is very easy to implement. However, the problem is that we change the physics and in particular the time and the resulting velocity distribution deviates from the canonical one. This method might seems to be very effective. A modification of this method makes use of an additional parameter t_T (relaxation time) which describes the coupling to heat bath. The scaling factor is then (Berendsen thermostat)

$$\alpha = \sqrt{1 + \frac{\Delta t}{t_T}\left(\frac{T}{\mathcal{T}} - 1\right)}. \tag{7}$$

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



Nosé-Hoover thermostat

In order to overcome the problem of the wrong velocity distribution, we are now going to discuss the Nosè-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.The corresponding potential and kinetic energy are

$$\begin{aligned} \mathcal{V}(s) &= (3N+1)\, k_B T \ln s, \\ K(s) &= \frac{1}{2} Q \dot{s}^2. \end{aligned} \tag{13}$$

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

$$\mathrm{d}t' = s\mathrm{d}t \qquad \text{and} \qquad \mathbf{p}_i' = s\mathbf{p}_i. \tag{14}$$

Similarly, velocities are also rescaled since

$$\mathbf{v}_i' = \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t'} = \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}t'} = \frac{\mathbf{v}_i}{s}. \tag{15}$$

Note that we also used the chain rule in the second step of the equation for the rescaling of momenta:

$$\mathbf{p}_i' = \nabla_{\mathbf{v}_i'}\left(\frac{1}{2}\sum_{i=1}^{N} m_i {\mathbf{v}_i'}^2\right) = s\nabla_{\mathbf{v}_i}\left(\frac{1}{2}\sum_{i=1}^{N} m_i {\mathbf{v}_i}^2\right) = s\mathbf{p}_i, \tag{16}$$

The Hamiltonian is thus

$$\mathcal{H} = \sum_{i=1}^{N} \frac{{\mathbf{p}_i'}^2}{2m_i s^2} + \frac{1}{2} Q\dot{s}^2 + V(\mathbf{x}_1, \ldots, \mathbf{x}_N) + \mathcal{V}(s), \tag{17}$$

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

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t'} &= \nabla_{\mathbf{p}_i'}\mathcal{H} = \frac{\mathbf{p}_i'}{m_i s^2}, \\ \frac{\mathrm{d}s}{\mathrm{d}t'} &= \frac{\partial \mathcal{H}}{\partial p_s} = \frac{p_s}{Q}. \end{aligned} \tag{18}$$

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

$$\mathbf{f}_i = \frac{\mathrm{d}\mathbf{p}_i'}{\mathrm{d}t'} = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}_i} = -\nabla_{\mathbf{x}_i} V(\mathbf{x}_1, \ldots, \mathbf{x}_N) = 2m_i s\dot{s}\dot{\mathbf{x}}_i + m_i s^2 \ddot{\mathbf{x}}_i \tag{19}$$

and

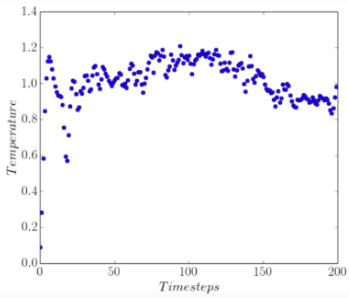
$$\frac{\mathrm{d}\mathbf{p}_s}{\mathrm{d}t'} = -\frac{\partial \mathcal{H}}{\partial s} = \frac{1}{s}\left[\sum_{i=1}^{N} \frac{{\mathbf{p}_i'}^2}{m_i s^2} - (3N+1)\, k_B T\right]. \tag{20}$$

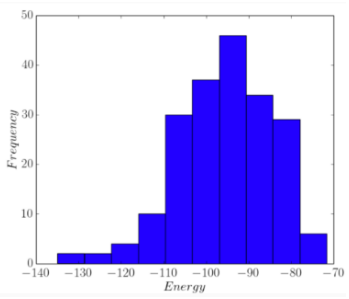
Based on the latter Hamilton equations, we find for the equations of motion in virtual time t'

$$m_i s^2 \ddot{\mathbf{x}}_i = \mathbf{f}_i - 2m_i \dot{s} s \dot{\mathbf{x}}_i \quad \text{with} \quad i \in \{1, \ldots, N\} \tag{21}$$

and

$$Q\ddot{s} = \sum_{i=1}^{N} m_i s \dot{\mathbf{x}}_i^2 - \frac{1}{s}(3N+1)\, k_B T. \tag{22}$$





In order to obtain the equations of motion in real time, we have to remind ourselfs that $\mathrm{d}t = \mathrm{d}t'/s$ and $\mathbf{p}_i' = s\mathbf{p}_i$. Thus, we find for the velocities

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} &= s\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t'} = \frac{\mathbf{p}_i'}{m_i s} = \frac{\mathbf{p}_i'}{m_i}, \\ \frac{\mathrm{d}s}{\mathrm{d}t} &= s\frac{\mathrm{d}s}{\mathrm{d}t'} = s\frac{\mathbf{p}_s}{Q}, \end{aligned} \tag{23}$$

and for the forces

$$\begin{aligned} \frac{\mathrm{d}\mathbf{p}_i}{\mathrm{d}t} &= s\frac{\mathrm{d}}{\mathrm{d}t'}\left(\frac{\mathbf{p}_i'}{s}\right) = \frac{\mathrm{d}\mathbf{p}_i'}{\mathrm{d}t'} - \frac{1}{s}\frac{\mathrm{d}s}{\mathrm{d}t'}\mathbf{p}_i' = \mathbf{f}_i - \frac{1}{s}\frac{\mathrm{d}s}{\mathrm{d}t}\mathbf{p}_i, \\ \frac{\mathrm{d}\mathbf{p}_s}{\mathrm{d}t} &= s\frac{\mathrm{d}p_s}{\mathrm{d}t'} = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{m_i} - (3N+1)\, kT. \end{aligned} \tag{24}$$

With $\xi = \frac{\mathrm{d}\ln(s)}{\mathrm{d}t} = \frac{\dot{s}}{s}$ representing a friction term, the equations of motions (65) and (66) are given in real time by

$$\ddot{\mathbf{x}}_i = \frac{\mathbf{f}_i}{m_i} - \xi\dot{\mathbf{x}}_i \tag{25}$$

and

$$Q\dot{\xi} = \sum_{i=1}^{N} m_i \dot{\mathbf{x}}_i^2 - (3N+1)\, k_B T. \tag{26}$$

The first term in Eq. (70) denotes the **measured** kinetic energy whereas the second one corresponds to the **desired** kinetic enery. The quantity Q represents the coupling to the heat bath and the higher the value of Q, the stronger the system reacts to temperature fluctuations. For $Q \to \infty$, we recover microcanonical MD.

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

$$\overline{\Delta T} = \sqrt{\frac{2}{Nd}}\overline{T}, \tag{27}$$

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

$$Z = \int \delta(\mathcal{H} - E)\, \mathrm{d}s\, \mathrm{d}p_s\, \mathrm{d}^3x'\, \mathrm{d}^3p', \tag{28}$$

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

$$\begin{aligned} Z &= \int \delta\left[(\mathcal{H}_1 - E) + (3N+1)\, kT \ln(s)\right] s^{3N} \mathrm{d}s\, \mathrm{d}p_s\, \mathrm{d}^3x\, \mathrm{d}^3p \\ &= \int \delta\left[s - e^{-\frac{\mathcal{H}_1 - E}{(3N+1)kT}}\right] \frac{s^{3N+1}}{(3N+1)\, kT} \mathrm{d}s\, \mathrm{d}p_s\, \mathrm{d}^3x'\, \mathrm{d}^3p', \end{aligned} \tag{29}$$

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. (73) over s yields

$$\begin{aligned} Z &= \int \frac{1}{(3N+1)\, kT} e^{-\frac{\mathcal{H}_1 - E}{kT}} \mathrm{d}p_s\, \mathrm{d}^3x\, \mathrm{d}^3p \\ &= \int e^{-\frac{\mathcal{H}_1 - E}{kT}} \mathrm{d}^3x\, \mathrm{d}^3p \int \frac{1}{(3N+1)\, kT} \mathrm{d}p_s, \end{aligned} \tag{30}$$

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.

Constant pressure

Another important situation is the one of constant pressure.

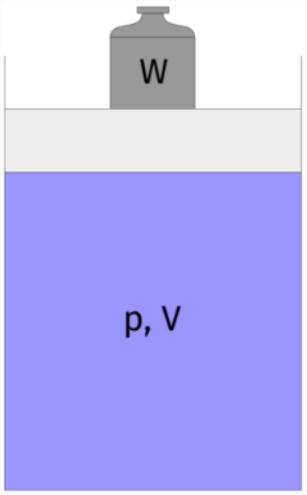


Figure 1: A weight of mass W exerts a pressure p on the system with volume V.

We will again consider the equipartition theorem (45) with the Hamiltonian

$$\mathcal{H} = K(\mathbf{p}) + V(\mathbf{x})$$

Taking the derivative of the $\mathcal{H}$ with respect to the spatial component yields

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

We define

$$w = -\frac{1}{3}\left\langle \sum_{i=1}^{N} \mathbf{x}_i \cdot \left(\mathbf{f}_i^{\text{part}}\right) \right\rangle \tag{34}$$

as the viral. Based on

$$\frac{1}{3}\left\langle \sum_{i=1}^{N} \mathbf{x}_i \cdot \left(\mathbf{f}_i^{\text{part}}\right) \right\rangle = -\frac{1}{3}\int_{\Gamma} p\mathbf{x}\mathrm{d}\mathbf{A} = -\frac{1}{3}p\int_{V} (\nabla \cdot \mathbf{x})\,\mathrm{d}\mathbf{V} = -pV \tag{35}$$

we define the instantaneous pressure $\mathcal{P}$ by

$$\mathcal{P}V \equiv Nk_BT + \langle w \rangle \tag{36}$$

Similarly to Nosé-Hoover thermostat, we introduce a sort of "pressure bath',', i.e a parameter W which adjusts the pressure of the system. The volume change can be written as:

$$V = 1 - \alpha_T \frac{\Delta}{t_p}(p - \mathcal{P}) \tag{37}$$

where α_T is the isothermal compressibility and t_p is a relaxation time for the pressure.

Elastic Collisions

One of the first examples for event-driven programming applied to molecular dynamics is a work by Alder in 1957.

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.

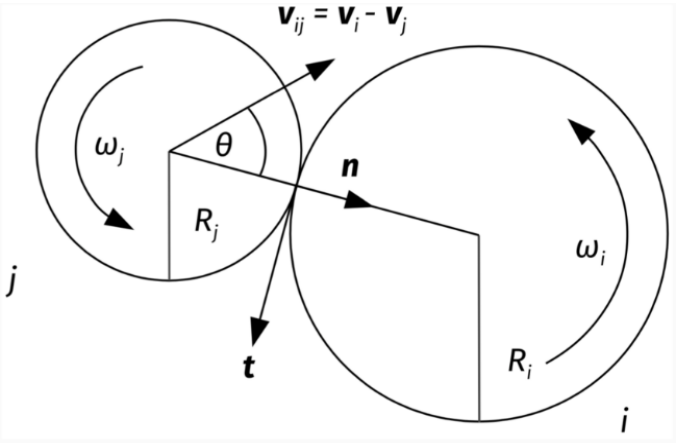


Figure 2: Two particles collide elastically.

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 the particle i and the particle j would occur. At time t_{ij}, the distance between the two particles is

$$|\mathbf{r}_{ij}(t_{ij})| = |R_i + R_j| \tag{38}$$

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

$$v_{ij}^2 t_{ij}^2 + 2\left[\mathbf{r}_{ij}(t_0)\,\mathbf{v}_{ij}\right] t_{ij} + \left[r_{ij}(t_0)\right]^2 - (R_i + R_j)^2 = 0. \tag{39}$$

We should bear in mind that Eq. (83) are only meaningful if the trajectories of particles i and j cross with each other. The time t_c when the next collision occurs, is the minimum over all pairs, i.e.,

$$t_c = \min_{ij}(t_{ij}). \tag{40}$$

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

$$\mathbf{r}_i(t_0 + t_c) = \mathbf{r}_i(t_0) + \mathbf{v}_i(t_0)\,t_c \quad \text{and} \quad \phi_i(t_0 + t_c) = \phi_i(t_0) + \omega_i(t_0)\,t_c. \tag{41}$$

Lubachevsky method

Instead of going through all the particle pairs ($\mathcal{O}\left(N^2\right)$), we create a list of events for each particle. The reordering of the event list takes a time in the order of $\mathcal{O}\left(N \log N\right)$.

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.,

$$t^{(i)} = \min_j \left(t_{ij}\right). \tag{42}$$

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 $\mathcal{O}\left(1\right)$ instead of $\mathcal{O}\left(N\right)$. The next collision occurs at time

$$t_c = \min_i \left(t^{(i)}\right). \tag{43}$$

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

- The vector $\mathrm{part}[m]$ points to particle i which is at position m in the stack. (Sometimes also a vector $\mathrm{pos}[i]$ 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.
- $\mathrm{part}[1]$ is the particle with minimal collision time: $t_c = t^{(\mathrm{part}[1])}$
- After the event for both particles all 6 entries (event times, new partners, positions and velocities) have to be updated. Additionally, the vector $\mathrm{part}[m]$ has to be reordered.

Reordering the times $t^{(i)}$ after each event is of order $\mathcal{O}\left(\log N\right)$ 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.

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

$$I\frac{\mathrm{d}\boldsymbol{\omega}_i}{\mathrm{d}t} = \mathbf{r} \wedge \mathbf{f}_i, \tag{50}$$

where I denotes the moment of inertia and $\mathbf{f}_i$ the forces exerted on particle i.

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

$$\begin{aligned} I\left(\boldsymbol{\omega}_i' - \boldsymbol{\omega}_i\right) &= -Rm\mathbf{n} \wedge \left(\mathbf{v}_i' - \mathbf{v}_i\right), \\ I\left(\boldsymbol{\omega}_j' - \boldsymbol{\omega}_j\right) &= Rm\mathbf{n} \wedge \left(\mathbf{v}_j' - \mathbf{v}_j\right), \end{aligned} \tag{51}$$

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

$$\mathbf{v}_i' + \mathbf{v}_j' = \mathbf{v}_i + \mathbf{v}_j, \tag{52}$$

we obtain the rule for computing the new angular velocities after the collision, i.e.,

$$\boldsymbol{\omega}_i' - \boldsymbol{\omega}_i = \boldsymbol{\omega}_j' - \boldsymbol{\omega}_j = -\frac{Rm}{I}\left(\mathbf{v}_i' - \mathbf{v}_i\right) \wedge \mathbf{n}. \tag{53}$$

The relative velocity between particles i and j is

$$\mathbf{u}_{ij} = \mathbf{v}_i - \mathbf{v}_j - R(\boldsymbol{\omega}_i + \boldsymbol{\omega}_j) \wedge \mathbf{n}. \tag{54}$$

We decompose the relative velocities $\mathbf{u}$ of the particles into their normal and tangential components $\mathbf{u}^n$ and $\mathbf{u}^t$, 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, the relevant quantity to consider is the relative velocity of the particle surfaces at the contact point. The normal and tangential velocities are given by

$$\begin{aligned} \mathbf{u}_{ij}^n &= \left(\mathbf{u}_{ij}\mathbf{n}\right)\mathbf{n}, \\ \mathbf{u}_{ij}^t &= \mathbf{u}_{ij} \wedge \mathbf{n} = \left[\left(\mathbf{v}_i - \mathbf{v}_j\right) - R\left(\boldsymbol{\omega}_i + \boldsymbol{\omega}_j\right)\right] \wedge \mathbf{n}. \end{aligned} \tag{55}$$

General slips are described by

$$\mathbf{u}_{ij}^{t\,\prime} = e_t \mathbf{u}_{ij}^t, \tag{56}$$

where the 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 $e_t = 1$. In the case of $e_t < 1$, energy is dissipated.

If we compute the difference of the relative tangential velocities before and after the slip we get

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

Combining the previous equation with Eq. (97), we obtain an expression without angular velocities

$$\begin{aligned} \mathbf{u}_{ij}^t - \mathbf{u}_{ij}^{t\,\prime} &= \left(1 - e_t\right)\mathbf{u}_{ij}^t && (58) \\ &= -\left[2\left(\mathbf{v}_i^{t\prime} - \mathbf{v}_i^t\right) + 2q\left(\mathbf{v}_i^{t\prime} - \mathbf{v}_i^t\right)\right] && (59) \end{aligned}$$

and finally

$$\mathbf{v}_i^{t\prime} = \mathbf{v}_i^t - \frac{\left(1 - e_t\right)\mathbf{u}_{ij}^t}{2\left(1 + q\right)} \quad \text{with} \quad q = \frac{mR^2}{I}. \tag{60}$$

Analogously, we find for the remaining quantities

$$\begin{aligned} \mathbf{v}_j^{t\prime} &= \mathbf{v}_j^t + \frac{\left(1 - e_t\right)\mathbf{u}_{ij}^t}{2\left(1 + q\right)}, \\ \boldsymbol{\omega}_i' &= \boldsymbol{\omega}_i - \frac{\left(1 - e_t\right)\mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R\left(1 + q^{-1}\right)}, \\ \boldsymbol{\omega}_j' &= \boldsymbol{\omega}_j - \frac{\left(1 - e_t\right)\mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R\left(1 + q^{-1}\right)}. \end{aligned} \tag{61}$$

And the updated velocities are

$$\begin{aligned} \mathbf{v}_i{}' &= \mathbf{v}_i - \mathbf{u}_{ij}^n - \frac{\left(1 - e_t\right)\mathbf{u}_{ij}^t}{2\left(1 + q\right)}, \\ \mathbf{v}_j{}' &= \mathbf{v}_j + \mathbf{u}_{ij}^n + \frac{\left(1 - e_t\right)\mathbf{u}_{ij}^t}{2\left(1 + q\right)}. \end{aligned} \tag{62}$$

Inelastic collisions (of rotating particles)

Elastic collisions correspond to $r = 1$ whereas perfect plasticity is described by $r = 0$. Similar to our previous discussion of collisions with rotations, we also distinguish between normal and tangential energy transfer and define the corresponding coefficients

$$e_n = \sqrt{r_n} = \frac{v_n^{\text{after}}}{v_n^{\text{before}}}, \tag{27}$$

$$e_t = \sqrt{r_t} = \frac{v_t^{\text{after}}}{v_t^{\text{before}}}. \tag{28}$$

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 energies involved in the events, the angle of impact and other factors. Usually, they are determined experimentally.

The relative velocity of the particles at their contact point is

$$\mathbf{u}_{ij}^n = (\mathbf{u}_{ij}\mathbf{n})\,\mathbf{n} = [(\mathbf{v}_i - \mathbf{v}_j)\,\mathbf{n}]\,\mathbf{n}. \tag{29}$$

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

$$\mathbf{u}_{ij}^{n\prime} = -e_n \mathbf{u}_{ij}^n \tag{30}$$

For $e_n = 1$, there is no dissipation whereas dissipation effects occur for $e_n < 1$.

Similar to Eq. (19) and following derivations, we obtain the expressions for the velocities of each particle after the collision

$$\begin{aligned} \mathbf{v}_i{}' &= \mathbf{v}_i - \frac{(1+e_n)}{2}\mathbf{u}_{ij}^n, \\ \mathbf{v}_j{}' &= \mathbf{v}_j + \frac{(1+e_n)}{2}\mathbf{u}_{ij}^n. \end{aligned} \tag{31}$$

In the case of perfect slip, the momentum exchange is

$$\Delta\mathbf{p}_n = -m_{\text{eff}}(1+e_n)\,[(\mathbf{v}_i - \mathbf{v}_j)\,\mathbf{n}]\,\mathbf{n}. \tag{32}$$

With $q = \frac{m_{\text{eff}}R^2}{I_{\text{eff}}}$, the equations for the velocities after the collision are

$$\begin{aligned} \mathbf{v}_i' &= \mathbf{v}_i - \frac{(1+e_n)}{2}\mathbf{u}_{ij}^n - \frac{(1-e_t)\,\mathbf{u}_{ij}^t}{2\,(1+q)}, \\ \mathbf{v}_j' &= \mathbf{v}_j + \frac{(1+e_n)}{2}\mathbf{u}_{ij}^n + \frac{(1-e_t)\,\mathbf{u}_{ij}^t}{2\,(1+q)}, \\ \boldsymbol{\omega}_i' &= \boldsymbol{\omega}_i - \frac{(1-e_t)\,\mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R\,(1+q^{-1})}, \\ \boldsymbol{\omega}_j' &= \boldsymbol{\omega}_j + \frac{(1-e_t)\,\mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R\,(1+q^{-1})}. \end{aligned} \tag{33}$$

These equations describe inelastic collisions of rotating particles.

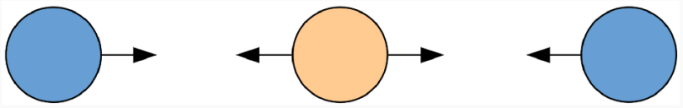
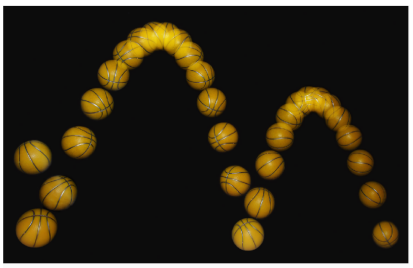


Figure 2: The orange particle at the center is bouncing between the blue particles which approach each other.



Every time the ball hits the surface its kinetic energy is lowered according to Eq.(26). 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 a 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*[1].

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

$$\begin{aligned} t_{\text{tot}} &= \sum_{i=1}^{\infty} t_i \\ &= 2\sqrt{\frac{2h^{\text{initial}}}{g}} \sum_{i=1}^{\infty} \sqrt{r^i} \\ &= 2\sqrt{\frac{2h^{\text{initial}}}{g}} \left(\frac{1}{1-\sqrt{r}} - 1\right). \end{aligned} \tag{34}$$

Luding and McNamara introduced in 1998 a coefficient of restitution that is dependent of the time elapsed since the last event occurred. If the time since the last collision of one of the interacting particles $t^{(i)}$ or $t^{(j)}$ is less than t_{contact}, then the coefficient is set to unity, i.e.,

$$r^{i,j} = \begin{cases} r, & \text{for} \quad t^{(i)} > t_{\text{contact}} \quad \text{or} \quad t^{(j)} > t_{\text{contact}} \\ 1, & \text{otherwise}\,. \end{cases} \tag{35}$$

With this redefinition of the restitution coefficient, the collision type changes from inelastic to elastic if too many collisions occur during t_{contact}.

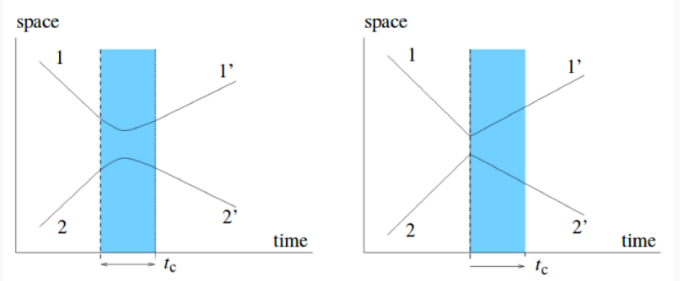


Figure 4: Trajectories of soft (left) and hard (right) particles. The figure

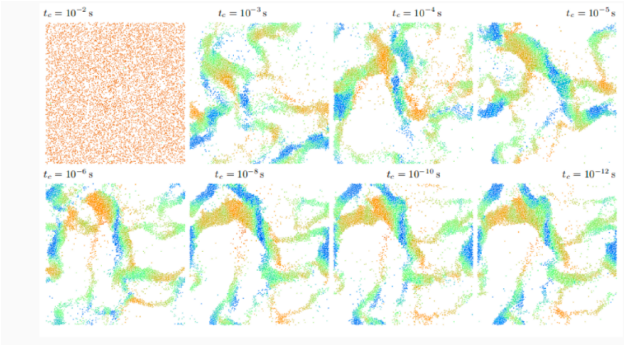


Figure 5: Examples of different contact times t_c. The figure is taken

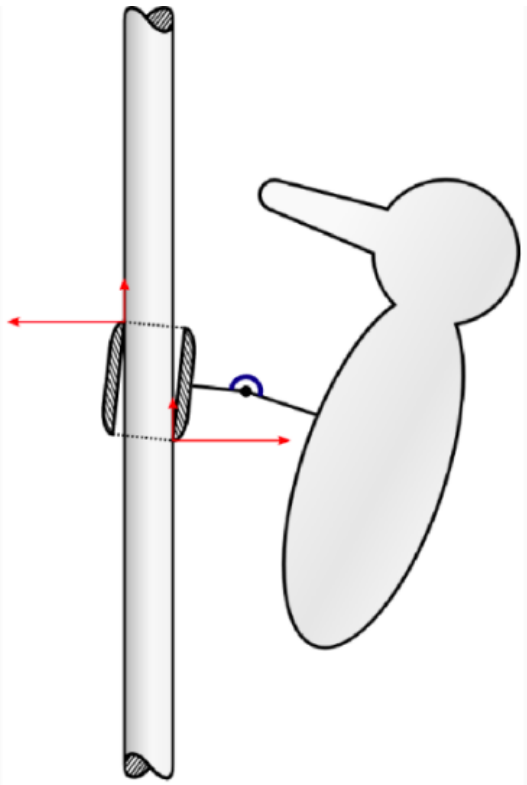


Figure 6: A woodpecker toy as a benchmark example for contact dynamics problems.

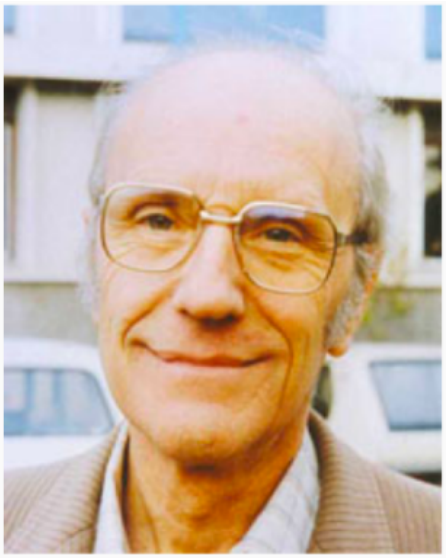


Figure 7: Per Lötstedt and Jean-Jacques Moreau contributed to development of contact dynamics.

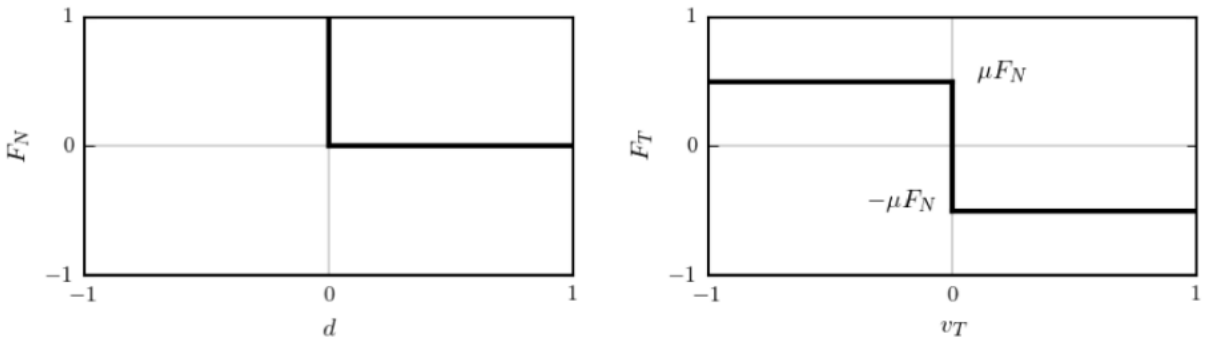


Figure 8: Signorini (left) and Coulomb (right) graphs.

1D Contact

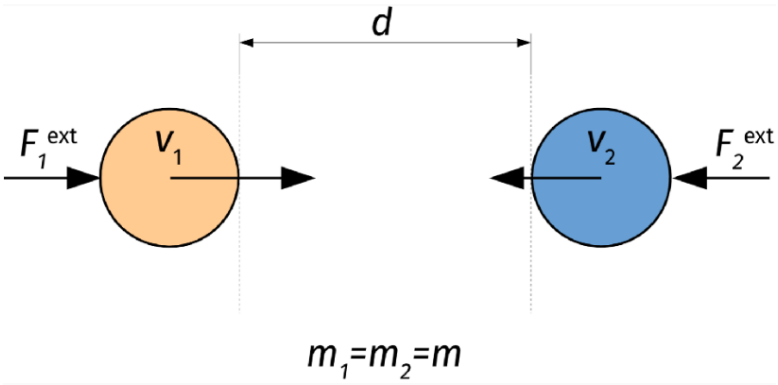


Figure 9: Illustration of a one-dimensional contact.

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. These 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

$$\begin{aligned} \mathbf{v}_i\left(t+\Delta t\right) &= \mathbf{v}_i\left(t\right) + \Delta t \frac{1}{m_i}\mathbf{F}_i\left(t+\Delta t\right), \\ \mathbf{r}_i\left(t+\Delta t\right) &= \mathbf{r}_i\left(t\right) + \Delta t \mathbf{v}_i\left(t+\Delta t\right), \end{aligned} \tag{36}$$

where the force consists of an external and a contact term, i.e., $\mathbf{F}_i\left(t\right) = \mathbf{F}_i^{\text{ext}}\left(t\right) + \mathbf{R}_i\left(t\right)$.

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.

This leads to

$$v_n^{\text{loc}} = v_2 - v_1 = \begin{pmatrix} -1 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \tag{37}$$

and local forces

$$\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} -R_n^{\text{loc}} \\ R_n^{\text{loc}} \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} R_n^{\text{loc}} = H R_n^{\text{loc}}. \tag{38}$$

The equations of motion for both particles are

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \frac{1}{m}\left[\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} + \begin{pmatrix} F_1^{\text{ext}} \\ F_2^{\text{ext}} \end{pmatrix}\right]. \tag{39}$$

Combining the last equation with the transformation rule of Eq. (38) we find

$$\begin{aligned} \frac{\mathrm{d}v_n^{\text{loc}}}{\mathrm{d}t} &= \begin{pmatrix} -1 & 1 \end{pmatrix} \frac{1}{m}\left[\begin{pmatrix} -1 \\ 1 \end{pmatrix} R_n^{\text{loc}} + \begin{pmatrix} F_1^{\text{ext}} \\ F_2^{\text{ext}} \end{pmatrix}\right] \\ &= \frac{1}{m_{\text{eff}}} R_n^{\text{loc}} + \frac{1}{m}\left(F_2^{\text{ext}} - F_1^{\text{ext}}\right), \end{aligned} \tag{40}$$

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

We integrate the last equation with an implicit Euler method and find:

$$\frac{v_n^{\text{loc}}\left(t+\Delta t\right) - v_n^{\text{loc}}\left(t\right)}{\Delta t} = \frac{1}{m_{\text{eff}}} R_n^{\text{loc}}\left(t+\Delta t\right) + \frac{1}{m}\left(F_2^{\text{ext}} - F_1^{\text{ext}}\right). \tag{41}$$

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

$$R_n^{\text{loc}}\left(t+\Delta t\right) = \frac{v_n^{\text{loc}}\left(t+\Delta t\right) - v_n^{\text{loc, free}}\left(t+\Delta t\right)}{\Delta t} \tag{42}$$

with

$$v_n^{\text{loc, free}}\left(t+\Delta t\right) = v_n^{\text{loc}}\left(t\right) + \Delta t \frac{1}{m}\left(F_2^{\text{ext}} - F_1^{\text{ext}}\right). \tag{43}$$

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.

3D Contact

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

$$\mathbf{v}_{12} = \begin{pmatrix} v_{12}^x \\ v_{12}^y \\ v_{12}^z \end{pmatrix} \quad \mathbf{R}_{12} = \begin{pmatrix} R_{12}^x \\ R_{12}^y \\ R_{12}^z \end{pmatrix} \quad \mathbf{F}_{12}^{\text{ext}} = \begin{pmatrix} F_{12}^{x,\text{ext}} \\ F_{12}^{y,\text{ext}} \\ F_{12}^{z,\text{ext}} \end{pmatrix}. \tag{44}$$

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

$$\mathbf{n} = \begin{pmatrix} n^x \\ n^y \\ n^z \end{pmatrix}, \tag{45}$$

and obtain

$$v_n^{loc} = \mathbf{n} \cdot (\mathbf{v}_2 - \mathbf{v}_1) \quad \mathbf{R}_1 = -\mathbf{n} R_n^{loc} \quad \mathbf{R}_2 = \mathbf{n} R_n^{loc}. \tag{46}$$

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

$$v_n^{loc} = H^T \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix}, \quad \begin{pmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{pmatrix} = H R_n^{loc}, \tag{47}$$

with

$$H^T = (-n_x, -n_y, -n_z, n_x, n_y, n_z) \tag{48}$$

Friction can be included by considering angular velocities and torques.

Particles in Fluids

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*

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\left(\nabla \mathbf{u}\right) = -\frac{1}{\rho}\nabla p + \mu \Delta \mathbf{u} \tag{1}$$

The velocity and pressure fields are denoted by $\mathbf{u}\left(\mathbf{x}\right)$ and $p\left(\mathbf{x}\right)$, respectively. In the case of constant density ρ, the continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla\left(\rho \mathbf{u}\right) = 0 \tag{2}$$

yields $\nabla \mathbf{u} = 0$.

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

$$\mathrm{Re} = \frac{uh}{\mu} = \begin{cases} \ll 1 & \text{Stokes limit,} \\ \gg 1 & \text{turbulent flow,} \end{cases} \tag{3}$$

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. 1 and 2. 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 methods
- Discrete simulation methods

Based on the fluid motion described by the Navier-Stokes equations, 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

$$\mathbf{F}_D = \int_\Gamma \Theta \mathrm{d}\mathbf{A}. \tag{4}$$

The stress tensor of the fluid is given by:

$$\Theta_{ij} = -p\delta_{i,j} + \eta\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{5}$$

where $\eta = \rho\mu$ is the static viscosity and p the hydrostatic pressure.

In the Stokes limit for $\mathrm{Re} \ll 1$, the drag law is given by:

$$F_D = 6\pi\eta R u, \tag{6}$$

where η is the viscosity of the fluid, R the radius of the particle, u the velocity of the fluid relative to the particle. The Stokes law is exact for $\mathrm{Re} = 0$. In the case of turbulent flow for $\mathrm{Re} \gg 1$, the drag force is (Newton's law)

$$F_D = 0.22\pi\rho R^2 u^2. \tag{7}$$

The general drag law is

$$F_D = \frac{\pi\eta^2}{8\rho} C_D \mathrm{Re}^2. \tag{8}$$

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.

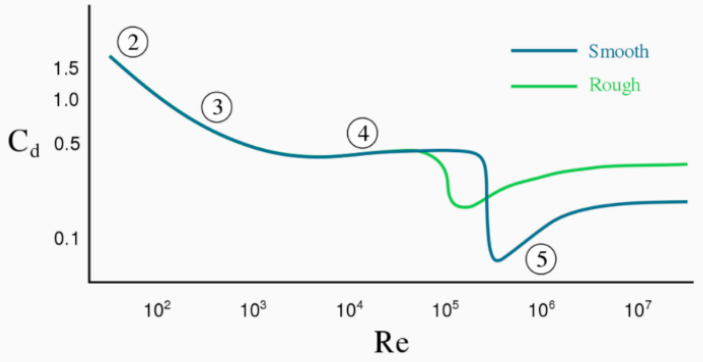


Figure 1: Dependence of the drag coefficient C_D on the Reynold's number.

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

$$F_L = \frac{1}{2} C_L \rho A u^2, \tag{9}$$

where C_L denotes the lift coefficient.

In addition to drag and lift forces, rotating particles experience a torque

$$T = \int_\Gamma \mathbf{r}_{\mathrm{cm}} \wedge \Theta \mathrm{d}\mathbf{A} \tag{10}$$

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

$$F_M = 2\pi R^2 \rho u \omega. \tag{11}$$

There exist empirical relation for drag coefficient in certain Reynold's numbers regimes. For example, one may addopt the following drag coefficient dependence:

$$C_D = \begin{cases} 1 & \mathrm{Re}_\rho < 1000, \\ 0.44 & \mathrm{Re}_\rho \geq 1000, \end{cases} \tag{12}$$

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

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

$$f(x + v_i, v_i, t + 1) - f(x, v_i, t) + F(v_i) = \frac{1}{\tau}[f_i^{\mathsf{eq}} - f(x, v_i, t)] \quad (17)$$

the equilibrium distribution is

$$f_i^{\mathsf{eq}} = n\omega_i \left[1 + \frac{3}{c^2}\mathbf{u}\mathbf{v}_i + \frac{9}{2c^4}(\mathbf{u}\mathbf{v}_i)^2 - \frac{3}{2c^2}\mathbf{u}\mathbf{u}\right]. \quad (18)$$

One possible choice of the weights in two dimensions is

$$\omega_i = \begin{cases} 4/9 & i = 0, \\ 1/9 & i = 1, 2, 3, 4, \\ 1/36 & i = 5, 6, 7, 8. \end{cases} \quad (19)$$

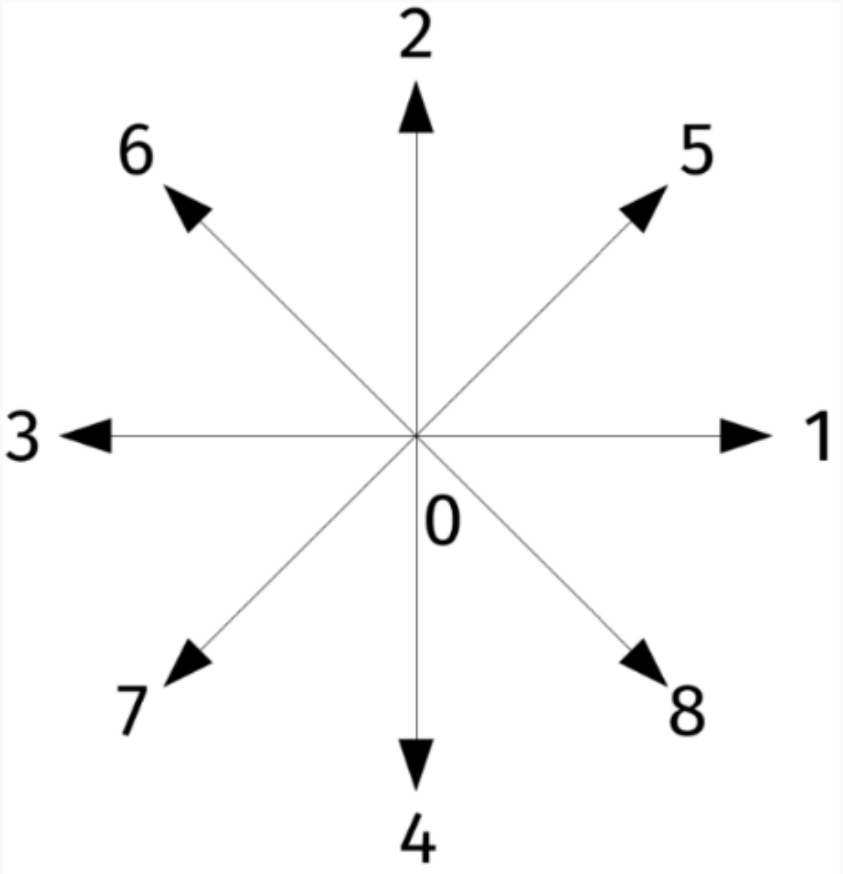


Figure 2: Lattice Boltzmann weights in 2 dimensions

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

$$\mathrm{Kn} = \frac{\lambda}{L} \tag{23}$$

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.

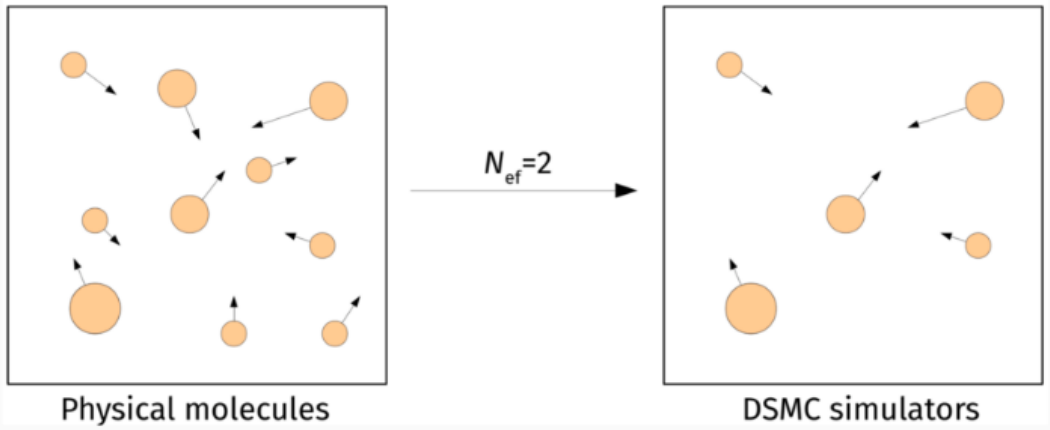
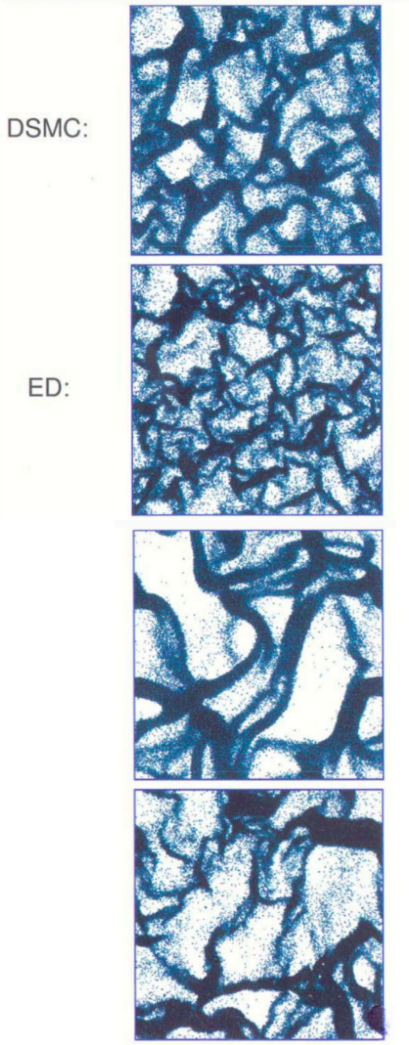


Figure 3: In DSMC N_{ef} simulators represent one physical particle.



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

$$P_{v_x}(v_x) = \pm\frac{m}{k_B T_W} v_x e^{-\frac{m v_x^2}{2k_B T_W}} \tag{24}$$

$$P_{v_y}(v_y) = \sqrt{\frac{m}{2\pi k_B T_W}} e^{-\frac{m(v_y - u_W)^2}{2k_B T_W}} \tag{25}$$

$$P_{v_x}(v_x) = \sqrt{\frac{m}{2\pi k_B T_W}} e^{-\frac{m v_z^2}{2k_B T_W}} \tag{26}$$