

FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Technical Report

**Optimization of Neighbor List Techniques
in Liquid Matter Simulations**

*Godehard Sutmann, Vladimir Stegailov**

FZJ-ZAM-IB-2004-13

December 2004

(last change: 02.12.2004)

(*) Moscow Institute of Physics and Technology
Institutskij per. 9, Dolgoprudnij, Moscow Region, Russia

Preprint: Submitted for publication

Optimization of Neighbor List Techniques in Liquid Matter Simulations

G. Sutmann^{*,1} and V. Stegailov[‡]

^{*} Central Institute for Applied Mathematics (ZAM)
and John von Neumann Institute for Computing (NIC)
Research Centre Jülich; D-52425 Jülich, Germany

[‡] Moscow Institute of Physics and Technology
Institutskij per. 9, Dolgoprudnij, Moscow Region, Russia

Abstract: The performance of neighbor list techniques in molecular dynamics simulations depend on a variety of parameters, which may be adjusted for maximum efficiency. Here, a model is presented, which allows to choose optimal parameters for the performance of Verlet- and linked-cell lists. In several cases an efficiency gain of $\approx 50\%$ is found if parameters are chosen adequately. Test cases are presented for Lennard-Jones systems at different state points. Good agreement between analytical model and simulation results is found.

1 Introduction

On a first level the CPU time required in molecular dynamics (MD) for the simulation of complex molecular systems is a rapidly growing function with increasing number of particles. This is due to the computation of interatomic potentials and forces which are required for the integration step in MD. The number of operations to calculate mutual distances between all particles N in the system is proportional to N^2 , which limits the size of the system to several thousand particles. In the case of short range interaction potentials, i.e. potentials which decay faster than r^{-d} (d is the physical dimension), it is possible to restrict the range of particle interactions to a limited area around a tagged particle, the so called *interaction sphere*. The radius is chosen in a way that contributions from particles outside R_c are rather small and errors in the total energy may be accounted for in a continuum way as long-range corrections [1]. For Lennard-Jones systems, often $R_c = 2.5\sigma$ is chosen, where σ is the diameter of a Lennard-Jones particle. Note, however, that also in the case of long-range potentials interaction spheres are often in use, e.g. for the short range real space part in Ewald sums [2].

If an interaction sphere is introduced, the number of interactions taken into account in the force computations are reduced to $N(N^* - 1)/2$. Here, $N^* = 4\pi/3\rho R_c^3$, where ρ is the number density in the system and R_c the radius of the interaction sphere, called the *cutoff radius*. The factor $1/2$ accounts for the principle of action and counteraction, Newton's third law. For short range interactions it becomes clear that the intrinsic complexity should be $O(N)$.

¹Corresponding author: g.sutmann@fz-juelich.de

However, it is not clear *a priori* which particles are in the neighborhood of a tagged particle. Therefore the mutual distances of all particle pairs in the system have to be checked whether a force calculation has to be performed or not according to the cutoff radius criterion, increasing the computational effort again to $O(N^2)$.

In 1967 Verlet [3] introduced a list technique which considerably speeded up simulations of particles with a short range interaction. The idea was to remember all particles at a time step t_0 which were located inside the cutoff sphere plus a certain larger region characterized by a radius $R = R_c + R_s$, where R_s is the so called *skin radius*. The skin region serves as a reservoir of particles, which has to be updated from time to time. If in subsequent time steps, $t_0 + n\delta t$, particles from outside R_c enter into the interaction sphere, they originate from the skin region and are therefore also stored in the list. Now, for a tagged particle, not all other particles in the system have to be checked for their relative distance, but only those which are stored in the list. This is usually a very small portion of the whole system and the efficiency gain between list updates is approximately given by N/N^+ , where $N^+ = 4\pi/3\rho(R_c + R_s)^3$. After a certain time, which depends on several parameters, e.g. the density or the temperature, the list has to be updated since particles from outside the radius $R_c + R_s$, which are not stored in the list, may enter into the interaction sphere and therefore have to be taken into account in the force routine. This step requires again a loop over all particle pairs and therefore conserves an $O(N^2)$ complexity for the Verlet list with a strongly reduced prefactor compared to the *brute force* method.

A list technique, reducing the complexity of the problem really to $O(N)$ was introduced in Refs. [4], [5]. The linked-cell method starts with subdividing the whole system into cubic cells and sorting all particles into these cells according to their position. The size of the cells, L_c , is chosen to be $L_c \leq L_{Box}/\text{floor}(L_{Box}/R_c)$, where L_{Box} is the length of the simulation box. All particles are then sorted into a list array of length N . The list is organized in a way that particles, belonging to the same cell are linked together, i.e. the entry in the list referring to a particle points directly to the entry of a next particle inside the same cell. A zero entry in the list stops the search in the cell and a next cell is checked for entries. This technique not only has computational complexity of $O(N)$, since the sorting into the cells and into the N -dimensional array is of $O(N)$, but also has memory requirements which only grow linearly with the number of particles. These features make this technique very appealing. However, the technique is not well vectorizable and also the addressing of next neighbors in the cells require indirect access (e.g. `i=index(i)`), which may lead to cache misses. In order not to miss any particle pair in the interactions every box has to have a neighbor region in each direction which extends to R_c . In the case, where $L_c \geq R_c$, every cell is surrounded by 26 neighbor cells in three dimensional systems. This gives rise to the fact that the method gives only efficiency gains if $L_{Box} \geq 4R_c$, i.e. subdividing each box direction into more than 3 cells.

Although list techniques offer a tremendous potential of speeding up molec-

ular simulations, there is no work known to the authors which focusses on the systematic optimization of parameters, determining the performance of simulations for a given system of particles. There are a few works in literature on performance measurements for neighbor lists as a function of parameters. In Ref. [6] effects of the size of the neighbor radius were discussed and a performance curve was presented. However, no systematic model was built to predict the optimum of the performance curve. Efficiency tests of the Verlet list technique and the linked-cell list as a function of particles N in the system were done in Ref. [7]. Also here no prediction was made for an arbitrary number of particles.

In the present article a parametrization of the performance curves of the Verlet- and linked-cell lists is developed. Performance characteristics depend on the number of particles in the system N , density ρ , temperature T or cutoff radius R_c . Optimization of performance can be obtained by adjusting the skin radius R_s or update interval n in the case of Verlet lists and the cell size L_c in the case of linked-cell lists. The parametrization will be done for *straight forward* implementations of list techniques which have a rather transparent code structure. More refined techniques, aiming to optimize code for special purpose computer architectures are beyond the scope of the present work. However, it will be straight forward to adopt the present formalism also to more complicated code implementations.

In section 2 the theoretical model will be described for the performance curve parametrization. In section 3 the model will be tested against simulations and section 4 will conclude our findings.

2 Theory

2.1 Verlet lists

A straight forward Verlet-list implementation starts with defining a skin radius, R_s , which serves as a reservoir of particles for the list. Interactions between particles are always calculated for each particle within the interaction sphere, characterized by the cutoff radius R_c . In an update step of the Verlet list, all distances $d = \sqrt{\sum_{\alpha} (r_{\alpha,i} - r_{\alpha,j})^2}$ ($\alpha = x, y, z$) between particle pairs i and j have to be checked in the system and those pairs are stored in the list, for which $d \leq R_c + R_s$. Since particles move in time, it may happen that particles from outside the region $R_c + R_s$ enter into the interaction sphere. Since these particles are not taken into account in the list, this would cause errors. Therefore the list has to be updated in certain intervals. Denoting the position of a particle at time t with $\mathbf{r}(t)$, a criterion for updating the list is

$$\max_i |\mathbf{r}(t_0 + n\delta t) - \mathbf{r}(t_0)| > \frac{R_s}{2} \quad (1)$$

where t_0 is the time of the last list update, δt the time step and n an integer. This criterion would include the worst case scenario, where two particles move

a distance $R_s/2$ towards each other on the particles' conacting line. It is clear that n is a stochastic variable with average value $\langle n \rangle$ and variance σ_n^2 . For brevity, n is used in the following to denote the average value, which depends on the skin radius R_s , i.e. $n = n(R_s)$.

Now, the question arises, which value to choose for R_s . Let us first consider two limiting cases: i) $R_s = 0$ implies that in each time step, particles from outside may enter into the list, i.e. the list has to be updated every time step which requires an all-pairs operation $N(N-1)/2$; ii) $R_s \approx L_{Box}$ implies that all particles are within the list. No update of the list has to be performed, but every simulation step requires again $N(N-1)/2$ operations for calculating distances.

These limiting cases imply that there is an optimum value for R_s , where the performance curve has a maximum. A first approximation for the time which has to be spent for the force computations in a Verlet cycle (one list generation and n list steps) is given by

$$T_{VL} = \frac{1}{2}N\tau_f \left[\frac{N-1}{n+1} + \frac{n}{n+1} \left(4\pi\rho \int_0^{R_c+R_s} dr r^2 g(r) - 1 \right) \right] \quad (2)$$

where ρ is the number density of the system, $g(r)$ the radial distribution function and τ_f the time spent in the force routine for each particle pair. Here the first term includes the update step in the list, which is done on average every $(n+1)$ steps. The second term includes the amount of time which is spent for force computations using the list.

In order to optimise the parameters for the Verlet list, an approximation for R_s has to be found in terms of system parameters. The size of the skin radius will determine the length of the update interval n . As is seen from Eq.1, a particle has to move a distance larger than $R_s/2$ in order to trigger an update. For relatively short time intervals one may consider the average distance $\langle x \rangle$, which a particle moves per time step. Knowing this length one may write

$$R_s = 2n\langle x \rangle \quad (3)$$

where the factor of two follows from Eq. 1. As a first approximation one could use $\langle x \rangle \approx l_D$, where $l_D = \sqrt{D\delta t}$ is the diffusional length scale and D is the diffusion constant. This length scale would apply for large skin radii, where the time for crossing the distance becomes large and the motion may be characterized by diffusional processes [8]. For relatively small skin radii this length scale is too small and a free flight approximation will be more appropriate $\langle x \rangle \approx l_{th}$, where $l_{th} = v_{th}\delta t$. The velocity is thereby given as the thermal velocity $v_{th} = \sqrt{3k_B T/m}$, where k_B is the Boltzmann constant, T the temperature and m the particle mass. This velocity is the average velocity of the particles and consequently the thermal length is written as

$$\langle x \rangle \approx c\delta t \sqrt{\frac{3k_B T}{m}} \quad (4)$$

Note that the update criterion, Eq. 1, becomes active if any particle satisfies it. Therefore the fastest particles in the system will determine the update interval.

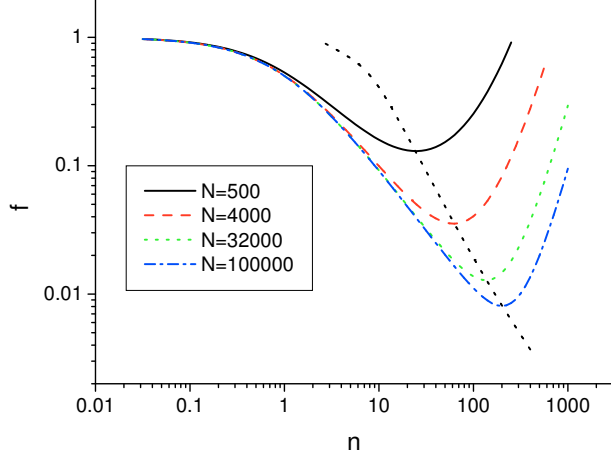


Figure 1: Timing curve for the Verlet list, based on Eq. 5 as function of particle number. The dashed line represents the analytical solution for the minimum position.

Considering those particles having their actual velocities in the wings of the Maxwell-Boltzmann distribution, a rough approximation of the constant c in Eq. 4 is $c \approx 3$.

For not too small values of the radius $R_c + R_s$, the integral may be evaluated within a mean field approximation, i.e. $\int_0^{R_c+R_s} dr r^2 g(r) \approx (R_c + R_s)^3/3$. Therefore, the relative time per particle for the Verlet list compared with a *brute force* approach, i.e. $T_{VL}/(N(N-1)/2\tau_f)$ can be written as

$$f_{VL} = \frac{1}{n+1} + \frac{n}{n+1} \frac{1}{N-1} \left[\frac{4\pi}{3} \rho (R_c + 2n\langle x \rangle)^3 - 1 \right] \quad (5)$$

The performance ratio will therefore depend parametrically on the number of particles N , the density ρ , the cutoff radius R_c and via $\langle x \rangle$ the temperature T , the mass m and the time step δt . For a given system these parameters are known. The only dependence will be on the update interval n .

In Fig.1 the relative performance curves as a function of update interval is shown for different numbers of particles. As expected there is a minimum of the function which is shifted to larger update intervals for larger number of particles. This reflects the fact that a list update is an $O(N^2)$ operation and therefore it is preferred to have smaller number of updates for larger systems.

The optimal update interval can be found via $df_{VL}/dn = 0$, which gives

$$\left(R_c^3 - \frac{3N}{4\pi\rho} \right) + 12R_c^2\langle x \rangle n + 36R_c\langle x \rangle^2 n^2 + 8(3R_c + 4\langle x \rangle)\langle x \rangle^2 n^3 + 24\langle x \rangle^3 n^4 = 0 \quad (6)$$

There is a lengthy analytical solution of Eq. 7 which is omitted here. In practice the solution is easily found by e.g. a Newton-Raphson method [9].

As was shown the update interval n will grow with increasing system size N . For very large systems the asymptotic solution is given by

$$\lim_{N \rightarrow \infty} n = \left(\frac{N}{32\pi\rho\langle x \rangle^3} \right)^{\frac{1}{4}} \quad \text{or} \quad \lim_{N \rightarrow \infty} R_s = \left(\frac{N\langle x \rangle}{2\pi\rho} \right)^{\frac{1}{4}} \quad (7)$$

2.2 Linked-cell lists

In the case of linked-cell lists, the simulation system is subdivided into (rectangular) cells which, in the most simple case, have a length $L_c = L_{Box}/\text{floor}(L_{Box}/R_c)$. First of all some general considerations are done about the efficiency of the simplest cell assignment.

Assuming that the box size $L_{Box} \geq 3R_c$, the maximum cell size varies between $R_c \leq L_c \lesssim 1.25R_c$, due to commensurability constraints. For very large systems L_c may always be chosen as $L_c \approx R_c$, since commensurability constraints become less important. This implies that every cell has 26 neighbor cells where a particle search has to be undertaken. In this case, a simple consideration shows, however, that the real volume V_a where partners of all particles inside a given cell are located is given by

$$V_a = \left(7 + \frac{4\pi}{3} + 3\pi \right) R_c^3 \quad (8)$$

which means that in the case, where $L_c = R_c$ only $\approx 76\%$ of the cell volume is relevant for a particle search. In the worst case, where $L_c = 1.25R_c$, only 39.1% of the cell volume is relevant. The same percentage applies for the relevant interactions between particles.

This estimation is, however, still very optimistic, since the integrated volume of particle interactions of all particles in a given cell is considered. If one asks for the relevant volume per particle, this ratio becomes even smaller. For each particle, interaction partners are located inside the sphere of radius R_c . The ideal case would be therefore if each particle would have to check only those distances of other particles located inside this sphere. However, for every particle distances are checked with all particles lying inside of 27 cells. This implies a volume ratio for each particle of $(4\pi/3)/27 = 0.155$ for the ideal case, where $L_c = R_c$. If $L_c > R_c$ this ratio may drop down to 0.08, which means that between 8% and 15% of the interparticle distances are only relevant for the force computation! Since the algorithm scales with $O(N)$, this affects the prefactor of the algorithm also linearly.

Therefore one may try to reduce the size of the cells, in order to get a better approximation to the interaction sphere with radius R_c . In the limit the cell size tends to zero, thus increasing the cell number density $\rho_c = L_{Box}^3/L_c^3$ to infinity. Since the algorithm is organized in a way that for each cell the distance of particles in all neighbor cells are checked, the problem is shifted from a redundant volume to redundant cells, which contain no particle information. Since all cells have to be checked, there will be a penalty for checking whether a particle is contained inside a cell or not. Therefore one may think about an

optimization of the cell size in order to find a compromise between overhead in distance checking and overhead in cell information checking.

If the cells are reduced, the question is how many neighbor cells of each cell have to be taken into account in the algorithm. The number of neighbor cells along a positive x, y, z -direction from a given cell is

$$f_c = \text{floor} \left(\frac{R_c}{L_c + \epsilon} + 1 \right) \quad (9)$$

where ϵ is the machine precision, which is introduced in order to have the correct result for $L_c = R_c$. The total number of neighbor cells is thereby given as

$$\begin{aligned} N_c = & 8 \left[f_c^3 - \sum_{i=0}^{f_c-1} \sum_{j=0}^{f_c-1} \sum_{k=0}^{f_c-1} \text{floor} \left(\sqrt{i^2 + j^2 + k^2} \frac{L_c}{R_c} \right) \right] \\ & + 12 \left[f_c^2 - \sum_{i=0}^{f_c-1} \sum_{j=0}^{f_c-1} \text{floor} \left(\sqrt{i^2 + j^2} \frac{L_c}{R_c} \right) \right] + 6f_c \end{aligned} \quad (10)$$

In this equation the discreteness of the cells was taken into account. A fairly simpler description is obtained in a spherical approximation, where no discontinuities in numbers of neighbor cells are taken into account. This can be written as

$$N_{c,app} = \frac{4\pi}{3} \left(\frac{R_c}{L_c} + \alpha \right)^3 + 3\pi \left(\frac{R_c}{L_c} + \alpha \right)^2 + 6 \left(\frac{R_c}{L_c} + \alpha \right) \quad (11)$$

Depending on the value of α ($\alpha \in [0, 1]$) this approximation may be considered as an upper and a lower envelope of N_c . Using a least squares approximation, one finds that $\alpha \approx 0.7$ for the best average of N_c . In practice this will act as an interpolation formula for the discontinuous version, Eq. 10.

It is interesting to consider the ratio of the volume which has to be taken into account for a particle search in order to calculate mutual distances between particle pairs and the volume of the interaction sphere. This is given as

$$\rho_r = \frac{(N_c + 1) L_c^3}{\frac{4\pi}{3} R_c^3} \quad (12)$$

Fig. 2 shows ρ_r as function of relative cell size L_c/R_c . Strong discontinuities dominate this function for large arguments. Assume that $L_c = R_c$, i.e. the length of the cells is fully compatible with the cutoff radius and there are only 26 neighbor cells. Now, if the cell size is very slightly reduced so that $L_c < R_c$ the next neighbor cells have to be taken into account for a particle search. Since the single cell volume is only reduced very slightly but the number of neighbor cells is increased from 26 to 124, the volume ratio increases dramatically by a factor ≈ 4.5 . Since the linked-list algorithm is of complexity $O(N)$ the computational time is proportional to the volume of the system (given a homogenous distribution of particles). Since with decreasing size of the cells, the cutoff

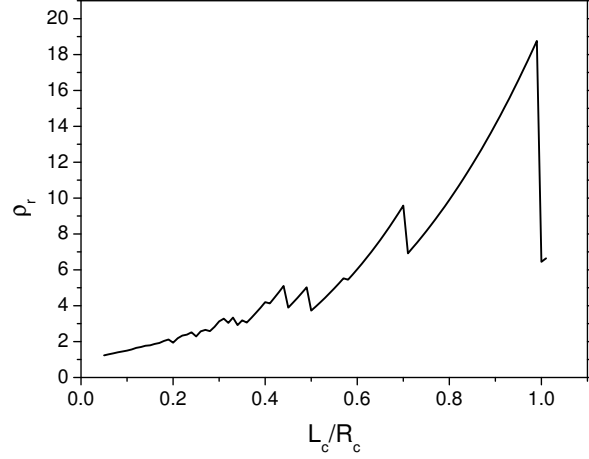


Figure 2: Volume ratio between cell volume, which has to be looked through for particle interactions, and volume of the interaction sphere. In the limiting case $L_c \rightarrow 0$ it is $\rho_r = 1$.

sphere is better and better approximated, the discontinuities are smoothed for very small arguments and it is

$$\lim_{L_c \rightarrow 0} \rho_r = 1 \quad (13)$$

Now let us consider the algorithm in more detail. In order to find a suitable function for the execution time, different parts of the algorithm have to be identified:

1. Search through all cells. For small cell sizes one has to check whether a particle is located inside the cell or not. For very small cell sizes the majority of cells will be empty and there will be a penalty for looking through empty cells.
2. Running over cells, checking how much particles are inside and initializing the loop over the linked cell algorithm.
3. Calculating distances between particles, found within neighbor cells.
4. Calculating forces between particles within the cutoff sphere.

The first two items are essentially conditional statements which have to be evaluated within a loop over all cells. The time, τ_c , for each statement may be approximated for both items to be the same. The time τ_d , spent in the third item will be proportional to the volume of the neighbor cells and therefore will decrease for small cell sizes. The time τ_f of the forth item will be independent

of the cell size and constitutes a system dependent task. Therefore a performance function per particle, normalized to the force evaluation time, τ_f , may be modelled as

$$t_{LCL} = \left(\frac{1}{\rho L_c^3} + \frac{1}{2}(N_c + 1) \right) r_\alpha + \frac{1}{2} (\rho L_c^3 (N_c + 1) - 1) r_\beta \quad (14) \\ + \frac{1}{2} \left(\frac{4\pi}{3} \rho R_c^3 - 1 \right) (1 - r_\alpha - r_\beta)$$

where $r_\alpha = \tau_c/\tau_f$ and $r_\beta = \tau_d/\tau_f$. The first term in Eq. 14 is essentially a penalty function. For cells, containing a lot of particles, it will be negligibly small. However, decreasing the average number of particles per cell below one, it will give an increasingly larger contribution, making the algorithm inefficient for very small cell sizes.

3 Results and Discussion

In order to validate the model for the performance curves, a large number of simulations for Lennard-Jones systems have been performed in the NVE ensemble. Specific parameters of the simulations were varied according to the test case and are reported in the following.

3.1 Verlet-lists

For a system with specified number of particles N , cutoff radius R_c and density ρ the performance function depends on the choice of the skin radius R_s . According to Eq.4 the skin radius, or equivalently the update interval n , depends parametrically on the time step of integration δt , temperature T and particle mass m . Considering those values as fixed, the only free parameter is R_s itself. Nevertheless, for consistency a check is necessary whether the model is robust against a variation of e.g. N, R_c, T, ρ .

Therefore a large number of simulations have been performed for systems where these parameters have been varied in a systematic way. For each single system a series of simulations have been performed where R_s was scanned through a specified interval. For all cases, presented here, the constant c , appearing in Eq. 4, was fixed to $c = 3$.

Variation of temperature: A crucial value in the performance function of the Verlet-list is the length scale $\langle x \rangle$ which gives a kind of upper estimate of the mean path of a particle per time step. Since for a fixed time step the length of the mean path depends on the velocity, a variation in temperature is one way to check the validity of the assumptions made for $\langle x \rangle$.

Therefore simulations for a system of $N = 4000$ particles at a density of $\rho = 0.45$ and a time step of integration $\delta t = 0.001$ were performed for temperatures ranging in the interval $[0.1, 4]$. Fig.3 a shows results obtained for the average

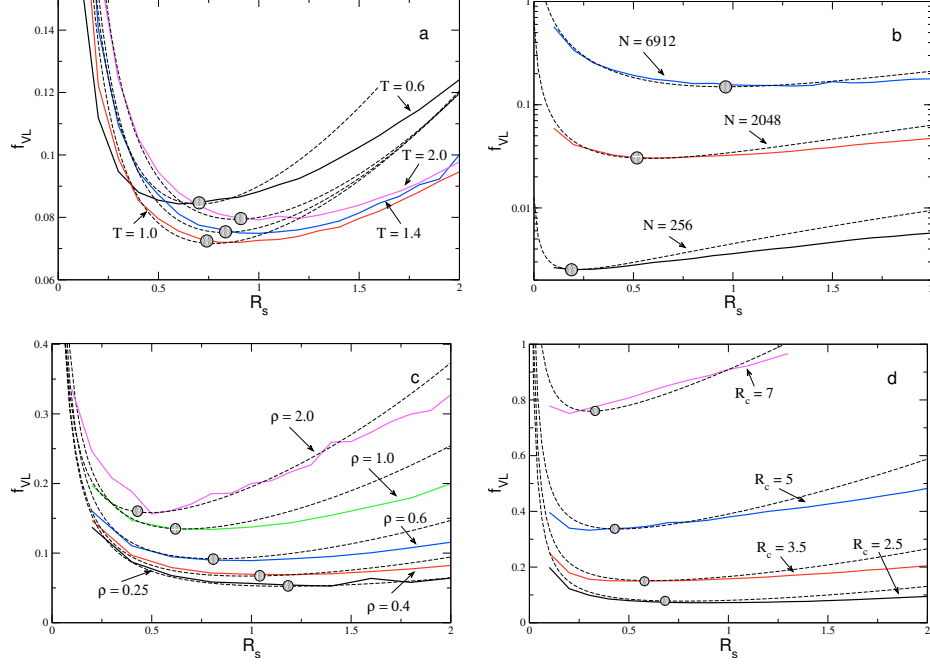


Figure 3: Comparison between run time results from simulation (solid lines) for a Lennard-Jones system and model predictions (dashed lines), based on Eq. 5. Big dots represent the position of the predicted minima. a) variation of temperature T ; b) variation of particle number N ; c) variation of density ρ ; d) variation of cutoff radius R_c .

time of the force routine call. Every symbol in the figure corresponds to a simulation where the size of the skin radius was varied. Large dots correspond to the predictions obtained on the basis of Eq.6.

Variation of particle number: Due to the update step, done every $n + 1$ time steps, the Verlet list technique is still of order $O(N^2)$. Between update steps, the complexity is reduced to $O(N)$. Therefore a variation of the particle number should also reveal weaknesses of the performance function. Simulations were performed for $N \in [500, 4000]$. Results for this test case are shown in Fig. 3 b, which again proves that the minimum value of the execution time is correctly predicted by the model.

Variation of density: The question here is whether for a variation of ρ the characteristic length $\langle x \rangle$ is still valid. Especially for very dense systems, the estimate based mainly on a free flight assumption is likely to break down. Therefore a set of simulations with parameters $N = 4000$, $T = 2$, $\delta t = 0.001$ were performed where a variation of density in the range of $\rho \in [0.25, 1]$ was scanned. It

is found (cf. Fig. 3 c) that in the density range where the system is in the liquid state, the minimum of the performance curve is predicted very accurately. Only for the largest density, where the system is in a state where particle motion is confined due to structural constraints, the prediction slightly underestimates the true minimum position.

The reason for this may be due to the approximation $g(r) = 1$ done in Eq.6, which neglects completely the structure of the system. Also the free flight assumption is invalid here since the motion of particles is highly localized, i.e. particles do not escape from the sphere with radius $R_c + R_s$ in the prescribed way.

Nevertheless, in all cases very good estimates for the best value of R_s are found from the model.

Variation of cutoff radius: To check the validity of the model against a variation of R_c , simulations were performed for a Lennard-Jones system with $N = 4000$, $T = 1$, $\rho = 0.5$ and $\delta t = 0.001$. Cutoff radii were varied in the interval $[2.5, 6.0]$. It is clear that for a given number of particles the *efficiency* for a list update is larger for a large cutoff radius (ratio of particles in the list with respect to total number of particles is large). Therefore the cost of updating the list is higher for small R_c . This will lead to the fact that the particle reservoir, i.e. the skin radius R_s , has to be chosen larger for small R_c . In Figure 3 d results for this test case are shown. Indeed the optimal value for the skin radius is found at larger values for small R_c . As in the previous cases, the model gives a fairly good description of the minimum position.

3.2 Linked-cell lists

In the case of linked-cell lists the performance function depends only on the number of particles in the central- and neighbor cells as well as the number of neighbor cells N_c . Therefore the total number of particles in the system does not affect the performance function for this algorithm. Also there is no characteristic length, i.e. no dynamical information is needed from the system. Therefore the algorithm does not depend on the temperature T . The only parameters which have to be checked are the dependency on density and size of the cutoff radius. To identify neighbor cells of a given cell a static list of this neighbor information was created in the beginning of the simulation. This technique was found to be superior with respect to the technique based on a relative index search [10].

Variation of density: In order to check the dependence on density three different systems at $T = 1$ and $R_c = 2.5$ were simulated with $\rho = 0.75, 2.0, 5.0$. A rather large density was chosen in order to check whether this approximation works as well for liquid as for solid states. It is found that the model describes nearly perfect the behavior of the simulation results (cf. Fig. 4 a). For all three state points the saw tooth shape is reproduced rather exactly. As is seen the minimum value of the performance curve is found between ≈ 0.25 and ≈ 0.4 units of length for large and small densities respectively.

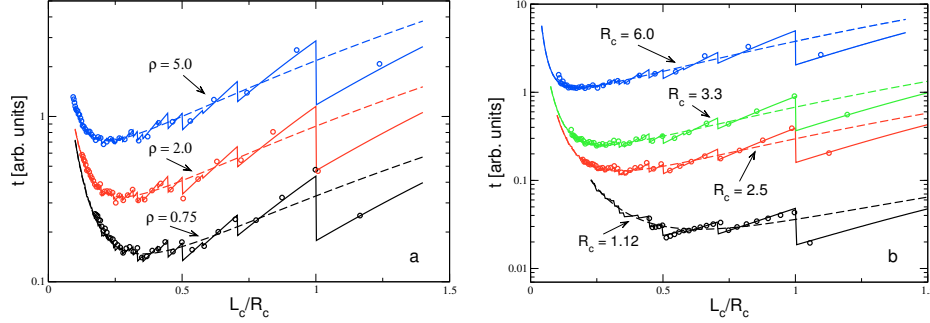


Figure 4: Comparison between run time results from simulation (open dots) for a Lennard-Jones system and model predictions (solid lines), based on Eq. 14. Also shown are model results in the spherical approximation (dashed lines), Eq. 11. a) variation of density ρ ; b) variation of cutoff radius R_c .

Variation of cutoff radius: The dependence on the cutoff radius was checked by simulations of four different systems at state points at $T = 1$ and $\rho = 0.5$. The values for the cutoff radius were $R_c = 2^{1/6}, 2.5, 3.3, 6.0$, ranging between values chosen for contact interactions and long range interactions. Again it is found that the model describes perfectly the behavior of the performance function (cf. Fig. 4 b).

Generally the performance function of the linked-cell algorithm does depend on the relative times r_α and r_β . These times are of course not known in the beginning of a simulation. Moreover, they will depend on the platform, the compiler and compile optimization flags. Therefore, in order to optimise the algorithm, these times have to be evaluated separately. Results, presented here, are based on fits for these time ratios. Attempts to measure these individual times separately, lead to similar values, which often do not fit as well to the model as is shown here, but which nevertheless give a quite good estimation for the minimum of the performance curve. A more detailed description and discussion of this point will be given elsewhere [11].

4 Conclusions

We have presented parametrized models for the optimization of neighbor list techniques, which are commonly used in molecular dynamics simulations. In the case of the Verlet list, essentially all parameters in the model are known in advance of the simulation and therefore the optimum of the skin radius, R_s , can be chosen without numerically adjusting the performance. The only parameter, which relies on an approximation is the length scale, $\langle x \rangle$. In the present work, it was estimated on the basis of a free flight approximation. It is clear that this approximation will work best for small skin radii, low density and high temperature. In the case, where the optimum value of R_s is to be found

at a value where diffusive motion dominates the motion across the distance $R_s/2$ (the list update criterion, Eq. 1), the predicted values for R_s will be underestimated. Nevertheless, the present results indicate that the optimum is located in the region, where the free flight approximation still gives rather good estimates for the true optimum. A more detailed analysis of this point will be given elsewhere [11].

In the case of the linked-cell algorithm it was shown that the model is very successful in describing the true performance of a simulation. However, the knowledge of the time ratios r_α and r_β in Eq. 14, leads to some complication in straight forwardly using the model for optimization. Using time routines or hardware performance counter libraries, it is in principle possible to measure these time ratios. A deeper discussion of this point will be also given elsewhere [11].

A straight forward extension of the present results will be the modeling of a combination of Verlet list and linked cell list. For updating the Verlet list, a linked cell list is used. This reduces the search for neighbors to the nearest neighbor cells and transforms the Verlet list technique into an $O(N)$ algorithm.

A further extension of the present models will be to the simulation of molecules and mixtures of liquids, especially when different masses and cut-off radii for different species are present. In this case several skin radii have to be optimized. Work in this direction is in progress.

References

- [1] M. P. Allen and D. J. Tildesley. *Computer simulation of liquids*. Oxford Science Publications, Oxford, 1987.
- [2] A. Y. Toukmaji and J. A. Board Jr. Ewald summation techniques in perspective: a survey. *Comp. Phys. Comm.*, 95:73–92, 1996.
- [3] L. Verlet. Computer experiments on classical fluids. I. Thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, 159:98, 1967.
- [4] R. W. Hockney, S. P. Goel, and J. W. Eastwood. Quite high-resolution computer models of a plasma. *J. Comp. Phys.*, 14:148, 1974.
- [5] R. W. Hockney and J. W. Eastwood. *Computer simulation using particles*. McGraw-Hill, New York, 1981.
- [6] S. M. Thompson. Use of neighbour lists in molecular dynamics. *CCP5 Newsletter*, 8:20–28, 1983.
- [7] J. J. Morales, L. F. Rull, and S. Toxvaerd. Efficiency test of the traditional MD and the link-cell methods. *Comp. Phys. Comm.*, 56:129–134, 1989.
- [8] A. M. Berezhkovskii and G. Sutmann. Time and length scales for diffusion in liquids. *Phys. Rev. E*, 65:060201, 2002.

- [9] E. Isaacson and H. B. Keller. *Analysis of numerical methods*. Dover Publications, New York, 1994.
- [10] W. Mattson and B. M. Rice. Near-neighbor calculations using a modified linked-list method. *Comp. Phys. Comm.*, 119:135, 1999.
- [11] G. Sutmann and V. Stegailov (in preparation).