Institute for Advanced Simulation

# Molecular Dynamics - Extending the Scale from Microscopic to Mesoscopic

Godehard Sutmann

published in

*Multiscale Simulation Methods in Molecular Sciences*,
J. Grotendorst, N. Attig, S. Blügel, D. Marx (Eds.),
Institute for Advanced Simulation, Forschungszentrum Jülich,
NIC Series, Vol. **42**, ISBN 978-3-9810843-8-2, pp. 1-49, 2009.

© 2009 by John von Neumann Institute for Computing
Permission to make digital or hard copies of portions of this work for personal or classroom use is granted provided that the copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise requires prior specific permission by the publisher mentioned above.

http://www.fz-juelich.de/nic-series/volume42

# Molecular Dynamics - Extending the Scale from Microscopic to Mesoscopic

**Godehard Sutmann**

Institute for Advanced Simulation (IAS)
Jülich Supercomputing Centre (JSC)
Forschungszentrum Jülich, 52425 Jülich, Germany
*E-mail: g.sutmann@fz-juelich.de*

An introduction to classical molecular dynamics simulation is presented. In addition to some historical notes, an overview is given over particle models, integrators and different ensemble techniques. In the end, methods are presented for parallelisation of short range interaction potentials. The efficiency and scalability of the algorithms on massively parallel computers is discussed with an extended version of Amdahl's law.

## 1 Introduction

Computer simulation methods have become a powerful tool to solve many-body problems in statistical physics[1], physical chemistry[2] and biophysics[3]. Although both the theoretical description of complex systems in the framework of statistical physics as well as the experimental techniques for detailed microscopic information are rather well developed it is often only possible to study specific aspects of those systems in great detail via simulation. On the other hand, simulations need specific input parameters that characterize the system in question, and which come either from theoretical considerations or are provided by experimental data. Having characterized a physical system in terms of model parameters, simulations are often used both to solve theoretical models beyond certain approximations and to provide a hint to experimentalists for further investigations. In the case of big experimental facilities it is often even required to prove the potential outcome of an experiment by computer simulations. In this sense it can be stated that the field of computer simulations has developed into a very important branch of science, which on the one hand helps theorists and experimentalists to go beyond their *inherent limitations* and on the other hand is a scientific field on its own. Therefore, simulation science has often been called the *third pillar* of science, complementing theory and experiment.

The traditional simulation methods for many-body systems can be divided into two classes, i.e. stochastic and deterministic simulations, which are largely represented by the Monte Carlo (MC) method[1,4] and the molecular dynamics[5,6] (MD) method, respectively. Monte Carlo simulations probe the configuration space by trial moves of particles. Within the so-called Metropolis algorithm, the energy change from step $n$ to $n+1$ is used as a trigger to accept or reject a new configuration. Paths towards lower energy are always accepted, those to higher energy are accepted with a probability governed by Boltzmann statistics. This algorithm ensures the correct limiting distribution and properties of a given system can be calculated by averaging over all Monte Carlo moves within a given statistical ensemble (where one move means that every degree of freedom is probed once on average). In contrast, MD methods are governed by the system Hamiltonian and consequently

Hamilton's equations of motion[7,8]

$$\dot{p}_i = -\frac{\partial \mathcal{H}}{\partial q_i} \qquad , \qquad \dot{q}_i = \frac{\partial \mathcal{H}}{\partial p_i} \tag{1}$$

are integrated to move particles to new positions and to assign new velocities at these new positions. This is an advantage of MD simulations with respect to MC, since not only the configuration space is probed but the whole phase space which gives additional information about the dynamics of the system. Both methods are complementary in nature but they lead to the same averages of static quantities, given that the system under consideration is ergodic and the same statistical ensemble is used.

In order to characterise a given system and to simulate its complex behavior, a model for interactions between system constituents is required. This model has to be tested against experimental results, i.e. it should reproduce or approximate experimental findings like distribution functions or phase diagrams, and theoretical constraints, i.e. it should obey certain fundamental or limiting laws like energy or momentum conservation.

Concerning MD simulations the ingredients for a program are basically threefold:
(i) As already mentioned, a model for the interaction between system constituents (atoms, molecules, surfaces etc.) is needed. Often, it is assumed that particles interact only pairwise, which is exact e.g. for particles with fixed partial charges. This assumption greatly reduces the computational effort and the work to implement the model into the program.
(ii) An integrator is needed, which propagates particle positions and velocities from time $t$ to $t + \delta t$. It is a finite difference scheme which propagates trajectories discretely in time. The time step $\delta t$ has properly to be chosen to guarantee stability of the integrator, i.e. there should be no drift in the system's energy.
(iii) A statistical ensemble has to be chosen, where thermodynamic quantities like pressure, temperature or the number of particles are controlled. The natural choice of an ensemble in MD simulations is the microcanonical ensemble (NVE), since the system's Hamiltonian without external potentials is a conserved quantity. Nevertheless, there are extensions to the Hamiltonian which also allow to simulate different statistical ensembles.

These steps essentially form the essential framework an MD simulation. Having this tool at hand, it is possible to obtain *exact* results within numerical precision. Results are only correct with respect to the model which enters into the simulation and they have to be tested against theoretical predictions and experimental findings. If the simulation results differ from *real system* properties or if they are incompatible with *solid* theoretical manifestations, the model has to be refined. This procedure can be understood as an adaptive refinement which leads in the end to an approximation of a model of the *real world* at least for certain properties. The model itself may be constructed from plausible considerations, where parameters are chosen from neutron diffraction or NMR measurements. It may also result from first principle *ab initio* calculations. Although the electronic distribution of the particles is calculated very accurately, this type of model building contains also some approximations, since many-body interactions are mostly neglected (this would increase the parameter space in the model calculation enormously). However, it often provides a good starting point for a realistic model.

An important issue of simulation studies is the accessible time- and length-scale which can be covered by microscopic simulations. Fig.1 shows a schematic representation for different types of simulations. It is clear that the more detailed a simulation technique

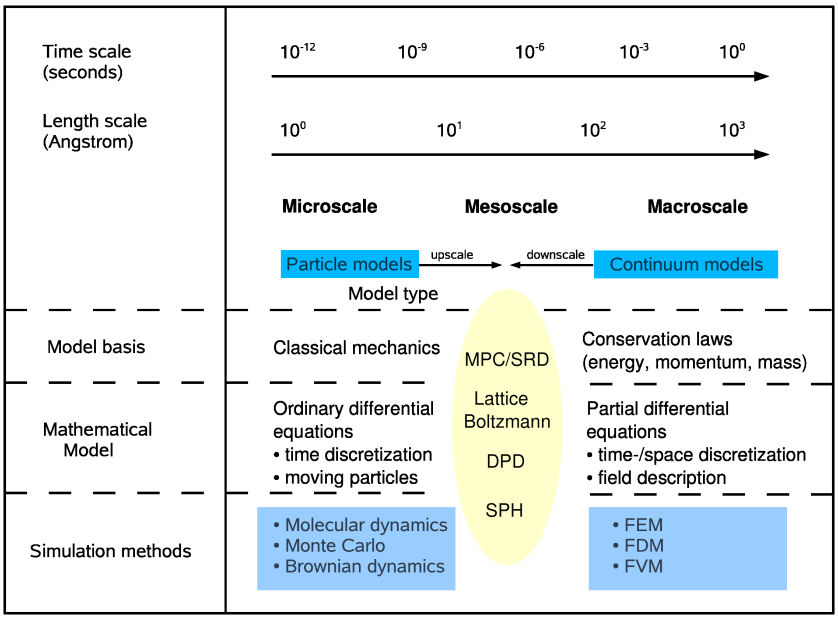


Figure 1. Schematic of different time- and length-scales, occurring from microscopic to macroscopic dimensions. Due to recent developments of techniques like Stochastic Rotation Dynamics (SRD) or Lattice Boltzmann techniques, which are designed to simulate the mesoscopic scales, there is the potential to combine different methods in a multiscale approach to cover a broad spectrum of times and lengths.

operates, the smaller is the accessibility of long times and large length scales. Therefore quantum simulations, where electronic fluctuations are taken into account, are located in the part of the diagram of very short time and length scales which are typically of the order of $\mathring{A}$ and $ps$. Classical molecular dynamics approximates electronic distributions in a rather coarse-grained fashion by putting either fixed partial charges on interaction sites or by adding an approximate model for polarization effects. In both cases, the time scale of the system is not dominated by the motion of electrons, but the time of intermolecular collision events, rotational motions or intramolecular vibrations, which are orders of magnitude slower than those of electron motions. Consequently, the time step of integration is larger and trajectory lengths are of order $ns$ and accessible lengths of order $10 - 100\ \mathring{A}$. If one considers tracer particles in a solvent medium, where one is not interested in a detailed description of the solvent, one can apply Brownian dynamics, where the effect of the solvent is hidden in average quantities. Since collision times between tracer particles is very long, one may apply larger timesteps. Furthermore, since the solvent is not simulated explicitly, the lengthscales may be increased considerably. Finally, if one is interested not in a microscopic picture of the simulated system but in macroscopic quantities, the concepts of hydrodynamics may be applied, where the system properties are hidden in effective numbers, e.g. density, viscosity or sound velocity.

It is clear that the performance of particle simulations strongly depends on the computer facilities at hand. The first studies using MD simulation techniques were performed in 1957

by B. J. Alder and T. E. Wainright[9] who simulated the phase transition of a system of hard spheres. The general method, however, was presented only two years later[10]. In these early simulations, which were run on an IBM-704, up to 500 particles could be simulated, for which 500 collisions per hour were calculated. Taking into account 200000 collisions for a production run, these simulations lasted for more than two weeks. Since the propagation of hard spheres in a simulation is event driven, i.e. it is determined by the collision times between two particles, the propagation is not based on an integration of the equations of motion, but rather the calculation of the time of the next collision, which results in a variable time step in the calculations.

The first MD simulation which was applied to atoms interacting via a continuous potential was performed by A. Rahman in 1964. In this case, a model system for Argon was simulated and not only binary collisions were taken into account but the interactions were modeled by a Lennard-Jones potential and the equations of motion were integrated with a finite difference scheme. This work may be considered as seminal for dynamical calculations. It was the first work where a numerical method was used to calculate dynamical quantities like autocorrelation functions and transport coefficients like the diffusion coefficient for a realistic system. In addition, more involved characteristic functions like the dynamic van Hove function and non-Gaussian corrections to diffusion were evaluated. The calculations were performed for 864 particles on a CDC 3600, where the propagation of all particles for one time step took $\approx 45\ s$. The calculation of $50000$ timesteps then took more than three weeks! [a]

With the development of faster and bigger massively parallel architectures the accessible time and length scales are increasing for all-atom simulations. In the case of classical MD simulations it is a kind of competition to break new world records by carrying out demonstration runs of larger and larger particle systems[11–14]. In a recent publication, it was reported by Germann and Kadau[15] that a trillion-atom ($10^{12}$ particles!) simulation was run on an IBM BlueGene/L machine at Lawrence Livermore National Laboratory with 212992 PowerPC 440 processors with a total of 72 TB memory. This run was performed with the memory optimised program SPaSM[16,17] (Scalable Parallel Short-range Molecular dynamics) which, in single-precision mode, only used 44 Bytes/particle. With these conditions a simulation of a Lennard-Jones system of $N = (10000)^3$ was simulated for 40 time steps, where each time step used $\approx 50secs$ wall clock time.

Concerning the accessible time scales of all-atom simulations, a numerical study, carried out by Y. Duan and P. A. Kollman in 1998 still may be considered as a milestone in simulation science. In this work the protein folding process of the subdomain HP-36 from the villin headpiece[18,19] was simulated up to $1\ \mu s$. The protein was modelled with a 596 interaction site model dissolved in a system of 3000 water molecules. Using a timestep of integration of $2 \times 10^{-15}s$, the program was run for $5 \times 10^8$ steps. In order to perform this type of calculation, it was necessary to run the program several months on a CRAY T3D and CRAY T3E with 256 processors. It is clear that such kind of a simulation is exceptional due to the large amount of computer resources needed, but it was nonetheless a kind of milestone pointing to future simulation practices, which are nowadays still not standard, but nevertheless exceptionally applied[20].

Classical molecular dynamics methods are nowadays applied to a huge class of prob-

[a] On a standard PC this calculation may be done within less than one hour nowadays!

lems, e.g. properties of liquids, defects in solids, fracture, surface properties, friction, molecular clusters, polyelectrolytes and biomolecules. Due to the large area of applicability, simulation codes for molecular dynamics were developed by many groups. On the internet homepage of the Collaborative Computational Project No.5 (CCP5)[21] a number of computer codes are assembled for condensed phase dynamics. During the last years several programs were designed for parallel computers. Among them, which are partly available free of charge, are, e.g., Amber/Sander[22], CHARMM[23], NAMD[24], NWCHEM[25], GROMACS[26] and LAMMPS[27].

Although, with the development of massively parallel architectures and highly scalable molecular dynamics codes, it has become feasible to extend the time and length scales to *relatively* large scales, a lot of processes are still beyond technical capabilities. In addition, the time and effort for running these simulations is enormous and it is certainly still far beyond of standard. A way out of this dilemma is the invention of new simulation of methodological approaches. A method which has attracted a lot of interest recently is to coarse grain all-atom simulations and to approximate interactions between individual atoms by interactions between whole groups of atoms, which leads to a smaller number of degrees of freedom and at the same time to a smoother energy surface, which on the one hand side increases the computation between particle interactions and on the other hand side allows for a larger time step, which opens the way for simulations on larger time and length scales of physical processes[28]. Using this approach, time scales of more than 1 $\mu secs$ can now be accessed in a fast way[29,30], although it has to be pointed out that coarse grained force fields have a very much more limited range of application than all-atom force fields. In principle, the coarse graining procedure has to be outlined for every different thermodynamic state point, which is to be considered in a simulation and from that point of view coarse grain potentials are not transferable in a straight forward way as it is the case for a wide range of all-atom force field parameters.

## 2 Models for Particle Interactions

A system is completely determined through it's Hamiltonian $\mathcal{H} = \mathcal{H}_0 + \mathcal{H}_1$, where $\mathcal{H}_0$ is the *internal* part of the Hamiltonian, given as

$$\mathcal{H}_0 = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + \sum_{i<j}^{N} u(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i<j}^{N} u^{(3)}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \ldots \tag{2}$$

where $\mathbf{p}$ is the momentum, $m$ the mass of the particles and $u$ and $u^{(3)}$ are pair and three-body interaction potentials. $\mathcal{H}_1$ is an external part, which can include time dependent effects or external sources for a force. All simulated objects are defined within a model description. Often a precise knowledge of the interaction between atoms, molecules or surfaces are not known and the model is constructed in order to describe the main features of some observables. Besides boundary conditions, which are imposed, it is the model which completely determines the system from the physical point of view. In classical simulations the *objects* are most often described by point-like centers which interact through pair- or multibody interaction potentials. In that way the highly complex description of electron dynamics is abandoned and an effective picture is adopted where the main features like the hard core of a particle, electric multipoles or internal degrees of freedom of a molecules are

modeled by a set of parameters and (most often) analytical functions which depend on the mutual position of particles in the configuration. Since the parameters and functions give a complete information of the system's energy as well as the force acting on each particle through $\mathbf{F} = -\nabla U$, the combination of parameters and functions is also called a *force field*[31]. Different types of force field were developed during the last ten years. Among them are e.g. MM3[32], MM4[33], Dreiding[34], SHARP[35], VALBON[36], UFF[37], CFF95[38], AMBER[39] CHARMM[40], OPLS[41] and MMFF[42].

There are major differences to be noticed for the potential forms. The first distinction is to be made between pair- and multibody potentials. In systems with no constraints, the interaction is most often described by pair potentials, which is simple to implement into a program. In the case where multibody potentials come into play, the counting of interaction partners becomes increasingly more complex and dramatically slows down the execution of the program. Only for the case where interaction partners are known in advance, e.g. in the case of torsional or bending motions of a molecule can the interaction be calculated efficiently by using neighbor lists or by an intelligent way of indexing the molecular sites.

A second important difference between interactions is the spatial extent of the potential, classifying it into short and long range interactions. If the potential drops down to zero faster than $r^{-d}$, where $r$ is the separation between two particles and $d$ the dimension of the problem, it is called short ranged, otherwise it is long ranged. This becomes clear by considering the integral

$$I = \int \frac{dr^d}{r^n} = \begin{cases} \infty & : n \leq d \\ \text{finite} & : n > d \end{cases} \tag{3}$$

i.e. a particles' potential energy gets contributions from *all particles of the universe* if $n \leq d$, otherwise the interaction is bound to a certain region, which is often modeled by a spherical interaction range. The long range nature of the interaction becomes most important for potentials which only have potential parameters of the same sign, like the gravitational potential where no screening can occur. For Coulomb energies, where positive and negative charges may compensate each other, long range effects may be of minor importance in some systems like molten salts.

There may be different terms contributing to the interaction potential between particles, i.e. there is no universal expression, as one can imagine for first principles calculations. In fact, contributions to interactions depend on the model which is used and this is the result of collecting various contributions into different terms, coarse graining interactions or imposing constraints, to name a few. Generally one can distinguish between bonded and non-bonded terms, or intra- and inter-molecular terms. The first class denotes all contributions originating between particles which are closely related to each other by constraints or potentials which guaranty defined particles as close neighbors. The second class denotes interactions between particles which can *freely* move, i.e. there are no defined neighbors, but interactions simply depend on distances.

A typical form for a (so called) force field (e.g. AMBER[22]) looks as follows

$$\begin{aligned} \mathcal{U} = & \sum_{\text{bonds}} K_r(r - r_{eq})^2 + \sum_{\text{angles}} K_\theta(\theta - \theta_{eq})^2 + \sum_{\text{dihedrals}} \frac{V_n}{2}[1 + \cos(n\phi - \gamma)] \\ & + \sum_{i<j} \left[ \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \right] + \sum_{\text{H}-\text{bonds}} \left[ \frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}} \right] + \sum_{i<j} \frac{q_i q_j}{r_{ij}} \end{aligned} \tag{4}$$

In the following, short- and long-range interaction potentials and methods are briefly described in order to show differences in their algorithmical treatment.

In the following two examples shall illustrate the different treatment of short- and long range interactions.

### 2.1 Short Range Interactions

Short range interactions offer the possibility to take into account only neighbored particles up to a certain distance for the calculation of interactions. In that way a cutoff radius is introduced beyond of which mutual interactions between particles are neglected. As an approximation one may introduce *long range corrections* to the potential in order to compensate for the neglect of explicit calculations. The whole short range potential may then be written as

$$U = \sum_{i<j}^{N} u(r_{ij}|r_{ij} < R_c) + U_{lrc} \tag{5}$$

The long-range correction is thereby given as

$$U_{lrc} = 2\pi N \rho_0 \int_{R_c}^{\infty} dr\ r^2 g(r) u(r) \tag{6}$$

where $\rho_0$ is the number density of the particles in the system and $g(r) = \rho(r)/\rho_0$ is the radial distribution function. For computational reasons, $g(r)$ is most often only calculated up to $R_c$, so that in practice it is assumed that $g(r) = 1$ for $r > R_c$, which makes it possible for many types of potentials to calculate $U_{lrc}$ analytically.

Besides internal degrees of freedom of molecules, which may be modeled with short range interaction potentials, it is first of all the excluded volume of a particle which is of importance. A finite diameter of a particle may be represented by a steep repulsive potential acting at short distances. This is either described by an exponential function or an algebraic form, $\propto r^{-n}$, where $n \geq 9$. Another source of short range interaction is the van der Waals interaction. For neutral particles these are the London forces arising from induced dipole interactions. Fluctuations of the electron distribution of a particle give rise to fluctuating dipole moments, which on average compensate to zero. But the instantaneous created dipoles induce also dipoles on neighbored particles which attract each other $\propto r^{-6}$. Two common forms of the resulting interactions are the Buckingham potential

$$u^B_{\alpha\beta}(r_{ij}) = A_{\alpha\beta} e^{-B_{\alpha\beta} r_{ij}} - \frac{D_{\alpha\beta}}{r^6_{ij}} \tag{7}$$

and the Lennard-Jones potential

$$u^{LJ}_{\alpha\beta}(r_{ij}) = 4\epsilon_{\alpha\beta} \left( \left( \frac{\sigma_{\alpha\beta}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{\alpha\beta}}{r_{ij}} \right)^{6} \right) \tag{8}$$

which are compared in Fig.2. In Eqs.7,8 the indices $\alpha, \beta$ indicate the species of the particles, i.e. there are parameters $A, B, D$ in Eq.7 and $\epsilon, \sigma$ in Eq.8 for intra-species interactions ($\alpha = \beta$) and cross species interactions ($\alpha \neq \beta$). For the Lennard-Jones potential the parameters have a simple physical interpretation: $\epsilon$ is the minimum potential energy,

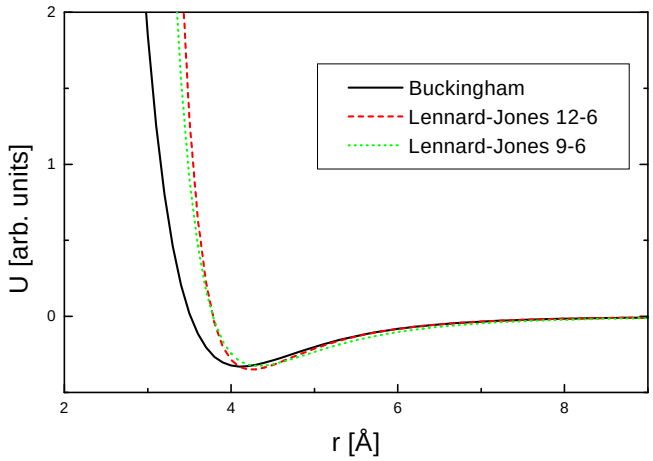


Figure 2. Comparison between a Buckingham-, Lennard-Jones (12-6) and Lennard-Jones (9-6) potential.

located at $r = 2^{1/6}\sigma$ and $\sigma$ is the diameter of the particle, since for $r < \sigma$ the potential becomes repulsive. Often the Lennard-Jones potential gives a reasonable approximation of a *true* potential. However, from exact quantum ab initio calculations an exponential type repulsive potential is often more appropriate. Especially for dense systems the too steep repulsive part often leeds to an overestimation of the pressure in the system. Since computationally the Lennard-Jones interaction is quite attractive the repulsive part is sometimes replaced by a weaker repulsive term, like $\propto r^{-9}$. The Lennard-Jones potential has another advantage over the Buckingham potential, since there are combining rules for the parameters. A common choice are the Lorentz-Berelot combining rules

$$\sigma_{\alpha\beta} = \frac{\sigma_{\alpha\alpha} + \sigma_{\beta\beta}}{2} \quad , \quad \epsilon_{\alpha\beta} = \sqrt{\epsilon_{\alpha\alpha}\epsilon_{\beta\beta}} \tag{9}$$

This combining rule is, however, known to overestimate the well depth parameter. Two other commonly known combining rules try to correct this effect, which are Kong[43] rules

$$\sigma_{\alpha\beta} = \left[ \frac{1}{2^{13}} \frac{\epsilon_{\alpha\alpha}\sigma_{\alpha\alpha}^{12}}{\sqrt{\epsilon_{\alpha\alpha}\sigma_{\alpha\alpha}^{6}\epsilon_{\beta\beta}\sigma_{\beta\beta}^{6}}} \left( 1 + \left( \frac{\epsilon_{\beta\beta}\sigma_{\beta\beta}^{12}}{\epsilon_{\alpha\alpha}\sigma_{\alpha\alpha}^{12}} \right)^{1/13} \right)^{13} \right]^{1/6} \tag{10}$$

$$\epsilon_{\alpha\beta} = \frac{\sqrt{\epsilon_{\alpha\alpha}\sigma_{\alpha\alpha}^{6}\epsilon_{\beta\beta}\sigma_{\beta\beta}^{6}}}{\sigma_{\alpha\beta}^{6}} \tag{11}$$

and the Waldman-Kagler[44] rule

$$\sigma_{\alpha\beta} = \left( \frac{\sigma_{\alpha\alpha}^{6} + \sigma_{\beta\beta}^{6}}{2} \right)^{1/6} \quad , \quad \epsilon_{\alpha\beta} = \frac{\sqrt{\epsilon_{\alpha\alpha}\sigma_{\alpha\alpha}^{6}\epsilon_{\beta\beta}\sigma_{\beta\beta}^{6}}}{\sigma_{\alpha\beta}^{6}} \tag{12}$$

In a recent study[45] of Ar-Kr and Ar-Ne mixtures, these combining rules were tested and it was found that the Kong rules give the best agreement between simulated and experimental

pressure-density curves. An illustration of the different combining rules is shown in Fig.3 for the case of an Ar-Ne mixture.

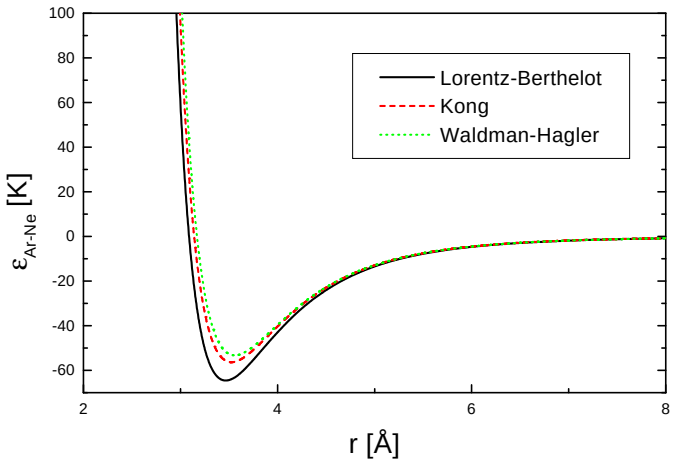


Figure 3. Resulting cross-terms of the Lennard-Jones potential for an Ar-Ne mixture. Shown is the effect of different combining rules (Eqs.9-12). Parameters used are $\sigma_{Ar} = 3.406\ \mathring{A}, \epsilon_{Ar} = 119.4\ K$ and $\sigma_{Ne} = 2.75\ \mathring{A}, \epsilon_{Ne} = 35.7\ K$.

Since there are only relatively few particles which have to be considered for the interaction with a tagged particle (i.e. those particles within the cutoff range), it would be a computational bottleneck if in any time step all particle pairs would have to be checked whether they lie inside or outside the interaction range. This becomes more and more a problem as the number of particles increases. A way to overcome this bottleneck is to introduce list techniques. The first implementation dates back to the early days of molecular dynamics simulations. In 1967, Verlet introduced a list[46], where at a given time step all particle pairs were stored within a range $R_c + R_s$, where $R_s$ is called the skin radius and which serves as a reservoir of particles, in order not to update the list in each time step (which would make the list redundant). Therefore, in a force routine, not all particles have to tested, whether they are in a range $r_{ij} < R_c$, but only those particle pairs, stored in the list. Since particles are moving during the simulation, it is necessary to update the list from time to time. A criterion to update the list could be, e.g.

$$\max_i |\mathbf{r}_i(t) - \mathbf{r}_i(t_0)| \geq \frac{R_s}{2} \tag{13}$$

where $t_0$ is the time from the last list update. This ensures that particles cannot move from the outside region into the cutoff sphere without being recognized. This technique, though efficient, has still complexity $\mathcal{O}(N^2)$, since at an update step, *all* particle pairs have to be checked for their mutual distances. Another problem arises when simulating many particles, since the memory requirements are relatively large (size of the list is $4\pi(R_c + R_s)^3 \rho\, N/3$). There is, of course also the question, how large the skin radius should be chosen. Often, it is chosen as $R_s = 1.5\sigma$. In Ref.[47] it was shown that an optimal choice

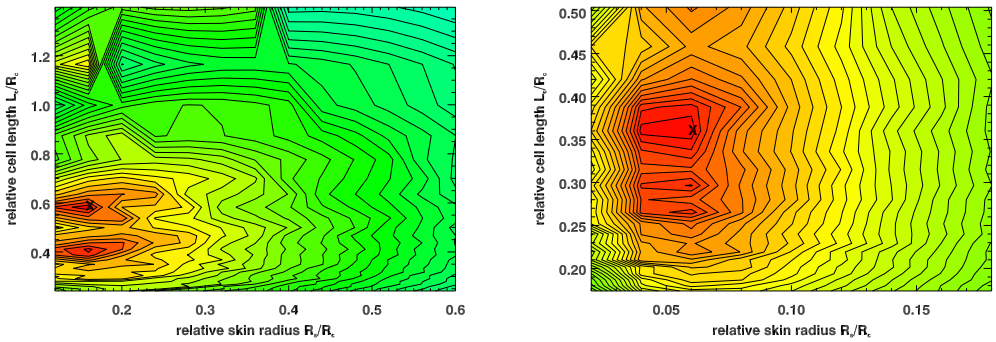


Figure 4. Contour plots of the performance for the combination of linked-cell and Verlet list as a function of the cell length and the size of the skin radius. Crosses mark the positions predicted from an optimization procedure[48]. Test systems were composed of 4000 Lennard-Jones particles with $R_c = 2.5\,\sigma$ at temperature $T = 1.4\,\epsilon/k_B$. Left: $\rho = 0.75/\sigma^3$. Right: $\rho = 2.0/\sigma^3$.

strongly depends on the number of particles in the system and an optimization procedure was outlined.

An alternative list technique, which scales linearly with the number of particles is the linked-cell method[49,50]. 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. In order to approximate the cutoff sphere in a better way by cubic cells, one may reduce the cell size and simultaneously increasing the total number of cells. In an optimization procedure[47], it was found that a reduction of cell sizes to $L_c = R_c/2$ or even smaller often gives very much better results.

It is, of course, possible to combine these list techniques, i.e. using the linked-cell technique in the update step of the Verlet list. This reduces the computational complexity of the Verlet list to $\mathcal{O}(N)$ while fully preserving the efficiency of the list technique. It is also possible to model the performance of this list combination and to optimize the length of the cells and the size of the skin radius. Figure 4 shows the result of a parameter study,

where the performance of the list was measured as a function of $(L_c, R_s)$. Also shown is the prediction of parameters coming out of an optimization procedure[48].

### 2.2 Long Range Interactions

Long range interactions essentially require to take all particle pairs into account for a proper treatment of interactions. This may become a problem, if periodic boundary conditions are imposed to the system, i.e. formally simulating an infinite number of particles (no explicit boundaries imply infinite extend of the system). Therefore one has to devise special techniques to treat this situation. On the other hand one also has to apply fast techniques to overcome the inherent $\mathcal{O}(N^2)$ complexity of the problem, since for large numbers of particles this would imply an intractable computational bottleneck. In general one can classify algorithms for long range interactions into the following system:

- Periodic boundary conditions
  - Grid free algorithms, e.g. Ewald summation method[51–53]
  - Grid based algorithms, e.g. Smoothed Particle Mesh Ewald[54,55], Particle-Particle Particle-Mesh method[56–58]
- Open boundary conditions
  - Grid free algorithms, e.g. Fast Multipole Method[59–64] (FMM), Barnes-Hut Tree method[65,66]
  - Grid based algorithms, e.g. Particle-Particle Particle-Multigrid method[67] ($P^3$Mg), Particle Mesh Wavelet method[68] (PMW)

In the following two important members of these classes will be described, the Ewald summation method and the Fast Multipole Method.

#### 2.2.1 Ewald Summation Method

The Ewald summation method originates from crystal physics, where the problem was to determine the Madelung constant[69], describing a factor for an effective electrostatic energy in a perfect periodic crystal. Considering the electrostatic energy of a system of $N$ particles in a cubic box and imposing periodic boundary conditions, leads to an equivalent problem. At position $\mathbf{r}_i$ of particle $i$, the electrostatic potential, $\phi(\mathbf{r}_i)$, can be written down as a lattice sum

$$\phi(\mathbf{r}_i) = \sum_{\mathbf{n}}^{\dagger} \sum_{j=1}^{N} \frac{q_j}{\|\mathbf{r}_{ij} + \mathbf{n}L\|} \tag{14}$$

where $\mathbf{n} = (n_x, n_y, n_z), n_\alpha \in \mathbb{Z}$ is a vector along cartesian coordinates and $L$ is the length of the simulation box. The sign "†" means that $i \neq j$ for $\|\mathbf{n}\| = 0$.

Eq. (14) is conditionally convergent, i.e. the result of the outcome depends on the order of summation. Also the sum extends over infinite number of lattice vectors, which means that one has to modify the procedure in order to get an absolute convergent sum and to get it fast converging. The original method of Ewald consisted in introducing a convergence

factor $e^{-ns}$, which makes the sum absolute convergent; then transforming it into different fast converging terms and then putting $s$ in the convergence factor to zero. The final result of the calculation can be easier understood from a physical picture. If every charge in the system is screened by a counter charge of opposite sign, which is smeared out, then the potential of this composite charge distribution becomes short ranged (it is similar in electrolytic solutions, where ionic charges are screened by counter charges - the result is an exponentially decaying function, the Debye potential[70]). In order to compensate for the added charge distribution it has to be subtracted again. The far field of a localized charge distribution is, however, again a Coulomb potential. Therefore this term will be long ranged. There would be nothing gained if one would simply sum up these different terms. The efficiency gain shows up, when one calculates the short range interactions as direct particle-particle contributions in real space, while summing up the long range part of the smeared charge cloud in reciprocal Fourier space. Choosing as the smeared charge distribution a Gaussian charge cloud of half width $1/\alpha$ the corresponding expression for the energy becomes

$$\phi(\mathbf{r}_i) = \sum_{\mathbf{n}}{}^{\dagger} \sum_{j=1}^{N} q_j \frac{\operatorname{erfc}(\alpha \|\mathbf{r}_{ij} + \mathbf{n}L\|)}{\|\mathbf{r}_{ij} + \mathbf{n}L\|} + \frac{4\pi}{L^3} \sum_{\mathbf{k}\neq 0} \sum_{j=1}^{N} \frac{q_j}{\|\mathbf{k}\|^2} e^{-\|\mathbf{k}\|^2/4\alpha^2} e^{i\mathbf{k}\mathbf{r}_{ij}} - q_i \frac{2\alpha}{\sqrt{\pi}} \tag{15}$$

The last term corresponds to a self energy contribution which has to be subtracted, as it is considered in the Fourier part. Eq. (15) is an exact equivalent of Eq. (14), with the difference that it is an absolute converging expression. Therefore nothing would be gained without further approximation. Since the complimentary error function can be approximated for large arguments by a Gaussian function and the k-space parts decreases like a Gaussian, both terms can be approximated by stopping the sums at a certain lattice vector $\mathbf{n}$ and a maximal $k$-value $k_{max}$. The choice of parameters depends on the error, $\epsilon = \exp(-p^2)$, which one accepts to tolerate. Setting the error tolerance $p$ and choosing the width of the counter charge distribution, one gets

$$R_c^2 + \frac{\log(R_c)}{\alpha^2} = \frac{1}{\alpha^2}(p^2 - \log(2)) \tag{16}$$

$$k_{max}^2 + 8\alpha^2 \log(k_{max}) = 4\alpha^2 p^2 + \log\left(\frac{4\pi}{L^3}\right) \tag{17}$$

This can be solved iteratively or if one is only interested in an approximate estimate for the error, i.e. neglecting logarithmic terms, one gets

$$R_c = \frac{p}{\alpha} \tag{18}$$

$$k_{max} = 2\alpha p \tag{19}$$

Using this error estimate and furthermore introducing execution times, spent for the real- and reciprocal-space part, it is possible to show that parameters $R_c$, $\alpha$ and $k_{max}$ can be chosen to get a complexity of $\mathcal{O}(N^{3/2})$ for the Ewald sum[71,72]. In this case, parameters

are

$$\frac{R_c}{L} \approx \sqrt{\frac{\pi}{N^{1/3}}} \quad , \quad \alpha L \approx \frac{L k_{max}}{2\pi} = \sqrt{\pi N^{1/3}} \tag{20}$$

Figure 5 shows the contributions of real- and reciprocal parts in Eq. (15), as a function of the spreading parameter $\alpha$, where an upper limit in both the real- and reciprocal-contributions was applied. In the real-space part usually one restricts the sum to $|\mathbf{n}| = 0$ and applies a spherical cutoff radius, $R_c$. For fixed values of $R_c$ and $k_{max}$ there is a broad plateau region, where the two terms add up to a constant value. Within this plateau region, a value for $\alpha$ should be chosen. Often it is chosen according to $\alpha = 5/L$. Also shown is the potential energy of a particle, calculated with the Ewald sum. It is well observed that due to the periodicity of the system the potential energy surface is not radial symmetric, which may cause problems for small numbers of particles in the system.

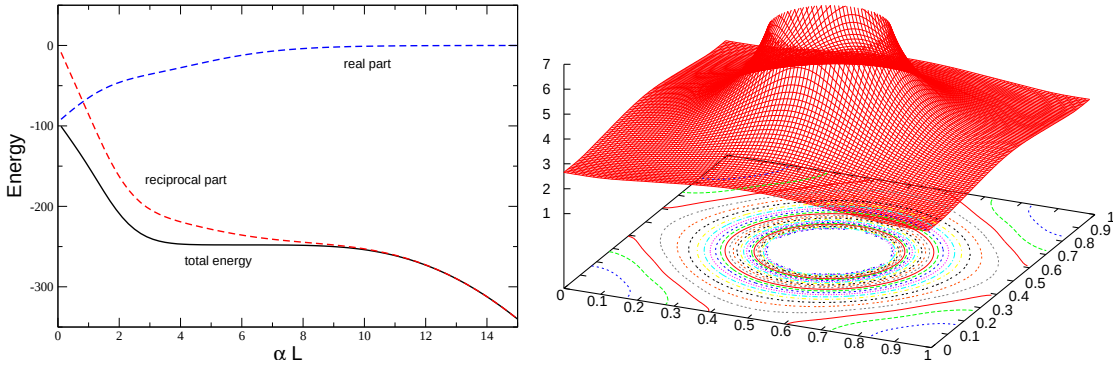


Figure 5. Left: Dependence of the calculated potential on the choice of the scaled inverse width, $\alpha L$, of the smeared counter charge distribution. Parameters for this test were $N = 152$, $R_c = 0.5\,L$ and $k_{max} L/2\pi = 6$. Right: Surface plot and contours for the electrostatic potential of a charge, located in the center of the simulation volume. Picture shows the xy-plane for $z = L/2$. Parameters were $R_c = 0.25\,L$, $\alpha L = 12.2$ and $k_{max} L/2\pi = 6$.

The present form of the Ewald sum gives an exact representation of the potential energy of point like charges in a system with periodic boundary conditions. Sometimes the charge distribution in a molecule is approximated by a point dipole or higher multipole moments. A more general form of the Ewald sum, taking into account arbitrary point multipoles was given in Ref.[73]. The case, where also electronic polarizabilities are considered is given in Ref.[74].

In certain systems, like in molten salts or electrolyte solutions, the interaction between charged species may approximated by a screened Coulomb potential, which has a Yukawa-like form

$$U = \frac{1}{2} \sum_{i,j=1}^{N} q_i q_j \frac{e^{-\kappa \|\mathbf{r}_{ij}\|}}{\|\mathbf{r}_{ij}\|} \tag{21}$$

The parameter $\kappa$ is the inverse Debye length, which gives a measure of screening strength in the system. If $\kappa < 1/L$ the potential is short ranged and usual cut-off methods may

be used. Instead, if $\kappa > 1/L$, or generally if $u(r = L/2)$ is larger than the prescribed uncertainties in the energy, the minimum image convention in combination with truncation methods fails and the potential must be treated in a more rigorous way, which was proposed in Ref.[75], where an extension of the Ewald sum for such Yukawa type potentials was developed.

### 2.2.2 The Fast Multipole Method

In open geometries there is no lattice summation, but only the sum over all particle pairs in the whole system. The electrostatic energy at a particle's position is therefore simply calculated as

$$\phi(\mathbf{r}_i) = \sum_{j=1}^{N} \frac{q_j}{\|\mathbf{r}_i - \mathbf{r}_j\|} \tag{22}$$

Without further approximation this is always an $\mathcal{O}(N^2)$ algorithm since there are $N(N-1)/2$ interactions to consider in the system (here Newton's third law was taken into account). The idea of a multipole method is to group particles which are far away from a tagged particle together and to consider an effective interaction of a particle with this particle group[76–78]. The physical space is therefore subdivided in a hierarchical way, where the whole system is considered as level 0. Each further level is constructed by dividing the length in each direction by a factor of two. The whole system is therefore subdivided into a hierarchy of boxes where each *parent box* contains eight *children boxes*. This subdivision is performed at maximum until the level, where each particle is located in an individual box. Often it is enough to stop the subdivision already at a lower level.

In the following it is convenient to work in spherical coordinates. The main principle of the method is that the interaction between two particles, located at $\mathbf{r} = r, \theta, \varphi$ and $\mathbf{a} = (a, \alpha, \beta)$ can be written as a multipole expansion[79]

$$\frac{1}{\|\mathbf{r} - \mathbf{a}\|} = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{(l-|m|)!}{(l+|m|)!} \frac{a^l}{r^{l+1}} P_{lm}(\cos\alpha)\, P_{lm}(\cos\theta)\, e^{-im(\beta-\varphi)} \tag{23}$$

where $P_{lm}(x)$ are associated Legendre polynomials[80]. This expression requires that $a/r < 1$ and this gives a lower limit for the so called *well separated* boxes. This makes it necessary to have at least one box between a tagged box and the zone, where contributions can be expanded into multipoles. Defining the operators

$$O_{lm}(\mathbf{a}) = a^l\, (l-|m|)!\, P_{lm}(\cos\alpha)\, e^{-im\beta} \tag{24}$$

$$M_{lm}(\mathbf{r}) = \frac{1}{r^{l+1}} \frac{1}{(l+|m|)!}\, P_{lm}(\cos\theta)\, e^{im\varphi} \tag{25}$$

with which Eq. (23) may simply be rewritten in a more compact way, it is possible to write further three operators, which are needed, in a compact scheme, i.e.

1.) a translation operator, which relates the multipole expansion of a point located at $\mathbf{a}$ to a multipole expansion of a point located at $\mathbf{a} + \mathbf{b}$

$$O_{lm}(\mathbf{a}+\mathbf{b}) = \sum_{j=0}^{l} \sum_{k=-l}^{j} A_{jk}^{lm}(\mathbf{b})\, O_{jk}(\mathbf{a}) \qquad , \qquad A_{jk}^{lm}(\mathbf{b}) = O_{l-j,m-k}(\mathbf{b}) \tag{26}$$

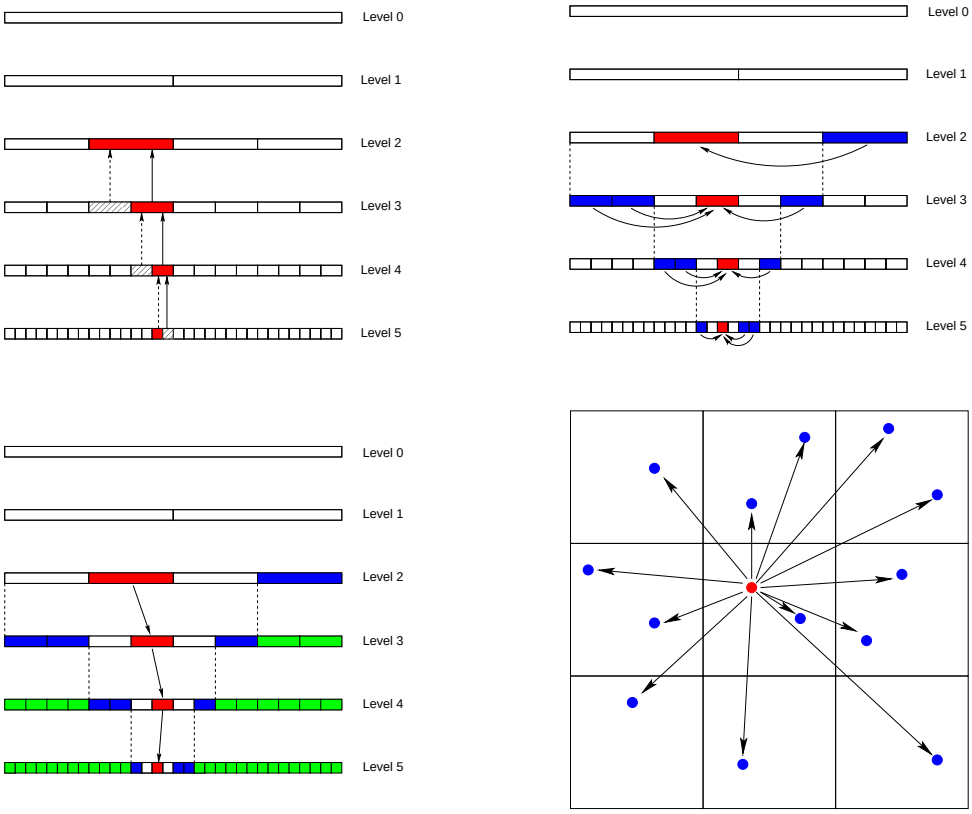


Figure 6. Schematic of different passes in the Fast Multipole Method. Upper left: Pass 1, evaluation of multipole terms in finest subdivision and translating information upwards the tree. Upper right: Pass 2, transforming multipole expansions in well separated boxes into local Taylor expansions. Lower left: Pass 3, transferring multipole expansions downwards the tree, thus collecting information of the whole system, except nearest neighbor boxes. Lower right: Pass 5, direct calculation of particle-particle interactions in local and nearest neighbor boxes.

2.) a transformation operator, which transforms a multipole expansion centered at the origin into a Taylor expansion centered at location $\mathbf{b}$

$$M_{lm}(\mathbf{a}-\mathbf{b}) = \sum_{j=0}^{l}\sum_{k=-l}^{j} B_{jk}^{lm}(\mathbf{b})\, O_{jk}(\mathbf{a}) \qquad , \qquad B_{jk}^{lm}(\mathbf{b}) = M_{l+j,m+k}(\mathbf{b}) \qquad (27)$$

3.) a translation operator, which translates a Taylor expansion of a point $\mathbf{r}$ about the origin into a Taylor expansion of $\mathbf{r}$ about a point $\mathbf{b}$

$$M_{lm}(\mathbf{r}-\mathbf{b}) = \sum_{j=0}^{l}\sum_{k=-l}^{j} C_{jk}^{lm}(\mathbf{b})\, M_{jk}(\mathbf{r}) \qquad , \qquad C_{jk}^{lm}(\mathbf{b}) = A_{jk}^{lm}(\mathbf{b}) \qquad (28)$$

The procedure to calculate interactions between particles is then subdivided into five passes. Figure 6 illustrates four of them. The first pass consists of calculating the multipole expansions in the lowest level boxes (finest subdivision). Using the translation operator $O_{lm}(\mathbf{a}+\mathbf{b})$, the multipole expansions are translated into the center of their parent boxes and summed up. This procedure is repeated then subsequently for each level, until level 2 is reached, from where no further information is passed to a coarser level. In pass 2, using operator $M_{lm}(\mathbf{a}-\mathbf{b})$, multipole expansions are translated into Taylor expansions in a box from well separated boxes, whose parent boxes are nearest neighbor boxes. Well separated means, that for all particles in a given box the multipole expansion in a separated box is valid. Since the applicability of Eq. (23) implies $r > a$, well separateness means on level $l$ that boxes should be separated by a distance $2^{-l}$. This also explains, why there is no need to transfer information higher than level 2, since from there on it is not possible to have well separated boxes anymore, i.e. multipole expansions are not valid any more. In pass 3, using the operator $M_{lm}(\mathbf{a}-\mathbf{b})$, this information is then translated downwards the tree, so that finally on the finest level all multipole information is known in order to interact individual particles with expansions, originating from all other particles in the system which are located in well separated boxes of the finest level. In pass 4 this interaction between individual particles and multipoles is performed. Finally in pass 5, explicit pair-pair interactions are calculated between particles in a lowest level box and those which are in nearest neighbor boxes, i.e. those boxes which are not called well separated.

It can be shown[61] that each of the steps performed in this algorithm is of order $\mathcal{O}(N)$, making it an optimal method. Also the error made by this method can be controlled rather reliably[64]. A very conservative error estimate is thereby given as[76,61,81]

$$\left|\phi(r) - \frac{q}{\|\mathbf{r}-\mathbf{a}\|}\right| \leq \frac{|q|}{r-a}\left(\frac{a}{r}\right)^{p+1} \tag{29}$$

At the current description the evaluation of multipole terms scales as $\mathcal{O}(l_{max}^4)$, when $l_{max}$ is the largest value of $l$ in the multipole expansion, Eq.(23). A faster version which scales as $\mathcal{O}(l_{max}^3)$ and therefore strongly reducing the prefactor of the overall scheme, was proposed in Ref.[62], where multipoles are evaluated in a rotated coordinate frame, which makes it possible to reduce calculations to Legendre polynomials and not requiring associated Legendre polynomials.

Also to mention is that there are approaches to extend the Fast Multipole Method to periodic systems[82,83].

### 2.3 Coarse Grain Methods

The force field methods mentioned so far treat molecules on the atomic level, i.e. resolving heavy atoms, in most cases also hydrogens, explicitly. In the case, where flexible molecular bonds, described e.g. by harmonic potentials, are considered the applied time step is of the order of $\delta t \approx 10^{-15}\, secs$. Considering physical phenomena like self assembling of lipid molecules[84,85], protein folding or structure formation in macromolecular systems[86–88], which take place on time scales of microseconds to seconds or even longer, the number of timesteps would exceed the current computational capacities. Although these phenomena all have an underlying microscopic background, the fast dynamics of e.g. hydrogen vibrations are not directly reflected in the overall process. This lead to the

idea to either freeze certain degrees of freedom, as it is done for e.g. rigid water models[89–92], or to take several degrees of freedom only into account effectively via a pseudo potential, which reflects the behavior of whole groups of atoms. It is the latter approach which is now known as coarse graining[28,93,94] of molecular potentials and which opens the accessibility of a larger time and length scale. Mapping groups of atoms to one pseudo atom, or interaction site, leads already to an effective increase of the specific volume of the degrees of freedom. Therefore, the same number of degrees of freedom of a coarse grain model, compared with a conventional force field model, would directly lead to larger spatial scale, due to the increase of volume of each degree of freedom. On the other hand, comparing a conventional system before and after coarse graining, the coarse grained system could cover time scales longer by a factor of 100-1000 or even longer compared with a conventional force field all-atom model (the concrete factor certainly depends on the level of coarse graining).

Methodologies for obtaining coarse grain models of a system often start from an atomistic all-atom model, which adequately describes phase diagrams or other physical properties of interest. On a next level, groups of atoms are collected and an effective non-bonded interaction potential may be obtained by calculating potential energy surfaces of these groups and to parametrize these potentials to obtain analytical descriptions. Therefore, distribution functions of small atomic groups are taken into account (at least implicitly) which in general depend on the thermodynamic state point. For bonded potentials between groups of atoms, a normal mode analysis may be performed in order to get the most important contributions to vibrational-, bending- or torsional-modes.

In principle, one is interested in reducing the number of degrees of freedom by separating the problem space into coordinates which are *important* and those which are *unimportant*. Formally, this may be expressed through a set of coordinates $\{\mathbf{r}\} \in \mathbb{R}^{n_i}$ and $\{\tilde{\mathbf{r}}\} \in \mathbb{R}^{n_u}$, where $n_i$ and $n_u$ are the number of degrees of important and unimportant degrees of freedom, respectively. Consequently, the system Hamiltonian may be written as $H = H(r_1, \ldots, r_{n_i}, \tilde{r}_1, \ldots, \tilde{r}_{n_i})$. From these considerations one may define a *reduced* partition function, which results from integrating out all unimportant degrees of freedom

$$Z = \int dr_1, \ldots, dr_{n_i}, d\tilde{r}_1, \ldots, d\tilde{r}_{n_i} \exp\left\{-\beta H(r_1, \ldots, r_{n_i}, \tilde{r}_1, \ldots, \tilde{r}_{n_u})\right\} \quad (30)$$

$$= \int dr_1, \ldots, dr_{n_i}, d\tilde{r}_1, \ldots, d\tilde{r}_{n_u} \exp\left\{-\beta H^{CG}(r_1, \ldots, r_{n_i})\right\} \quad (31)$$

where a coarse grain Hamiltonian has been defined

$$H^{CG}(r_1, \ldots, r_{n_i}) = -\log \int \tilde{r}_1, \ldots, d\tilde{r}_{n_i} \exp\left\{-\beta H(r_1, \ldots, r_{n_i}, \tilde{r}_1, \ldots, \tilde{r}_{n_u})\right\} \quad (32)$$

which corresponds to the potential of mean force and which is the free energy of the non-important degrees of freedom. Since the Hamiltonian describes only a subset of degrees of freedom, thermodynamic properties, derived from this Hamiltonian will be different than obtained from the full Hamiltonian description (e.g. pressure will correspond to the osmotic pressure and not to the thermodynamic pressure). This has to be taken into account when simulating in different ensembles or if experimental thermodynamic properties should be reproduced by simulation.

The coarse grained Hamiltonian is still a multi-body description of the system, which is hard to obtain numerically. Therefore, it is often approximated by a pair-potential, which

is considered to contribute the most important terms

$$H^{CG}(r_1, \ldots, r_{n_i}) \approx \sum_{i>j} V_{ij}(r_{ij}) \qquad , \qquad r_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\| \tag{33}$$

According to the uniqueness theorem of Henderson[95], in a liquid where particles interact only through pair interactions, the pair distribution function $g(r)$ determines up to a constant uniquely the pair interaction potential $V_{ij}$. Therefore, $V_{ij}$ may be obtained pointwise by reverting the radial pair distribution function[96–98], e.g. by reverse Monte Carlo techniques[99] or dynamic iterative refinement[100]. This approach directly confirms what was stated in Sec. 1 about the limited applicability of coarse grained potentials. It is clear that for different temperatures, pressures or densities the radial distribution functions of e.g. cation-cation, cation-anion and anion-anion distributions in electrolytic solutions will be different. If one wants to simulate ions in an effective medium (continuum solvent), the potential, which is applied in the simulation will depend on the thermodynamic state point and therefore has to be re-parametrized for every different state point.

## 3 The Integrator

The propagation of a classical particle system can be described by the temporal evolution of the phase space variables $(\mathbf{p}, \mathbf{q})$, where the phase space $\Gamma(\mathbf{p}, \mathbf{q}) \in \mathbb{R}^{6N}$ contains all possible combinations of momenta and coordinates of the system. The exact time evolution of the system is thereby given by a flow map

$$\Phi_{\delta t, \mathcal{H}} : \mathbb{R}^{6N} \rightarrow \mathbb{R}^{6N} \tag{34}$$

which means

$$\Phi_{\delta t, \mathcal{H}}(\mathbf{p}(t), \mathbf{q}(t)) = (\mathbf{p}(t) + \delta\mathbf{p}, \mathbf{q}(t) + \delta\mathbf{q}) \tag{35}$$

where

$$\mathbf{p} + \delta\mathbf{p} = \mathbf{p}(t + \delta t) \qquad , \qquad \mathbf{q} + \delta\mathbf{q} = \mathbf{q}(t + \delta t) \tag{36}$$

For a nonlinear many-body system, the equations of motion cannot be integrated exactly and one has to rely on numerical integration of a certain order. Propagating the coordinates by a constant step size $h$, a number of different finite difference schemes may be used for the integration. But there are a number of requirements, which have to be fulfilled in order to be useful for molecular dynamics simulations. An integrator, suitable for many-body simulations should fulfill the following requirements:

- Accuracy, i.e. the solution of an analytically solvable test problem should be as close as possible to the numerical one.
- Stability, i.e. very long simulation runs should produce physically relevant trajectories, which are not governed by numerical artifacts
- Conservativity, there should be no drift or divergence in conserved quantities, like energy, momentum or angular momentum

- Reversibility, i.e. it should have the same temporal structure as the underlying equations
- Effectiveness, i.e. it should allow for large time steps without entering instability and should require a minimum of force evaluations, which usually need about 95 % of CPU time per time step
- Symplecticity, i.e. the geometrical structure of the phase space should be conserved

It is obvious that the numerical flow, $\phi_{\delta t,\mathcal{H}}$, of a finite difference scheme will not be fully equivalent to $\Phi_{\delta t,\mathcal{H}}$, but the system dynamics will be described correctly if the items above will be fulfilled.

In the following the mentioned points will be discussed and a number of different integrators will be compared.

### 3.1 Basic Methods

The most simple integration scheme is the Euler method, which may be constructed by a first order difference approximation to the time derivative of the phase space variables

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \delta t \frac{\partial}{\partial \mathbf{q}} \mathcal{H}(\mathbf{p}_n, \mathbf{q}_n) \tag{37}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \delta t \frac{\partial}{\partial \mathbf{p}} \mathcal{H}(\mathbf{p}_n, \mathbf{q}_n) \tag{38}$$

where $\delta t$ is the step size of integration. This is equivalent to a Taylor expansion which is truncated after the first derivative. Therefore, it is obvious that it is of first order. Knowing all variables at step $n$, this scheme has all relevant information to perform the integration. Since only information from one time step is required to do the integration, this scheme is called the one-step explicit Euler scheme. The basic scheme, Eqs. (37,38) may also be written in different forms.

The implicit Euler method

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \delta t \frac{\partial}{\partial \mathbf{q}} \mathcal{H}(\mathbf{p}_{n+1}, \mathbf{q}_{n+1}) \tag{39}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \delta t \frac{\partial}{\partial \mathbf{p}} \mathcal{H}(\mathbf{p}_{n+1}, \mathbf{q}_{n+1}) \tag{40}$$

can only be solved iteratively, since the derivative on the right-hand-side (*rhs*) is evaluated at the coordinate positions on the left-hand-side (*lhs*).

An example for a so called partitioned Runge-Kutta method is the *velocity implicit method*

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \delta t \frac{\partial}{\partial \mathbf{q}} \mathcal{H}(\mathbf{p}_{n+1}, \mathbf{q}_n) \tag{41}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \delta t \frac{\partial}{\partial \mathbf{p}} \mathcal{H}(\mathbf{p}_{n+1}, \mathbf{q}_n) \tag{42}$$

Since the Hamiltonian usually splits into kinetic $\mathcal{K}$ and potential $\mathcal{U}$ parts, which only depend on one phase space variable, i.e.

$$\mathcal{H}(\mathbf{p}, \mathbf{q}) = \frac{1}{2} \mathbf{p}^T \, \mathbf{M}^{-1} \, \mathbf{p} + \mathcal{U}(\mathbf{q}) \tag{43}$$

where $\mathbf{M}^{-1}$ is the inverse of the diagonal mass matrix, this scheme may also be written as

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \delta t \frac{\partial}{\partial \mathbf{q}} \mathcal{U}(\mathbf{q}_n) \tag{44}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \frac{\delta t}{m} \mathbf{p}_{n+1} \tag{45}$$

showing that it is not necessary to solve it iteratively.

Obviously this may be written as a *position implicit method*

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \delta t \frac{\partial}{\partial \mathbf{q}} \mathcal{U}(\mathbf{q}_{n+1}) \tag{46}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \frac{\delta t}{m} \mathbf{p}_n \tag{47}$$

Applying first Eq. (47) and afterwards Eq. (46) also this variant does not require an iterative procedure.

All of these schemes are first order accurate but have different properties, as will be shown below. Before discussing these schemes it will be interesting to show a higher order scheme, which is also based on a Taylor expansion. First write down expansions

$$\mathbf{q}(t+\delta t) = \mathbf{q}(t) + \delta t\, \dot{\mathbf{q}}(t) + \frac{1}{2}\, \delta t^2\, \ddot{\mathbf{q}}(t) + O(\delta t^3) \tag{48}$$

$$= \mathbf{q}(t) + \frac{\delta t}{m}\, \mathbf{p}(t) + \frac{1}{2m}\, \delta t^2\, \dot{\mathbf{p}}(t) + O(\delta t^3) \tag{49}$$

$$\mathbf{p}(t+\delta t) = \mathbf{p}(t) + \delta t\, \dot{\mathbf{p}}(t) + \frac{1}{2}\, \delta t^2\, \ddot{\mathbf{p}}(t) + O(\delta t^3) \tag{50}$$

$$= \mathbf{p}(t) + \frac{\delta t}{2}\, (\dot{\mathbf{p}}(t) + \dot{\mathbf{p}}(t+\delta t)) + O(\delta t^3) \tag{51}$$

where in Eq. (49), the relation $\dot{\mathbf{q}} = \mathbf{p}/m$ was used and in Eq. (51) a first order Taylor expansion for $\dot{\mathbf{p}}$ was inserted. From these expansions a simple second order, one-step splitting scheme may be written as

$$\mathbf{p}_{n+1/2} = \mathbf{p}_n + \frac{\delta t}{2}\, \mathbf{F}(\mathbf{q}_n) \tag{52}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \frac{\delta t}{m}\, \mathbf{p}_{n+1/2} \tag{53}$$

$$\mathbf{p}_{n+1} = \mathbf{p}_{n+1/2} + \frac{\delta t}{2}\, \mathbf{F}(\mathbf{q}_{n+1}) \tag{54}$$

where the relation $\dot{\mathbf{p}} = -\partial \mathcal{H}/\partial \mathbf{q} = \mathbf{F}$ was used. This scheme is called the *Velocity Verlet* scheme. In a pictorial way it is sometimes described as half-kick, drift, half-kick, since the first step consists in applying forces for half a time step, second step consists in free flight of a particle with momentum $\mathbf{p}_{n+1/2}$ and the last step applies again a force for half a time step. In practice, forces only need to be evaluated once in each time step. After having calculated the new positions, $\mathbf{q}_{n+1}$, forces are calculated for the last integration step. They are, however, stored to be used in the first integration step as *old* forces in the next time step of the simulation.

This algorithm comes also in another flavor, called the *Position Verlet* scheme. It can be expressed as

$$\mathbf{q}_{n+1/2} = \mathbf{q}_n + \frac{\delta t}{2m}\,\mathbf{p}_n \tag{55}$$

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \delta t\,\mathbf{F}(\mathbf{q}_{n+1/2}) \tag{56}$$

$$\mathbf{q}_{n+1/2} = \mathbf{q}_{n+1/2} + \frac{\delta t}{2m}\,\mathbf{p}_{n+1} \tag{57}$$

In analogy to the description above this is sometimes described as half-drift, kick, half-drift. Using the relation $\mathbf{p} = \dot{\mathbf{q}}/m$ and expressing this as a first order expansion, it is obvious that $\mathbf{F}(\mathbf{q}_{n+1/2}) = \mathbf{F}((\mathbf{q}_n + \mathbf{q}_{n+1})/2)$ which corresponds to an implicit midpoint rule.

### 3.2 Operator Splitting Methods

A more rigorous derivation, which in addition leads to the possibility of splitting the propagator of the phase space trajectory into several time scales, is based on the phase space description of a classical system. The time evolution of a point in the $6N$ dimensional phase space is given by the Liouville equation

$$\mathbf{\Gamma}(t) = e^{i\mathcal{L}t}\,\mathbf{\Gamma}(0) \tag{58}$$

where $\mathbf{\Gamma} = (\mathbf{q}, \mathbf{p})$ is the $6N$ dimensional vector of generalized coordinates, $\mathbf{q} = \mathbf{q}_1, \ldots, \mathbf{q}_N$, and momenta, $\mathbf{p} = \mathbf{p}_1, \ldots, \mathbf{p}_N$. The Liouville operator, $\mathcal{L}$, is defined as

$$i\mathcal{L} = \{\ldots, \mathcal{H}\} = \sum_{j=1}^{N}\left(\frac{\partial \mathbf{q}_j}{\partial t}\frac{\partial}{\partial \mathbf{q}_j} + \frac{\partial \mathbf{p}_j}{\partial t}\frac{\partial}{\partial \mathbf{p}_j}\right) \tag{59}$$

In order to construct a discrete timestep integrator, the Liouville operator is split into two parts, $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2$, and a Trotter expansion[101] is performed

$$e^{i\mathcal{L}\delta t} = e^{i(\mathcal{L}_1+\mathcal{L}_2)\delta t} \tag{60}$$

$$= e^{i\mathcal{L}_1\delta t/2}e^{i\mathcal{L}_2\delta t}e^{i\mathcal{L}_1\delta t/2} + \mathcal{O}(\delta t^3) \tag{61}$$

The partial operators can be chosen to act only on positions and momenta. Assuming usual cartesian coordinates for a system of $N$ free particles, this can be written as

$$i\mathcal{L}_1 = \sum_{j=1}^{N}\mathbf{F}_j\frac{\partial}{\partial \mathbf{p}_j} \tag{62}$$

$$i\mathcal{L}_2 = \sum_{j=1}^{N}\mathbf{v}_j\frac{\partial}{\partial \mathbf{r}_j} \tag{63}$$

Applying Eq.60 to the phase space vector $\mathbf{\Gamma}$ and using the property $e^{a\partial/\partial x}f(x) = f(x+a)$ for any function $f$, where $a$ is independent of x, gives

$$\mathbf{v}_i(t+\delta t/2) = \mathbf{v}(t) + \frac{\mathbf{F}_i(t)}{m_i}\frac{\delta t}{2} \tag{64}$$

$$\mathbf{r}_i(t+\delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t+\delta t/2)\delta t \tag{65}$$

$$\mathbf{v}_i(t+\delta t) = \mathbf{v}_i(t+\delta t/2) + \frac{\mathbf{F}_i(t+\delta t)}{m_i}\frac{\delta t}{2} \tag{66}$$

which is the velocity Verlet algorithm, Eqs. 52-54. In the same spirit, another algorithm may be derived by simply changing the definitions for $\mathcal{L}_1 \rightarrow \mathcal{L}_2$ and $\mathcal{L}_2 \rightarrow \mathcal{L}_1$. This gives the so called *position Verlet algorithm*

$$\mathbf{r}_i(t + \delta t/2) = \mathbf{r}_i(t) + \mathbf{v}(t)\frac{\delta t}{2} \tag{67}$$

$$\mathbf{v}_i(t + \delta t) = \mathbf{v}(t) + \frac{\mathbf{F}_i(t + \delta t/2)}{m_i} \tag{68}$$

$$\mathbf{r}_i(t + \delta t) = \mathbf{r}_i(t + \delta t/2) + (\mathbf{v}(t) + \mathbf{v}_i(t + \delta t))\frac{\delta t}{2} \tag{69}$$

Here the forces are calculated at intermediate positions $\mathbf{r}_i(t+\delta t/2)$. The equations of both the velocity Verlet and the position Verlet algorithms have the property of propagating velocities or positions on half time steps. Since both schemes decouple into an applied force term and a *free flight* term, the three steps are often called *half-kick*/*drift*/*half kick* for the velocity Verlet and correspondingly *half-drift*/*kick*/*half-drift* for the position Verlet algorithm.

Both algorithms, the velocity and the position Verlet method, are examples for symplectic algorithms, which are characterized by a volume conservation in phase space. This is equivalent to the fact that the Jacobian matrix of a transform $x' = f(x, p)$ and $p' = g(x, p)$ satisfies

$$\begin{pmatrix} f_x & f_p \\ g_x & g_p \end{pmatrix} \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \begin{pmatrix} f_x & f_p \\ g_x & g_p \end{pmatrix} = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} \tag{70}$$

Any method which is based on the splitting of the Hamiltonian, is symplectic. This does not yet, however, guarantee that the method is also time reversible, which may be also be considered as a strong requirement for the integrator. This property is guaranteed by symmetric methods, which also provide a better numerical stability[102]. Methods, which try to enhance the accuracy by taking into account the particles' history (multi-step methods) tend to be incompatible with symplecticness[103, 104], which makes symplectic schemes attractive from the point of view of data storage requirements. Another strong argument for symplectic schemes is the so called *backward error analysis*[105–107]. This means that the trajectory produced by a discrete integration scheme, may be expressed as the solution of a perturbed ordinary differential equation whose *rhs* can formally be expressed as a power series in $\delta t$. It could be shown that the system, described by the ordinary differential equation is Hamiltonian, if the integrator is symplectic[108, 109]. In general, the power series in $\delta t$ diverges. However, if the series is truncated, the trajectory will differ only as $\mathcal{O}(\delta t^p)$ of the trajectory, generated by the symplectic integrator on timescales $\mathcal{O}(1/\delta t)$[110].

### 3.3 Multiple Time Step Methods

It was already mentioned that the rigorous approach of the decomposition of the Liouville operator offers the opportunity for a decomposition of time scales in the system. Supposing that there are different time scales present in the system, e.g. fast intramolecular vibrations and slow domain motions of molecules, then the factorization of Eq.60 may be written in

a more general way

$$e^{i\mathcal{L}\Delta t} = e^{i\mathcal{L}_1^{(s)}\Delta t/2} e^{i\mathcal{L}_1^{(f)}\Delta t/2} e^{i\mathcal{L}_2\delta t} e^{i\mathcal{L}_1^{(f)}\Delta t/2} e^{i\mathcal{L}_1^{(s)}\Delta t/2} \tag{71}$$

$$= e^{i\mathcal{L}_1^{(s)}\Delta t/2} \left\{ e^{i\mathcal{L}_1^{(f)}\delta t/2} e^{i\mathcal{L}_2\delta t} e^{i\mathcal{L}_1^{(f)}\delta t/2} \right\}^p e^{i\mathcal{L}_1^{(s)}\Delta t/2} \tag{72}$$

where the time increment is $\Delta t = p\delta$. The decomposition of the Liouville operator may be chosen in the convenient way

$$i\mathcal{L}_1^{(s)} = \mathbf{F}_i^{(s)} \frac{\partial}{\partial \mathbf{p}_i} \quad , \quad i\mathcal{L}_1^{(f)} = \mathbf{F}_i^{(f)} \frac{\partial}{\partial \mathbf{p}_i} \quad , \quad i\mathcal{L}_2 = \mathbf{v}_i \frac{\partial}{\partial \mathbf{q}_i} \tag{73}$$

where the superscript $(s)$ and $(f)$ mean slow and fast contributions to the forces. The idea behind this decomposition is simply to take into account contributions from slowly varying components only every $p$'th timestep with a large time interval. Therefore, the force computation may be considerably speeded up in the the $p-1$ intermediate force computation steps. In general, the scheme may be extended to account for more time scales. Examples for this may be found in Refs.[111–113]. One obvious problem, however, is to separate the timescales in a proper way. The scheme of Eq.72 is *exact* if the time scales decouple completely. This, however, is very rarely found and most often timescales are coupled due to nonlinear effects. Nevertheless, for the case where $\Delta t$ is not very much larger than $\delta t$ ($p \approx 10$), the separation may be often justified and lead to stable results. Another criteria for the separation is to distinguish between long range and short range contributions to the force. Since the magnitude and the fluctuation frequency is very much larger for the short range contributions this separation makes sense for speeding up computations including long range interactions[114].

The method has, however, its limitations[115,116]. As described, a particle gets every $n$'th timestep a *kick* due to the slow components. It was reported in literature that this may excite a system's resonance which will lead to strong artifacts or even instabilities[117,118]. Recently different schemes were proposed to overcome these resonances by keeping the property of symplecticness[119–125].

### 3.4 Stability

Performing simulations of stable many-body systems for long times should produce configurations which are in thermal equilibrium. This means that system properties, e.g. pressure, internal energy, temperature etc. are fluctuating around constant values. To measure these equilibrium properties it should not be relevant where to put the time origin from where configurations are considered to calculate average quantities. This requires that the integrator should propagate phase space variables in such a way that small fluctuations do not lead to a diverging behavior of a system property. This is a kind of minimal requirement in order to simulate any physical system without a domination of numerical artifacts. It is clear, however, that any integration scheme will have its own stability range depending on the step size $\delta t$. This is a kind of sampling criterion, i.e. if the step size is too large, in order to resolve details of the energy landscape, an integration scheme may end in instability.

For linear systems it is straight forward to analyze the stability range of a given numerical scheme. Consider e.g. the harmonic oscillator, for which the equations of motion may be written as $\dot{q}(t) = p(t)$ and $\dot{p}(t) = -\omega^2 q(t)$, where $\omega$ is the vibrational frequency and it

is assumed that it oscillates around the origin. The exact solution of this problem may be written as

$$\begin{pmatrix} \omega\, q(t) \\ p(t) \end{pmatrix} = \begin{pmatrix} \cos\omega t & \sin\omega t \\ -\sin\omega t & \cos\omega t \end{pmatrix} \begin{pmatrix} \omega\, q(0) \\ p(0) \end{pmatrix} \tag{74}$$

For a numerical integrator the stepwise solution may be written as

$$\begin{pmatrix} \omega\, q_{n+1} \\ p_{n+1} \end{pmatrix} = \mathbf{M}(\delta t) \begin{pmatrix} \omega\, q_n \\ p_n \end{pmatrix} \tag{75}$$

where $\mathbf{M}(\delta t)$ is a propagator matrix. It is obvious that any stable numerical scheme requires eigenvalues $|\lambda(\mathbf{M})| \leq 1$. For $|\lambda| > 1$ the scheme will be unstable and divergent, for $|\lambda| < 1$ it will be stable but will exhibit friction, i.e. will loose energy. Therefore, in view of the conservativity of the scheme, it will be required that $|\lambda(\mathbf{M})| = 1$.

As an example the propagator matrices for the Implicit Euler (IE) and Position Verlet (PV) algorithms are calculated as

$$\mathbf{M}_{IE}(\delta t) = \frac{1}{1+\omega^2\delta t^2} \begin{pmatrix} 1 & \omega\delta t \\ -\omega\delta t & 1 \end{pmatrix} \tag{76}$$

$$\mathbf{M}_{PV}(\delta t) = \begin{pmatrix} 1 - \frac{1}{2}\omega^2\delta t^2 & \omega\delta t\left(1 - \frac{1}{4}\omega^2\delta t^2\right) \\ -\omega\delta t & 1 - \frac{1}{2}\omega^2\delta t^2 \end{pmatrix} \tag{77}$$

It is then straight forward to calculate the eigenvalues as roots of the characteristic polynomials. The eigenvalues are then calculated as

$$\lambda_{EE} = 1 \pm i\omega\delta t \tag{78}$$

$$\lambda_{IE} = \frac{1}{1+\omega^2\delta t^2}(1 \pm i\omega\delta t) \tag{79}$$

$$\lambda_{PV} = \lambda_{VV} = \lambda_{VIE} = \lambda_{PIE} = 1 - \frac{1}{2}\omega^2\delta t^2 \left(1 \pm \sqrt{1 - \frac{4}{\omega^2\delta t^2}}\right) \tag{80}$$

This shows that the absolute values for the Explicit Euler (EE) and the Implicit Euler methods never equals one for $\delta t \neq 0$, i.e. both methods do not produce stable trajectories. This is different for the Position Verlet, the Velocity Verlet (VV), the Position Implicit Euler (PIE) and the Velocity Implicit Euler (VIE), which all have the same eigenvalues. It is found that the range of stability for all of them is in the range $\omega^2\delta t^2 < 2$. For larger values of $\delta t$ the absolute values of the eigenvalues bifurcates, getting larger and smaller values than one. In Figure 7 the absolute values are shown for all methods and in in Figure 8 the imaginary versus real parts of $\lambda$ are shown. For EE it is clear that the imaginary part diverges linearly with increase of $\delta t$. The eigenvalues of the stable methods are located on a circle until $\omega^2\delta t^2 = 2$. From there one branch diverges to $-\infty$, while the other decreases to zero.

As a numerical example the phase space trajectories of the harmonic oscillator for $\omega = 1$ are shown for the different methods in Figure 9. For the stable methods, results

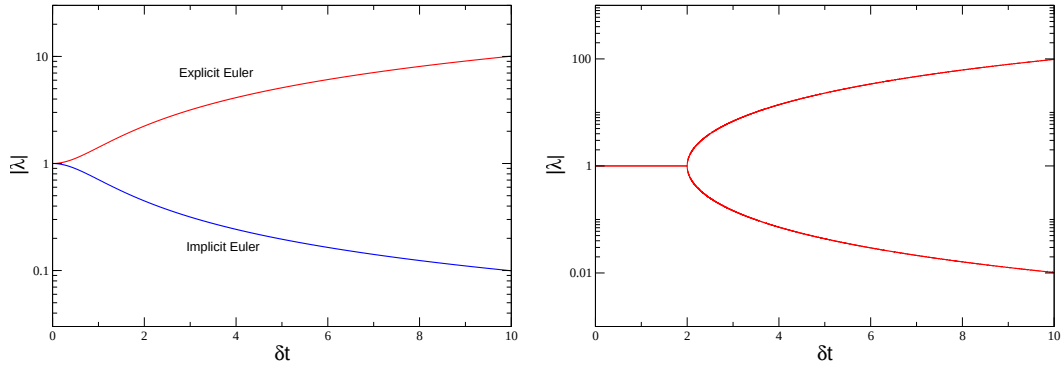


Figure 7. Absolute value of the eigenvalues $\lambda$ as function of the time step $\delta t$. Left: Explicit and implicit Euler method. Right: Velocity and Position Verlet as well as Velocity Implicit and Position implicit Euler method. All methods have the eigenvalues.

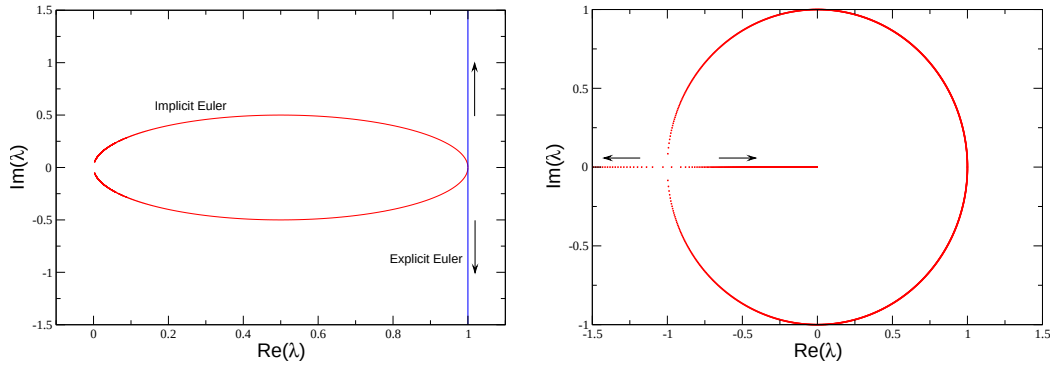


Figure 8. Imaginary versus real part of eigenvalues $\lambda$ of the propagator matrices. Left: Implicit and Explicit Euler. Right: Velocity and Position Verlet as well as Velocity Implicit and Position implicit Euler method.

for a time step close to instability is shown. All different methods produce closed, stable orbits, but it is seen on the other hand that they strongly deviate from the exact solution, which is shown for reference. This demonstrates that stability is a necessary, but only a weak criterion for correct results. Numerically correct results are only obtained for much smaller time steps in the range of $\delta t \approx 0.01$. Also shown are the results for EE and IE. Here a very much smaller time step, $\delta t = 0.01$ is chosen. It is seen that the phase space trajectory of EE spirals out while the one of IE spirals in with time, showing the instable or evanescent character of the methods.

Another issue related to stability is the effect of a trajectory perturbation. If initial conditions are slightly perturbed, will a good integrator keep this trajectory close to the reference trajectory? The answer is No and it is even found that the result is not that strong dependent on the integrator. Even for integrators of high order, trajectories will not stay close to each other. The time evolution of the disturbance may be studied similar to the system trajectory. Consider the time evolution for $\Gamma + \delta\Gamma$, where $\Gamma = (\mathbf{p}, \mathbf{q})$ and

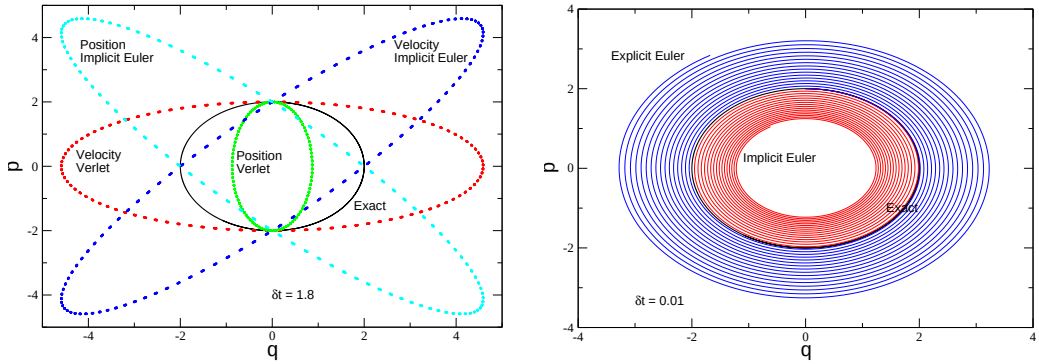


Figure 9. Phase space trajectories for the one-dimensional harmonic oscillator, integrated with the Velocity Implicit Euler, Position Implicit Euler, Velocity Verlet, Position Verlet and integration step size of $\delta t = 1.8$ (left) and the Implicit Euler and Explicit Euler and step size $\delta t = 0.01$ (right).

$\delta\Gamma = (\delta\mathbf{p}, \delta\mathbf{q})$ is a small disturbance. Then

$$\frac{d\Gamma}{dt} = \nabla_\Gamma \mathcal{H}(\Gamma) \tag{81}$$

Similarly one can write for small $\delta\Gamma$

$$\frac{d}{dt}(\Gamma + \delta\Gamma) = \nabla_\Gamma \mathcal{H}(\Gamma + \delta\Gamma) \tag{82}$$

$$= \nabla_\Gamma \mathcal{H}(\Gamma) + \nabla_\Gamma(\nabla_\Gamma \mathcal{H}(\Gamma))\delta\Gamma \tag{83}$$

where the second line is a truncated Taylor series. Comparing terms one simply gets as equation of motion for a perturbation

$$\frac{d\delta\Gamma}{dt} = \nabla_\Gamma^2 \mathcal{H}(\Gamma)\delta\Gamma \tag{84}$$

It is found that the disturbance develops exponentially, with a characteristic, system dependent exponent, which is the Ljapunov exponent[126, 127].

Now consider the following situation where identical starting configurations are taken for two simulations. They will be carried out by different yet exact algorithms, therefore leading formally to the same result. Nevertheless it may happen that different orders of floating-point operations are used in both algorithms. Due to round off errors, floating-point arithmetic is not necessarily associative, i.e. in general

$$a \mathbin{\hat{\circ}} (b \mathbin{\hat{\circ}} c) \neq (a \mathbin{\hat{\circ}} b) \mathbin{\hat{\circ}} c \tag{85}$$

where $\mathbin{\hat{\circ}}$ is a floating-point machine operation $(+, -, /, *)$. Therefore, both simulations will be different by round off errors. According to the above discussion, this may be considered as the slightest disturbance of a system trajectory, $\delta\Gamma_{\min}$, and the question is, what effect such a round off error will have. A different method to study difference in system trajectories is the calculation of the difference

$$\gamma_x(t) = \frac{1}{3N}\sum_{i=1}^{N}\sum_{\alpha=x,y,z}(x(t) - \tilde{x}(t))^2 \tag{86}$$

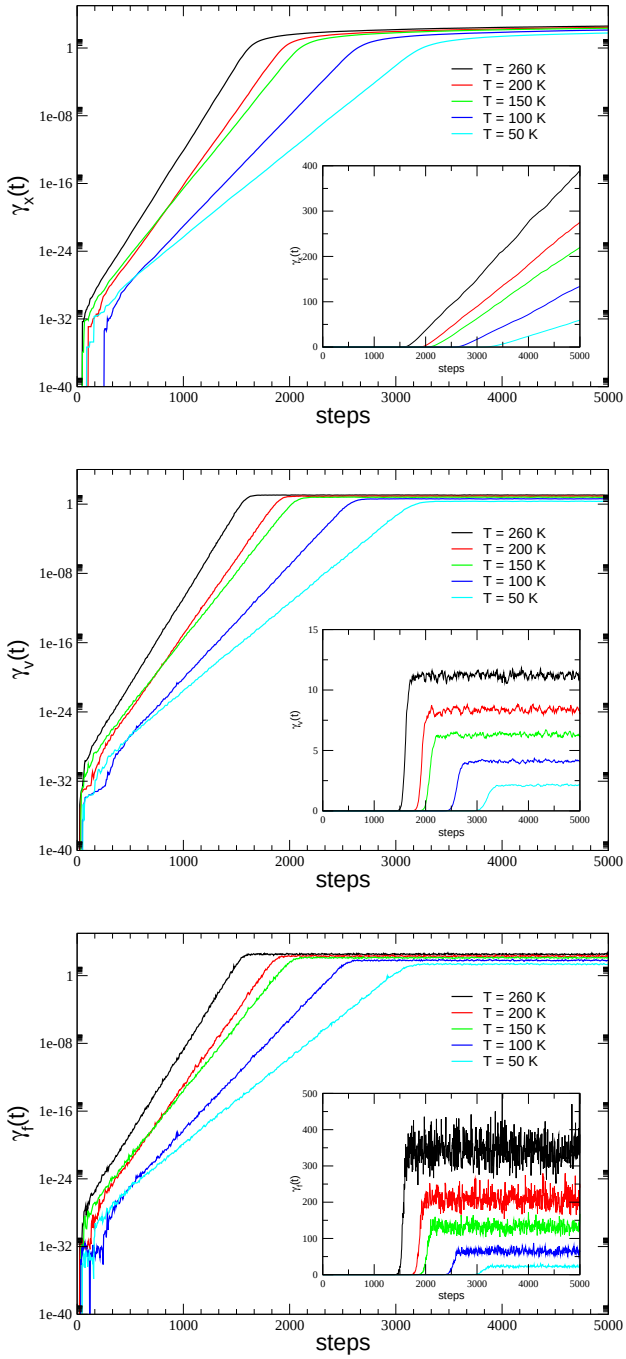


Figure 10. Divergent behavior of trajectories due to round off errors, induced by different summation order in the force routine. From top to bottom: coordinates, velocities, forces. The insets show on a linear scale the long time behavior of the trajectory differences, i.e. when the two systems get uncorrelated.

where $N$ is the number of particles, $x(t)$ a certain property, e.g. the coordinates or momenta, and $\tilde{x}$ the same property of a disturbed trajectory. In Figure 10 results are shown for a system of Lennard-Jones particles, where the disturbance was induced by reversing the order of summation in the force routine, thereby provoking round off errors in the first time step. Shown are results for the coordinates, the velocities and the forces and it is seen that all quantities diverge exponentially from machine accuracy up to a certain behavior at long times, which is shown in the inset. To understand the long time behavior, $\gamma_x(t)$ can be written as average property

$$\begin{aligned}\gamma_x(t) &= \langle (x(t) - x(0) - \tilde{x}(t) + x(0))^2 \rangle & (87)\\ &= \langle |x(t) - x(0)|^2 \rangle + \langle |\tilde{x}(t) - x(0)|^2 \rangle & (88)\\ &\quad -2\langle x(t)\tilde{x}(t) \rangle + 2\langle x(0)\tilde{x}(t) \rangle + 2\langle x(t)x(0) \rangle - 2\langle x(0)^2 \rangle\end{aligned}$$

In the second equation the first two terms are mean square displacements of $x$ in the two systems (note that $\tilde{x}(0) = x(0)$ since the same starting configurations are used), the next term is a cross correlation between the systems. This will vanish if the systems become independent of each other. The next two systems consist of auto-correlation functions of $x$ in each system. For long times they will also decrease to zero. Finally, the last term gives a constant offset which does not depend on time. Therefore the long time behavior will be governed for coordinates, momenta and forces by

$$\lim_{t\to\infty} \gamma_q(t) = 2\langle |\mathbf{q}(t) - \mathbf{q}(0)|^2 \rangle = 12Dt \quad (89)$$

$$\lim_{t\to\infty} \gamma_p(t) = 2\langle \mathbf{p}(t)^2 \rangle = mk_BT \quad (90)$$

$$\lim_{t\to\infty} \gamma_f(t) = 2\langle \mathbf{F}(t)^2 \rangle = 2(\nabla\mathcal{W})^2 \quad (91)$$

where $D$ is the diffusion coefficient, $T$ the temperature and $\mathcal{W}$ the potential of mean force.

That the divergent behavior of neighbored trajectories is a system dependent property is shown in Figure 10 where results for Lennard-Jones systems at different temperatures are shown.

In conclusion, the individual trajectories of a physical complex system will end up at different places in phase space when introducing round off errors or small perturbations. Round off errors cannot be avoided with simple floating-point arithmetic (only discrete calculations are able to avoid round off errors; but then the physical problem is transformed into a different space). Since one cannot say anything about a *true* summation order, the location in phase space cannot have an absolute meaning. Therefore, the solution to come out of this dilemma is to interpret the phase space location as a *possible* and *allowed* realization of the system, which makes it necessary, however, to average over a lot of possible realizations.

### 3.5 Accuracy

For an integrator of order $p \geq 1$, the local error may be written as an upper bound[8]

$$\|\Phi_{\delta t,\mathcal{H}} - \phi_{\delta t}\| \leq M\delta t^{p+1} \quad (92)$$

where $M > 0$ is a constant, $\Phi_{\delta t,\mathcal{H}}$ is the exact and $\phi_{\delta t}$ the numerical flow of the system. The global error, i.e. the accumulated error for larger times, is thereby bound for stable

methods by[8]

$$\|\Gamma(t_n) - \Gamma_n\| \leq K\left(e^{Lt_n} - 1\right)\, \delta t^p \qquad , \qquad t_n = n\delta t \tag{93}$$

where $K > 0$ is a constant, $L > 0$ the Lipschitz constant, $\Gamma(t_n) = (\mathbf{p}(t_n), \mathbf{q}(t_n))$ the exact and $\Gamma_n = (\mathbf{p}_n, \mathbf{q}_n)$ the numerically computed trajectory at time $t_n$. This estimate gives of course not too much information for $Lt_n 1$ unless $\delta t$ is chosen very small. Nevertheless, qualitatively this estimate shows a similar exponential divergent behavior of numerical and exact solution for a numerical scheme, as was observed in Section 3.4.

A different approach to the error behavior of a numerical scheme is backward error analysis, first mentioned in Ref.[128] in the context of differential equations. The idea is to consider the numerical solution of a given scheme as the exact solution of a modified equation. The comparison of the original and the modified equation then gives qualitative insight into the long time behavior of a given scheme.

It is assumed that the numerical scheme can be expressed as a series of the form

$$\phi_{\delta t}(\Gamma_n) = \Gamma_n + \delta t f(\Gamma) + \delta t^2 g_2(\Gamma) + \delta t^3 g_3(\Gamma) \pm \dots \tag{94}$$

where the $g_i$ are known coefficients and for consistency of the differential equation it must hold

$$f(\Gamma) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \nabla_p \\ \nabla_q \end{pmatrix} \mathcal{H}(\mathbf{p}, \mathbf{q}) \tag{95}$$

On the other hand it is assumed that there exists a modified differential equation of the form

$$\frac{d}{dt}\tilde{\Gamma} = f(\tilde{\Gamma}) + \delta t f_2(\tilde{\Gamma}) + \delta t^2 f_3(\tilde{\Gamma}) + \dots \tag{96}$$

where $\tilde{\Gamma}$ will be equivalent to the numerically obtained solution. In order to construct the modified equation, the solution of Eq. (96) is Taylor expanded, i.e.

$$\begin{aligned}
\tilde{\Gamma}(t+\delta t) = \tilde{\Gamma}(t) &+ \delta t \left(f(\tilde{\Gamma}) + \delta t f_2(\tilde{\Gamma}) + \delta t^2 f_3(\tilde{\Gamma}) + \dots\right) \\
&+ \frac{\delta t^2}{2!}\left(f'(\tilde{\Gamma}) + \delta t f_2'(\tilde{\Gamma}) + \dots\right)\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\left(f(\tilde{\Gamma}) + \delta t f_2(\tilde{\Gamma}) + \dots\right) \\
&+ \frac{\delta t^3}{3!}\Bigg\{\left(f''(\tilde{\Gamma}) + \delta t f_2''(\tilde{\Gamma}) + \dots\right)\left(\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\left(f(\tilde{\Gamma}) + \delta t f_2(\tilde{\Gamma}) + \dots\right)\right)^2 \\
&\quad + \left(f'(\tilde{\Gamma}) + \delta t f_2'(\tilde{\Gamma}) + \dots\right)\left(\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\left(f'(\tilde{\Gamma}) + \delta t f_2'(\tilde{\Gamma}) + \dots\right)\right) \\
&\quad \times \left(f(\tilde{\Gamma}) + \delta t f_2(\tilde{\Gamma}) + \dots\right)\Bigg\} \\
&+ \dots
\end{aligned} \tag{97}$$

The procedure to construct the unknown functions $f_i$ proceeds in analogy to perturbation theory, i.e. coefficients with same powers of $\delta t$ are collected which leads to a recursive scheme to solve for all unknowns.

To give an example the Lennard-Jones oscillator is considered, i.e. a particle performing stable motions in negative part of a Lennard-Jones potential. As was observed already for the harmonic oscillator, the Explicit Euler method will gain energy during the time,

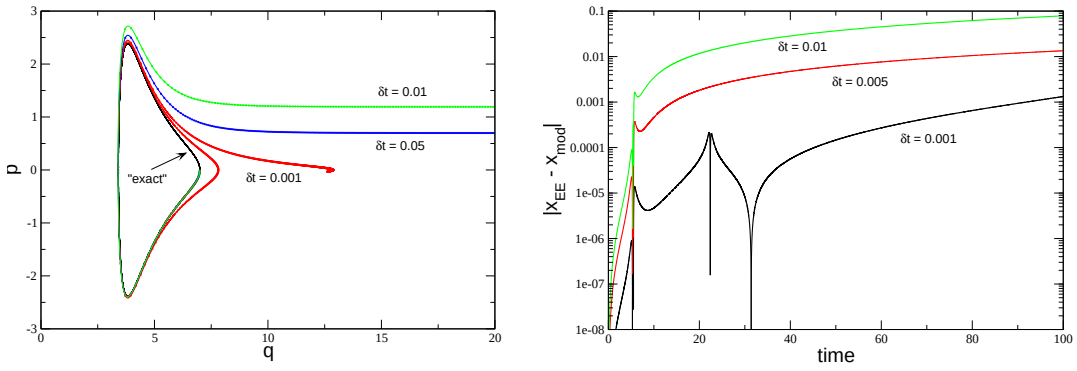


Figure 11. Phase space trajectories of the Lennard-Jones oscillator calculated with the Explicit Euler scheme and different time steps of integration. The *exact* solution (numerical solution of a high order composition scheme with small time step) is shown as a reference - it forms closed orbits. Superimposed to the solutions are results, obtained with a Velocity Verlet scheme, applied to the modified equations, Eqs. (98,99). The right figure shows the differences in coordinates between the calculation with Explicit Euler scheme applied to Lennard-Jones oscillator and Velocity Verlet applied to the modified equation, $|\mathbf{q}_{EE}(t) - \mathbf{q}_{mod}(t)|$.

i.e. the particle will increase kinetic energy which finally will lead to an escape of the Lennard-Jones potential well. Solving for the modified equation of the Explicit Euler, one gets as a first correction

$$\dot{\mathbf{q}} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}} + \frac{\delta t}{2} \frac{\partial \mathcal{H}}{\partial \mathbf{q}} \tag{98}$$

$$\dot{\mathbf{p}} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} + \frac{\delta t}{2} \mathbf{p} \frac{\partial^2 \mathcal{H}}{\partial \mathbf{p}^2} \tag{99}$$

Figure 11 shows results for the integration of equations of motion with the Explicit Euler scheme. Different time steps for integration were applied which show a faster escape from a stable orbit with increasing time step. Also plotted in the same figure is the solution of the modified equations with a high order symplectic scheme, which can be considered as *exact* on these time scales. It is found that the trajectories more or less coincide and cannot be distinguished by eye. A more quantitative analysis (Figure 11) shows that for relatively long times the solution is rather well approximated by the modified equation, although with increasing time the differences between solutions become more pronounced. This means that for longer times it would be necessary to include more terms of higher order in $\delta t$ into the modified equation. It should be mentioned that, in general, the series expansion of the modified equation diverges.

## 4 Simulating in Different Ensembles

In MD simulations it is possible to realize different types of thermodynamic ensembles which are characterized by the control of certain thermodynamic quantities. If one knows how to calculate a thermodynamic quantity, e.g. the temperature or pressure, it is often possible to formulate an algorithm which fixes this property to a desired value. However, it is

not always clear whether this algorithm describes the properties of a given thermodynamic ensemble.

One can distinguish four different types of control mechanisms:

*Differential control* : the thermodynamic quantity is fixed to the prescribed value and no fluctuations around an average value occur.

*Proportional control* : the variables, coupled to the thermodynamic property $f$, are corrected in each integration step through a coupling constant towards the prescribed value of $f$. The coupling constant determines the strength of the fluctuations around $\langle f \rangle$.

*Integral control* : the system's Hamiltonian is extended and variables are introduced which represent the effect of an external system which fix the state to the desired ensemble. The time evolution of these variables is determined by the equations of motion derived from the extended Hamiltonian.

*Stochastic control* : the values of the variables coupled to the thermodynamic property $f$ are propagated according to modified equations of motion, where certain degrees of freedom are additionally modified stochastically in order to give the desired mean value of $f$.

In the following, different statistical ensembles are presented and all methods will be discussed via examples.

### 4.1 The Microcanonical Ensemble

The microcanonical ensemble (NVE) may be considered as the *natural* ensemble for molecular dynamics simulations (as it is the canonical ensemble (NVT) for Monte Carlo simulations). If no time dependent external forces are considered, the system's Hamiltonian is constant, implying that the system's dynamics evolves on a constant energy surface. The corresponding probability density in phase space is therefore given by

$$\rho(\mathbf{q}, \mathbf{p}) = \delta(\mathcal{H}(\mathbf{q}, \mathbf{p}) - E) \tag{100}$$

In a computer simulation this theoretical condition is generally violated, due to limited accuracy in integrating the equations of motion and due to roundoff errors resulting from a limited precision of number representation. In Ref.[129] a numerical experiment was performed showing that tiny perturbations of the initial positions of a trajectory are doubled about every picosecond. This would mean even for double precision arithmetic that after about $50\ ps$ roundoff errors will be dominant[117]. This is, however, often not a too serious restriction, since most time correlation functions drop to zero on a much shorter time scale. Only for the case where long time correlations are expected one does have to be very careful in generating trajectories.

### 4.2 The Canonical Ensemble

The simplest extension to the microcanonical ensemble is the canonical one (N,V,T), where the number of particles, the volume and the temperature are fixed to prescribed values. The temperature $T$ is, in contrast to $N$ and $V$, an intensive parameter. The extensive counterpart would be the kinetic energy of the system. In the following, different control mechanisms, introduced in Sec. 4 are described.

### 4.2.1 The Differential Thermostat

Different methods were proposed to fix the temperature to a fixed value during a simulation without allowing fluctuations of $T$. The first method was introduced by Woodcock[130], where the velocities were scaled according to $\mathbf{p}_i \rightarrow \sqrt{T_0/T}\mathbf{p}_i$, where $T_0$ is the reference temperature and $T$ the actual temperature, calculated from the velocity of the particles. This method leads to discontinuities in the momentum part of the phase space trajectory due to the rescaling procedure.

An extension of this method implies a constraint of the equations of motion to keep the temperature fixed[131–133]. The principle of least constraint by Gauss states that a force added to restrict a particle motion on a constraint hypersurface should be normal to the surface in a realistic dynamics. From this principle the equations of motion are derived

$$\frac{\partial \mathbf{q}_i}{\partial t} = \mathbf{p}_i \tag{101}$$

$$\frac{\partial \mathbf{p}_i}{\partial t} = -\frac{\partial V}{\partial \mathbf{q}_i} - \zeta \mathbf{p}_i \tag{102}$$

where $\zeta$ is a constraint force term, calculated as

$$\zeta = -\frac{\sum_{i=1}^{N} \frac{\mathbf{p}_i}{m_i} \frac{\partial V}{\partial \mathbf{q}_i}}{\sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{m_i}} \tag{103}$$

Since the principle of least constraint by Gauss is used, this algorithm is also called *Gaussian thermostat*. It may be shown for this method that the configurational part of the phase space density is of canonical form, i.e.

$$\rho(\mathbf{q}, \mathbf{p}) = \delta(T - T_0)\, e^{-\beta U(\mathbf{q})} \tag{104}$$

### 4.2.2 The Proportional Thermostat

The proportional thermostat tries to correct deviations of the actual temperature $T$ form the prescribed one $T_0$ by multiplying the velocities by a certain factor $\lambda$ in order to move the system dynamics towards one corresponding to $T_0$. The difference with respect to the differential control is that the method allows for fluctuations of the temperature, thereby not fixing it to a constant value. In each integration step it is insured that the $T$ is corrected to a value more close to $T_0$. A thermostat of this type was proposed by Berendsen et al.[134,135] who introduced *weak coupling methods to an external bath*. The weak coupling thermostat was motivated by the minimization of local disturbances of a stochastic thermostat while keeping the global effects unchanged. This leads to a modification of the momenta $\mathbf{p}_i \rightarrow \lambda \mathbf{p}_i$, where

$$\lambda = \left[1 + \frac{\delta t}{\tau_T}\left(\frac{T_0}{T} - 1\right)\right]^{\frac{1}{2}} \tag{105}$$

The constant $\tau_T$, appearing in Eq.105, is a so called coupling time constant which determines the time scale on which the desired temperature is reached. It is easy to show that the

proportional thermostat conserves a Maxwell distribution. However, the method cannot be mapped onto a specific thermodynamic ensemble. In Ref.[136] the phase space distribution could be shown to be

$$\rho(\mathbf{q},\mathbf{p}) = f(\mathbf{p})\, e^{-\beta\left(U(\mathbf{q})-\alpha\beta\delta U(\mathbf{q})^2/3N\right)} \tag{106}$$

where $\alpha \simeq (1-\delta E/\delta U)$ and $\delta U,\ \delta E$ are the mean fluctuations of the potential and total energy. $f(\mathbf{p})$ is in general an unknown function of the momenta, so that the full density cannot be determined. For $\alpha = 0$, which corresponds in Eq.105 to $\tau_T = \delta t$, the fluctuations in the kinetic energy vanish and Eq.106 reduces to Eq.104, i.e. it represents the canonical distribution. The other extreme of $\tau_T \rightarrow \infty$ corresponds to an isolated system and the energy should be conserved, i.e. $\delta E = \delta K + \delta U = 0$ and $\alpha = 1$. In this case, Eq.106 corresponds to the microcanonical distribution[136]. Eq.106 may therefore be understood as an interpolation between the canonical and the microcanonical ensemble.

### 4.2.3 The Stochastic Thermostat

In the case of a stochastic thermostat, all or a subset of the degrees of freedom of the system are subject to collisions with *virtual* particles. This method can be motivated by a Langevin stochastic differential equation which describes the motion of a particle due to the thermal agitation of a heat bath

$$\frac{\partial \mathbf{p}_i}{\partial t} = -\frac{\partial U}{\partial \mathbf{q}_i} - \gamma \mathbf{p}_i + \mathbf{F}^+ \tag{107}$$

where $\gamma$ is a friction constant and $\mathbf{F}^+$ a Gaussian random force. The amplitude of $\mathbf{F}^+$ is determined by the second fluctuation dissipation theorem

$$\langle \mathbf{F}_i^+(t_1)\mathbf{F}_j^+(t_2)\rangle = 2\gamma k_B T \delta_{ij}\delta(t_1-t_2) \tag{108}$$

A larger value for $\gamma$ will increase thermal fluctuations, while $\gamma = 0$ reduces to the microcanonicle ensemble. This method was applied to molecular dynamics in Ref.[137]. A more direct way was followed in Refs.[138,139] where particles collide occasionally with virtual particles from a Maxwell distribution corresponding to a temperature $T_0$ and after collisions loose their memory completely, i.e. the motion is totally randomized and the momenta become discontinuous. In order not to disturb the phase space trajectory too much, the collision frequency has to be chosen not too high. Since a large collision frequency will lead to a strong loss of the particle's memory, it will lead to a fast decay of dynamic correlation functions[140]. The characteristic decay time of correlation functions should therefore be a measure for the collision time. It was proved for the stochastic thermostat[138] that it leads to a canonical distribution function.

A slightly different method which is able to control the coupling to an external bath was suggested in Refs.[141,142]. In this approach the memory of the particle is not completely destroyed but the new momenta are chosen to be

$$\mathbf{p}_{i,n} = \sqrt{1-\alpha^2}\ \mathbf{p}_{i,o} + \alpha\ \mathbf{p}_r \tag{109}$$

where $\mathbf{p}_r$ is chosen a momentum, drawn from a Maxwell distribution corresponding to $T_0$. Similar to the proportional thermostat, the parameter $\alpha$ may be tuned to give distributions ranging from the microcanonical to the canonical ensemble.

### 4.2.4 The Integral Thermostat

The integral method is also often called *extended system method* as it introduces additional degrees of freedom into the system's Hamiltonian for which equations of motion can be derived. They are integrated in line with the equations for the spatial coordinates and momenta. The idea of the method invented by Nosé[143,144], is to reduce the effect of an external system acting as heat reservoir to keep the temperature of the system constant, to one additional degree of freedom. The thermal interactions between a heat reservoir and the system result in a change of the kinetic energy, i.e. the velocity of the particles in the system. Formally it may therefore be expressed a scaling of the velocities. Nosé introduced two sets of variables: real and so called virtual ones. The virtual variables are consistently derived from a Sundman transformation[145] $d\tau/dt = s$, where $\tau$ is a virtual time and $s$ is a resulting scaling factor, which is treated as dynamical variable. The transformation from virtual to real variables is then performed as

$$\mathbf{p}_i = \boldsymbol{\pi}_i s \qquad , \qquad \mathbf{q}_i = \boldsymbol{\rho}_i \tag{110}$$

The introduction of the *effective mass*, $M_s$, connects also a momentum to the additional degree of freedom, $\pi_s$. The resulting Hamiltonian, expressed in virtual coordinates reads

$$\mathcal{H}^* = \sum_{i=1}^{N} \frac{\boldsymbol{\pi}_i^2}{2m_i s^2} + U(\boldsymbol{\rho}) + \frac{\pi_s^2}{2M_s} + g k_B T \ln s \tag{111}$$

where $g = 3N + 1$ is the number of degrees of freedom (system of $N$ free particles). The Hamiltonian in Eq.111 was shown to lead to a probability density in phase space, corresponding to the canonical ensemble.

The equations of motion drawn from this Hamiltonian are

$$\frac{\partial \boldsymbol{\rho}_i}{\partial \tau} = \frac{\boldsymbol{\pi}_i}{s^2} \tag{112}$$

$$\frac{\partial \boldsymbol{\pi}_i}{\partial \tau} = -\frac{\partial U(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}_i} \tag{113}$$

$$\frac{\partial s}{\partial \tau} = \frac{\pi_s}{M_s} \tag{114}$$

$$\frac{\partial \pi_s}{\partial \tau} = \frac{1}{s^3} \sum_{i=1}^{N} \frac{\boldsymbol{\pi}_i^2}{m_i} - \frac{g k_B T}{s} \tag{115}$$

If one transforms these equations back into real variables, it is found[146] that they can be simplified by introducing the new variable $\zeta = \partial s/\partial t = s p_s/M_s$ ($p_s$ is *real* momentum connected to the heat bath)

$$\frac{\partial \mathbf{q}_i}{\partial t} = \frac{\mathbf{p}_i}{m_i} \tag{116}$$

$$\frac{\partial \mathbf{p}_i}{\partial t} = -\frac{\partial U(\mathbf{q})}{\partial \mathbf{q}_i} - \zeta \mathbf{p}_i \tag{117}$$

$$\frac{\partial \ln s}{\partial t} = \zeta \tag{118}$$

$$\frac{\partial \zeta}{\partial t} = \frac{1}{M_s} \left( \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{m_i} - g k_B T \right) \tag{119}$$

These equations describe the so called Nosé-Hoover thermostat.

### 4.3 The Constant-Pressure Constant-Enthalpy Ensemble

In order to control the pressure in an MD simulation cell, it is necessary to allow for volume variations. A simple picture for a constant pressure system is a box the walls of which are coupled to a piston which controls the pressure. In contrast to the case where the temperature is controlled, no coupling to the dynamics of the particles (timescales) is performed but the length scales of the system will be modified. In the following, different algorithms are described for a constant pressure ensemble. The conserved quantity will not be the system's energy, since there will be an energy transfer to or from the *external* system (piston etc.), but the enthalpy $H$ will be constant. In line with the constant temperature methods there are also differential, proportional, integral and stochastic methods to achieve a constant pressure situation in simulations. The differential method, however, is not discussed here, since there are problems with that method related to the *correct initial* pressure[147, 148]. A scheme for the calculation of the pressure in MD simulations for various model systems is given in the appendix.

#### 4.3.1 The Proportional Barostat

The proportional thermostat in Sec. 4.2.2 was introduced as an extension for the equation of the momentum, since it couples to the kinetics of the particles. Since the barostat acts on a volume change, which may be expressed in a scaling of particles' positions, a phenomenological extension for the equation of motion of the coordinates may be formulated[134]

$$\frac{\partial \mathbf{q}_i}{\partial t} = \frac{\mathbf{p}_i}{m_i} + \alpha \mathbf{q}_i \tag{120}$$

while a change in volume is postulated as

$$\dot{V} = 3\alpha V \tag{121}$$

A change in pressure is related to the isothermal compressibility $\kappa_T$

$$\dot{P} = -\frac{1}{\kappa_T V}\frac{\partial V}{\partial t} = -\frac{3\alpha}{\kappa_T} \tag{122}$$

which is approximated as

$$\frac{(P_0 - P)}{\tau_P} = -\frac{3\alpha}{\kappa_T} \tag{123}$$

and therefore Eq.120 can be written as

$$\frac{\partial \mathbf{q}_i}{\partial t} = \frac{\mathbf{p}_i}{m_i} - \frac{\kappa_T}{3\tau_P}(P_0 - P) \tag{124}$$

which corresponds to a scaling of the boxlength and coordinates $\mathbf{q} \rightarrow s\mathbf{q}$ and $L \rightarrow sL$, where

$$s = 1 - \frac{\kappa_T \delta t}{3\tau_P}(P_0 - P) \tag{125}$$

The time constant $\tau_P$ was introduced into Eq.123 as a characteristic timescale on which the system pressure will approach the desired pressure $P_0$. It also controls the strength of the coupling to the barostat and therefore the strength of the volume/pressure fluctuations. If the isothermal compressibility, $\kappa_T$, is not known for the system, the constant $\tau_P' = \tau_P/\kappa_T$ may be considered as a phenomenological coupling time which can be adjusted to the system under consideration. As for the proportional thermostat, a drawback for this method is that the statistical ensemble is not known. In analog to the thermostat, it may be assumed to *interpolate* between the microcanonical and the constant-pressure/constant-enthalpy ensemble, depending on the coupling constant $\tau_P$.

### 4.3.2 The Integral Barostat

In line with the integral thermostat one can introduce a new degree freedom into the systems Hamiltonian which controls volume fluctuations. This method was first proposed by Andersen[138]. The idea is to include the volume as an additional degree of freedom and to write the Hamiltonian in a scaled form, where lengths are expressed in units of the boxlength $L = V^{1/3}$, i.e. $\mathbf{q}_i = L\ \boldsymbol{\rho}_i$ and $\mathbf{p}_i = L\ \boldsymbol{\pi}_i$. Since $L$ is also a dynamical quantity, the momentum is not related to the simple time derivative of the coordinates but $\partial_t \mathbf{q}_i = L\ \partial_t \boldsymbol{\rho}_i + \boldsymbol{\rho}_i\ \partial_t L$. The extended system Hamiltonian is then written as

$$\mathcal{H}^* = \frac{1}{V^{2/3}} \sum_{i=1}^{N} \frac{\boldsymbol{\pi}_i}{2m_i} + U(V^{1/3}\boldsymbol{\rho}) + P_{ex}V + \frac{\boldsymbol{\pi}_V^2}{2M_V} \tag{126}$$

where $P_{ex}$ is the prescribed external pressure and $\boldsymbol{\pi}_V$ and $M_V$ are a momentum and a mass associated with the fluctuations of the volume.

The equations of motion which are derived from this Hamiltonian are

$$\frac{\partial \boldsymbol{\rho}_i}{\partial t} = \frac{1}{V^{2/3}} \frac{\boldsymbol{\pi}_i}{m_i} \tag{127}$$

$$\frac{\partial \boldsymbol{\pi}_i}{\partial t} = \frac{\partial U(V^{1/3}\boldsymbol{\rho})}{\partial \boldsymbol{\rho}_i} \tag{128}$$

$$\frac{\partial V}{\partial t} = \frac{\boldsymbol{\pi}_V}{M_V} \tag{129}$$

$$\frac{\partial \boldsymbol{\pi}_V}{\partial t} = \frac{1}{3V} \left( \frac{1}{V^{2/3}} \sum_{i=1}^{N} \frac{\boldsymbol{\pi}_i}{m_i} - V^{1/3} \boldsymbol{\rho}_i \frac{\partial U(\mathbf{q})}{\partial \mathbf{q}_i} \right) \tag{130}$$

A transformation to real variables then gives

$$\frac{\partial \mathbf{q}_i}{\partial t} = \frac{\mathbf{p}_i}{m_i} + \frac{1}{3V} \frac{\partial V}{\partial t} \mathbf{q}_i \tag{131}$$

$$\frac{\partial \mathbf{p}_i}{\partial t} = -\frac{\partial U(\mathbf{q})}{\partial \mathbf{q}_i} - \frac{1}{3V} \frac{\partial V}{\partial t} \mathbf{p}_i \tag{132}$$

$$\frac{\partial V}{\partial t} = \frac{\boldsymbol{p}_V}{M_V} \tag{133}$$

$$\frac{\partial \boldsymbol{p}_V}{\partial t} = \underbrace{\frac{1}{3V} \left( \sum_{i=1}^{N} \frac{\mathbf{p}_i}{m_i} - \mathbf{q}_i \frac{\partial U(\mathbf{q})}{\partial \mathbf{q}_i} \right)}_{P} - P_{ex} \tag{134}$$

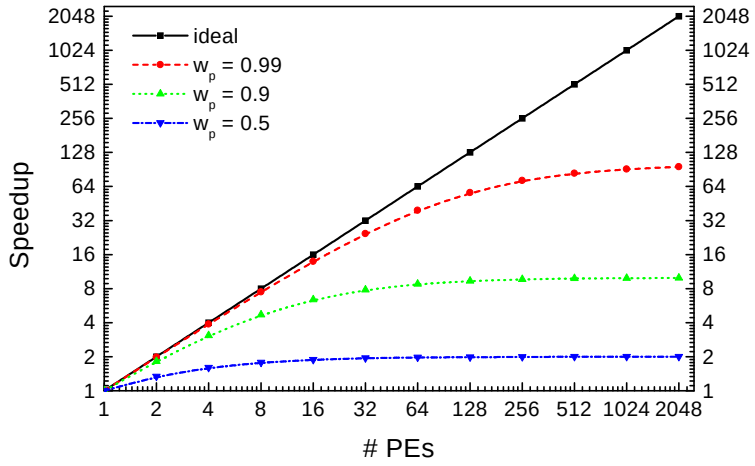


Figure 12. The ideal speedup for parallel applications with 50%, 90%, 99% and 100% (ideal) parallel work as a function of the number of processors.

In the last equation the term in brackets corresponds to the pressure, calculated from the virial theorem. The associated volume force, introducing fluctuations of the box volume is therefore controlled by the internal pressure, originating from the particle dynamics and the external pressure, $P_{ex}$.

## 5 Parallel Molecular Dynamics

With the advent of massively parallel computers, where thousands of processors may work on a single task, it has become possible to increase the size of the numerical problems considerably. As has been already mentioned in Sec.1 it is in principle possible to treat multi-billion particle systems. However, the whole success of parallel computing strongly depends both on the underlying problem to be solved and the optimization of the computer program. The former point is related to a principle problem which is manifested in the so called Amdahl's law[149]. If a problem has inherently certain parts which can be solved only in serial, this will give an upper limit for the parallelization which is possible. The speedup $\sigma$, which is a measure for the gain of using multiple processors with respect to a single one, is therefore bound

$$\sigma = \frac{N_p}{w_p + N_p w_s}. \tag{135}$$

Here, $N_p$ is the number of processors, $w_p$ and $w_s$ is the amount of work, which can be executed in parallel and in serial, i.e. $w_p + w_s = 1$. From Eq.135 it is obvious that the maximum efficiency is obtained when the problem is completely parallelizable, i.e. $w_p = 1$ which gives an $N_p$ times faster execution of the program. In the other extreme, when $w_s = 1$ there is no gain in program execution at all and $\sigma = 1$, independent of $N_p$. In Fig.12 this limitation is illustrated for several cases, where the relative amount for the serial work was modified. If the parallel work is 50%, the maximum speedup is bound to $\sigma = 2$.

If one aims to execute a program on a real massively parallel computer with hundreds or thousands of processors, the problem at hand must be inherently parallel for 99.99...%. Therefore, not only big parallel computers guarantee a fast execution of programs, but the problem itself has to be chosen properly.

Concerning MD programs there are only a few parts which have to be analysed for parallelization. As was shown, an MD program consists essentially of the force routine, which costs usually more than 90% of the execution time. If one uses neighbor lists, these may be also rather expensive while reducing the time for the force evaluation. Other important tasks are the integration of motion, the parameter setup at the beginning of the simulation and the file input/output (I/O). In the next chapter it will be shown how to parallelize the force routine. The integrator may be naturally parallelized, since the loop over $N$ particles may be subdivided and performed on different processors. The parameter setup has either to be done in serial so that every processor has information about relevant system parameters, or it may be done in parallel and information is distributed from every processor via a broadcast. The file I/O is a more complicated problem. The message passing interface MPI I does not offer a parallel I/O operation. In this case, if every node writes some information to the same file there is, depending on the configuration of the system, often only one node for I/O, to which internally the data are sent from the other nodes. The same applies for reading data. Since on this node the data from/for the nodes are written/read sequentially, this is a serial process which limits the speedup of the execution. The new MPI II standard offers parallel read/write operations, which lead to a considerable efficiency gain with respect to MPI. However, the efficiency obtained depend strongly on the implementation on different architectures. Besides MPI methods, there are other libraries, which offer more efficient parallel I/O with respect to native programming. To name a few, there are PnetCDF[150,151], an extension towards parallelism of the old *network Common Data Form*, netCDF-4[152,153], which is in direct line of netCDF development, which now has parallel functionality and which is built on top of MPI-I/O, or SIONlib, a recently developed high performance library for parallel I/O[154].

Another serious point is the implementation into the computer code. A problem which is inherently 100% parallel will not be solved with maximum speed if the program is not 100% mapped onto this problem. Implementation details for parallel algorithms will be discussed in the following sections. Independent of the implementation of the code, Eq.135 gives only an upper theoretical limit which will only be reached in very rare cases. For most problems it is necessary to communicate data from one processor to another or even to all other processors in order to take into account data dependencies. This implies an overhead which depends on the latency and the bandwidth of the interprocessor network, which strongly depends on the hardware.

### 5.1 Domain Decomposition

The principle of spatial decomposition methods is to assign geometrical domains to different processors. This implies that particles are no longer bound to a certain processor but will be transfered from one PE to another, according to their spatial position. This algorithm is especially designed for systems with short range interactions or to any other algorithm where a certain cut-off in space may be applied. Since neighbored processors contain all relevant data needed to compute forces on particles located on a given PE,

this algorithm avoids the problem of global communications. Given that the range of interaction between particles is a cut-off radius of size $R_c$, the size, $D$ of the domains is preferentially chosen to be $D > R_c$, so that only the $3^d - 1$ neighbored processors have to communicate data ($d$ is the dimension of the problem). Whether this can be fulfilled depends on the interplay between size of the system and the numbers of processors. If a small system is treated with a large number of processors, the domains will be small and $D < R_c$. In this case not only the next but also the second or even higher order neighbor PEs have to send their coordinates to a given PE. For simplicity we assume here $D > R_c$. Algorithms, which treat efficiently the general case were developed recently[155–157].

The algorithm then works as follows. Particles are distributed in the beginning of the simulation to a geometrical region. The domains are constructed to have a rather homogeneous distribution of particles on each processor, e.g. for homogeneous bulk liquids the domains can be chosen as equally sized cuboids which fill the simulation box. In order to calculate forces between particles on different processors, coordinates of the so called *boundary particles* (those which are located in the outer region of size $R_b \geq R_c$ of the domains) have to be exchanged. Two types of lists are constructed for this purpose. The one contains all particle indices, which have left the local domain and which have consequently to be transferred to the neighbored PE. The other one contains all particle indices, which lie in the outer region of size $R_b$ of a domain. The first list is used to update the particles' *address*, i.e. all information like positions, velocities, forces etc. are sent to the neighbored PE and are erased in the old domain. The second list is used to send temporarily position coordinates which are only needed for the force computation. The calculation of forces then operates in two steps. First, the forces due to local particles are computed using Newton's 3rd law. In a next step, forces due to the boundary particles are calculated. The latter forces are thus calculated twice: on the local PE and the neighbored PE. This extra computation has the advantage that there is no communication step for forces. A more elaborate scheme has nevertheless been proposed which includes also Newton's 3rd law for the boundary particles and thus the communication of forces[158, 159]. Having finished the evaluation of forces, the new positions and velocities are evaluated only for local particles.

A naive method would require $3^d - 1$ send/receive operations. However, this may be reduced to $2\log_d(3^d - 1)$ operations with a similar tree-like method. The method is described here for the case of $d = 2$. It may be generalized rather easily. The 4 processors, located directly at the edges of a given one are labeled as left/right and up/down. Then in a first step, information is sent/received to/from the left and the right PE, i.e. each processor now stores the coordinates of three PEs (including local information). The next step proceeds in sending/receiving the data to the up and down PEs. This step finishes already the whole communication process.

The updating process is not necessarily done in each time step. If the width of the boundary region is chosen as $R_b = R_c + \delta r$, it is possible to trigger the update automatically via the criterion $\max(|\mathbf{x}(t_0 + t) - \mathbf{x}(t_0)|) \leq \delta r$, which is the maximum change in distance of any particle in the system, measured from the last update.

A special feature of this algorithm is the fact that it shows a theoretical superlinear speed-up if Verlet neighbor lists are used. The construction of the Verlet list requires $N'(N' - 1)/2 + N'\delta N$ operations, where $\delta N$ is the number of boundary particles and $N'$ is the number of particles on a PE. If the number of PEs is increased as twice as large,

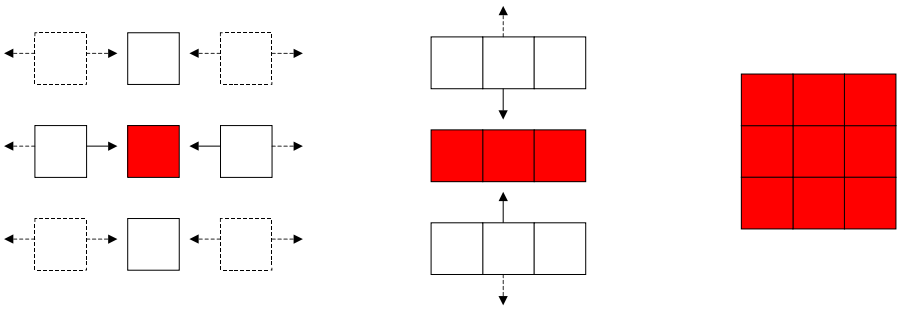

Figure 13. Communication pattern for the domain decomposition algorithm in 2 dimensions.

there are $N'/2$ particles on each processor which therefore requires $N'/2(N'/2-1)/2+N'/2\delta N$ operations. If $N' \gg \delta N$ and $N'^2 \gg N'$ one gets a speed-up factor of $\approx 4$!

### 5.2 Performance Estimations

In order to estimate the performance of the different algorithms on a theoretical basis it is useful to extend the ideal Amdahl's law to a more realistic case. The ideal law only takes into account the degree of parallel work. From that point of view all parallel algorithms for a given problem should work in the same way. However the communication between the processors is also a limiting factor in parallel applications and so it is natural to extend Amdahl's law in the following way

$$\sigma = \frac{1}{w_p/N_p + w_s + c(N_p)} \tag{136}$$

where $c(N_p)$ is a function of the number of processors which will characterize the different parallel algorithms. The function will contain both communication work, which depends on the bandwidth of the network and the effect of the latency time, i.e. how fast the network responds to the communication instruction. The function $c(N_p)$ expresses the relative portion of communication with respect to computation. Therefore it will depend in general also on the number of particles which are simulated.

In the following a model analysis for the domain decomposition algorithm is presented. It is assumed that the work is strictly parallel, i.e. $w_p = 1$.

Spatial decomposition algorithm is based on local communication. As was described in Sec.5.1, only six communication steps are required to distribute the data to neighbored PEs. Therefore the latency time part is constant whereas the amount of data to be sent and consequently the communication part is decreased with larger $N_p$. The communication function reads therefore

$$c(N_p) = f(N_p)\left(\lambda + \frac{\chi}{N_p^{2/3}}\right) \quad , \quad f(N_p) = \begin{cases} 0 & N_p = 1 \\ 2 & N_p = 2 \\ 4 & N_p = 4 \\ 6 & N_p \leq 8 \end{cases} \tag{137}$$

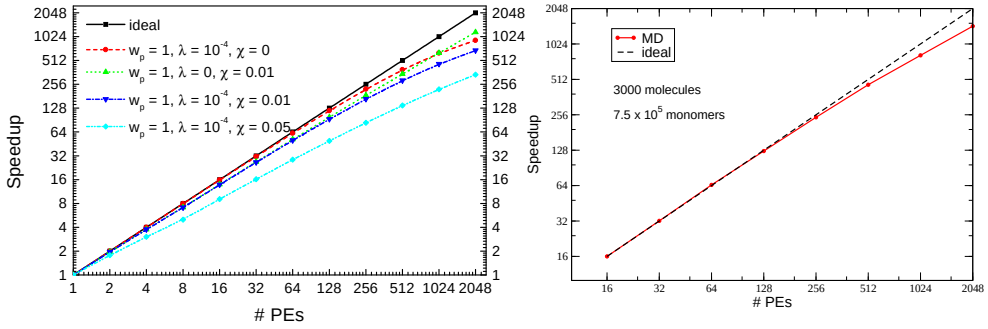


Figure 14. Left: Estimation of realistic speedup curves if one includes the latency time and bandwidth of the processor interconnect. It is assumed that the problem can potentially be parallelized 100%. Different parameter values are compared for the latency time $\lambda$ and bandwidth $\chi$ for a local nearest neighbor communications. The ideal curve neglects communication completely. Right: Realistic benchmark for a domain decomposition program, simulating a system consisting of 3000 polymers with 250 monomers each.

Here the function $f(N_p)$ was introduced to cover also the cases for small numbers of PEs, where a data exchange is not necessary in each spatial direction. As seen from Fig.14 the speedup curves are nearly linear with a slightly smaller slope than unity. However, for very large numbers of PEs the curves will also flatten. Nevertheless, the local communication model provides the best speedup behavior from all parallel algorithms and seems to be best suited for large parallel architectures.

*Remark* Note that the local communication model in its present form is only valid for short range interaction potentials. If the potential is longer ranged than one spatial domain, the function $f(N_p)$ has to be modified. For long range interactions, all-to-all communications are generally required. In that case the tree-method may be mostly preferred.

This theoretical analysis demonstrates the importance of a fast interconnect between processors for the case of molecular dynamics simulations. Not included in the communication function $c(N_p)$ is the bandwidth function of the network. This, however, will only slightly change Fig.14.

### 5.3 Comparison with Simulation

In order to verify the theoretical model, one may perform real MD simulations for model systems, which are as large as the principal features, appearing in the analysis are fulfilled. This includes that domains are large enough in order to restrict particle interactions to neighbored domains and to have a nearly homogenous particle distribution, which avoids unbalanced computational work on the processors.

In the current case, the program MP2C[160] was used, which implements both a mesoscopic solvent method, based on the Muli-Particle-Collision (MPC) dynamics and a molecular dynamics part. The program is based on a domain decomposition approach and allows to couple MD and MPC simulations or to decouple them, in order to run either MD or MPC in a simulation for e.g. all-atom force-field simulations without hydrodynamic coupling or e.g. fluid dynamics without solvated particles, respectively. In the present case a simulation of a polymer system, consisting of 3000 polymeric chains with 250 monomers each

was simulated. The monomers were coupled within the chain by a harmonic bond potential and the non-bonded part of the potential was set to the repulsive part of a Lennard-Jones potential which was applied to all particle pairs which were not coupled within bonds.

The program was run on an IBM BlueGene/P at Jülich Supercomputing Centre. Fig. 14 shows the scaling up to $N_p = 2048$ processors, which is qualitatively comparable and shows the same behavior as prescribed by the simple model. A better scaling is to be expected, when more particles are simulated, which moves the ratio of communication/computation to smaller values, which reduces the relative overhead in the parallel execution.

## References


1. K. Binder and D. Landau. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, Cambridge, 2000.
2. A.R. Leach. *Molecular Modelling - Principles and Applications*. Pearson Education Ltd., Essex, England, 2001.
3. T. Schlick. *Molecular Modeling and Simulation*. Springer, New York, 2002.
4. K. Binder and D.W. Heermann. *Monte Carlo Simulation in Statistical Physics*. Springer, Berlin, 1997.
5. D. Frenkel and B. Smit. *Understanding molecular simulation. From algorithms to applications.* Academic Press, San Diego, 1996.
6. J. M. Haile. *Molecular Dynamics Simulation*. Wiley, New York, 1997.
7. H. Goldstein, Ch. Poole, and J. Safko. *Classical Mechanics*. Addison Wesley, San Francisco, CA, 2002.
8. B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics*. Cambridge University Press, Cambridge, 2004.
9. B. J. Alder and T. E. Wainwright. Phase transition for a hard sphere system. *J. Chem. Phys.*, 27:1208–1209, 1957.
10. B. J. Alder and T. E. Wainwright. Studies in molecular dynamics. I. General method. *J. Chem. Phys.*, 31:459, 1959.
11. J. Roth, F. Gähler, and H.-R. Trebin. A molecular dynamics run with 5.180.116.000 particles. *Int. J. Mod. Phys. C*, 11:317–322, 2000.
12. K. Kadau, T. C. Germann, and P. S. Lomdahl. Large-scale molecular-dynamics simulation of 19 billion particles. *Int. J. Mod. Phys. C*, 15:193, 2004.
13. K. Kadau, T. C. Germann, and P. S. Lomdahl. World record: Large-scale molecular-dynamics simulation of 19 billion particles. Technical Report LA-UR-05-3853, Los Alamos NAtional Laboratory, 2005.
14. K. Kadau, T. C. Germann, and P. S. Lomdahl. Molecular-Dynamics Comes of Age: 320 Billion Atom Simulation on BlueGene/L. *Int. J. Mod. Phys. C*, 17:1755, 2006.
15. T. C. Germann and K. Kadau. Trillion-atom molecular dynamics becomes a reality. *Int. J. Mod. Phys. C*, 19:1315–1319, 2008.
16. P.S. Lomdahl, P. Tamayo, N. Gronbach-Jensen, and D.M. Beazley. In G.S. Ansell, editor, *Proc. Supercomputing 93*, page 520, Los Alamitos CA, 1993. IEEE Computer Society Press.
17. D.M. Beazley and P.S. Lomdahl. *Comput. Phys.*, 11:230, 1997.
18. Y. Duan, L. Wang, and P. A. Kollman. The early stage of folding of villin headpiece subdomain observed in 200-nanosecond fully solvated molecular dynamics simultion.

*Proc. Natl. Acad. Sci. USA*, 95:9897, 1998.
19. Y. Duan and P. A. Kollman. Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science*, 282:740, 1998.
20. C. Mura and J.A. McCammon. Molecular dynamics of a $\kappa$B DNA element: base flipping via cross-strand intercalative stacking in a microsecond-scale simulation. *Nucl. Acids Res.*, 36:4941–4955, 2008.
21. http://www.ccp5.ac.uk/.
22. http://amber.scripps.edu/.
23. http://www.charmm.org.
24. http://www.ks.uiuc.edu/Research/namd/.
25. http://www.emsl.pnl.gov/docs/nwchem/nwchem.html.
26. http://www.gromacs.org.
27. http://www.cs.sandia.gov/ sjplimp/lammps.html.
28. Gregory A. Voth. *Coarse-Graining of Condensed Phase and Biomolecular Systems*. CRC Press, 2008.
29. A. Arkhipov A.Y. Shih, P.L. Freddolino, and K. Schulten. Coarse grained protein-lipid model with application to lipoprotein particles. *J. Phys. Chem. B*, 110:3674–3684, 2006.
30. P.M. Kasson, A. Zomorodian, S. Park, N. Singhal, L.J. Guibas, and V.S. Pande. Persistent voids: a new structural metric for membrane fusion. *Bioinformatics*, 23:1753–1759, 2007.
31. A. J. Stone. Intermolecular potentials. *Science*, 321:787–789, 2008.
32. W. L. Cui, F. B. Li, and N. L. Allinger. *J. Amer. Chem. Soc.*, 115:2943, 1993.
33. N. Nevins, J. H. Lii, and N. L. Allinger. *J. Comp. Chem.*, 17:695, 1996.
34. S. L. Mayo, B. D. Olafson, and W. A. Goddard. *J. Phys. Chem.*, 94:8897, 1990.
35. M. J. Bearpark, M. A. Robb, F. Bernardi, and M. Olivucci. *Chem. Phys. Lett.*, 217:513, 1994.
36. T. Cleveland and C. R. Landis. *J. Amer. Chem. Soc.*, 118:6020, 1996.
37. A. K. Rappé, C. J. Casewit, K. S. Colwell, W. A. Goddard, and W. M. Skiff. *J. Amer. Chem. Soc.*, 114:10024, 1992.
38. Z. W. Peng, C. S. Ewig, M.-J. Hwang, M. Waldman, and A. T. Hagler. Derivation of class ii force fields. 4. van der Waals parameters of Alkali metal cations and Halide anions. *J. Phys. Chem.*, 101:7243–7252, 1997.
39. W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J. Amer. Chem. Soc.*, 117:5179–5197, 1995.
40. A. D. Mackerell, J. Wiorkiewiczkuczera, and M. Karplus. *J. Amer. Chem. Soc.*, 117:11946, 1995.
41. W. L. Jorgensen, D. S. Maxwell, and J. Tiradorives. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *J. Amer. Chem. Soc.*, 118:11225–11236, 1996.
42. T. A. Halgren. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *J. Comp. Chem.*, 17:490–519, 1996.
43. J. Kong. Combining rules for intermolecular potential paramters. II. Rules for the Lennard-Jones (12-6) potential and the Morse potential. *J. Chem. Phys.*, 59:2464–

2467, 1973.
44. M. Waldman and A. T. Hagler. New combining rules for rare gas van der Waals parameters. *J. Comp. Chem.*, 14:1077, 1993.
45. J. Delhommelle and P. Millié. Inadequacy of the Lorentz-Bertelot combining rules for accurate predictions of equilibrium properties by molecular simulation. *Molec. Phys.*, 99:619–625, 2001.
46. L. Verlet. Computer experiments on classical fluids. I. Thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, 159:98, 1967.
47. G. Sutmann and V. Stegailov. Optimization of neighbor list techniques in liquid matter simulations. *J. Mol. Liq.*, 125:197–203, 2006.
48. G. Sutmann and V. Stegailov (to be published).
49. R. W. Hockney. The potential calculation and some applications. *Meth. Comput. Phys.*, 9:136–211, 1970.
50. R. W. Hockney, S. P. Goel, and J. W. Eastwood. Quite high-resolution computer models of a plasma. *J. Comp. Phys.*., 14:148, 1974.
51. P. Ewald. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann. Phys.*, 64:253, 1921.
52. S. W. de Leeuw, J. M. Perram, and E. R. Smith. Simulation of electrostatic systems in periodic boundary conditions. I. Lattice sums and dielectric constants. *Proc. R. Soc. London*, A373:27, 1980.
53. S. W. de Leeuw, J. M. Perram, and E. R. Smith. Simulation of electrostatic systems in periodic boundary conditions. II. Equivalence of boundary conditions. *Proc. R. Soc. London*, A373:57, 1980.
54. T. Darden, D. York, and L. Pedersen. A NlogN method for Ewald sums in large systems. *J. Chem. Phys.*, 98:10089, 1993.
55. U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen. A smooth particle mesh ewald method. *J. Chem. Phys.*, 103:8577, 1995.
56. R. W. Hockney and J. W. Eastwood. *Computer simulation using particles*. McGraw-Hill, New York, 1981.
57. M. Deserno and C. Holm. How to mesh up Ewald sums. I. a theoretical and numerical comparison of various particle mesh routines. *J. Chem. Phys.*, 109:7678, 1998.
58. M. Deserno and C. Holm. How to mesh up Ewald sums. II. an accurate error estimate for the P3M algorithm. *J. Chem. Phys.*, 109:7694, 1998.
59. L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comp. Phys.*, 73:325, 1987.
60. H. Cheng, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *J. Comp. Phys.*., 155:468–498, 1999.
61. C. A. White and M. Head-Gordon. Derivation and efficient implementation of the fast multipole method. *J. Chem. Phys.*, 101:6593–6605, 1994.
62. C. A. White and M. Head-Gordon. Rotating around the quartic angular momentum barrier in fast multipole method calculations. *J. Chem. Phys.*, 105:5061–5067, 1996.
63. C. A. White and M. Head-Gordon. Fractional tiers in fast multipole method calculations. *Chem. Phys. Lett.*, 257:647–650, 1996.
64. H. Dachsel. An improved implementation of the fast multipole method. In *Proceedings of the 4th MATHMOD Vienna*, Vienna, 2003.
65. J. E. Barnes and P. Hut. A hierarchical O($N \log N$) force calculation algorithm.

*Nature*, 324:446, 1986.
66. S. Pfalzner and P. Gibbon. *Many Body Tree Methods in Physics*. Cambridge University Press, New York, 1996.
67. G. Sutmann and B. Steffen. A particle-particle particle-multigrid method for long-range interactions in molecular simulations. *Comp. Phys. Comm.*, 169:343–346, 2005.
68. G. Sutmann and S. Wädow. A Fast Wavelet Based Evaluation of Coulomb Potentials in Molecular Systems. In U.H.E. Hansmann, editor, *From Computational Biophysics to Systems Biology 2006*, volume 34, pages 185–189, Jülich, 2006. John von Neumann Institute for Computing.
69. N. W. Ashcroft and N. D. Mermin. *Solid State Physics*. Saunders College Publishing, Fort Worth, 1976.
70. R. A. Robinson and R. H. Stokes. *Electrolyte Solutions*. Butterworth, London, 1965.
71. J. W. Perram, H. G. Petersen, and S. W. de Leeuw. An algorithm for the simulation of condensed matter which grows as the 3/2 power of the number of particles. *Molec. Phys.*, 65:875–893, 1988.
72. D. Fincham. Optimisation of the Ewald sum for large systems. *Molec. Sim.*, 13:1–9, 1994.
73. W. Smith. Point multipoles in the Ewald sum. *CCP5 Newsletter*, 46:18–30, 1998.
74. T. M. Nymand and P. Linse. Ewald summation and reaction field methods for potentials with atomic charges, dipoles and polarizabilities. *J. Chem. Phys.*, 112:6152–6160, 2000.
75. G. Salin and J. P. Caillol. Ewald sums for yukawa potentials. *J. Chem. Phys.*, 113:10459–10463, 2000.
76. L. Greengard. *The rapid evaluation of potential fields in particle systems*. MIT press, Cambridge, 1988.
77. L. Greengard. The numerical solution of the N-body problem. *Computers in Physics*, 4:142–152, 1990.
78. L. Greengard. Fast algorithms for classical physics. *Science*, 265:909–914, 1994.
79. J. D. Jackson. *Classical Electrodynamics*. Wiley, New York, 1983.
80. M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions*. Dover Publ. Inc., New York, 1972.
81. R. K. Beatson and L. Greengard. A short course on fast multipole methods. In M. Ainsworth, J. Levesley, W.A. Light, and M. Marletta, editors, *Wavelets, Multilevel Methods and Elliptic PDEs*, pages 1–37. Oxford University Press, 1997.
82. C. G. Lambert, T. A. Darden, and J. A. Board. A multipole-based algorithm for efficient calculation of forces and potentials in macroscopic periodic assemblies of particles. *J. Comp. Phys..*, 126:274–285, 1996.
83. M. Challacombe, C. A. White, and M. Head-Gordon. Periodic boundary conditions and the fast multipole method. *J. Chem. Phys.*, 107:10131–10140, 1997.
84. S. J. Marrink, E. Lindahl, O. Edholm, and A. E. Mark. Simulation of the spontaneous aggregation of phospholipids into bilayers. *J. Am. Chem. Soc.*, 123:8638, 2001.
85. Michael L. Klein and Wataru Shinoda. Large-scale molecular dynamics simulations of self-assembling systems. *Science*, 321:798 – 800, 2008.
86. A. H. de Vries, A. E. Mark, and S. J. Marrink. Molecular dynamics simulation of the spontaneous formation of a small dppc vesicle in water in atomistic detail. *J. Am.*

*Chem. Soc.*, 126:4488, 2004.
87. G. Srinivas, S. O. Nielsen, P. B. Moore, and M. L. Klein. Molecular dynamics simulations of surfactant self-organization at a solid-liquid interface. *J. Am. Chem. Soc.*, 128:848, 2006.
88. S. J. Marrink X. Periole, Th. Huber and Th. P. Sakmar. Protein-coupled receptors self-assemble in dynamics simulations of model bilayers. *J. Am. Chem. Soc.*, 129:10126, 2007.
89. W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.*, 79:926–935, 1983.
90. W. L. Jorgensen and J. D. Madura. Temperature and size dependence for Monte Carlo simulations of TIP4P water. *Mol. Phys.*, 56:1381–1392, 1985.
91. H. J. C. Berendsen, J. R. Grigera, and T. P. Straatsma. The missing term in effective pair potentials. *J. Phys. Chem.*, 91:6269, 1987.
92. M. W. Mahoney and W. L. Jorgensen. A five-site model for liquid water and the reproduction of the density anomaly by rigid, nonpolarizable potential functions. *J. Chem. Phys.*, 112:8910–8922, 2000.
93. S. J. Marrink, A. H. de Vries, and A. E. Mark. Coarse grained model for semiquantitative lipid simulations. *J. Phys. Chem. B*, 108:750, 2004.
94. S. J. Marrink, H. J. Risselada, S. Yefimov, D. P.Tieleman, and A. H. de Vries. Coarse grained model for semiquantitative lipid simulations. *J. Phys. Chem. B*, 108:750, 2004.
95. R.L. Henderson. A uniqueness theorem for fluid pair correlation functions. *Phys. Lett.*, 49 A:197–198, 1974.
96. A.P.Lyubartsev and A.Laaksonen. Reconstruction of pair interaction potentials from radial distribution functions. *Comp. Phys. Comm.*, 121-122:57–59, 1999.
97. A.P.Lyubartsev and A.Laaksonen. Determination of pair potentials from ab-initio simulations: Application to liqid water. *Chem. Phys. Lett.*, 325:15–21, 2000.
98. V.Lobaskin, A.P.Lyubartsev, and P.Linse. Effective macroion-macroion potentials in asymmetric electrolytes. *Phys. Rev. E*, 63:020401, 2001.
99. A.P.Lyubartsev and A.Laaksonen. Calculation of effective interaction potentials from radial distribution functions: A reverse Monte Carlo approach. *Comp. Phys. Comm.*, 121-122:57–59, 1999.
100. T.C. Terwilliger. Improving macromolecular atomic models at moderate resolution by automated iterative model building, statistical density modification and refinement. *Acta Cryst.*, D59:1174–1182, 2003.
101. H. F. Trotter. On the product of semi-groups of operators. *Proc. Am. Math. Soc.*, 10:545–551, 1959.
102. O. Buneman. Time-reversible difference procedures. *J. Comp. Phys..*, 1:517–535, 1967.
103. E. Hairer and P. Leone. Order barriers for symplectic multi-value methods. In D. Griffiths, D. Higham, and G. Watson, editors, *Pitman Research Notes in Mathematics*, volume 380, pages 133–149, 1998.
104. D. Okunbor and R. D. Skeel. Explicit canonical methods for Hamiltonian systems. *Math. Comput.*, 59:439–455, 1992.
105. E. Hairer. Backward error analysis of numerical integrators and symplectic methods.

*Ann. Numer. Math.*, 1:107–132, 1994.
106. E. Hairer and C. Lubich. The lifespan of backward error analysis for numerical integrators. *Numer. Math.*, 76:441–462, 1997.
107. S. Reich. Backward error analysis for numerical integrators. *SIAM J. Numer. Anal.*, 36:1549–1570, 1999.
108. D. M. Stoffer. *Some geometrical and numerical methods for perturbed integrable systems*. PhD thesis, Swiss Federal Institute of Technology, Zürich, 1988.
109. J. M. Sanz-Serna M. Calvo. *Numerical Hamiltonian Problems*. Chapman and Hall, London, 1994.
110. R. D. Skeel. Integration schemes for molecular dynamics and related applications. In M. Ainsworth, J. Levesley, and M. Marletta, editors, *The Graduate Student's Guide to Numerical Analysis*, pages 119–176, New York, 1999. Springer.
111. M. E. Tuckerman and W. Langel. Multiple time scale simulation of a flexible model of $co_2$. *J. Chem. Ohys.*, 100:6368, 1994.
112. P. Procacci, T. Darden, and M. Marchi. A very fast Molecular Dynamics method to simulate biomolecular systems with realistic electrostatic interactions. *J. Phys. Chem.*, 100:10464–10468, 1996.
113. P. Procacci, M. Marchi, and G. L. Martyna. Electrostatic calculations and multiple time scales in molecular dynamics simulation of flexible molecular systems. *J. Chem. Phys.*, 108:8799–8803, 1998.
114. P. Procacci and M. Marchi. Taming the Ewald sum in molecular dynamics simulations of solvated proteins via a multiple time step algorithm. *J. Chem. Phys.*, 104:3003–3012, 1996.
115. J. J. Biesiadecki and R. D. Skeel. Dangers of multiple time step methods. *J. Comp. Phys..*, 109:318–328, 1993.
116. J. L. Scully and J. Hermans. Multiple time steps: limits on the speedup of molecular dynamics simulations of aqueous systems. *Molec. Sim.*, 11:67–77, 1993.
117. B. J. Leimkuhler and R. D. Skeel. Symplectic numerical integrators in constrained Hamiltonian systems. *J. Comp. Phys..*, 112:117–125, 1994.
118. T. Schlick. Some failures and success of long timestep approaches to biomolecular simulations. In P. Deuflhard, J. Hermans, B. J. Leimkuhler, A. Mark, S. Reich, and R. D. Skeel, editors, *Lecture notes in computational science and engineerung. Algorithms for macromolecular modelling*, volume 4, pages 221–250, New York, 1998. Springer.
119. E. Barth and T. Schlick. Overcoming stability limitations in biomolecular danamics. I. Combining force splitting via extrapolation with Langevin dynamics. *J. Chem. Phys.*, 109:1617–1632, 1998.
120. E. Barth and T. Schlick. Extrapolation versus impulse in multiple-timestepping schemes. II. Linear analysis and applications to Newtonian and Langevin dynamics. *J. Chem. Phys.*, 109:1633–1642, 1998.
121. B. Garcia-Archilla, J. M. Sanz-Serna, and R. D. Skeel. Long-time-step methods for oscillatory differential equations. *SIAM J. Sci. Comp.*, 20:930–963, 1998.
122. B. Garcia-Archilla, J. M. Sanz-Serna, and R. D. Skeel. The mollified impulse method for oscillatory differential equations. In D. F. Griffiths and G. A. Watson, editors, *Numerical analysis 1997*, pages 111–123, London, 1998. Pitman.
123. B. Garcia-Archilla, J. M. Sanz-Serna, and R. D. Skeel. The mollified impulse method

for oscillatory differential equations. *SIAM J. Sci. Comp.*, 20:930–963, 1998.
124. J. A. Izaguirre. *Longer time steps for molecular dynamics.* PhD thesis, University of Illinois at Urbana-Champaign, 1999.
125. J. A. Izaguirre, S. Reich, and R. D. Skeel. Longer time steps for molecular dynamics. *J. Chem. Phys.*, 110:9853, 1999.
126. B. V. Chirikov. A universal instability of many-dimensional oscillator systems. *Phys. Rep.*, 52:264–379, 1979.
127. F. Calvo. Largest Lyapunov exponent in molecular systems: Linear molecules and application to nitrogen clusters. *Phys. Rev. E*, 58:5643–5649, 1998.
128. R. F. Warming and B. J. Hyett. The modified equation approach to the stability and accuracy analysis of finite difference methods. *J. Comp. Phys..*, 14:159–179, 1974.
129. M. P. Allen and D. J. Tildesley. *Computer simulation of liquids.* Oxford Science Publications, Oxford, 1987.
130. L. V. Woodcock. Isothermal molecular dynamics calculations for liquid salt. *Chem. Phys. Lett.*, 10:257–261, 1971.
131. W. G. Hoover, A. J. C. Ladd, and B. Moran. High strain rate plastic flow studied via nonequilibrium molecular dynamics. *Phys. Rev. Lett.*, 48:1818–1820, 1982.
132. D. J. Evans, W. G. Hoover, B. H. Failor, B. Moran, and A. J. C. Ladd. Nonequilibrium molecular dynamics via Gauss's principle of least constraint. *Phys. Rev. A*, 28:1016–1021, 1983.
133. D. J. Evans. Computer experiment for nonlinear thermodynamics of Couette flow. *J. Chem. Phys.*, 78:3298–3302, 1983.
134. H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.*, 81:3684, 1984.
135. H. J. C. Berendsen. Transport properties computed by linear response through weak coupling to a bath. In M. Meyer and V. Pontikis, editors, *Computer Simulation in Materials Science*, pages 139–155, Amsterdam, 1991. Kluwer Academic Publishers.
136. T. Morishita. Fluctuation formulas in molecular dynamics simulations with the weak coupling heat bath. *J. Chem. Phys.*, 113:2976–2982, 2000.
137. T. Schneider and E. Stoll. Molecular dynamics study of a three dimensional one-component model for distortive phase transitions. *Phys. Rev. B*, 17:1302–1322, 1978.
138. H. C. Andersen. Molecular dynamics simulations at constant pressure and/or temperature. *J. Chem. Phys.*, 72:2384, 1980.
139. E. Bonomi. *J. Stat. Phys.*, 39:167, 1985.
140. H. Tanaka, K. Nakanishi, and N. Watanabe. *J. Chem. Phys.*, 78:2626, 1983.
141. M. E. Riley, M. E. Coltrin, and D. J. Diestler. A velocity reset method of simulating thermal motion and damping in gas-solid collisions. *J. Chem. Phys.*, 88:5934–5942, 1988.
142. G. Sutmann and B. Steffen. Correction of finite siye effects in molecular dynamics simulation applied to friction. *Comp. Phys. Comm.*, 147:374–377, 2001.
143. S. Nosé. A unified formulation of the constant temperature molecular dynamics methods. *J. Chem. Phys.*, 81:511–519, 1984.
144. S. Nosé. A molecular dynamics methood for simulations in the canonical ensemble. *Molec. Phys.*, 52:255–268, 1984.

145. K. Zare and V. Szebehely. Time tranformations for the extended phase space. *Celestial Mech.*, 11:469, 1975.
146. W. G. Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A*, 31:1695–1697, 1985.
147. D. J. Evans and G. P. Morris. The isothermal isobaric molecular dynamics ensemble. *Phys. Lett. A*, 98:433–436, 1983.
148. D. J. Evans and G. P. Morris. Isothermal isobaric molecular dynamics. *Chem. Phys.*, 77:63–66, 1983.
149. G. M. Amdahl. Validity of the single-processor approach to achieving large scale computing capabilities. In *AFIPS Conference Proceedings*, volume 30, pages 483–485, Reston, Va., 1967. AFIPS Press.
150. http://trac.mcs.anl.gov/projects/parallel-netcdf.
151. http://trac.mcs.anl.gov/projects/parallel-netcdf/netcdf-api.ps.
152. www.unidata.ucar.edu/packages/netcdf/.
153. http://www.hdfgroup.uiuc.edu/HDF5/projects/archive/WRF-ROMS/Parallel_NetCDF4_Performance.pdf.
154. W. Frings, F. Wolf, and V. Petkov. SIONlib: Scalable parallel I/O for task-local files, 2009. (to be submitted).
155. M. Snir. A note on n-body computations with cutoffs. *Theor. Comput. Syst.*, 37:295–318, 2004.
156. D.E. Shaw. A fast, scalable method for the parallel evaluation of distance limited pairwise particle interactions. *J. Comp. Chem.*, 26:1318–1328, 2005.
157. K.E. Bowers, R.O. Dror, and D.E. Shaw. The midpoint method for parallelization of particle simulations. *J. Chem. Phys.*, 124:184109, 2006.
158. D. Brown, J. H. R. Clarke, M. Okuda, and T. Yamazaki. A domain decomposition parallel processing algorithm for molecular dynamics simulations of polymers. *Comp. Phys. Comm.*, 83:1, 1994.
159. M. Pütz and A. Kolb. Optimization techniques for parallel molecular dynamics using domain decomposition. *Comp. Phys. Comm.*, 113:145–167, 1998.
160. G. Sutmann, R. Winkler, and G. Gompper. Multi-particle collision dynamics coupled to molecular dynamics on massively parallel computers, 2009. (to be submitted).