1 Introduction

This course aims at providing the student with the ability to write her/his own Monte Carlo and/or Molecular Dynamics code, having mastered the fundamental ideas behind the implemented algorithms. We will limit ourselves to describing the collective behaviour of classical atoms and molecules (particles in general) which propagate in phase-space under the laws of classical mechanics. From this basic knowledge, it is expected that the student can start using the available packages for numerical simulations, even of more complicated systems like polymers and macromolecules of biological interest.

We will assume that the motion of the nuclei can be described by Newton's law. This approximation breaks down clearly for very light atoms (Helium, Proton, Deuterium) and for all vibrational motions whose characteristic frequency, translated in energy, is comparable or larger than the thermal energy k_BT.

Performing numerical simulations, we sample the system phase-space and hence we have access to positions $\mathbf{r}_N$ and momenta $\mathbf{p}_N$ of all atoms in the system. From this information. the microscopic value of all observables which are function of $(\mathbf{r}_N, \mathbf{p}_N)$ can be evaluated and, eventually, averaged over time or phase-space.

Numerical simulations can be considered as a tool for performing numerical experiments. In this respect, simulations can be used for different applications: (i) reproduce an existing experimental result, to complement the macroscopic information with the atomistic level provided by the simulation (ii) to produce a numerical prediction of a gedanken experiment or (iii) for checking theoretical approximations based on model systems.

2 An historical perspective (Tuckerman)

One of the earliest examples of such a numerical thought experiment was the Fermi-Pasta-Ulam calculation (1955), in which the equations of motion for a one-dimensional chain of nonlinear oscillators were integrated numerically in order to quantify the degree of ergodicity and energy equipartitioning in the system. Later, Alder and Wainwright carried out the first condensed-phase molecular dynamics calculation on a hard-sphere system (Alder and Wainwright, 1957; Alder and Wainwright, 1959), showing that a solid-liquid phase transition exists. Following this, Rahman (1964) and Verlet (1967) carried out the first simulations using a realistic continuous potential for systems of 864 argon atoms. The next major milestone came when Berne and coworkers (Harp and Berne, 1968; Berne et al., 1968; Harp and Berne, 1970; Berne, 1971) carried out molecular dynamics simulations of diatomic liquids and characterized the time dependence of molecular reorientation in these systems. Following these studies, Stillinger and Rahman (1971, 1972, 1974) carried out the first molecular dynamics simulations of liquid water. Soon thereafter, Karplus and coworkers reported the first molecular dynamics calculations of proteins (McCammon et al., 1976; McCammon et al., 1977). Explicit treatment of molecular systems was enabled by the introduction of techniques for maintaining specific bonding patterns either by stiff intramolecular forces (Berne and Harp, 1970a) or by imposing holonomic constraints into the simulation (Ryckaert et al., 1977). The evolution of the field of molecular dynamics has benefitted substantially by advances in high-performance computing. The original Alder and Wainwright calculations required the use of a "supercomputer" at Lawrence Livermore National Lab- oratory in California, namely, the UNIVAC system. Nowadays, molecular dynamics calculations with force fields can be carried out on desktop computers. Nevertheless, another major milestone in molecular dynamics, the technique now known as ab initio or first-principles molecular dynamics (Car and Parrinello, 1985), currently requires large-scale high-performance supercomputing resources. In an ab initio molecular dynamics calculation, the interatomic interactions are computed directly from the electronic structure on the fly as the simulation proceeds, thereby allowing chemical bonding breaking and forming events to be treated explicitly. The computational overhead of solving the electronic Schroedinger equation using widely employed ap- proximation schemes is considerable, which is why such calculations demand the use of these resources. The field of molecular dynamics is an exciting and rapidly evolving one, and the immediate availability of free software packages capable of performing many different types of molecular dynamics calculations has dramatically increased the number of users of the methodology.

3 The potentials

Let's start with reviewing the most common inter particle interaction potentials encountered in numerical simulation of simple systems.

The potential $V(\mathbf{r}^N)$ describes the way particles interact among themselved. In general, if we focus on the nuclei position, it should be calculated from the electronic density distribution. In this respect it is a many body term. In quantum calculations, the electronic density distribution is indeed what is considered. In classical simulations one assumes that the effect of the electronic density can be condensed in a known functional dependence, which is taken for granted. In addition, very rarely calculations extent beyond three-body interactions. Indeed, the interction potential can be expanded as

$$V(\mathbf{r}^N) = \sum_{i,j} V_2(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i,j,k} V_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + ...$$

where, defining the relative distance $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$

$$V_2(\mathbf{r}_i, \mathbf{r}_j) = V_2(|\mathbf{r}_{ij}|)$$

and

$$V_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) = V_3(\mathbf{r}_{ij}, \mathbf{r}_{ik}, \mathbf{r}_{jk})$$

Let's take a look at the most commonly used functional forms. In the molecular physics course you have already encountered the Morse potential

$$V(r) = V_\infty + D_e \left\{ [1 - e^{-\alpha(r - r_0)}]^2 - 1 \right\}$$

where r_0 indicates the equilibrium distance, V_∞ the value of the potential energy at infinite distance, $V_\infty - D_e$ the value at $r = r_0$. α controls the curvature of the potential around the minimum.

You have also encountered the mostly used Lennard-Jones (LJ) potential

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right]$$

commonly chosen to model noble gases. ϵ is the energy scale (the depth of the minimum) and σ the lenght scale (the diameter of the particle). The position of the minimum coincides with $r = 2^{1/6}\sigma \approx 1.122\sigma$.

Sometime, to modify the range of interaction the exponents $6 - 12$ of the LJ potential are changed in a generic m, n pair

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{m} - \left(\frac{\sigma}{r}\right)^{n} \right]$$

Another well studied potential is the soft-sphere potential, e.g. the repulsive part of the LJ potential for a generic exponent n

$$V(r) = \epsilon \left(\frac{\sigma}{r}\right)^{n}$$

In soft-spheres the (excess) thermodynamics of the model depend on the temperature T and on the density $\rho^* = \frac{N\sigma^3}{V}$ only via the combination

$$\Gamma = \rho\sigma^3 \left(\frac{\epsilon}{k_B T}\right)^{\frac{3}{n}}$$

Indeed, the partition function of the system requires the integration of

$$Z \sim \int_V d\mathbf{r}_1 d\mathbf{r}_2 .... d\mathbf{r}_N exp[-\beta V(\mathbf{r}_1\mathbf{r}_2....\mathbf{r}_N)]$$

and by defining scaled quantities $\xi \equiv \mathbf{r}V^{-1/3}$

$$Z \sim V^N \int_1 d\xi_1 d\xi_2 .... d\xi_N exp[-\beta\epsilon \frac{\sigma^n}{V^{n/3}} \sum_{ij} V(\xi_{ij})]$$

that can be rewritten as

$$Z \sim V^N \int_1 d\xi_1 d\xi_2 .... d\xi_N exp[- \left(\frac{\Gamma}{N}\right)^{n/3} \sum_{ij} V(\xi_{ij})]$$

In theory and in the colloidal field, very often step-wise potentials are chosen. The most famous is the hard-sphere potential

$$V(r) = \begin{cases} \infty, & \text{if } r \leq \sigma \\ 0, & \text{otherwise} \end{cases}$$

where only the repulsion due to the hard-core is included and the square well potential

$$V(r) = \begin{cases} \infty, & r \leq \sigma \\ -\epsilon, & \sigma \leq r \leq \sigma + \delta \\ 0 & \text{otherwise} \end{cases}$$

As an example of a three-body potential let's consider the Stillinger-Weber potential for network forming liquids. The functional form is

$$V_{triplet} = \sum_{i,j,k} g(r_{ij}) g(r_{ik}) \left(\cos\theta_{jik} + \frac{1}{3} \right)$$

and the torsianal potential in the typical polymer (or ammino acid) chains

$$V(r) = k(\theta - \theta_0)^2$$

In several cases, particles are charged. This adds to the hard-core repulsion and the van der Walls attraction an additional term, modeling the electrostatic potential. In vacuum, indicating with q_i and q_j the charges of the two particles it takes the form (with the proper unit-dependent constant)

$$V(r) = A \frac{q_i q_j}{r}$$

If the two charges are immersed in a salt-solution, the typical effective interaction potential that model the screened electrostatic contribution is named Yukawa potential

$$V(r) = A \frac{q_i q_j e^{-r/\lambda}}{r}$$

where λ is the screening lenght.

For molecules, the interaction potential is usually expressed as a sum over all sites compositing the molecule.

4 The Lennard-Jones potential

Undeniable, the most common potential for simulations is the Lennard-Jones (LJ) potential, a model devised to describe the interaction between the nuclei of noble glasses. It is defined by two parameters, the interaction depth ϵ and the diameter of the atom σ. The functional form is

$$V_{LJ}(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{1}$$

$$\frac{dV_{LJ}(r)}{dr} = 4\epsilon\frac{1}{r}\left[-12\left(\frac{\sigma}{r}\right)^{12} + 6\left(\frac{\sigma}{r}\right)^{6}\right] \tag{2}$$

This functional form shows that atoms with different σ and ϵ behave in the same way if the density is measured in units of σ and the temperature in units of ϵ, implementing at the microscopic level the law of corresponding states.

The LJ potential exactly vanishes at $r = \infty$. Still, assuming that beyond a certain distance r_c the correlation between different atoms have vanished, it is reasonable to include in the force direct sum only pairs of atoms with relative distance smaller than r_c.

The cut-off at r_c generates a discontinuity in the potential which is typically handled with a shift of the entire functional form by $V_{LJ}(r_c)$. In this way one obtains the so-called cut-and-shifted Lennard Jones potential $V_{LJ}^{CS}(r)$

$$V_{LJ}^{CS}(r) = V_{LJ}(r) - V_{LJ}(r_c) \qquad r < r_c \tag{3}$$

$$0 \qquad r > r_c \tag{4}$$

To compare the $V_{LJ}^{CS}(r)$ and the $V_{LJ}(r)$ potential one can add to the numerical results obtained for the $V_{LJ}^{CS}(r)$ the theoretically estimated contribution arising from the presence of the finite radial cut-off. Indeed, consistent with the hypothesis of absence of correlation beyond r_c, the energy contribution per particle associated to all interaction beyond r_c can be evaluated as

$$E^{correction} = \frac{1}{2}\int_{r_c}^{\infty} V_{LJ}(r)\rho 4\pi r^2 dr =$$

$$\frac{1}{2}16\pi\rho\epsilon\int_{r_c}^{\infty} r^2\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] =$$

$$\frac{8}{3}\pi\rho\epsilon\sigma^3\left[\frac{1}{3}\left(\frac{\sigma}{r_c}\right)^{9} - \left(\frac{\sigma}{r_c}\right)^{3}\right]$$

where ρ is the number density. and the factor 1/2 accounts for double counting. A similar correction can be estimated for the pressure.

The $V_{LJ}^{CS}(r)$ potential still presents an inconvenience: the non continuity of the first derivative at r_c. If r_c is rather large, this may not be a problem, being the force at r_c already vanishing. On the other hand, for the often used $r_c = 2.5\sigma$ value, the discontinuity originates a non-conservation of the energy which may introduce a significant drift in the total energy in long-lasting simulations.

To overcome this problem, often people prefer to add an additional linear contribution to the LJ potential which preserves continuity of the potential and of its first derivative at r_c.

$$V_{LJ}^{LM}(r) = V_{LJ}(r) + ar + b \tag{5}$$
$$a = -\frac{dV}{dr}|_{r_c} \tag{6}$$
$$b = -V_{LJ}(r_c) - ar_c \tag{7}$$

Another possibility is offered by the use of switching functions. In this case the original potential is multiplied by a switching function $S(r)$, often defined as

$$S(r) = \begin{cases} 1 & r < r_c - \lambda \\ 1 + R^2(2R-3) & r_c - \lambda < r < r_c + \lambda \\ 0 & r > r_c + \lambda \end{cases} \tag{8}$$

where $R = [r - (r_c - \lambda)]/2\lambda$. $R = 1$ when $r = r_c + \lambda$ and $R = 0$ when $r = r_c - \lambda$. The derivative of this function is

$$\frac{dS(r)}{dr} = \begin{cases} 0 & r < r_c - \lambda \\ 6(R^2 - R)\frac{dR}{dr} & r_c - \lambda < r < r_c + \lambda \\ 0 & r > r_c + \lambda \end{cases} \tag{9}$$

The LJ potential, in all his forms, has been carefully studied and it is now a day a benchmark for testing a new code as well as the reference system for numerically tackling new open problems (glass transition, crystallisation, kinetic of phase separation, gelation).

One note of caution need to be given for people using simple truncation. In this case, two different corrections need to be applied to the pressure. One is consistent with the integrated tail correction to the energy discussed previously

$$\Delta P^{tail} = \frac{1}{2}4\pi\rho^2 \int_{r_c}^{\infty} V_{LJ}(r)\mathbf{r}\cdot\left(-\frac{dV}{d\mathbf{r}}\right) r^2 dr =$$
$$\frac{16}{3}\pi\rho^2\epsilon\sigma^3\left[\frac{2}{3}\left(\frac{\sigma}{r_c}\right)^9 - \left(\frac{\sigma}{r_c}\right)^3\right]$$

On top of this contribution, one need to evaluate the contribution arising from the discontinuity at r_c. This additional contribution should be included even if the long-range part of the energy and of the pressure (the tail) is not included. Indeed the simply cut potential is equivalent to a SW potential of depth $V_{LJ}(r_c)$, added on the cut and shifted potential. To evaluate this contribution, let's consider the definition of pressure

$$P =< \frac{\partial U}{\partial V} >$$

Let us assume we change the volume of a configuration by isotropically scaling all coordinates by ξ. Similarly, all relative distances are scaled by a factor ξ. If $\xi > 1$ (expansion) all pairs which where within a shell of width $r_c(1-1/\xi)$ will cross the cut off and will loose the SW depth energy. The number of these particles will be (assuming a constant density around r_c)

$$M = 4\pi r_c^2[r_c(1-1/\xi)]\rho$$

so that, summing over all particles and dividing by two to avoid over-counting

$$\delta E = -V_{LJ}(r_c)M\frac{N}{2}$$

The corresponding δV is

$$\delta V = (\xi L)^3 - L^3 = V(\xi^3-1) = V(\xi-1)(\xi^2+\xi+1)$$

So that

$$P^{SW} = \lim_{\xi\to 1}\frac{V_{LJ}(r_c)4\pi r_c^2[r_c(1-1/\xi)]\rho(r_c)\frac{N}{2}}{V(\xi-1)(\xi^2+\xi+1)} =$$

$$\lim_{\xi\to 1} V_{LJ}(r_c)4\pi r_c^2\rho(r_c)\frac{N}{2V}\frac{\xi-1}{\xi}\frac{1}{(\xi-1)(\xi^2+\xi+1)} = \lim_{\xi\to 1} V_{LJ}(r_c)4\pi r_c^3\rho(r_c)\frac{N}{2V}\frac{1}{\xi}\frac{1}{(\xi^2+\xi+1)}$$

$$P^{SW} = \frac{8}{3}\pi\epsilon\rho(r_c^-)\rho r_c^3\left[\left(\frac{\sigma}{r_c}\right)^{12} - \left(\frac{\sigma}{r_c}\right)^{6}\right] = \frac{8}{3}\pi\epsilon\rho(r_c^-)\rho\sigma^3\left[\left(\frac{\sigma}{r_c}\right)^{9} - \left(\frac{\sigma}{r_c}\right)^{3}\right]$$

and approximating $\rho(r_c^-)$ (e.g. approaching r_c from the left) with ρ

$$P^{SW} \approx \frac{8}{3}\pi\epsilon\rho^2\sigma^3\left[\left(\frac{\sigma}{r_c}\right)^{9} - \left(\frac{\sigma}{r_c}\right)^{3}\right]$$

5 What do we need

Our goal is, generically, to calculate static (via Monte Carlo or Molecular Dynamics methods) and dynamic quantities (via Molecular Dynamics methods) of a system for which we know the interaction potential $V(\mathbf{r}^N)$.

To calculate static quantities we need to evaluate the probability $\mathcal{N}(\mathbf{r}^N)$ that the system composed by N particles (atoms, not molecules, for the time being) explores the phase-space volume centered around $\mathbf{r}^N$, e.g. in the canonical ensamble (constant N,V,T)

$$\mathcal{N}(\mathbf{r}^N)d\mathbf{r}^N = \frac{1}{N!\lambda^{3N}}\frac{\exp[-\beta V(\mathbf{r}^N)]d\mathbf{r}^N}{Z_N(V,T)}$$

where the configurational part of the partition function (where momenta have been integrated out) is

$$Z_N(V,T) = \frac{1}{N!\lambda^{3N}}\int d\mathbf{r}^N \exp[-\beta\mathcal{U}(\mathbf{r}^N)]$$

6 The basic of Monte Carlo: The Metropolis Method (Frenkel-Smit)

Monte Carlo is a generic method to evaluate integrals. In our case, we would like to use it to evaluate the partition function. Unfortunately it is in general not possible to evaluate an integral, such as $\int \exp[-\beta\mathcal{U}(\mathbf{r}^N)]d\mathbf{r}^N$ by direct Monte Carlo sampling. Very few points with a large Boltzmann weight may control the integral value and it is rather difficult to sample them.

However, in many cases, we are not interested in the configurational part of the partition function itself but in averages of the type

$$< A >= \frac{1}{N!\lambda^{3N}} \frac{\int d\mathbf{r}^N \exp[-\beta\mathcal{U}(\mathbf{r}^N)]A(\mathbf{r}^N)}{Z}$$

Ideally one could partition space in minute hypercubes and transform the integration in a sum over $\#_P$ points as

$$< A >= \frac{\sum_{i=1}^{\#_P} A_i \exp[-\beta\mathcal{U}_i]}{\sum_{i=1}^{\#_P} \exp[-\beta\mathcal{U}_i]}$$

This procedure, while feasible for systems with a very small number of degrees of freedom, it is not conceivable for realistic systems, even with a number of particles significantly smaller than Avogadro's number.

If it is possible to devise scheme for generating point which are distributed proportionally to $\exp[-\beta\mathcal{U}]$ (even if the proportionality constant is not known !), e.g.

$$\mathcal{N}(\mathbf{r}^N) \sim \exp[-\beta\mathcal{U}(\mathbf{r}^N)]$$

then one can rewrite the integral

$$< A >= \frac{\int d\mathbf{r}^N \mathcal{N}(\mathbf{r}^N)A(\mathbf{r}^N)}{\int d\mathbf{r}^N \mathcal{N}(\mathbf{r}^N)}$$

and discretize it as

$$< A >= \frac{\sum_{i=1}^{\#_P} A_i\mathcal{N}_i}{\sum_{i=1}^{\#_P} \mathcal{N}_i} = \frac{1}{\#_P}\sum_{i=1}^{\#_P} A_i\mathcal{N}_i$$

Note that in the previous equation we know only $\exp[-\beta\mathcal{U}(\mathbf{r}^N)]$ that is, we know only the relative but not the absolute probability of visiting different points in configuration space.

Let us next consider how to generate points in configuration space with a relative probability proportional to the Boltzmann factor. The general approach is first to prepare the system in a configuration $\mathbf{r}^N$, which we denote by o (old), that has a nonvanishing Boltzmann factor $\exp[-\beta\mathcal{U}(o)]$. This configuration, for example, may correspond to a regular crystalline lattice with no hard-core overlaps. Next, we generate a new trial configuration $\mathbf{r'}^N$ which we denote by n (new), by adding a small random displacement Δ to o. The Boltzmann factor of this trial configuration is $\exp[-\beta\mathcal{U}(n)]$ We must now decide whether we will accept or reject the trial configuration. Many rules for making this decision satisfy the constraint that on average the probability of finding the system in a configuration n is proportional to $\mathcal{N}(n)$. Here we discuss only the Metropolis scheme, because it is simple and generally applicable. Let us now "derive" the Metropolis scheme to determine the transition probability $\pi(o \to n)$ to go from configuration o to n. It is convenient to start with a thought experiment (actually a thought simulation). We carry out a very large number (say M) Monte Carlo simulations in parallel, where M is much larger than the total number of accessible configurations. We denote the number of points in any configuration o by $m(o)$. We wish that, on average $m(o)$ is proportional to $\mathcal{N}(o)$. The matrix elements $\pi(o \to n)$ must satisfy one obvious condition: they do not destroy such an equilibrium distribution once it is reached. This means that, in equilibrium, the average number of accepted trial moves that result in the system leaving state o must be exactly equal to the number of accepted trial moves from all other states to state o (the balance condition). It is convenient to impose a much stronger condition; namely, that in equilibrium the average number of accepted moves from o to any other state n is exactly canceled by the number of reverse moves. This detailed balance condition implies the following:

$$\mathcal{N}(o)\pi(o \to n) = \mathcal{N}(n)\pi(n \to o)$$

Many possible forms of the transition matrix $\pi(o \to n)$ satisfy the previous equation Let us look how $\pi(o \to n)$ is constructed in practice. We recall that a Monte Carlo move consists of two stages. First, we perform a trial move from state o to state n. We denote the transition matrix that determines the probability of performing a trial move from o to n by $\alpha(o \to n)$, where α is usually referred to as the underlying matrix of the Markov chain [47]. The next stage is the decision to either accept or reject this trial move. Let us denote the probability of accepting a trial move from o to n by $acc(o \to n)$. Clearly,

$$\pi(o \to n) = \alpha(o \to n) \times acc(o \to n)$$

In the original Metropolis scheme, α is chosen to be a symmetric matrix $\alpha(o \to n) = \alpha(n \to o)$. However, in later sections we shall see several examples where α is not symmetric. If α is symmetric, we can rewrite the previous equation in terms of the $acc(o \to n)$

$$\mathcal{N}(o) \times acc(o \to n) = \mathcal{N}(n) \times acc(n \to o)$$

such that

$$\frac{acc(o \to n)}{acc(n \to o)} = \frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \exp(-\beta[\mathcal{U}(n) - \mathcal{U}(o)]) \tag{10}$$

Again, many choices for $acc(o \to n)$ satisfy this condition (and the obvious condition that the probability $acc(o \to n)$ cannot exceed 1). The choice of Metropolis et al. is

$$
\begin{aligned}
acc(o \to n) &= \mathcal{N}(n)/\mathcal{N}(o) \quad && \text{if} \quad \mathcal{N}(n) < \mathcal{N}(o) \\
acc(o \to n) &= 1 && \text{if} \quad \mathcal{N}(n) \geq \mathcal{N}(o)
\end{aligned}
$$

To prove that this choice satisfies Eq. 10 let's consider that

$$acc(o \to n) = \begin{cases} 1 & E_n \leq E_o \\ e^{-\beta(E_n - E_o)} & E_n > E_o \end{cases}$$

while

$$acc(n \to o) = \begin{cases} 1 & E_o \leq E_n \\ e^{-\beta(E_o - E_n)} & E_o > E_n \end{cases}$$

Thus, if $E_n \leq E_o$

$$\frac{acc(o \to n)}{acc(n \to o)} = \frac{1}{e^{-\beta(E_o - E_n)}}$$

while if $E_n > E_o$

$$\frac{acc(o \to n)}{acc(n \to o)} = \frac{e^{-\beta(E_n - E_o)}}{1}$$

In both cases

$$\frac{acc(o \to n)}{acc(n \to o)} = \frac{\mathcal{N}(n)}{\mathcal{N}(o)}$$

Other choices for $acc(o \to n)$ are possible but the original choice of Metropolis et al. appears to result in a more efficient sampling of configuration space than most other strategies that have been proposed. In summary, then, in the Metropolis scheme, the transition probability for going from state o to state n is given by

$$
\begin{aligned}
\pi(o \to n) &= \alpha(o \to n) \quad && \text{if} \quad \mathcal{N}(n) \geq \mathcal{N}(o) \\
\pi(o \to n) &= \alpha(o \to n)\mathcal{N}(n)/\mathcal{N}(o) && \text{if} \quad \mathcal{N}(n) < \mathcal{N}(o)
\end{aligned}
$$

and the probability from n to o is one minus the probability to go to any state different from o

$$\pi(n \to o) = 1 - \sum_{n \neq o} \pi(o \to n)$$

Note that we still have not specified the matrix acc, except for the fact that it must be symmetric. This reflects considerable freedom in the choice of our trial moves. We will come back to this point in subsequent sections. One thing that we have not yet explained is how to decide whether a trial move is to be accepted or rejected. The usual procedure is as follows. Suppose that we have generated a trial move from state o to state n, with $\mathcal{U}(n) > \mathcal{U}(o)$. This trial move should be accepted with a probability

$$acc(o \rightarrow n) = exp(-\beta[\mathcal{U}(n) - \mathcal{U}(o)] < 1$$

In order to decide whether to accept or reject the trial move, we generate a random number, denoted by Ranf, from a uniform distribution in the interval [0, 1]. Clearly, the probability that Ranf is less than $acc(o \rightarrow n)$ is equal to $acc(o \rightarrow n)$. We now accept the trial move if Ranf $<$ $acc(o \rightarrow n)$ and reject it otherwise. This rule guarantees that the probability to accept a trial move from o to n is indeed equal to $acc(o \rightarrow n)$. Obviously, it is very important that our random-number generator does indeed generate numbers uniformly in the interval [0, 1]. Otherwise the Monte Carlo sampling will be biased. The quality of random-number generators should never be taken for granted. A good discussion of random-number generators can be found in Numerical Recipes and in Monte Carlo Methods by Kalos and Whitlock. Thus far, we have not mentioned another condition that $\pi(o \rightarrow n)$ should satisfy, namely that it is ergodic (i.e., every accessible point in configuration space can be reached in a finite number of Monte Carlo steps from any other point). Although some simple MC schemes are guaranteed to be ergodic, these are often not the most efficient schemes. Conversely, many efficient Monte Carlo schemes have either not been proven to be ergodic or, worse, been proven to be nonergodic. The solution is usually to mix the efficient, nonergodic scheme with an occasional trial move of the less-efficient but ergodic scheme. The method as a whole will then be ergodic (at least, in principle). At this point, we should stress that, in the present book, we focus on Monte Carlo methods to model phenomena that do not depend on time. In the literature one can also find dynamic Monte Carlo schemes. In such dynamic algorithms, Monte Carlo methods are used to generate a numerical solution of the master equation that is supposed to describe the time evolution of the system under study. These dynamic techniques fall outside the scope of this book. The reader interested in dynamic MC schemes is referred to the relevant literature, for example Ref. [48] and references therein.

6.1 The code

In the approach introduced by Metropolis et al. [6], the following scheme is proposed:

- 1. Select a particle at random, and calculate its energy $\mathcal{U}(o)$
- 2. Give the particle a random displacement, $\mathbf{r}' = \mathbf{r} + \mathbf{\Delta}$ and calculate its new energy $\mathcal{U}(n)$.
- 3. Accept the move from o to n with probability $acc(o \rightarrow n) = min(1, exp[-\beta(\mathcal{U}(n) - \mathcal{U}(o))]$

We start our discussion with trial moves of the molecular centers of mass. A perfectly acceptable method for creating a trial displacement is to add random numbers between $-\Delta/2$ and $\Delta/2$ to the x, y and z coordinates of the molecular center of mass:

$$\begin{aligned} x_i' &= x_i + \Delta(Ranf - 0.5) \\ y_i' &= y_i + \Delta(Ranf - 0.5) \\ z_i' &= z_i + \Delta(Ranf - 0.5) \end{aligned} \tag{11}$$

where Ranf are random numbers uniformly distributed between 0 and 1. Clearly, the reverse trial move is equally probable (hence, α is symmetric).

We are now faced with two questions: how large should we choose Δ? and should we attempt to move all particles simultaneously or one at a time? In the latter case we should pick the molecule that is to be moved at random to ensure that the underlying Markov chain remains symmetric.

All other things being equal, we should choose the most efficient sampling procedure. But, to this end, we must first define what we mean by efficient sampling. In very vague terms, sampling is efficient if it gives you good value for money. Good value in a simulation corresponds to high statistical accuracy, and "money" is simply money: the money that buys your computer time and even your own time. For the sake of the argument, we assume the average scientific programmer is poorly paid. In that case we have to worry only about your computer budget. Then we could use the following definition of an optimal sampling scheme: a Monte Carlo sampling scheme can be considered optimal if it yields the lowest statistical error in the quantity to be computed for a given expenditure of computing budget. Usually, computing budget is equivalent to CPU time. From this definition it is clear that, in principle, a sampling scheme may be optimal for one quantity but not for another. Actually, the preceding definition is all but useless in practice (as are most definitions). For instance, it is just not worth the effort to measure the error estimate in the pressure for a number of different Monte Carlo sampling schemes in a series of runs of fixed length. However, it is reasonable to assume that the mean-square error in the observables is inversely proportional to the number of uncorrelated configurations visited in a given amount of CPU time. And the number of independent configurations visited is a measure for the distance covered in configuration space. This suggests a more manageable, albeit rather ad hoc, criterion to estimate the efficiency of a Monte Carlo sampling scheme: the sum of the squares of all accepted trial displacements divided by computing time. This quantity should be distinguished from the mean-squared displacement per unit of computing time, because the latter quantity goes to 0 in the absence of diffusion (e.g., in a solid or a glass), whereas the former does not. Using this criterion it is easy to show that for simulations of condensed phases it is usually advisable to perform random displacements of one particle at a time (as we shall see later, the situation is different for correlated displacements). In brief, if you move all particles, there is a non negligible probablility that one of the particles will contribute to an energy change significantly larger than $k_B T$, determining a rejection of all displacements.

Next, consider the choice of the parameter Δ which determines the size of the trial move. How large should Δ be? If it is very large, it is likely that the resulting configuration will have a high energy and the trial move will probably be rejected. If it is very small, the change in potential energy is probably small and most moves will be accepted. In the literature, one often finds the mysterious statement that an acceptance of approximately 50% should be optimal. This statement is not necessarily true. The optimum acceptance ratio is the one that leads to the most efficient sampling of configuration space. If we express efficiency as mean-squared displacement per CPU time, it is easy to see that different Monte Carlo codes will have different optimal acceptance ratios. The reason is that it makes a crucial difference whether the amount of computing required to test whether a trial move is accepted depends on the magnitude of the move (see Figure 3.4). In the conventional Metropolis scheme, all continuous interactions have to be computed before a move can be accepted or rejected. Hence, for continuous potentials, the amount of computation does not depend on the size of a trial move. In contrast, for simulations of molecules with hard repulsive cores, a move can be rejected as soon as overlap with any neighbor is detected. In that case, a rejected move is cheaper than an accepted one, and hence the average computing time per trial move goes down as the step size is increased. As a result, the optimal acceptance ratio for hard-core systems is appreciably lower than for systems with continuous interactions. Exactly how much depends on the nature of the program, in particular on whether it is a scalar or a vector code (in the latter case, hard-core systems are treated much like continuous systems), on how the information about neighbor lists is stored, and even on the computational "cost" of random numbers and exponentiation. The consensus seems to be that for hard-core systems the optimum acceptance ratio is closer to 20 than to 50%, but this is just another rule of thumb that should be checked? A distinct disadvantage of the efficiency criterion discussed previously is that it does not allow us to detect whether the sampling of configuration space is ergodic. To take a specific example, suppose that our system consists of a number of particles that are trapped in different potential energy minima. Clearly, we can sample the vicinity of these minima quite well and still have totally inadequate sampling of the whole of the configuration space.

If we are simulating molecules rather than atoms we must also generate trial moves that change the molecular orientation. As we discussed already, it almost requires an effort for generating translational trial moves with a distribution that does not satisfy the symmetry requirement of the underlying Markov chain. For rotational moves, the situation is very different. It is only too easy to introduce a systematic bias in the orientational distribution function of the molecules by using a nonsymmetrical orientational sampling scheme. Several different strategies to generate rotational displacements are discussed in [19].

7 Generating the starting configuration

There are several alternatives to generate the starting configuration. If the system is disordered (fluid) one can generate N random position $\mathbf{r}_i$ (inside the simulation box) (and later on N random velocities $\mathbf{v}_i$,with some arbitrary average value). This is equivalent to progressively fill the simulation box of particles. This simple method suffers from the possibility to generate pairs of random coordinates which are so close to each other to give rise to extremely large energies. In the case of hard-core potentials, this simple method generates unphysical overlaps, that need to be resolved. An improvement of the method requires a calculation of the energy every time a particle is added in the box, disregarding all insertion that result in an increase of the system energy beyond a previously defined threshold.

In the cae of hard-core interactions, this improved method allows one to generate disordered configurations when the required density of the system is not particularly large. If one is interested in liquid like densities, it becomes convenient to generate a (cubic) crystalline starting configuration fixing the lattice distance to match the desired density. The number of particles can be conveniently selected as the number of unit cells times their occupancy. Again, depending on the required density, one can generate a cubic lattice (lower density), body-centered-cubic (intermediate) or a face-centered-cubic (higher density) lattice. In this case, the generation of the initial coordinates must be followed by a simulation at high temperature such that all particles have lost memory of their ordered state. This typically correspond to a mean-square displacement of the order of the nearest neighbour distance (squared).

In the case of MD, we will see that it is also important to check, after all momenta have been generated, that the total momentum of the system is strictly zero, to avoid artificial drift of the coordinates and a systematic error in the evaluation of the temperature. When building a crystal, the position of the atoms are given by

$$\mathbf{r} = l\mathbf{v_1} + m\mathbf{v_2} + n\mathbf{v_3}$$

where l, m, n are integer numbers and $\mathbf{v_1}$, $\mathbf{v_2}$ and $\mathbf{v_3}$ are the primitive vectors defining the unit cell. If the base contains more than one particle, for each l, m, n one need to generate all the base particles.

7.1 SC

In the case of the simple cubic (SC), the three primitive vectors are $\mathbf{v_1} = (a, 0, 0)$, $\mathbf{v_2} = (0, a, 0)$ and $\mathbf{v_3} = (0, 0, a)$. The particles can be inserted in the box of side $L = na$, where a is the lattice distance and $N = n^3$ according to the following algorithm

```
np=0
for i=1,n
for j=1,n
for k=1,n
np=np+1
x(np)=i*a
y(np)=j*a
z(np)=k*a
next k
next j
next i
```

The resulting number density is $\rho = n^3/L^3 = a^{-3}$. The nearest neighbour closest distance is a and hence the packing fraction of touching spheres (e.g. of diameter a) on the SC lattice is

$$\phi = \frac{\pi}{6} a^3 \frac{1}{a^3} = \frac{\pi}{6} = 0.52$$

7.2 BCC

In the case of a body centred cubic (BCC) lattice, the three primitive vector of the lattice are $\mathbf{v}_1 = (a/2, a/2, -a/2)$, $\mathbf{v}_2 = (-a/2, a/2, a/2)$ and $\mathbf{v}_3 = (a/2, -a/2, a/2)$. The volume of the unit cell is $V = |\mathbf{v}_1 \cdot \mathbf{v}_2 \times \mathbf{v}_3| = a^3/2$ and hence the number density $2a^{-3}$. To exploit the cubic symmetry it is easier to view the crystal as a simple cubic structure of side a but with two atoms in the unit cell. The location of the two atoms is $\mathbf{a}_1 = (0, 0, 0)$, $\mathbf{a}_2 = (a/2, a/2, a/2)$. The number density is, as before, $2/a^3$. The nearest neighbour closest distance is $\sqrt{3}a/2$ (the distance between the corner and the centre of the cube) and hence the packing fraction of touching spheres on the BCC lattice is

$$\phi = \frac{\pi}{6} \left(\frac{\sqrt{3}a}{2} \right)^3 \frac{2}{a^3} = \frac{\sqrt{3}}{8} \pi = 0.68$$

```
np=0
for i=1,n
for j=1,n
for k=1,n
for l=1,2
np=np+1
x(np)=i*a + (a_l)_x
y(np)=j*a + (a_l)_y
z(np)=k*a + (a_l)_z
next l
next k
next j
next i
```

7.3 FCC

In the case of a FCC lattice, the three primitive vector of the lattice are $\mathbf{v}_1 = (a/2, a/2, 0)$, $\mathbf{v}_2 = (0, a/2, a/2)$ and $\mathbf{v}_3 = (a/2, 0, a/2)$. The volume of the unit cell is $V = |\mathbf{v}_1 \cdot \mathbf{v}_2 \times \mathbf{v}_3| = a^3/4$ and hence the number density $4a^{-3}$. To exploit the cubic symmetry it is easier to view the crystal as a simple cubic structure of side a but with four atoms in the unit cell. The location of the four atoms is $\mathbf{a}_1 = (0, 0, 0)$, $\mathbf{a}_2 = (a/2, a/2, 0)$, $\mathbf{a}_3 = (0, a/2, a/2)$ and $\mathbf{a}_4 = (a/2, 0, a/2)$. The density is, as before, $4/a^3$.

```
np=0
for i=1,n
for j=1,n
for k=1,n
for l=1,4
np=np+1
x(np)=i*a + (a_l)_x
y(np)=j*a + (a_l)_y
z(np)=k*a + (a_l)_z
next l
next k
next j
next i
```

The number of particles $N = 4n^3$, where n^3 is the number of cubic cell composing the simulation box. We note that, in the case of FCC, the closest distance between two lattice points is the diagonal of the square of side a. Hence, if we would convert the number density to a packing fraction, assuming the crystal is composed by touching spheres, we would get

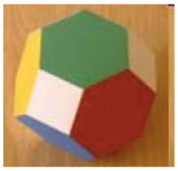
$$\phi = \frac{\pi}{6}\left(\frac{a}{\sqrt{2}}\right)^3 \frac{4}{a^3} = \frac{\sqrt{2}}{6}\pi = 0.7404$$

8 Periodic Boundary Conditions (PBC)

Small systems suffer from size effects. In a spherical cluster, the number of surface particles indeed scales with $N^{2/3}$ and hence the ratio surface to bulk particles goes as $N^{-1/3}$. For $N \approx 100$, 22 per cent of the particles are surface particle. To ease the problem, simulations implement the so-called periodic boundary conditions (PBC), by replicating the simulation box in space along the three directions, generating an infinite system of identical boxes. The shape of a periodic simulation cell must fill all space by translational operations of the central box in 3D and hence only a few geometries are possible:

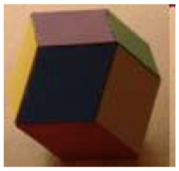
Cube

Truncated Octahedron

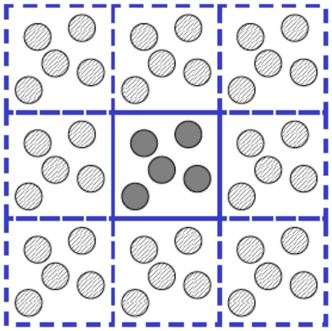


Hexagonal Prism

Rhombic Dodecahedron



The primary cell is thus replicated in all simulated direction as image cells. Primary and image cells have the same number, position, momentum of atoms, same size and same shape.



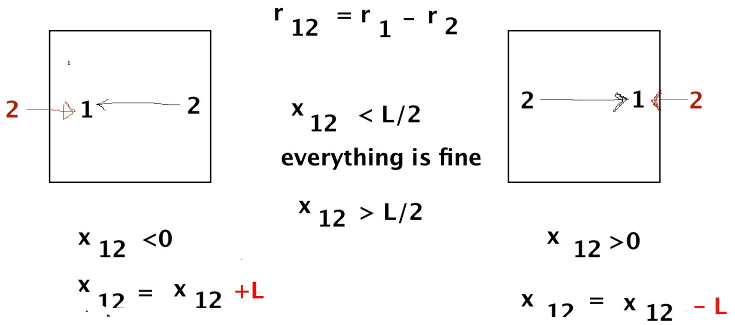
The periodic boundary conditions implies that a particle, crossing the box boundaries, re-enters from the opposite side. When the interaction potential has a radial cut-off (e.g. vanishes beyond a radius r_c), PBC require the side L of the simulation box be at least twice the interaction range, so that, within a sphere of radius r_c no copies of the same particle are present. Indeed, atom i (on the primary cell) can experience a force from atom j in the primary cell as well as from atom j in any of the image cell (26 are the closest in the cubic geometry) but only one image is a distance less than $L/2$. Thus if the pair potential is truncated at $r_c < L/2$ either atom j or only one of its images can exert force on atom i. For this reason, with PBCs, interactions are truncated at least to this minimum image distance.

The implementation of PBC depends if the code assumes that the position of the particles follows the trajectory or if the trajectory of each particles is constantly folded back in the simulation box (and if the box is defined between 0 and L or between $-L/2$ and $L/2$, being L the side of the simulation cube) In both cases, the evaluation of the distance between two particles requires considering PBC.

In the following we assume that all coordinates are always within the original box ($0 < x_i < L$). Under this hypothesis, the minimum distance between two particles can be calculated as

```
r_{ij}=r_i - r_j
d = L*int(r_{ij}/(L/2))  ~~~or‘~~~ d= L*nint(r_{ij}/L)
r_{ij}=r_{ij}-d
```

Inspection of the code shows that if the distance is less than $L/2$, $d = 0$ and the distance is unmodified. If the distance is larger than $L/2$ then $d = \pm 1$ depending on the sign of r_{ij}. In this case $r_{ij} = r_{ij} \mp L$.



9 Calculating the energy

The basic MC code thus requires the evaluation of the system energy before and after the displacement. In the case of pair-wise addititive interactions, it is sufficient to calculate the interaction potential of the randomly selected particle i before and after the displacement. In principle, the energy before is already know (if information are properly stored). To calculate the energy of a particle i

$$E_i = \sum_{j=1}^{N} V_2(r_{ij})$$

e.g. N evaluation of distances.

10 Neighbour list

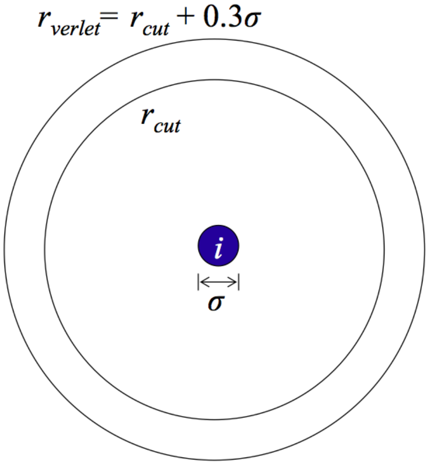
The energy or force calculation is the most time-consuming part of almost all Molecular Dynamics and Monte Carlo simulations. If we consider a model system with pairwise additive interactions (as is done in many molecular simulations), we have to consider the contribution to the force on particle i by all its neighbors. If we do not truncate the interactions, this implies that, for a system of N particles, we must evaluate N(N-1)/2 pair interactions. And even if we do truncate the potential, we still would have to compute all N(N-1)/2 pair distances to describe which pairs can interact. This implies that, if we use no tricks, the time needed for the evaluation of the energy scales as N^2. There exist efficient techniques for speeding up the evaluation of both short-range and long-range interactions in such a way that the computing time scales as $N^{3/2}$, rather than N^2. The techniques for the long-range interactions will be discussed in due time. Here, we discuss some of the techniques used for the short-range interactions. These techniques are:

- 1. Verlet list

- 2. Cell (or linked) list
- 3. Combination of Verlet and cell lists

10.1 Verlet List

If we simulate a large system and use a cutoff that is smaller than the simulation box, many particles do not contribute to the energy of a particle i. It is advantageous therefore to exclude the particles that do not interact from the expensive energy calculation. Verlet [13] developed a bookkeeping technique, commonly referred to as the Verlet list or neighbor list. In this method a second cutoff radius $r_\nu > r_c$ is introduced.

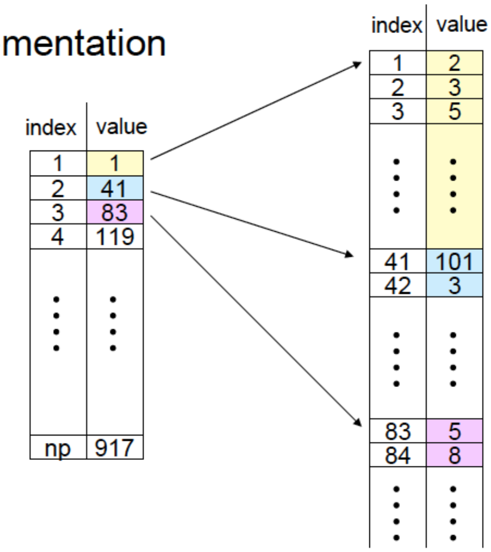


- □ LJ fluid with $\rho\sigma^3 = 0.8$
- □ Typically ~75 atoms with $r_{ij} < r_{verlet}$
- □ Need only track $j > i$
- □ So on average the neighbor list of atom i contains ~40 atoms

Before we calculate the interactions, a list is made (the Verlet list) of all particles within a radius r_ν of particle i and the position of all particles at the time of the calculation of the list are saved. In the subsequent calculation of the interactions, only those particles in this list have to be considered.

An efficient way to handle the list of the interactions without having to create a large matrix is to prepare a list of all interactions and keep track of the initial and final position in this list of the neightbour of particle i, as shown in the figure.

Simple implementation



index	value
1	1
2	41
3	83
4	119
⋮	⋮
np	917

index	value
1	2
2	3
3	5
⋮	⋮
41	101
42	3
⋮	⋮
83	5
84	8
⋮	⋮

Until now we have not saved any CPU time. We gain such time when we next calculate the interactions; if the maximum displacement of the particles is less than $r - r_c$, then we have to consider only the particles in the Verlet list of particle i. This is a calculation of order N. As soon as any of the particles is displaced more than $(r_\nu - r_c)/2$, we have to update the Verlet list. The latter operation is of order N^2, and although this step is not performed each time an interaction is calculated, it will dominate for a very large number of particles.

The Verlet list can be used for both Molecular Dynamics and Monte Carlo simulations. However, there are some small differences in the implementation. For example, in a Molecular Dynamics simulation, the force on all particles is calculated at the same time. It is sufficient therefore to have a Verlet list with half the number of particles for each particle as long as the interaction i-j is accounted for in either the list of particle i, or that of j. In a Monte Carlo simulation each particle is considered separately, therefore it is convenient to have for each particle the complete list. To calculate the force or potential energy of particle i, one has to locate the nearest image of the particles in the Verlet list of particle j.

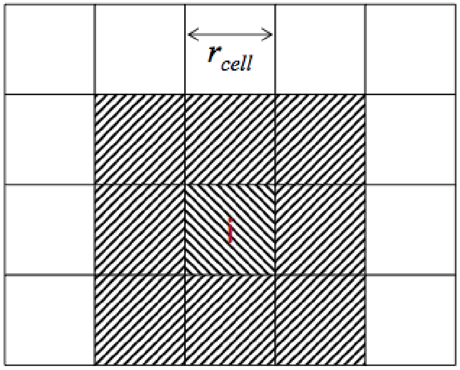
In summary, in the MC code we will need

```
select the particle to move
calculate its energy
move the particle
calculate its energy
accept/reject the move
```

```
if accept check if  Verlet list need to be updated (if so, update it).
```

10.2 Linked Cell

An algorithm that scales with N is the cell list or linked-list method [24]. The idea of the cell list is illustrated in Figure.



The simulation box is divided into cells with a size equal to or slightly larger than the cutoff radius r_c; each particle in a given cell interacts with only those particles in the same or neighboring cells. Since the allocation of a particle to a cell is an operation that scales with N and the total number of cells that needs to be considered for the calculation of the interaction is independent of the system size, the cell list method scales as N.

In the MC code we will need

```
select the particle to move
calculate its energy
move the particle
calculate its energy
accept/reject the move
if accept verify if the particle has changed cell, if so update cell list
```

At the beginning, one need to define the cells (for a cubic box, the optimal number of cells is $[int(L/r_c)]^3$) and for each cell make a list of all particles included in the cell. During the energy calculation of i , one need to evaluate all the interactions with particles in the same cella as i as well as with the 26 connected cells (a ”topology” list which needs to be prepared at the beginning and that does not change during the simulation) One also need to write a subroutine to eliminate a particle from a cell list and a subroutine to add a particle in a cell list.

10.3 Combining Verlet and linked cells

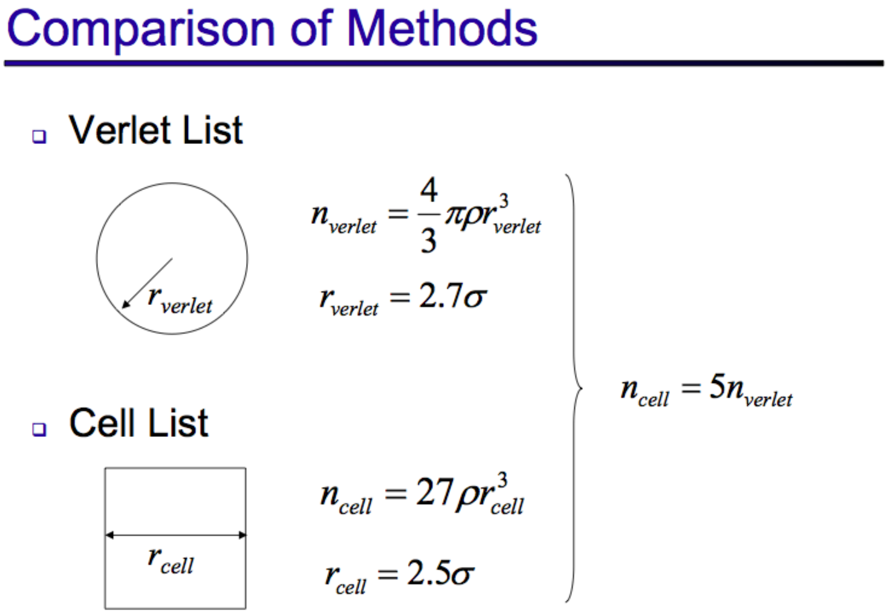
It is instructive to compare the efficiency of the Verlet list and cell list in more detail. In the Verlet list the number of particles for which the distance needs to be calculated is in three dimensions, given by

$$n_\nu = \frac{4}{3}\pi\rho r_\nu^3$$

for the cell list the corresponding number is

$$n_l = 27\rho r_c^3$$

If we use typical values for the parameters in these equations (Lennard-Jones potential with $r_c = 2.5\sigma$ and $r_\nu = 2.7\sigma$), we find that n_l is five times larger than n_ν.



Comparison of Methods

- Verlet List

$$n_{verlet} = \frac{4}{3}\pi\rho r_{verlet}^3$$

$$r_{verlet} = 2.7\sigma$$

- Cell List

$$n_{cell} = 27\rho r_{cell}^3$$

$$r_{cell} = 2.5\sigma$$

$$n_{cell} = 5n_{verlet}$$

As a consequence, in the Verlet scheme, the number of pair distances that needs to be calculated is 16 times less than in the cell list. The observation that the Verlet scheme is more efficient in evaluating the interactions motivated Auerbach et al. [531] to use a combination of the two lists: use a cell list to construct a Verlet list. The use of the cell list removes the main disadvantage of the Verlet list for a large number of particles scales as N^2–but keeps the advantage of an efficient energy calculation.

11 mixtures of particles with different interaction ranges

In this case, optimization can be favoured by combining different verlet lists for the different type of interactions.

12 E before

Computer time can be saved in a MC step if the energy of the selected particle i is calculated ONLY after the random displacement. For this to be possible one need to keep track of all interactions of each particles. In this way, when particle i is randomly selected, E_i can be simply calculated by summing all interactions of i. When a move is accepted, then one need not only to re-generate the list of interaction of i but also to update all previous interactions of particle j with i, for all j. Naturally, this list of interaction need to be created at the beginning of the simulation.

12.1 Linear Molecules

Consider a system consisting of N linear molecules. We specify the orientation of the i-th molecule by a unit vector $\mathbf{u}_i$ One possible procedure to change $\mathbf{u}_i$ by a small, random amount is the following. First, we generate a unit vector $\mathbf{v}$ with a random orientation. This is quite easy to achieve

12.1.1 Random vectors on the surface of a sphere

There are a number of suitable methods for generating a vector on the surface of a unit sphere. Marsaglia [1972] has suggested an interesting method:

- (a) Generate two uniform random numbers ξ_1 and ξ_2 between zero and one.
- (b) Trasform $\xi_1 = 1 - 2\xi_1$ and $\xi_2 = 1 - 2\xi_2$
- (c) Form the sum $\xi^2 = \xi_1^2 + \xi_2^2$
- (d) If $\xi < 1$ than calculate the vector $\vec{\xi} = (2\xi_1\sqrt{1-\xi^2}, 2\xi_2\sqrt{1-\xi^2}, 1-2\xi^2)$
- (e) If $\xi > 1$ reject and return to step (a).

This method requires on average 2.55 uniform variates and a square root. The method can be readily extended to choosing points on a four-sphere and Marsaglia gives an appropriate algorithm. To obtain random vectors in a plane normal to a given unit vector $\vec{e}$, simply subtract that part of $\vec{\xi}$ parallel to $\vec{e}$, $\vec{\xi} - \vec{\xi} \cdot \vec{e}$ and renormalize.

12.2 ... cont...

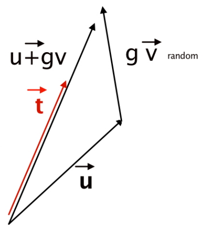
Next we multiply this random unit vector $\mathbf{v}$ by a scale factor γ.

The magnitude of γ determines the magnitude of the trial rotation. We now add $\gamma\mathbf{v}$ to $\mathbf{u}_i$. Let us denote the resulting sum vector by $\mathbf{t}$: $\mathbf{t} = \gamma\mathbf{v} + \mathbf{u}_i$. Note that $\mathbf{t}$ is not a unit vector. Finally, we normalize $\mathbf{t}$, and the result is our trial orientation vector fit. We still have to fix γ, which determines the acceptance probability for the orientational trial move. The optimum value of γ is determined by essentially the same criteria as for translational moves. In summary, for linear molecule, calling $\hat{u}$ the original direction of the linear molecule and $\hat{u}^{ROT}$ the rotated direction of $\hat{u}$ we can implement the following algorithm

$$\hat{u}^{ROT} = \frac{\hat{u} + \gamma\hat{v}}{|\hat{u} + \gamma\hat{v}|}$$

where γ is a fixed number that plays the role of Δ in the translational moves and $\hat{v}$ is a random verson on the unit sphere.

We have not yet indicated whether the translational and orientational trial moves should be performed simultaneously. Both procedures are acceptable. However, if rotation and translation correspond to separate moves, then the selection of the type of move should be probabilistic rather than deterministic.



Another possibility is to select a random versor an a sphere $\hat{\mathbf{v}}$ and then perform a rotation of all atoms composing the molecule arount the center of mass by a random angle θ between $-\gamma$ and γ around the random direction $\hat{\mathbf{v}}$ In both cases, the optimum value of γ is determined by essentially the same criteria as for translational moves. The rotational formula is

$$\mathbf{r'}_i = \mathbf{r}_i \cos(\theta) + \hat{\mathbf{v}}(\hat{\mathbf{v}} \cdot \mathbf{r}_i)(1 - \cos(\theta) + \mathbf{r} \times \hat{\mathbf{v}} \sin(\theta)$$

One should always verify that the selected algorithm generates with equal probability all directions on the surface of a sphere.

We have not yet indicated whether the translational and orientational trial moves should be performed simultaneously. Both procedures are acceptable. However, if rotation and translation correspond to separate moves, then the selection of the type of move should be probabilistic rather than deterministic.

13 In the case of non-linear rigid molecules

The situation for non-linear molecules is more complex. There one need to make sure that the Euler angle defining the orientation of the molecule are all properly sampled. Instead of using the Euler angle, it is convenient to encode the orientation of the molecule in a four dimensional versor $\mathbf{Q} \equiv (q_0, q_1, q_2, q_3)$ (with $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$) whose relation with the Euler angle is given by

$$\begin{aligned} q_0 &= \cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi+\psi}{2}\right) \\ q_1 &= \sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\phi-\psi}{2}\right) \\ q_2 &= \sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi-\psi}{2}\right) \\ q_3 &= \cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\phi+\psi}{2}\right) \end{aligned}$$

When the molecule is in the "fixed" frame of reference, $\mathbf{Q} = (1, 0, 0, 0)$, since all Euler angles are zero. The rotated position of the atom (x^R, y^R, z^R) composing the molecule can be calculated, knowing it position in the "fixed" frame (x, y, z) and $\mathbf{Q}$ via the following relation

$$\begin{bmatrix} x^R \\ y^R \\ z^R \end{bmatrix} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{12}$$

The molecule rotation can be achieved, as done for the linear molecule, by rotating $\mathbf{Q}$

$$\mathbf{Q}^{ROT} = \frac{\mathbf{Q} + \gamma\hat{V}}{|\mathbf{Q} + \gamma\hat{V}|}$$

where now $\hat{V}$ is a random vector in a four-dimensional unit sphere.

13.1 Generate uniform random number of a four-dimensional unit sphere

One possibility is offered by Marsaglia, annals of mathematical statistics, 43, 645 (1972).

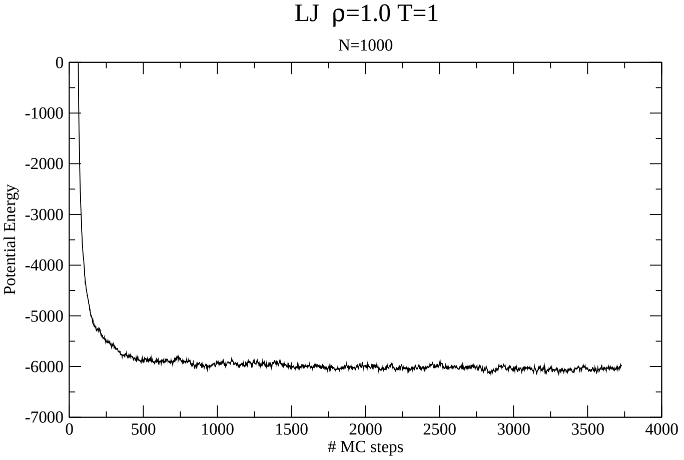
Choose V_1 and V_2 independent uniform on (-1,1) until $S_1 = V_1^2 + V_2^2 < 1$. Chose V_3 and V_4 independent uniform on (-1,1) until $S_2 = V_3^2 + V_4^2 < 1$. Then

$$(V_1, V_2, V_3[(1 - S_1)/S_2]^{1/2}, V_4[(1 - S_1)/S_2]^{1/2})$$

is uniform on the surface of the unit 4-dimensional sphere

14 Equilibrazione

Once the starting configuration has been selected (fixing in this way the density), the MC code can be put in production. It is highly recommended to save configurations periodically during the MC simulation, to be able or to perform a succesfull data analysis or to restart a simulation from the saved configuration at the same or at a different T (which should thus be written in the same format as the starting configuration). It is also recommended to save the total potential energy during the course of the simulation, to be able to monitor in real time the evolution of the system. Starting from a random configuration, the potential energy typically behaves as in the figure.



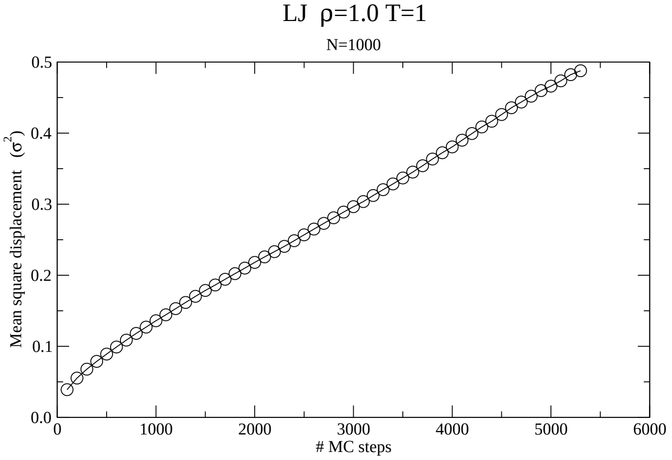
When the system is considered equilibrated (e.g. when the energy fluctuates around a non-drifting average) the equilibration phase can be considered completed and the following configurations can be considered as a proper canonical representation of the system.

15 From a set of equilibrium configurations:MSD

The configurations can then be used to evaluate statistical properties of the system at the selected T and density. It is always recommended to further test if equilibration is indeed achieved. To this aim one can calculate the mean square-displacement (in MC steps), averaged over all different particles and all different initial time 0 (time is in unit of MC steps)

$$MSD(t) = \frac{1}{N} < \sum_i |\mathbf{r}_i(t) - \mathbf{r}_i(0)|^2 >_{t=0}$$

to make sure the particles have diffuse a reasonable distance (of the order of their diameter) during the entire simulation. An example for the same system as before is reported in the following figure.



16 From a set of equilibrium configurations: The structure of the system in real space

The most important function in the theory of simple liquids is the the so-called radial distribution function $g(r)$. The radial distribution function is of interest for two reasons: first of all, neutron and X-ray scattering experiments on simple fluids, and light-scattering experiments on colloidal suspensions, yield information about $g(r)$. Second, $g(r)$ plays a central role in theories of the liquid state. Numerical results for $g(r)$ can be compared with theoretical predictions and thus serve as a criterion to test a particular theory. In a simulation, it is straightforward to measure $g(r)$: it is simply the ratio between the average number density $n(r)$ at a distancer$r$ from any given atom (for sim- plicity we assume that all atoms are identical) and the density at a distance r from an atom in an ideal gas at the same overall density. By construction, $g(r) = 1$ in an ideal gas. Any deviation of $g(r)$ from unity reflects correlations between the particles due to the intermolecular interactions.

To calculate $g(r)$ one need to build the histogram of the relative distances between all pairs of particles. A typical bin mesh δ is of the order of 1/20 of the nearest neighbour distance. To avoid spurious effect due to the PBC, the histogram is calculated (at most) between 0 and half of the box length. Each bin of the histogram then need to be normalized to the expected ideal gas case. If the j bin includes all found particle pairs between $r_j - \delta/2$ and $r_j + \delta/2$, then the expected pairs for an ideal gas of density $(N/-1)L^3$ ($N-1$ enters instead of N since it is the density of an ideal gas excluded the particle in the origin) is

$$\mathcal{N}_j = \frac{N-1}{L^3}\frac{4\pi}{3}[(r_j + \delta/2)^3 - (r_j - \delta/2)^3]$$

The histogram thus need to be normalized by N (to account for the number of particles assumed to be in the origin), by $\mathcal{N}_j$ and by the number of configurations analized to calculate $g(r)$. Note that if distances are calculated with the $i > j$ condition, then each found relative distance must increase the histogram by two (ij and ji).

17 From a set of equilibrium configurations: The structure of the system in Fourier space

The structure factor $S(q)$ is related to the Fourier transform in wavevector space of the radial distribution function. It is the quantity that is measured in all static scattering experiments. If the quality of the $g(r)$ is very high and the box is so large that $g(r)$ has endeed reached the asintothic value for $r < L/2$, then $S(q)$ can be calculated by Fourier transform of the $g(r)$. If this is not the case, then it is convenient to calculate $S(q)$ directly in q space according to the definition

$$S(\mathbf{q} = \frac{1}{N} < \rho_{\mathbf{q}}\rho^*_{\mathbf{q}} >$$

where < ..... > indicates a thermodynamic average (e.g. and average over all different configurations). The complex quantity $\rho_{\mathbf{q}}$ is the Fourier transform of the density, e.g.

$$\rho_{\mathbf{q}} = \sum_{i=1}^{N} e^{i\mathbf{q}\cdot\mathbf{r}_i}$$

being $\rho(\mathbf{r}) = \sum_{i=1}^{N} \delta(\mathbf{r} - \mathbf{r}_i)$ For isotropic systems it is possible to average over all different wavevectors with identical (within a pre-fixed bin width) modulus $|\mathbf{q}|$.

Due to PBC, the accessible wavevectors can be written as

$$\mathbf{q} = \frac{2\pi}{L}(n_x, n_y, n_z)$$

where n_x, n_y, n_z are integer numbers. Note that $-n_x, -n_y, -n_z$ and n_x, n_y, n_z are identical and thus the averaging in q should involve only the upper semi-space. Thus the smallest accessible wavevector is $q = \frac{2\pi}{L}$ and it can be averaged over three independent directions $(1, 0, 0)$,$(0, 1, 0)$ and $(0, 0, 1)$. A typical bin width is $\frac{1}{2}\frac{2\pi}{L}$.

18 From a set of equilibrium configurations: The pressure (atoms and molecules)

The thermodynamic definition of pressure is

$$P = -\frac{\partial A}{\partial V}_T = kT\frac{\partial lnQ}{\partial V}_T \tag{13}$$

To highlight the V dependence it is convenient to write the partition function in term of scaled variables $\vec{r} = \vec{\xi}L$, where $L^3 = V$.

$$Q_N = \frac{V^N}{N!\Lambda^{3N}} \int_0^1 .... \int_0^1 exp\left[-\beta U(\vec{\xi}^N, V)\right] d\vec{\xi}^N \tag{14}$$

so that

$$P = kT\frac{1}{Q_N}\frac{\partial Q_N}{\partial V}_T = \frac{NkT}{V} - < \frac{\partial U}{\partial V} > \tag{15}$$

In addition to the ideal gas contribution, one thus find a term that, configuration by configuration, express the energetic cost associated to an isotropic volume scaling. Hence each configuration $\vec{r}^N$ can be associated to a microscopic pressure $\mathcal{P} \equiv \frac{\partial U}{\partial V}$ whose thermodynamic average is the excess pressure For pair-wise potentials ($U = \frac{1}{2}\sum_{ij} v(|r_{ij}|)$)

$$\mathcal{P} = -\frac{\partial U}{\partial V} = \frac{1}{3L^2}\frac{\partial U}{\partial L} = \frac{1}{3L^2}\frac{1}{2}\sum_{ij}\left[\frac{\partial v(|r_{ij}|)}{\partial x_{ij}}\frac{\partial x_{ij}}{\partial L} + \frac{\partial v(|r_{ij}|)}{\partial y_{ij}}\frac{\partial y_{ij}}{\partial L} + \frac{\partial v(|r_{ij}|)}{\partial z_{ij}}\frac{\partial z_{ij}}{\partial L}\right] =$$

$$\frac{1}{3L^2}\sum_{i<j}\vec{F}_{ij}\cdot\vec{\xi}_{ij} = \frac{1}{3L^3}\sum_{i<j}\vec{F}_{ij}\cdot\vec{r}_{ij}$$

such that

$$P^{ex} = \frac{1}{3V} < \sum_{i<j}\vec{F}_{ij}\cdot\vec{r}_{ij} > \tag{16}$$

This expression is commonly implemented in the numerical codes to evaluate the pressure.

We note on passing that the excess pressure is related to $g(r)$

$$P^{ex} = -\frac{1}{3V}\frac{N}{2}\int 4\pi r^2 \rho \frac{dv(r)}{dr} r g(r) dr \tag{17}$$

$$P = \frac{NkT}{V}\left[1 - \frac{\rho}{6kT}\int r g(r) \frac{dv(r)}{dr} 4\pi r^2 dr\right] \tag{18}$$

18.0.1 Viral

An alternative derivation of the microscopic pressure is based on the temporal average of a quantity named "virial".

$$\mathcal{V} \equiv \sum_1^N \vec{r}_i \cdot \vec{F}_i$$

The time average is

$$< \mathcal{V} >_t = \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau d\tau \sum_1^N \vec{r}_i(t) \cdot \vec{F}_i(t) = \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau d\tau \sum_1^N m\vec{r}_i(t) \cdot \ddot{\vec{r}}_i(t)$$

Integrating by parts

$$\lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau d\tau \sum_1^N m\vec{r}_i(t) \cdot \ddot{\vec{r}}_i(t) = \lim_{\tau\to\infty} \frac{1}{\tau}\left[\sum_1^N m\vec{r}_i(t) \cdot \dot{\vec{r}}_i(t)|_0^\tau - \int_0^\tau d\tau \sum_1^N m|\dot{\vec{r}}_i(t)|^2\right]$$

The first term vanishes since positions and velocities are uncorrelated. Hence

$$< \mathcal{V} >_t = -\lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau d\tau \sum_1^N m|\dot{\vec{r}}_i(t)|^2 = -3Nk_BT$$

Considering that $\vec{F}_i = \vec{F}_i^{int} + \vec{F}_i^{ext}$, and that external forces are applied on the sample holder, and assuming that the holder is a cube of size L

$$< \mathcal{V} >_t = -3Nk_BT = \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau d\tau \sum_1^N \vec{r}_i(t) \cdot \vec{F}_i^{int}(t) + \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau d\tau \sum_1^N \vec{r}_i(t) \cdot \vec{F}_i^{ext}(t) =$$

The second integral is zero on the three sides with coordinates $x = 0$, $y = 0$ and $z = 0$ and it is L times $-PL^2$ on the three sides located in L,

$$< \mathcal{V} >_t = -3Nk_BT = \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau d\tau \sum_1^N \vec{r}_i(t) \cdot \vec{F}_i^{int}(t) - 3PV$$

so that

$$PV = Nk_BT + \frac{1}{3} < \sum_1^N \vec{r}_i(t) \cdot \vec{F}_i^{int}(t) >_t$$

or, remembering the minus sign connecting the force to the potential

$$PV = Nk_BT - \frac{1}{3} < \sum_1^N \vec{r}_i(t) \cdot \vec{\nabla}_i V_N[\vec{r}_i(t)] >_t$$

18.1 Molecular Virial

Although mathematically correct and physically sensible for purely atomic systems, the atomic virial might seem to be an overkill for molecular systems. In a collection of molecules, assuming no constraints, the force $\mathbf{F}_i$ appearing in the atomic virial contains both intramolecular and intermolecular components. If the size of the molecule is small compared to its container, it is more intuitive to think of the coordinate scaling as acting only on the centers of mass of the molecules rather than on each atom individually. That is, the scaling should only affect the relative positions of the molecules rather than the bond lengths and angles within each molecule. In fact, an alternative pressure estimator can be derived by scaling only the positions of the molecular centers of mass rather than individual atomic positions. Consider a system of N molecules with centers of mass at positions $\mathbf{R}_1, ..., \mathbf{R}_N$. For isotropic volume fluctuations, we would define the scaled coordinates $\mathbf{S}_1, ..., \mathbf{S}_N$ of the centers of mass by

$$\mathbf{S}_i = V^{-1/d}\mathbf{R}_i$$

If each molecule has n atoms with masses $m_{i,1}, ..., m_{i,n}$ and atomic positions $\mathbf{r}_{i,1}....\mathbf{r}_{i,N}$ then the center-of-mass position is

$$\mathbf{R}_i = \frac{\sum_{\alpha=1}^N m_{i,\alpha}\mathbf{r}_{i,\alpha}}{\sum_{\alpha=1}^N m_{i,\alpha}}$$

Repeating the same type of calculations done for the atomic virial, (at fixed molecular orientations !) one get again

$$P = kT\frac{1}{Q_N}\frac{\partial Q_N}{\partial V}_T = \frac{NkT}{V} - < \frac{\partial U(r^N, \Omega_N)}{\partial V} > \tag{19}$$

This time

$$\mathcal{P} = -\frac{\partial U(r^N, \Omega_N)}{\partial V} = \frac{1}{3L^2}\frac{\partial U(r^N, \Omega_N)}{\partial L}|_{\Omega_N} = \frac{1}{3L^2}\frac{1}{2}\sum_{ij}\left[\frac{\partial v(|R_{ij}|)}{\partial X_{ij}}\frac{\partial X_{ij}}{\partial L} + \frac{\partial v(|R_{ij}|)}{\partial Y_{ij}}\frac{\partial Y_{ij}}{\partial L} + \frac{\partial v(|R_{ij}|)}{\partial Z_{ij}}\frac{\partial Z_{ij}}{\partial L}\right] =$$

where $\vec{F}_{ij}$ is the force between the com of molecule i and j (the sum over all pair of interaction sites of i and j)

$$\frac{1}{3L^2}\sum_{i<j}\vec{F}_{ij}\cdot\vec{\xi}_{ij} = \frac{1}{3L^3}\sum_{i<j}\vec{F}_{ij}\cdot\vec{R}_{ij}$$

such that

$$P^{ex} = \frac{1}{3V} < \sum_{i<j}\vec{F}_{ij}\cdot\vec{R}_{ij} > \tag{20}$$

The virial term

$$\sum_{i=1}^{N}\mathbf{R}_i\cdot\mathbf{F}_i$$

where, calling α all sites composing molecule i

$$\mathbf{F}_i = \sum_{\alpha}\mathbf{F}_{i,\alpha}$$

is known as the molecular virial.

18.2 Proof of Equivalence between atomic and molecular virial for a polymer

Let's define the coordinates of the sites $\{\vec{r}_i\}$ with respect to the first one

$$H = H(\{\vec{r}_i\})$$

Since

$$\sum_i\sum_{j>i}\vec{F}_{ij}\cdot(\vec{r}_i-\vec{r}_j) = \frac{1}{2}\sum_i\sum_j\left(\vec{F}_{ij}\cdot\vec{r}_i+\vec{F}_{ji}\cdot\vec{r}_j\right) = \frac{1}{2}2\sum_i\sum_j\vec{F}_{ij}\cdot\vec{r}_i = \sum_i\vec{F}_i\cdot\vec{r}_i$$

the pressure P is

$$P = -\frac{1}{3V}\frac{1}{Z_N}\sum_i\int d\vec{r}_1...d\vec{r}_N\vec{F}_i\cdot\vec{r}_i e^{-\beta H(\vec{r}_i)}$$

Let's examine one element of the previous sum, for example the x component of particle 2 (including the minus sign)

$$-\frac{1}{Z_N}\int d\vec{r}_1...d\vec{r}_N\frac{\partial H}{\partial x_2}x_2 e^{-\beta H(\vec{r}_i)} = \frac{1}{Z_N}\int d\vec{r}_1...d\vec{r}_N x_2\frac{\partial}{\partial x_2}e^{-\beta H(\vec{r}_i)}\frac{1}{-\beta} =$$

$$= \frac{1}{-\beta}\frac{1}{Z_N}\left[x_2 e^{-\beta H(\vec{r}_i)}|_{\infty}^{\infty} - \int e^{-\beta H(\vec{r}_i)}\right]$$

and since $H(\pm\infty) = \pm\infty$, being the polymer of finite length,

$$= \frac{1}{-\beta}[-\frac{1}{Z}Z] = k_BT$$

There are $3N-3$ of these contributions (the first monomer is fixed in the origin) and hence

$$P_{virial} = 3(N-1)\frac{k_BT}{3V} \qquad P_{ig} = N\frac{k_BT}{V}$$

such that

$$P_{TOT} = \frac{k_BT}{V}$$