Molecular Dynamics - Vision and Reality

Godehard Sutmann

published in

Computational Nanoscience: Do It Yourself!,
J. Grotendorst, S. Blügel, D. Marx (Eds.),
John von Neumann Institute for Computing, Jülich,
NIC Series, Vol. **31**, ISBN 3-00-017350-1, pp. 159-194, 2006.

Molecular Dynamics - Vision and Reality

Godehard Sutmann

John von Neumann Institute for Computing
Central Institute for Applied Mathematics
Forschungszentrum Jülich
52425 Jülich, Germany
E-mail: g.sutmann@fz-juelich.de

> “*We may regard the present state of the universe as the effect of its past and the cause of its future. An intellect which at any given moment knew all of the forces that animate nature and the mutual positions of the beings that compose it, if this intellect were vast enough to submit the data to analysis, could condense into a single formula the movement of the greatest bodies of the universe and that of the lightest atom; for such an intellect nothing could be uncertain and the future just like the past would be present before its eyes.*“[1]
>
> Marquis Pierre Simon de Laplace, 1814

1 Introduction

With the age of enlightenment and the development of mathematical tools the vision of computability and predictability of natural phenomena arose among scientists and philosophers. Pierre Simon de Laplace phrased this vision in terms of a controlling, omniscient instance which would be able to look into the future as well as into the past due to the deterministic nature of processes, governed by the solution of differential equations. This omniscient instance, introduced by Laplace was henceforth called the *Laplace demon*. This rational view of Laplace, however, had to be corrected with the advent of chaos theory, starting with the work of Poincaré, which states that minimal changes in initial configurations of nonlinear differential equations might lead to a diverging behavior between solutions. Although the general view of Laplace's vision is corrected nowadays by chaos theory and quantum mechanics, it expresses two main features of classical mechanics, i.e. (i) determinism of processes and (ii) time reversibility of the fundamental equations. His understanding of nature was one of the first ideas for doing molecular dynamics simulations, i.e. considering an isolated system of particles, the behavior of which is fully determined by the solution of the classical equations of motion

$$\dot{\mathbf{p}}_i = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i} \quad , \quad \dot{\mathbf{q}}_i = \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} \tag{1}$$

where $\mathcal{H}$ is the Hamiltonian of the system and $\mathbf{p}_i$, $\mathbf{q}_i$ are the generalized momenta and coordinates of particle with index i. However, as is shown in Section 3, even for small systems, which are precisely described by initial and boundary conditions, the vision of Laplace is not fulfilled.

Nevertheless, the main line, which governs the idea of *Laplace's demon* is found in modern simulation methodologies and computer simulations based on this principle have become a powerful tool to treat the dynamical behavior of nonlinear many-body systems. The initial conditions as well as boundary conditions enter either from theoretical considerations (limiting laws) or from experimental values. If the physical system is fully characterized, simulations may be an indispensable tool to solve theoretical models beyond certain approximations or to provide additional help for experimentalists to get a deeper view into phenomena or to answer questions which are not possible to treat with current experimental facilities.

Although there are different methods to obtain information about complex systems, particle simulations always require a model for the interaction between system constituents. The model can only be considered as an approximation to reality, but computer simulations should provide a tool to solve this model exactly (within numerical precision). In order to get a connection to real world systems, 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 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 moves trajectories discretely in time. The time step δt 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 define 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 the *real system* properties or 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 investigations, like quantum *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.

The problem which is often met in molecular simulations is the existence of a variety of time scales, which govern the physical system. Since molecular dynamics consists of an integration of differential equations, one often has to take into account the fastest motion in the system by imposing a small time step of integration in order to sample the movements correctly and conserve first integrals. Considering e.g. simulations of DNA, there is a huge gap between time scales which govern certain phenomena, i.e. fast atomic vibrations take place on a sub-picosecond time scale (10^{-14} s), twisting of the molecule on a picosecond scale (10^{-12} s), bending of the whole molecule on a nanosecond scale ($10^{-7} - 10^{10}$ s) and supercoiling of the molecule on a broad time scale band up to seconds (10^{-6} – 10^{0} s)[2] (similar range of timescales holds for protein dynamics[3]). Therefore, *brute-force* calculations have to be done on the most powerful computers at hand, or approximations have to be done, reducing system components in order to coarsen the dynamics of the system. In the second approach it is of course very important to keep the essential ingredients which still reproduce global motions or long time dynamics. Examples for this approach may be found e.g. in Ref. 4. An impressive example for the first approach was given by the group of Peter Kollman, where a small protein (subdomain HP-36 from the villin headpiece) was studied in an all-atom simulation for a micro-second time interval[5,6]. Although this is still a small system, consisting of about 10000 atoms (600 protein atoms and 3000 water molecules), CRAY T3D and CRAY T3E machines with 256 processors had to be kept busy (continuously) for several months. This huge computational work had to be spent because 5×10^8 steps of integration had to be performed because of the small time step of integration of $\delta t = 2 \times 10^{-15} s$ which was necessary to apply in order to resolve high frequency motions in the system. Although with the development of computer hardware the time scales as well as the system sizes can be extended more and more it is still a current field of research how to model and simulate complex systems on a long time scale[7].

It is clear that the performance of particle dynamics 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[8,9] who simulated systems of hard spheres. In this early simulation, which was run on an IBM-704, up to 500 particles could be simulated, for which 500 collisions per hour could be calculated. Taking into account 200000 collisions for a production run, these simulations lasted for more than two weeks. Nowadays, systems for several million or even billion particles are performed[10–12]. These huge systems make parallel computing an indispensible tool. Fortunately, molecular dynamics is in principle possible to parallelize to 100% (not taking into account input/output operations). Therefore, the problem sizes and timescales are usually extended with the inauguration of more powerful parallel systems. Nevertheless it is still challenging to develop programs which are able to scale up to thousands of processors. In principle, new architectures, like the IBM Blue Gene/L[13,14] should provide a platform for tackling the grand challenge problems, e.g. protein folding. Nevertheless, practical experience tells that parallel programs do not scale good enough to use numbers of processors up to 10000 or more. This makes it necessary to improve parallel algorithms and to think in new algorithms, which overcome different time and length scales[7,15].

Classical molecular dynamics methods are nowadays applied to a huge class of problems, 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)[16] there are a lot of computer codes 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[17], CHARMM[18], NAMD[19], NWCHEM[20] and LAMMPS[21].

2 Models and Methods for Particle Interactions

The part of a simulation where physics come into play is the modeling of interactions between particles. This is the part of a molecular dynamics program which makes the difference between simulating a galaxy or simualting a droplet of water molecules. In general the physical system is determined by its Hamiltonian (or other way around, the Hamiltonian is written down according to the system under consideration). It can be written as *intrinsic* part $\mathcal{H}_0$ and *external* part $\mathcal{H}_1(t)$

$$\mathcal{H} = \mathcal{H}_0 + \mathcal{H}_1(t) \tag{2}$$

where the time-dependence of $\mathcal{H}_1$ indicates that time-varying external fields, e.g. sinusoidal laser beams, may enter into the energetic description. If the external part of the Hamiltonian is omitted then it is clear from classical mechanics that the system Hamiltonian is a conserved quantity.

The intrinsic part of the Hamiltonian can often be written as

$$\begin{aligned} \mathcal{H}_0 = \frac{1}{2}\mathbf{p}^T\mathbf{M}^{-1}\mathbf{p} + \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\theta}^{-1}\boldsymbol{\omega} + \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} u(r_{ij}) \\ + \sum_{i=1}^{N-2}\sum_{j=i+1}^{N-1}\sum_{k=j+1}^{N} u^{(3)}(r_{ij}, r_{ik}, r_{jk}) + \dots \end{aligned} \tag{3}$$

where $\mathbf{p}$ is a 3N-dimensional vector of particle momenta, $\mathbf{M}$ a diagonal mass matrix, $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ the distance between particle pairs and $u(r)$ a pair potential function. Furthermore, if simulating rigid bodies or molecules with fixed atomic distances, $\boldsymbol{\omega}$ is the angular velocity and $\boldsymbol{\theta}$ the tensor of inertia. If not only pair interactions are to be considered, 3-body potentials $u^{(3)}$, or multi-body potentials $u^{(n)}$ can be included into the Hamiltonian. Mainly this is avoided, since it is not easy to model and also it is rather time consuming to evaluate potentials and forces originating from these many-body terms.

Roughly speaking the potential functions can be classified into two main groups, namely short-range and long-range interactions. One can estimate the range of a potential function by considering the leading term of its expansion in powers of r^{-n}. The integral over the leading term gives an estimate of the range of influence, i.e.

$$\int \frac{1}{r^n}\, dr^d = \begin{cases} \text{finite} & : \text{short range} \\ \infty & : \text{long range} \end{cases} \tag{4}$$

where d is the dimension of the physical space. Therefore, in 3-dimensional space, potentials dropping as r^{-3}, e.g. dipole-dipole interactions, are still long ranged.

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[17]) looks as follows

$$\mathcal{U} = \sum_{\text{bonds}} K_r(r - r_{eq})^2 + \sum_{\text{angles}} K_\theta(\theta - \theta_{eq})^2 + \sum_{\text{dihedrals}} \frac{V_n}{2}[1 + \cos(n\phi - \gamma)] \quad (5)$$
$$+ \sum_{i<j} \left[\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}}\right] + \sum_{\text{H-bonds}} \left[\frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}}\right] + \sum_{i<j} \frac{q_i q_j}{r_{ij}}$$

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

2.1 Short Range Interactions

In Eq. (5) the first three terms are examples for bonded, short range interactions. They consist of bond vibrations (i.e. two molecules oscillating in a potential), angular vibrations (i.e. varying bond angle between three molecules) and dihedral motion (i.e. torsional deformation of four bonded particles). It is clear by definition that these interactions are short ranged. Since the bond partners between molecules are fixed in a simulation (in principle it is possible to include bond breaking in an approximate way[22] also in a classical simulation), one can set up fixed lists in the beginning of the simulation, containing all indices of particle-pairs, -triples, -quadruples etc. which exhibit bonded interactions. This omits lengthy checks, which particles take part in an interaction.

The next two terms are examples for short range, non-bonded terms. The first is the famous Lennard-Jones potential, which includes a strongly repelling term, modeling the Pauli exclusion, i.e. hard core repulsion. The attracting term on the other hand models the effect of interactions between induced dipoles, attraction due to fluctuating charge distributions. The other term is sometimes included to give a more realistic description of hydrogen bonds. This term is only evaluated between special particle pairs, i.e. hydrogens and electronegative atoms, which are able to form hydrogen bonds. The constants, $A_{ij}, B_{ij}, C_{ij}, D_{ij}$ are in general different for different atom types or pairs, i.e. they have to be adjusted by proper modeling. Often the constants are evaluated for single atom types and the parameters for cross terms between different atoms are calculated by combination rules.

The Lennard-Jones and the hydrogen bond terms can be written in a different way

$$u(r_{i\alpha,j\beta}) = 4\epsilon_{\alpha\beta}\left(\left(\frac{\sigma_{\alpha\beta}}{r_{i\alpha,j\beta}}\right)^a - \left(\frac{\sigma_{\alpha\beta}}{r_{i\alpha,j\beta}}\right)^b\right) \quad (6)$$

where i, j denotes different particle indices and α, β different types of atoms (for Lennard-Jones e.g. it is $\sigma = (A/B)^{-6}$, $\epsilon = B^2/4A$, $a = 12$ and $b = 6$). The most frequently applied combining rule is the Lorentz-Berthelot formula[23]

$$\sigma_{\alpha\beta} = \frac{\sigma_{\alpha\alpha} + \sigma_{\beta\beta}}{2} \quad , \quad \epsilon_{\alpha\beta} = \sqrt{\epsilon_{\alpha\alpha}\epsilon_{\beta\beta}} \quad (7)$$

It is simply noted here that there are also more complicated rules, e.g. the Kong rule[24] or the Waldman-Hagler rule[25] which sometimes give better results in the calculation of pressure-density profiles of liquids[26].

Since these potentials are short ranged one can restrict the evaluation to limited region of space. Usually one introduces a cutoff radius, R_c, beyond which the interactions are set to zero. A common choice for the cutoff radius is given by $R_c = 2.5\sigma$. Since at R_c the interaction is not identical zero, it is clear that particles entering from outside R_c into inside region exhibit a jump in potential and forces. Formally, the force at the cutoff distance ($\mathbf{F}_{ij} = -\nabla u(r_{ij})$) is infinitely large, since the potential exhibits a step due to truncation. This sudden acceleration of particles usually leads to a heating of the system, since the motion is not reversible. Consider, e.g. a two-particle system, where one particle comes from outside and moving with velocity v_0 to an interparticle distance $r = R_c - \varepsilon$, with $\varepsilon << 1$. Then it gets an abrupt force contribution, accelerating it, leading to a velocity $v_1 > v_0$. On the other hand, if a particle starts from $R_c - \varepsilon$ with velocity v_1 to leave the sphere with radius R_c, then it still has velocity $v_1 > v_0$. For a many-particle system this means that $\langle v^2 \rangle$ increases due to many crossings and recrossings of the interaction sphere, i.e. the temperature increases.

In order to avoid this statistical effect, one may either introduce smoothing functions, which continuously drop the potential and the forces to zero. The disadvantage with this approach is that there will be a zone of large forces at the cutoff distance if the forces are properly evaluated as derivatives of the potential. Therefore one has to smooth forces in this region, leading however to a non-conservative system. Therefore a different method is most often used, which consists in shifting the whole potential and force by a certain amount, which guarantees that both the potential and the force are exactly zero at the cutoff distance, i.e.

$$u^{(sfp)}(r_{ij}) = \begin{cases} u(r_{ij}) - u(R_c) + (r_{ij} - R_c)\, F(R_c) & : r_{ij} \le R_c \\ 0 & : r_{ij} > R_c \end{cases} \tag{8}$$

$$\mathbf{F}^{(sfp)}(\mathbf{r}_{ij}) = \begin{cases} \mathbf{F}(\mathbf{r}_{ij}) - F(R_c)\, \hat{\mathbf{r}}_{ij} & : r_{ij} \le R_c \\ 0 & : r_{ij} > R_c \end{cases} \tag{9}$$

This ensures a smooth transition from outside to inside the cutoff region and vice versa.

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[27], 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

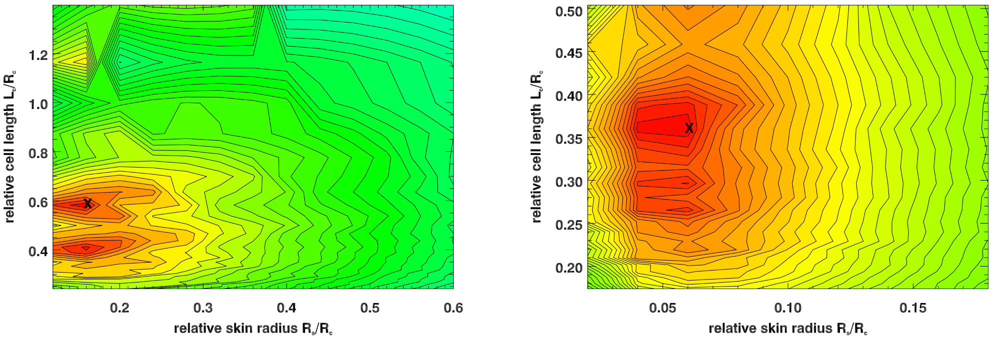

Figure 1. Contour plots of the performance for the combination of linked-cell and Verlet list as a function of the cell length and the size of the skin radius. Crosses mark the positions predicted from an optimization procedure[29]. Test systems were composed of 4000 Lennard-Jones particles with $R_c = 2.5\,\sigma$ at temperature $T = 1.4\,\epsilon/k_B$. Left: $\rho = 0.75/\sigma^3$. Right: $\rho = 2.0/\sigma^3$.

time to time. A criterion to update the list could be, e.g.

$$\max_i |\mathbf{r}_i(t) - \mathbf{r}_i(t_{upd})| \geq \frac{R_s}{2} \tag{10}$$

where t_{upd} 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. 28 it was shown that an optimal choice 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[30, 31]. 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[28], 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 1 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[29].

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[32–34]
  - Grid based algorithms, e.g. Smoothed Particle Mesh Ewald[35, 36], Particle-Particle Particle-Mesh method[37–39]
- Open boundary conditions
  - Grid free algorithms, e.g. Fast Multipole Method[40–45] (FMM), Barnes-Hut Tree method[46, 47]
  - Grid based algorithms, e.g. Particle-Particle Particle-Multigrid method[48] (P^{3}Mg), Particle Mesh Wavelet method[49] (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[50], 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

$$\phi(\mathbf{r}_i) = \sum_{\mathbf{n}}{}^{\dagger} \sum_{j=1}^{N} \frac{q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|} \tag{11}$$

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. (11) 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[51]). 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

$$\begin{aligned}\phi(\mathbf{r}_i) = & \sum_{\mathbf{n}}{}^{\dagger} \sum_{j=1}^{N} q_j \frac{\text{erfc}(\alpha|\mathbf{r}_{ij} + \mathbf{n}L|)}{|\mathbf{r}_{ij} + \mathbf{n}L|} \\ & + \frac{4\pi}{L^3} \sum_{\mathbf{k}\neq 0} \sum_{j=1}^{N} \frac{q_j}{|\mathbf{k}|^2} e^{-|\mathbf{k}|^2/4\alpha^2} e^{i\mathbf{k}\mathbf{r}_{ij}} - q_i \frac{2\alpha}{\sqrt{\pi}}\end{aligned} \tag{12}$$

The last term corresponds to a self energy contribution which has to be subtracted, as it is considered in the Fourier part. Eq. (12) is an exact equivalent of Eq. (11), 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

$$R_c^2 + \frac{\log(R_c)}{\alpha^2} = \frac{1}{\alpha^2}(p^2 - \log(2)) \tag{13}$$

$$k_{max}^2 + 8\alpha^2 \log(k_{max}) = 4\alpha^2 p^2 + \log\left(\frac{4\pi}{L^3}\right) \tag{14}$$

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

$$R_c = \frac{p}{\alpha} \tag{15}$$

$$k_{max} = 2\alpha p \tag{16}$$

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, α and k_{max} can be chosen to get a complexity of $\mathcal{O}(N^{3/2})$ for the Ewald sum[52,53]. In this case, parameters are

$$\frac{R_c}{L} \approx \sqrt{\frac{\pi}{N^{1/3}}} \quad , \quad \alpha L \approx \frac{L k_{max}}{2\pi} = \sqrt{\pi N^{1/3}} \tag{17}$$

Figure 2 shows the contributions of real- and reciprocal parts in Eq. (12), as a function of the spreading parameter α, 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 α 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.

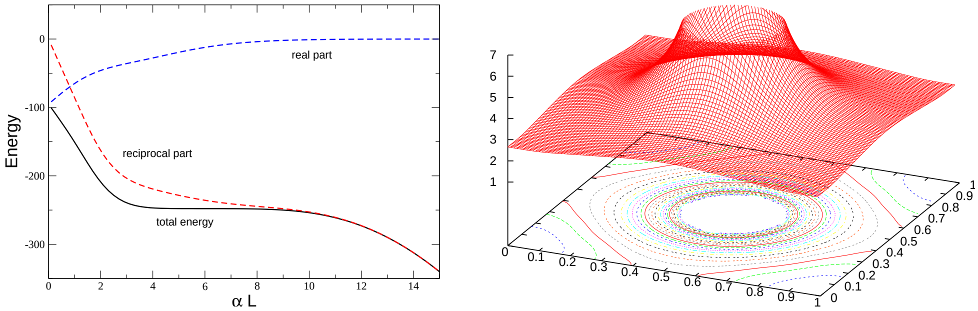

Figure 2. Left: Dependence of the calculated potential on the choice of the scaled inverse width, $\alpha\, L$, of the smeared counter charge distribution. Parameters for this test were $N = 152$, $R_c = 0.5\, L$ and $k_{max} L/2\pi = 6$. Right: Surface plot and contours for the electrostatic potential 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 $k_{max} L/2\pi = 6$.

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

$$\phi(\mathbf{r}_i) = \sum_{j=1}^{N} \frac{q_j}{|\mathbf{r}_i - \mathbf{r}_j|} \tag{18}$$

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[54–56]. 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[57]

$$\frac{1}{|\mathbf{r}-\mathbf{a}|} = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{(l-|m|)!}{(l+|m|)!} \frac{a^l}{r^{l+1}} P_{lm}(\cos\alpha)\, P_{lm}(\cos\theta)\, e^{-im(\beta-\varphi)} \tag{19}$$

where $P_{lm}(x)$ are associated Legendre polynomials[58]. 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

$$O_{lm}(\mathbf{a}) = a^l\, (l-|m|)!\, P_{lm}(\cos\alpha)\, e^{-im\beta} \tag{20}$$

$$M_{lm}(\mathbf{r}) = \frac{1}{r^{l+1}} \frac{1}{(l+|m|)!}\, P_{lm}(\cos\theta)\, e^{im\varphi} \tag{21}$$

with which Eq. (19) 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 $\mathbf{a}+\mathbf{b}$

$$O_{lm}(\mathbf{a}+\mathbf{b}) = \sum_{j=0}^{l} \sum_{k=-l}^{j} A_{jk}^{lm}(\mathbf{b})\, O_{jk}(\mathbf{a}) \qquad , \qquad A_{jk}^{lm}(\mathbf{b}) = O_{l-j,m-k}(\mathbf{b}) \tag{22}$$

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

$$M_{lm}(\mathbf{a}-\mathbf{b}) = \sum_{j=0}^{l} \sum_{k=-l}^{j} B_{jk}^{lm}(\mathbf{b})\, O_{jk}(\mathbf{a}) \qquad , \qquad B_{jk}^{lm}(\mathbf{b}) = M_{l+j,m+k}(\mathbf{b}) \tag{23}$$

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 $\mathbf{b}$

$$M_{lm}(\mathbf{r}-\mathbf{b}) = \sum_{j=0}^{l} \sum_{k=-l}^{j} C_{jk}^{lm}(\mathbf{b})\, M_{jk}(\mathbf{r}) \qquad , \qquad C_{jk}^{lm}(\mathbf{b}) = A_{jk}^{lm}(\mathbf{b}) \tag{24}$$

The procedure to calculate interactions between particles is then subdivided into five passes. Figure 3 illustrates four of them. The first pass consists of calculating the multipole

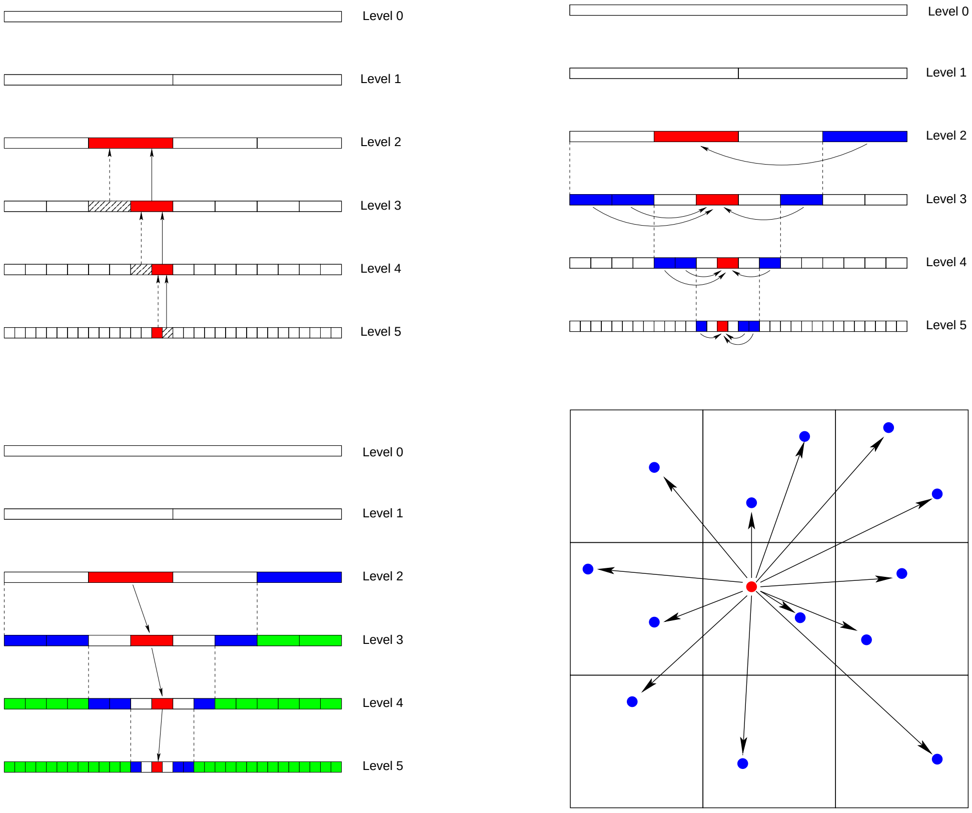

Figure 3. Schematic of different passes in the Fast Multipole Method. Upper left: Pass 1, evaluation of multipole terms in finest subdivision and translating information upwards the tree. Upper right: Pass 2, transforming multipole expansions in well separated boxes into local Taylor expansions. Lower 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.

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. (19) 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[42] 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[45]. A very conservative error estimate is thereby given as[54, 42, 59]

$$\left| \phi(r) - \frac{q}{|\mathbf{r} - \mathbf{a}|} \right| \le \frac{|q|}{r - a} \left(\frac{a}{r} \right)^{p+1} \tag{25}$$

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. (19). 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. 43, 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[60, 61].

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

$$\Phi_{\delta t, \mathcal{H}} : \mathbb{R}^{6N} \to \mathbb{R}^{6N} \tag{26}$$

which means

$$\Phi_{\delta t, \mathcal{H}}(\mathbf{p}(t), \mathbf{q}(t)) = (\mathbf{p}(t) + \delta\mathbf{p}, \mathbf{q}(t) + \delta\mathbf{q}) \tag{27}$$

where

$$\mathbf{p} + \delta\mathbf{p} = \mathbf{p}(t + \delta t) \qquad , \qquad \mathbf{q} + \delta\mathbf{q} = \mathbf{q}(t + \delta t) \tag{28}$$

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

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \delta t \frac{\partial}{\partial \mathbf{q}} \mathcal{H}(\mathbf{p}_n, \mathbf{q}_n) \tag{29}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \delta t \frac{\partial}{\partial \mathbf{p}} \mathcal{H}(\mathbf{p}_n, \mathbf{q}_n) \tag{30}$$

where δt 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. (29,30) may also be written in different forms.

The implicit Euler method

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \delta t \frac{\partial}{\partial \mathbf{q}} \mathcal{H}(\mathbf{p}_{n+1}, \mathbf{q}_{n+1}) \tag{31}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \delta t \frac{\partial}{\partial \mathbf{p}} \mathcal{H}(\mathbf{p}_{n+1}, \mathbf{q}_{n+1}) \tag{32}$$

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*

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \delta t \frac{\partial}{\partial \mathbf{q}} \mathcal{H}(\mathbf{p}_{n+1}, \mathbf{q}_n) \tag{33}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \delta t \frac{\partial}{\partial \mathbf{p}} \mathcal{H}(\mathbf{p}_{n+1}, \mathbf{q}_n) \tag{34}$$

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

$$\mathcal{H}(\mathbf{p},\mathbf{q}) = \frac{1}{2}\mathbf{p}^T\,\mathbf{M}^{-1}\,\mathbf{p} + \mathcal{U}(\mathbf{q}) \tag{35}$$

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

$$\mathbf{p}_{n+1} = \mathbf{p}_n - \delta t \frac{\partial}{\partial \mathbf{q}}\mathcal{U}(\mathbf{q}_n) \tag{36}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \frac{\delta t}{m}\mathbf{p}_{n+1} \tag{37}$$

showing that it is not necessary to solve it iteratively.

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

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

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \frac{\delta t}{m}\,\mathbf{p}_n \tag{39}$$

Applying first Eq. (39) and afterwards Eq. (38) 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

$$\mathbf{q}(t+\delta t) = \mathbf{q}(t) + \delta t\,\dot{\mathbf{q}}(t) + \frac{1}{2}\,\delta t^2\,\ddot{\mathbf{q}}(t) + O(\delta t^3) \tag{40}$$

$$= \mathbf{q}(t) + \frac{\delta t}{m}\,\mathbf{p}(t) + \frac{1}{2m}\,\delta t^2\,\dot{\mathbf{p}}(t) + O(\delta t^3) \tag{41}$$

$$\mathbf{p}(t+\delta t) = \mathbf{p}(t) + \delta t\,\dot{\mathbf{p}}(t) + \frac{1}{2}\,\delta t^2\,\ddot{\mathbf{p}}(t) + O(\delta t^3) \tag{42}$$

$$= \mathbf{p}(t) + \frac{\delta t}{2}\,(\dot{\mathbf{p}}(t) + \dot{\mathbf{p}}(t+\delta t)) + O(\delta t^3) \tag{43}$$

where in Eq. (41), the relation $\dot{\mathbf{q}} = \mathbf{p}/m$ was used and in Eq. (43) 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

$$\mathbf{p}_{n+1/2} = \mathbf{p}_n + \frac{\delta t}{2}\,\mathbf{F}(\mathbf{q}_n) \tag{44}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \frac{\delta t}{m}\,\mathbf{p}_{n+1/2} \tag{45}$$

$$\mathbf{p}_{n+1} = \mathbf{p}_{n+1/2} + \frac{\delta t}{2}\,\mathbf{F}(\mathbf{q}_{n+1}) \tag{46}$$

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

$$\mathbf{q}_{n+1/2} = \mathbf{q}_n + \frac{\delta t}{2m}\,\mathbf{p}_n \tag{47}$$

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \delta t\,\mathbf{F}(\mathbf{q}_{n+1/2}) \tag{48}$$

$$\mathbf{q}_{n+1/2} = \mathbf{q}_{n+1/2} + \frac{\delta t}{2m}\,\mathbf{p}_{n+1} \tag{49}$$

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 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 δt. 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 ω is the vibrational frequency and it is assumed that it oscillates around the origin. The exact solution of this problem may be written as

$$\begin{pmatrix} \omega\, q(t) \\ p(t) \end{pmatrix} = \begin{pmatrix} \cos\omega t & \sin\omega t \\ -\sin\omega t & \cos\omega t \end{pmatrix} \begin{pmatrix} \omega\, q(0) \\ p(0) \end{pmatrix} \tag{50}$$

For a numerical integrator the stepwise solution may be written as

$$\begin{pmatrix} \omega\, q_{n+1} \\ p_{n+1} \end{pmatrix} = \mathbf{M}(\delta t) \begin{pmatrix} \omega\, q_n \\ p_n \end{pmatrix} \tag{51}$$

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

$$\mathbf{M}_{IE}(\delta t) = \frac{1}{1+\omega^2\delta t^2} \begin{pmatrix} 1 & \omega\delta t \\ -\omega\delta t & 1 \end{pmatrix} \tag{52}$$

$$\mathbf{M}_{PV}(\delta t) = \begin{pmatrix} 1 - \frac{1}{2}\omega^2\delta t^2 & \omega\delta t\left(1 - \frac{1}{4}\omega^2\delta t^2\right) \\ -\omega\delta t & 1 - \frac{1}{2}\omega^2\delta t^2 \end{pmatrix} \tag{53}$$

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

$$\lambda_{EE} = 1 \pm i\omega\delta t \tag{54}$$

$$\lambda_{IE} = \frac{1}{1 + \omega^2\delta t^2}(1 \pm i\omega\delta t) \tag{55}$$

$$\lambda_{PV} = \lambda_{VV} = \lambda_{VIE} = \lambda_{PIE} = 1 - \frac{1}{2}\omega^2\delta t^2\left(1 \pm \sqrt{1 - \frac{4}{\omega^2\delta t^2}}\right) \tag{56}$$

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 δt the absolute values of the eigenvalues bifurcates, getting larger and smaller values than one. In Figure 4 the absolute values are shown for all methods and in in Figure 5 the imaginary versus real parts of λ are shown. For EE it is clear that the imaginary part diverges linearly with increase of δt. 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 6. For the stable methods, results 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 $\delta\Gamma = (\delta\mathbf{p}, \delta\mathbf{q})$ is a small disturbance. Then

$$\frac{d\Gamma}{dt} = \nabla_\Gamma \mathcal{H}(\Gamma) \tag{57}$$

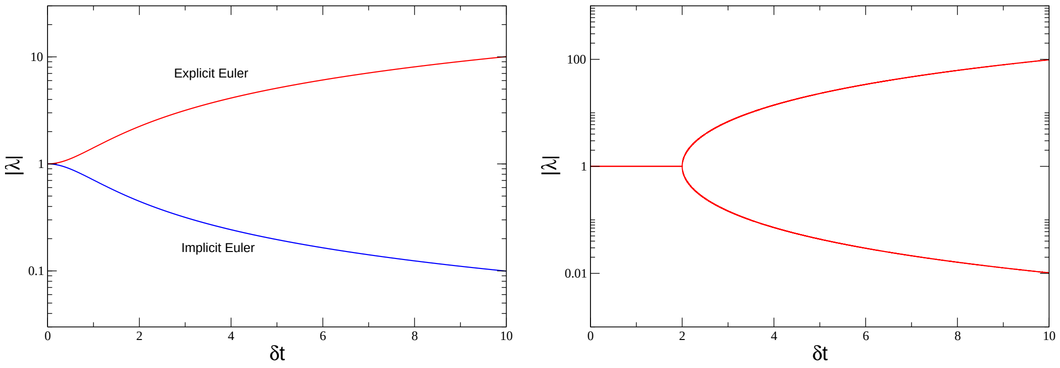

Figure 4. Absolute value of the eigenvalues λ as function of the time step δt. Left: Explicit and implicit Euler method. Right: Velocity and Position Verlet as well as Velocity Implicit and Position implicit Euler method. All methods have the eigenvalues.

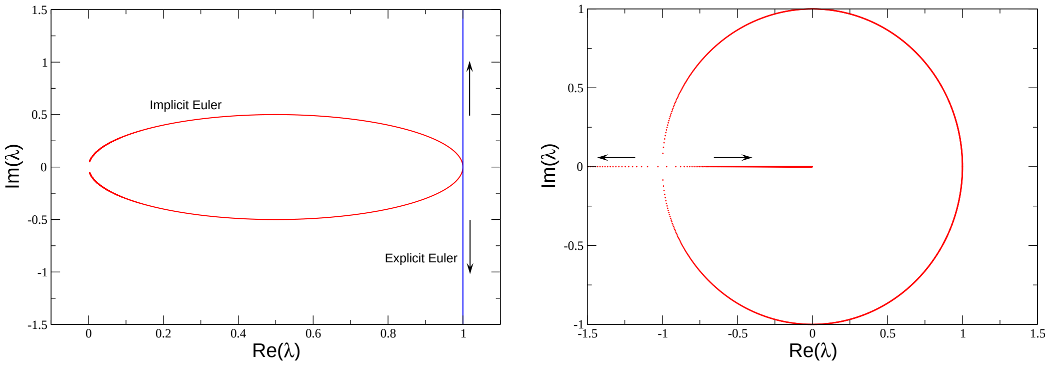

Figure 5. Imaginary versus real part of eigenvalues λ of the propagator matrices. Left: Implicit and Explicit Euler. Right: Velocity and Position Verlet as well as Velocity Implicit and Position implicit Euler method.

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

$$\frac{d}{dt}(\Gamma + \delta\Gamma) = \nabla_\Gamma \mathcal{H}(\Gamma + \delta\Gamma) \tag{58}$$

$$= \nabla_\Gamma \mathcal{H}(\Gamma) + \nabla_\Gamma(\nabla_\Gamma \mathcal{H}(\Gamma))\delta\Gamma \tag{59}$$

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

$$\frac{d\delta\Gamma}{dt} = \nabla_\Gamma^2 \mathcal{H}(\Gamma)\delta\Gamma \tag{60}$$

It is found that the disturbance develops exponentially, with a characteristic, system dependent exponent, which is the Ljapunov exponent[62, 63].

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

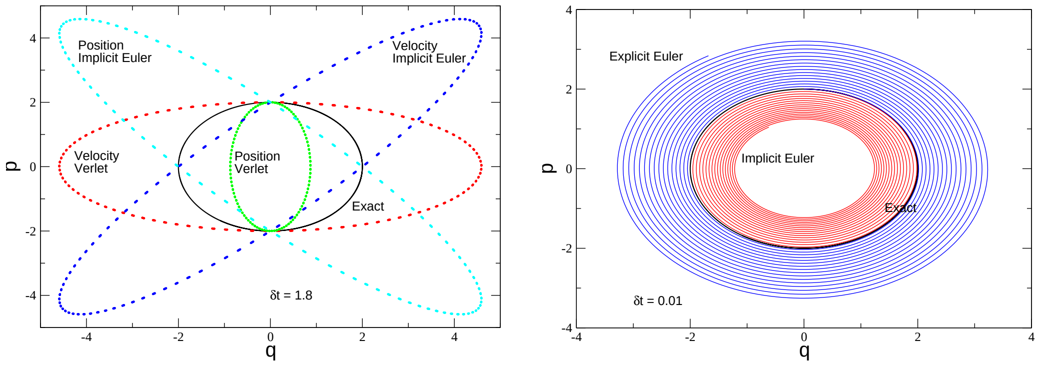

Figure 6. Phase space trajectories for the one-dimensional harmonic oscillator, integrated with the Velocity Implicit Euler, Position Implicit Euler, Velocity Verlet, Position Verlet and integration step size of $\delta t = 1.8$ (left) and the Implicit Euler and Explicit Euler and step size $\delta t = 0.01$ (right).

floating-point operations are used in both algorithms. Due to round off errors, floating-point arithmetic is not necessarily associative, i.e. in general

$$a \overset{\wedge}{\circ} (b \overset{\wedge}{\circ} c) \neq (a \overset{\wedge}{\circ} b) \overset{\wedge}{\circ} c \tag{61}$$

where $\overset{\wedge}{\circ}$ 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, $\delta\Gamma_{\rm min}$, 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

$$\gamma_x(t) = \frac{1}{3N} \sum_{i=1}^{N} \sum_{\alpha=x,y,z} (x(t) - \tilde{x}(t))^2 \tag{62}$$

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 7 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

$$\begin{aligned} \gamma_x(t) &= \langle (x(t) - x(0) - \tilde{x}(t) + x(0))^2 \rangle & (63) \\ &= \langle |x(t) - x(0)|^2 \rangle + \langle |\tilde{x}(t) - x(0)|^2 \rangle & (64) \\ &\quad -2\langle x(t)\tilde{x}(t) \rangle + 2\langle x(0)\tilde{x}(t) \rangle + 2\langle x(t)x(0) \rangle - 2\langle x(0)^2 \rangle \end{aligned}$$

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

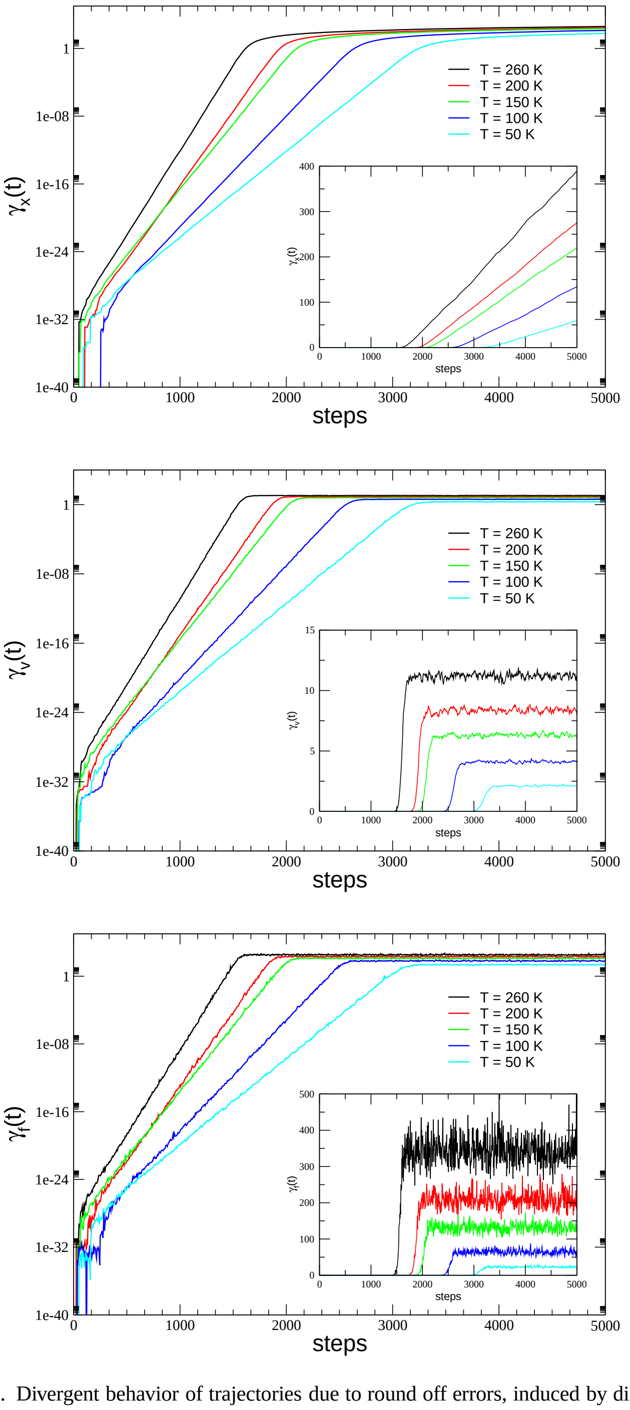

Figure 7. Divergent behavior of trajectories due to round off errors, induced by different summation order in the force routine. From top to bottom: coordinates, velocities, forces. The insets show on a linear scale the long time behavior of the trajectory differences, i.e. when the two systems get decorrelated.

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

$$\lim_{t\to\infty} \gamma_q(t) = 2\langle|\mathbf{q}(t) - \mathbf{q}(0)|^2\rangle = 12Dt \tag{65}$$

$$\lim_{t\to\infty} \gamma_p(t) = 2\langle\mathbf{p}(t)^2\rangle = mk_BT \tag{66}$$

$$\lim_{t\to\infty} \gamma_f(t) = 2\langle\mathbf{F}(t)^2\rangle = 2(\nabla\mathcal{W})^2 \tag{67}$$

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 7 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.

3.3 Time Reversibility

Considering the classical equations of motion for the case of a time independent Hamiltonian, i.e. no externally applied time-dependent potentials, it becomes clear that the dynamics of conservative particle systems should be time reversible. Since the momentum $\mathbf{p}$ has dimension [length/time], a time transformation will lead to

$$t \to -t \qquad , \qquad \mathbf{p} \to -\mathbf{p} \tag{68}$$

Therefore this transform will lead to

$$\frac{\partial \mathbf{p}}{\partial t} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} \quad \overset{t\to -t}{\to} \quad \frac{\partial(-\mathbf{p})}{\partial(-t)} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} \tag{69}$$

$$\frac{\partial \mathbf{q}}{\partial t} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}} = \frac{\mathbf{p}}{m} \quad \overset{t\to -t}{\to} \quad \frac{\partial(\mathbf{q})}{\partial(-t)} = -\frac{\partial \mathcal{H}}{\partial(-\mathbf{p})} = -\frac{\mathbf{p}}{m} \tag{70}$$

showing the equations of motion are unchanged under a time reverse transformation.

Therefore it is not a qualitative change to calculate the mechanical system towards the past and not towards the future. This means that an integrator should be able to propagate a system trajectory up to a certain time step and from there, by a time reversal transformation $\delta t \to -\delta t$, should go exactly the same trajectory back to the starting configuration, i.e.

$$\phi^*_{\delta t} = \phi^{-1}_{-\delta t} \overset{!}{=} \phi_{\delta t} \tag{71}$$

The map $\phi^*_{\delta t}$ is called the adjoint method[64] and if Eq. (71) holds, $\phi_{\delta t}$ is symmetric.

The time reversibility can easily be checked by inspection for each integrator. As an example consider the Position Implicit Euler method. The reverse of this method, to move positions and momenta from step $n+1$ back to step n or by a time reversal would be

$$\begin{aligned} &\text{Reverse} && \text{Transform } \delta t \to -\delta t \\ \mathbf{q}_n &= \mathbf{q}_{n+1} - \frac{\delta t}{m}\mathbf{p}_n & \mathbf{q}_n &= \mathbf{q}_{n+1} - \frac{\delta t}{m}\mathbf{p}_{n+1} \qquad (72) \\ \mathbf{p}_n &= \mathbf{p}_{n+1} - \delta t \mathbf{F}(\mathbf{q}_{n+1}) & \mathbf{p}_n &= \mathbf{p}_{n+1} - \delta t \mathbf{F}(\mathbf{q}_n) \qquad (73) \end{aligned}$$

As it is seen the method with time reversal is not identical to the inverse map, since the derivatives $\partial_q \mathcal{H}$ and $\partial_p \mathcal{H}$ are evaluated at the wrong time step. Therefore the Position Implicit Euler method is not time reversible. The same is true for all other Euler methods, shown in Section 3.1. However, it is instructive to note that the reverse map corresponds to the Velocity Implicit Euler. Applying a proper time reversal transform $\mathbb{T}$ to the algorithms yields the following

$$\mathbb{T}\left\{\phi_{\delta t}^{EE}\right\} = \phi_{\delta t}^{IE} \qquad (74)$$

$$\mathbb{T}\left\{\phi_{\delta t}^{IE}\right\} = \phi_{\delta t}^{EE} \qquad (75)$$

$$\mathbb{T}\left\{\phi_{\delta t}^{PIE}\right\} = \phi_{\delta t}^{VIE} \qquad (76)$$

$$\mathbb{T}\left\{\phi_{\delta t}^{VIE}\right\} = \phi_{\delta t}^{PIE} \qquad (77)$$

Therefore one can simply combine different methods in order to construct a new scheme. In order to propagate the system by a time step δt, apply each scheme for $\delta t/2$. For example consider the combination $(\phi_{\delta t/2}^{EE} \circ \phi_{\delta t/2}^{IE})$, which leads to the following steps

$$\mathbf{q}_{n+1/2} = \mathbf{q}_n + \frac{\delta t}{2m}\mathbf{p}_{n+1/2} \qquad (78)$$

$$\mathbf{p}_{n+1/2} = \mathbf{p}_n + \frac{\delta t}{2}\mathbf{F}(\mathbf{q}_{n+1/2}) \qquad (79)$$

$$\mathbf{q}_{n+1} = \mathbf{q}_{n+1/2} + \frac{\delta t}{2m}\mathbf{p}_{n+1/2} \qquad (80)$$

$$\mathbf{p}_{n+1} = \mathbf{p}_{n+1/2} + \frac{\delta t}{2}\mathbf{F}(\mathbf{q}_{n+1/2}) \qquad (81)$$

A simple comparison of terms yields that $\mathbf{q}_{n+1/2} = (\mathbf{q}_n + \mathbf{q}_{n+1})/2$ and $\mathbf{p}_{n+1/2} = (\mathbf{p}_n + \mathbf{p}_{n+1})/2$ which leads to

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \frac{\delta t}{2m}(\mathbf{p}_n + \mathbf{p}_{n+1}) \qquad (82)$$

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \frac{\delta t}{2}\mathbf{F}((\mathbf{q}_n + \mathbf{q}_{n+1})/2) \qquad (83)$$

which is the Implicit Midpoint Euler scheme (IME).

Combining different types of methods, it comes out that the following relations hold

$$\phi_{\delta t/2}^{EE} \circ \phi_{\delta t/2}^{IE} = \phi_{\delta t}^{IME} \qquad (84)$$

$$\phi_{\delta t/2}^{IE} \circ \phi_{\delta t/2}^{EE} = \phi_{\delta t}^{ITE} \qquad (85)$$

$$\phi_{\delta t/2}^{PIE} \circ \phi_{\delta t/2}^{VIE} = \phi_{\delta t}^{VV} \qquad (86)$$

$$\phi_{\delta t/2}^{VIE} \circ \phi_{\delta t/2}^{PIE} = \phi_{\delta t}^{PV} \qquad (87)$$

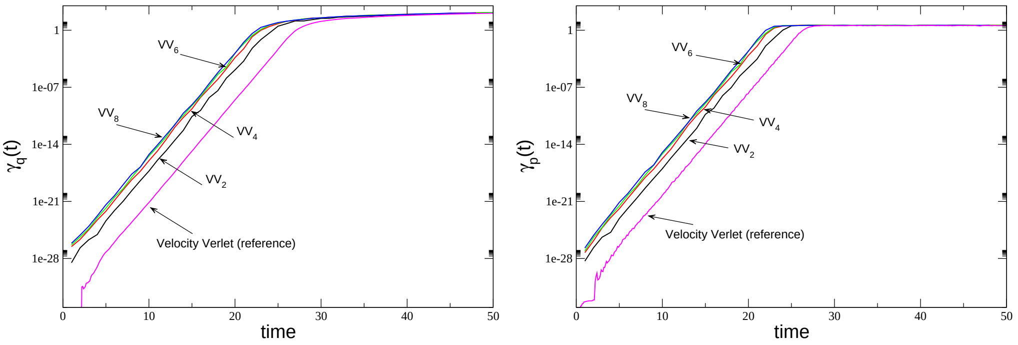

Figure 8. Results for calculations with time reversal transformation, $\delta t \to -\delta t$ after time t. Results show $\overset{\rightleftarrows}{\gamma}_q(t)$ and $\overset{\rightleftarrows}{\gamma}_p(t)$, Eq. (88), for the Velocity Verlet integrator (VV2) and higher order composition schemes (VV4, VV6, VV8, cmp. Section 3.4 and the Appendix). As a reference, results are shown for $\gamma_q(t)$ and $\gamma_p(t)$, obtained for two trajectories with different round off errors at $t = 0$.

where $\phi_{\delta t}^{ITE}$ is the Implicit Trapezoidal Euler. This result shows some interesting properties: (i) symmetric methods can be constructed by composition of non-symmetric methods (example for the adjoint method), (ii) since PIE and VIE are both first order methods, while VV is of second order, this shows that composition of methods increases the order of a method.

Now that time reversibility is established in principle for four different methods (VV, PV, IME, ITE), the question is how good time reversibility is achieved in a molecular dynamics computer simulation. In practice one can test this by simulating $M/2$ time steps and then reversing the sign of δt. For a symmetric method, the trajectories should end up exactly in the initial configuration after M time steps. E.g. if particles started from a regular lattice with zero momentum, this should be exactly the final configuration. Here now it is important to consider the effect of round off errors in the simulation which accumulate during the run. It was already discussed that a very small perturbation of the initial configuration may lead to a diverging behavior of two (initially very close) trajectories. If in the time reverse motion of particles the round off errors are not fully compensated, the trajectory will finally end up a distance apart, which corresponds to an initial perturbation of the size of the cumulated round off error. In Figure 8 the results for such a numerical experiment is shown. Calculated is the norm of the trajectory differences

$$\overset{\rightleftarrows}{\gamma}_x(t) = \|\mathbf{x}(0) - \mathbf{x}_{\leftarrow}(0|t)\|^2 \tag{88}$$

where $\mathbf{x} = (\mathbf{q}, \mathbf{p})$ and t denotes the time step, from where the trajectory was reversed. Figure 8 shows results for this calculation for a system of 864 Lennard-Jones particles. In addition to the trajectory difference, calculated from time reversal, a *reference* curve is shown, which is the result from a different summation order. This shows that true time reversibility cannot be achieved easily for a trajectory in molecular dynamics, although a true time reversible integrator is used. This problem can be traced back to the round off errors. The largest number of floating-point operations usually is done in the force routine of an MD program. This usually takes about 95% of CPU time of the whole simulation time. Since forces are used to update the velocities, which in turn are needed to propagate the coordinates, it seems to be clear that the main source of round off errors is to be found in the force loop. Therefore one attempt would be to reduce round off errors in this part of the program. Several techniques were proposed for this task, e.g. up- or down-sorting for force contributions[65,66], tree like summation or compensated summation[67–74]. These methods have in common that they are able to reduce the round off error, but they do not avoid it. Therefore there will be still a slight deviation from the *true* summation, which may act as a slight perturbation for the trajectory.

Another approach, which avoids round off errors is the use of integer arithmetic[75–78]. In this approach all relevant quantities, e.g. length, time, charge, energy, are scaled in a way that they can be represented by large integer values. The size of the values thus determines the resolution with which differences in lengths etc. can be resolved.

A different approach, which still relies on floating-point arithmetic was introduced by Skeel[79]. If a variable is denoted by x and its upper limit is known, i.e. $\max |x| \leq \bar{x}$, where $\bar{x}$ is a power of 2 and if μ is denoted the target precision, one can construct an equally spaced floating-point grid, with grid spacing $\delta x = 2^{-\mu}\,\bar{x}$. If ν denotes the number of significant bits in the floating-point representation of a machine (often $\nu = 24$ for single precision, $\nu = 53$ for double precision and $\nu = 113$ for quadruple precision), then use the following procedures to transform a machine number x_i with upper bound $\bar{x}$:
If $\mu \leq \nu - 2$ define

$$\text{round}_x(x_i) = (x_i \overset{\wedge}{+} (0.75 \times 2^{\nu-\mu}\,\bar{x})) \overset{\wedge}{-} (0.75 \times 2^{\nu-\mu}\,\bar{x}) \tag{89}$$

If $\mu = \nu - 1$ define

$$\text{round}_x(x_i) = (x_i \overset{\wedge}{+} \text{sign}(x_i)\,\bar{x})) \overset{\wedge}{-} \text{sign}(x_i)\,\bar{x} \tag{90}$$

If $\mu = \nu$ define

$$\text{round}_x(x_i) = (x_i \overset{\wedge}{-} \text{sign}(x_i)\,\bar{x})) \overset{\wedge}{+} \text{sign}(x_i)\,\bar{x} \tag{91}$$

where $\overset{\wedge}{\circ}$ denotes a machine operation.

E.g. the Velocity Verlet algorithm then reads as

$$\mathbf{p}_{n+1/2} = \mathbf{p}_n + \text{round}_p\left((0.5 \overset{\wedge}{\times} \delta t \overset{\wedge}{\div} m) \overset{\wedge}{\times} \mathbf{F}(\mathbf{q}_n)\right) \tag{92}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \text{round}_q\left((\delta t \overset{\wedge}{\div} m) \overset{\wedge}{\times} \mathbf{p}_n\right) \tag{93}$$

$$\mathbf{p}_{n+1} = \mathbf{p}_{n+1/2} + \text{round}_p\left((0.5 \overset{\wedge}{\times} \delta t \overset{\wedge}{\div} m) \overset{\wedge}{\times} \mathbf{F}(\mathbf{q}_{n+1})\right) \tag{94}$$

Varying the value of μ it is possible to change the accuracy of floating-point operations gradually while preserving exact machine operations, i.e. avoiding round off errors. This is shown in Figure 9 where the quadratic displacement between trajectories is shown for interval floating-point arithmetic with different values of μ and a reference trajectory with ordinary floating-point arithmetic. In practical applications it will be expensive to apply the round-function in every force evaluation. Therefore, the forces are calculated in the usual way, accepting round off errors. μ has to be chosen then in such a way that the last digits, affected by round off errors in the forces are truncated in the integration step.

The interest of such round off free algorithms is not only an academic one to prove a theoretical property of an integration scheme. Since round off errors accumulate, the calculated trajectory is dominated by round off errors from a certain time on. This does not mean that the trajectory does not carry any information anymore, but it may lead to the fact that correlations between particles over this time interval cannot be seen in a simulation. Avoiding round off errors this should not happen and small effects, which are usually covered by round off errors show up.

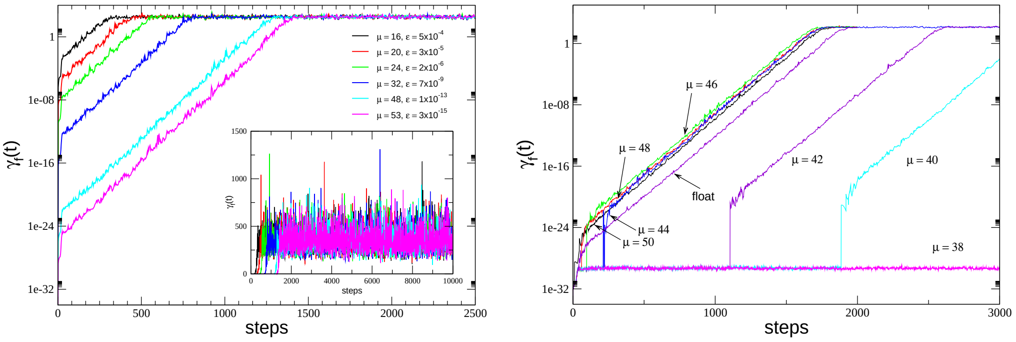

Figure 9. Left: Divergent behavior of system trajectories, calculated with the usual floating-point arithmetic and round off free interval floating-point arithmetic. In both cases a Velocity Verlet integration scheme is used. Right: Result for $\gamma_f(t)$ of floating-point interval arithmetic where trajectories were calculated with different summation sequences.

A numerical test shows that there is no accumulation of numerical errors in such a simulation. Reversing the time step δt after $M/2$ time steps and comparing the two trajectory pieces in the intervals $t \in [1, M/2]\,\delta t$ and $t \in [M/2+1, M]\,\delta t$ leads to differences in trajectories identical to zero (no fluctuations in position-, velocity- or force-differences)! This shows that complete reversibility is possible.

What about the effect of different summation order? If forces are not subject to floating-point interval arithmetic, there will be round off errors and the result of the forces will depend on the order of summing up individual force contributions. This does not do any harm to the integrator, unless the value of $\mathrm{round}_p\left((0.5 \overset{\wedge}{\times} \delta t \overset{\wedge}{\div} m) \overset{\wedge}{\times} \mathbf{F}(\mathbf{q}_n)\right)$ lies in different intervals for different summation orders. It is clear that with a wider floating-point interval the probability of such an event becomes smaller and consequently, for smaller μ it is to be expected that round off errors due to summation order show up later than for large μ.

In Figure 9 the quadratic displacement, $\gamma_f(t)$, of forces is shown due to a different order of summation in the force routine. It is found that the divergent behavior of trajectories starts at different times depending on the value of μ, as can be expected from the above discussion. For $\mu = 38$ there are only small fluctuations on the level of machine accuracy. Nevertheless, also this function will start to diverge at larger times. Compared to floating-point interval arithmetic is the result for usual double precision floating-point arithmetic. It is found that for large μ the diverging behavior is even more pronounced (although the Ljapunov exponent is for all cases the same of course). This effect is due to a stronger effect of initial disturbance in the case of interval arithmetic. This effect, induced by different order of summation, may be avoided by simply using interval arithmetic also in the force routine, i.e. rounding properly every individual force contribution before summing up. This introduces, however, a stronger computational overhead since the round-function has to be applied very much more often (in the case of short range interactions 10-100 times for each particle and time step).

3.4 Accuracy

For an integrator of order $p \geq 1$, the local error may be written as an upper bound[80]

$$||\Phi_{\delta t,\mathcal{H}} - \phi_{\delta t}|| \leq M \delta t^{p+1} \tag{95}$$

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[80]

$$||\Gamma(t_n) - \Gamma_n|| \leq K \left(e^{Lt_n} - 1\right) \delta t^p \qquad , \qquad t_n = n\delta t \tag{96}$$

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 δt 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 3.2.

A different approach to the error behavior of a numerical scheme is backward error analysis, first mentioned in Ref. 81 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

$$\phi_{\delta t}(\Gamma_n) = \Gamma_n + \delta t f(\Gamma) + \delta t^2 g_2(\Gamma) + \delta t^3 g_3(\Gamma) \pm \ldots \tag{97}$$

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

$$f(\Gamma) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \nabla_p \\ \nabla_q \end{pmatrix} \mathcal{H}(\mathbf{p}, \mathbf{q}) \tag{98}$$

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

$$\frac{d}{dt}\tilde{\Gamma} = f(\tilde{\Gamma}) + \delta t f_2(\tilde{\Gamma}) + \delta t^2 f_3(\tilde{\Gamma}) + \ldots \tag{99}$$

where $\tilde{\Gamma}$ will be equivalent to the numerically obtained solution. In order to construct the

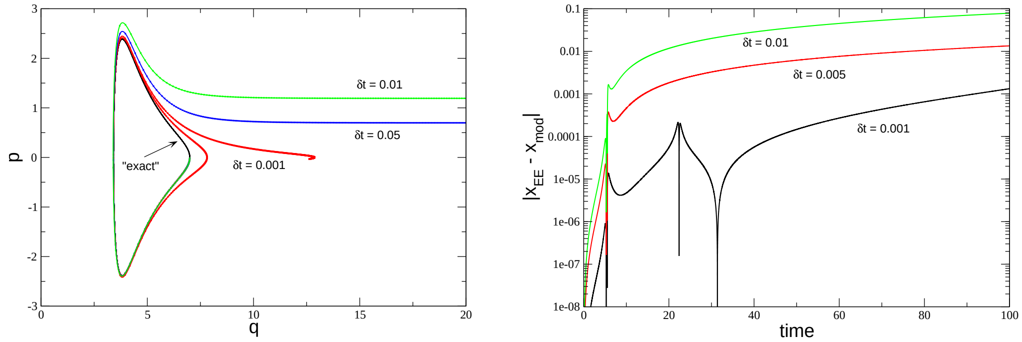

Figure 10. Phase space trajectories of the Lennard-Jones oscillator calculated with the Explicit Euler scheme and different time steps of integration. The *exact* solution (numerical solution of a high order composition scheme with small time step) is shown as a reference - it forms closed orbits. Superimposed to the solutions are results, obtained with a Velocity Verlet scheme, applied to the modified equations, Eqs. (101,102). 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, $|\mathbf{q}_{EE}(t) - \mathbf{q}_{mod}(t)|$.

modified equation, the solution of Eq. (99) is Taylor expanded, i.e.

$$\begin{aligned}\tilde{\Gamma}(t+\delta t) = \tilde{\Gamma}(t) &+ \delta t\left(f(\tilde{\Gamma}) + \delta t f_2(\tilde{\Gamma}) + \delta t^2 f_3(\tilde{\Gamma}) + \ldots\right) \\ &+ \frac{\delta t^2}{2!}\left(f'(\tilde{\Gamma}) + \delta t f_2'(\tilde{\Gamma}) + \ldots\right)\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\left(f(\tilde{\Gamma}) + \delta t f_2(\tilde{\Gamma}) + \ldots\right) \\ &+ \frac{\delta t^3}{3!}\Bigg\{\left(f''(\tilde{\Gamma}) + \delta t f_2''(\tilde{\Gamma}) + \ldots\right)\left(\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\left(f(\tilde{\Gamma}) + \delta t f_2(\tilde{\Gamma}) + \ldots\right)\right)^2 \\ &\qquad + \left(f'(\tilde{\Gamma}) + \delta t f_2'(\tilde{\Gamma}) + \ldots\right)\left(\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\left(f'(\tilde{\Gamma}) + \delta t f_2'(\tilde{\Gamma}) + \ldots\right)\right) \\ &\qquad \times\left(f(\tilde{\Gamma}) + \delta t f_2(\tilde{\Gamma}) + \ldots\right)\Bigg\} \\ &+ \ldots\end{aligned} \tag{100}$$

The procedure to construct the unknown functions f_i proceeds in analogy to perturbation theory, i.e. coefficients with same powers of δt 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, 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

$$\dot{\mathbf{q}} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}} + \frac{\delta t}{2}\frac{\partial \mathcal{H}}{\partial \mathbf{q}} \tag{101}$$

$$\dot{\mathbf{p}} = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} + \frac{\delta t}{2}\mathbf{p}\frac{\partial^2 \mathcal{H}}{\partial \mathbf{p}^2} \tag{102}$$

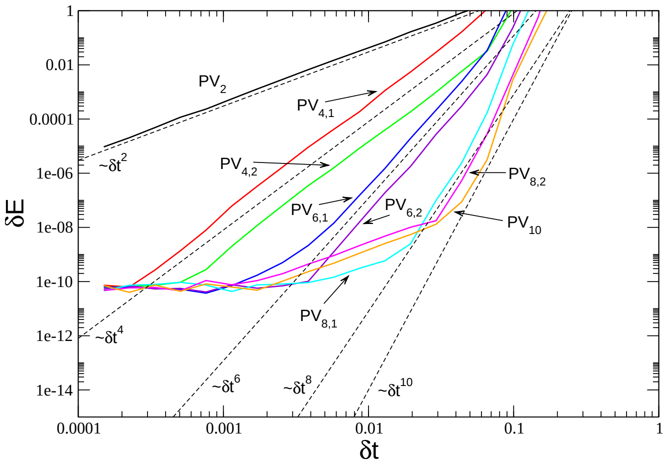

Figure 11. Comparison of symmetric composition high order integrators, based on the Position Verlet scheme. Coefficients of composition methods are given in the Appendix.

Figure 10 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 10) 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 δt into the modified equation. It should be mentioned that, in general, the series expansion of the modified equation diverges.

This gives a method to analyze the error. A better way, of course, is to avoid errors. A natural approach would be to increase the order p of the integrator at hand. As was shown already in Section 3.3 in the context of time reversibility, one can compose different methods in order to get (i) symmetric methods and (ii) increase the order of the method. A method for constructing higher order symplectic methods was proposed by Yoshida[82, 83] and Suzuki[84, 85], where an s-fold composition of symmetric methods was applied. The resulting flow can thus be written as

$$\psi_{\delta t} = \phi_{w_s \delta t} \circ \phi_{w_{s-1}\delta t} \circ \ldots \circ \phi_{w_2 \delta t} \circ \phi_{w_1 \delta t} \tag{103}$$

where $\phi_{\delta t}$ is a symmetric basic method, e.g. Velocity Verlet ($\phi_{\delta t} = \phi_{\delta t/2,\mathcal{T}} \circ \phi_{\delta t,\mathcal{U}} \circ \phi_{\delta t/2,\mathcal{T}}$, Eqs. (44-46,86)) or Implicit Midpoint Rule ($\phi_{\delta t} = \phi_{\delta t/2,\mathcal{H}} \circ \phi^*_{\delta t,\mathcal{H}}$, Eq. (84)) and w_i are chosen parameters which determine the width of a substep. For consistency and if the composition method itself is symmetric, it is required that

$$\sum_{i=1}^{s} w_i = 1 \quad , \quad w_{s+1-k} = w_k \quad , \quad k = 1, \ldots, s \tag{104}$$

In order to calculate the coefficients one has the additional condition

$$\sum_{i=1}^{s} w_i^{p+1} = 0 \tag{105}$$

If the basic method is of order p it can be shown that the composed method is at least of order $p+1$[64]. A symmetric method has an even order since all odd terms in the Taylor expansion cancel. If the composition method is constructed as a symmetric method according to Eq. (104), then $\tilde{\psi}_{\delta t}$ is even (at least) of order order $p+2$. This procedure can be repeated recursively in order to construct symmetric methods of any order. Of course, the number of stages increases each time the order is increased, e.g. starting with a symmetric method of order $p = 2$ as basic method and taking $s = 3$, this will result in a method of order $p = 4$. This order-four method can again be taken as a basic method and one ends up with a resulting order of $p = 6$ and $s = 9$. For even higher methods the number of stages will strongly increase, making this type of high order methods computationally demanding (cmp. Appendix). It is a matter of choice with which number of steps the procedure is started. For $s = 3$, one gets the scheme of Yoshida[82], corresponding to

$$w_1 = w_3 = \frac{1}{2 - 2^{1/(p+1)}} \quad , \quad w_2 = -\frac{2^{1/(p+1)}}{2 - 2^{1/(p+1)}} \tag{106}$$

For $s = 5$ one gets the scheme of Suzuki[84], corresponding to

$$w_1 = w_2 = w_4 = w_5 = \frac{1}{4 - 4^{1/(p+1)}} \quad , \quad w_3 = -\frac{4^{1/(p+1)}}{4 - 4^{1/(p+1)}} \tag{107}$$

There are also refined composition schemes, which do not need that large number of substeps. They are also based on Eq. (103) with symmetric basic methods like the Velocity Verlet. References and coefficients are given in the Appendix. Figure 11 shows the error energy fluctuation

$$\delta E = \sqrt{\frac{1}{N_t} \sum_{k=1}^{N_t} \left(1 - \frac{E_k}{E_0}\right)^2} \tag{108}$$

for a test case of a system of Lennard-Jones particles, where E_k is the total energy in the system at time step k and N_t the number of MD steps. As a basic method the Position Verlet scheme was applied. It is found that all integration schemes obey the prescribed order. Due to numerical round off errors the methods saturate for very small time steps.

4 Conclusions

The old vision of Laplace as it was sketched in Section 1 even cannot be met in computer simulations, where all initial conditions are known *exactly*. Slight disturbances, which may even be introduced by simple round off errors, as was shown by different orders of summation, lead to Lyapunov instabilities of close trajectories. Only in the case, where really discrete lattice maps are introduced, reversibility can be achieved and no effects due to round off errors will be seen. Nevertheless, also in such a discrete system, there will be Lyapunov instabilities if disturbing the initial conditions in the slightest way. These instabilities, however, have no dramatic consequences for the overall stability of trajectories of the system. The system develops into different parts of accessible phase space but if symplectic, time reversible integrators are used to solve the equations of motions, the geometrical structure of phase space is conserved. Physically this behavior can be interpreted as interference of a system with an uncontrollable, slight disturbance, which could happen within an experimental setup e.g. by thermal fluctuations or impurities etc.. This also should not drive a system out of equilibrium. Actually the fact that a disturbance develops an exponential drift of two nearby trajectories is often used to impose artificial initial conditions, i.e. start a simulation from an ordered regular grid and impose a uniform velocity distribution corresponding to a desired temperature. Equilibration will then lead a loss of memory to the initial conditions and to a proper statistical distribution of physical quantities. From that point of view an individual trajectory of particles is of no principal importance. It can only be considered as a possible, allowed realization of the system and lots of different configurations have to be obtained to get a proper averaging over different trajectories.

Appendix

Composition Methods of Order p

As an example a composition is done with a Velocity Verlet integrator as basic method, i.e.

$$\phi_{\delta t,\mathcal{H}} = \phi_{\delta t/2,\mathcal{U}} \circ \phi_{\delta t,\mathcal{T}} \circ \phi_{\delta t/2,\mathcal{U}}$$

Then the composition scheme is performed in a step wise fashion. In the program an s-stage composition scheme is then implemented as follows (here for the Velocity Verlet integrator)

```
do is = 1, s
  do i = 1, n_part
     do k = 1, 3
        v(k,i) = v(k,i) + 0.5*w(is)*h*f(k,i)/m(i)
        x(k,i) = x(k,i) + w(is)*h*v(k,i)
     enddo
  enddo

  ...  calculate forces  ...

  do i = 1, n_part
     do k = 1, 3
        v(k,i) = v(k,i) + 0.5*w(is)*h*f(k,i)/m(i)
     enddo
  enddo
enddo
```

In the following, coefficients for composition methods, based on symmetric schemes, with different number of stages, s and of order p are given. For all cases, the symmetry relation

$$w_{s+1-k} = w_k \qquad , \qquad k = 1, \ldots, s$$

holds. Therefore only coefficients up to $[s/2]+1$ are shown.

stages	order	coefficients	reference
3	4	$w_1 = 1.35120719195965777181 8$ $w_2 = -1.70241440387583820026 4$	Yoshida[82]
5	4	$w_1 = w_2 = 0.4144907717943757119 44$ $w_3 = -0.6579630931394397919 12$	Suzuki[84]
7	6	$w_1 = 0.7845136104775572638 2$ $w_2 = 0.2355732133593581336 8$ $w_3 = -1.1776799841788710069 5$ $w_4 = 1 - 2\ \sum_{s=1}^{3} w_s$	Yoshida[82]
9	6	$w_1 = 0.3921614440073141392 8$ $w_2 = 0.3325991367893594386 0$ $w_3 = -0.7062461725576393598 1$ $w_4 = 0.0822135962935508002 3$ $w_5 = 1 - 2\ \sum_{s=1}^{4} w_s$	Kahan & Li[86]
15	8	$w_1 = 0.7416703643506129534 5$ $w_2 = -0.4091008258000315940 0$ $w_3 = 0.1907547102962383799 5$ $w_4 = -0.5738624711160822666 6$ $w_5 = 0.2990641813036559238 4$ $w_6 = 0.3346249182452981837 8$ $w_7 = 0.3152930923967665966 3$ $w_8 = 1 - 2\ \sum_{s=1}^{7} w_s$	McLachlan[87]
17	8	$w_1 = 0.13020248308889008087881763$ $w_2 = 0.56116298177510838456196441$ $w_3 = -0.38947496264484728640807860$ $w_4 = 0.15884190655515560089621075$ $w_5 = -0.39590389413323757733623154$ $w_6 = 0.18453964097831570709183254$ $w_7 = 0.25837438768632204729397911$ $w_8 = 0.29501172360931029887096624$ $w_8 = 1 - 2\ \sum_{s=1}^{8} w_s$	Kahan & Li[86]
33	10	$w_1 = 0.09040619368607278492161150$ $w_2 = 0.53591815953030120213784983$ $w_3 = 0.35123257547493978187517736$	Kahan & Li[86]

$$
\begin{aligned}
w_4 &= -0.31116802097815835426086544 \\
w_5 &= -0.52556314194263510431065549 \\
w_6 &= 0.14447909410225247647345695 \\
w_7 &= 0.02983588609748235818064083 \\
w_8 &= 0.17786179923739805133592238 \\
w_9 &= 0.09826906939341637652532377 \\
w_{10} &= 0.46179986210411860873242126 \\
w_{11} &= -0.33377845599881851314531820 \\
w_{12} &= 0.07095684836524793621031152 \\
w_{13} &= 0.23666960070126868771909819 \\
w_{14} &= -0.49725977950660985445028388 \\
w_{15} &= -0.30399616617237257346546356 \\
w_{16} &= 0.05246957188100069574521612 \\
w_{17} &= 1 - 2 \sum_{s=1}^{16} w_s
\end{aligned}
$$

References

1. P. S. de Laplace. Théorie analytique des probabilités. In *Oeuvres Complètes de Laplace*, volume VII., Paris, 1820. Gauthier-Villars.
2. T. Schlick. *Molecular Modeling and Simulation*. Springer, New York, 2002.
3. C. L. Brooks III, M. Karplus, and B. M. Pettitt. Proteins: A theoretical perspective of dynamics, structure and thermodynamics. In *Advances in Chemical Physics*, volume LXXI, New York, 1988. John Wiley.
4. J. Baschnagel, K. Binder, P. Doruker, A. A. Gusev, O. Hahn, K. Kremer, W. L. Mattice, F. Müller-Plathe, M. Murat, W. Paul, S. Santos, U. W. Suter, and V. Tries. Bridging the gap between atomistic and coarse-grained models of polymers: Status and perspectives. *Adv. Polym. Sci.*, 152:41, 2000.
5. 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.
6. 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.
7. In P. Nielaba, M. Mareschal, and C. Ciccotti, editors, *Bridging the Time Scales: Molecular Simulations for the next Decade*, Heidelberg, 2002. Springer.
8. B. J. Alder and T. E. Wainwright. Phase transition for a hard sphere system. *J. Chem. Phys.*, 27:1208–1209, 1957.
9. B. J. Alder and T. E. Wainwright. Studies in molecular dynamics. I. General method. *J. Chem. Phys.*, 31:459, 1959.
10. K. Kadau, T. C. Germann, P. S. Lomdahl, and B. L. Holian. Microscopic view of structural phase transitions induced by shock waves. *Science*, 296:1681–1684, 2002.
11. 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.
12. V. Springel, S. D. M. White, A. Jenkins, C. S. Frenk, N. Yoshida, L. Gao, J. Navarro, R. Thacker, D. Croton, J. Helly, J. A. Peacock, S. Cole, P. Thomas, H. Couchman,

A. Evrard, J. Colberg, and F. Pearce. Simulations of the formation, evolution and clustering of galaxies and quasars. *Nature*, 435:629–636, 2005.

13. http://www.fz-juelich.de/zam/ibm-bgl.
14. http://www.research.ibm.com/bluegene/.
15. G. Lu and E.Kaxiras. Overview of multiscale simulations of materials. In M. Rieth and W. Schommers, editors, *Handbook of Theoretical and Computational Nanotechnology*, volume X, pages 1–33. American Scientific Publishers, 2005.
16. http://www.ccp5.ac.uk/.
17. http://amber.scripps.edu/.
18. http://www.charmm.org/.
19. http://www.ks.uiuc.edu/Research/namd/.
20. http://www.emsl.pnl.gov/docs/nwchem/nwchem.html.
21. http://www.cs.sandia.gov/∼sjplimp/lammps.html.
22. S. Walbran and A. A. Kornyshev. Proton transport in polarizable water. *J. Chem. Phys.*, 114:10039–10048, 2001.
23. G. C. Maitland, M. Rigby, E. B. Smith, and W. A. Wakeham. *Intermolecular Forces*. Clarendon, Oxford, 1981.
24. 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.
25. M. Waldman and A. T. Hagler. New combining rules for rare gas van der Waals parameters. *J. Comp. Chem.*, 14:1077, 1993.
26. 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.
27. L. Verlet. Computer experiments on classical fluids. I. Thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, 159:98, 1967.
28. G. Sutmann and V. Stegailov. Optimization of neighbor list techniques for molecular dynamics simulations. J. Mol. Liq. (in press), 2005.
29. G. Sutmann and V. Stegailov (to be published).
30. R. W. Hockney. The potential calculation and some applications. *Meth. Comput. Phys.*, 9:136–211, 1970.
31. R. W. Hockney, S. P. Goel, and J. W. Eastwood. Quite high-resolution computer models of a plasma. *J. Comp. Phys..*, 14:148, 1974.
32. P. Ewald. Die berechnung optischer und elektrostatischer gitterpotentiale. *Ann. Phys.*, 64:253, 1921.
33. 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.
34. 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.
35. T. Darden, D. York, and L. Pedersen. A NlogN method for Ewald sums in large systems. *J. Chem. Phys.*, 98:10089, 1993.
36. 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.

37. R. W. Hockney and J. W. Eastwood. *Computer simulation using particles*. McGraw-Hill, New York, 1981.
38. 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.
39. 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.
40. L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comp. Phys.*, 73:325, 1987.
41. H. Cheng, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *J. Comp. Phys..*, 155:468–498, 1999.
42. C. A. White and M. Head-Gordon. Derivation and efficient implementation of the fast multipole method. *J. Chem. Phys.*, 101:6593–6605, 1994.
43. 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.
44. C. A. White and M. Head-Gordon. Fractional tiers in fast multipole method calculations. *Chem. Phys. Lett.*, 257:647–650, 1996.
45. H. Dachsel. An improved implementation of the fast multipole method. In *Proceedings of the 4th MATHMOD Vienna*, Vienna, 2003.
46. J. E. Barnes and P. Hut. A hierarchical $\mathrm{O}(N \log N)$ force calculation algorithm. *Nature*, 324:446, 1986.
47. S. Pfalzner and P. Gibbon. *Many Body Tree Methods in Physics*. Cambridge University Press, New York, 1996.
48. 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.
49. G. Sutmann and S. Wädow. Wavelet based summation of long-range interactions in molecular systems. (in preparation).
50. N. W. Ashcroft and N. D. Mermin. *Solid State Physics*. Saunders College Publishing, Fort Worth, 1976.
51. R. A. Robinson and R. H. Stokes. *Electrolyte Solutions*. Butterworth, London, 1965.
52. 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.
53. D. Fincham. Optimisation of the Ewald sum for large systems. *Molec. Sim.*, 13:1–9, 1994.
54. L. Greengard. *The rapid evaluation of potential fields in particle systems*. MIT press, Cambridge, 1988.
55. L. Greengard. The numerical solution of the N-body problem. *Computers in Physics*, 4:142–152, 1990.
56. L. Greengard. Fast algorithms for classical physics. *Science*, 265:909–914, 1994.
57. J. D. Jackson. *Classical Electrodynamics*. Wiley, New York, 1983.
58. M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions*. Dover Publ. Inc., New York, 1972.
59. 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.

60. 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.
61. M. Challacombe, C. A. White, and M. Head-Gordon. Periodic boundary conditions and the fast multipole method. *J. Chem. Phys.*, 107:10131–10140, 1997.
62. B. V. Chirikov. A universal instability of many-dimensional oscillator systems. *Phys. Rep.*, 52:264–379, 1979.
63. F. Calvo. Largest Lyapunov exponent in molecular systems: Linear molecules and application to nitrogen clusters. *Phys. Rev. E*, 58:5643–5649, 1998.
64. E. Hairer, Ch. Lubich, and G. Wanner. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, Heidelberg, 2002.
65. J. H. Wilkinson. Error analysis of floating point computation. *Numer. Math.*, 2:319–340, 1960.
66. D. J. Higham. The accuracy of floating point summation. *SIAM J. Sci. Comput.*, 14:783–799, 1993.
67. W. Kahan. Further remarks on reducing truncation errors. *Comm. ACM*, 8:40, 1965.
68. O. Møller. Quasi double-precision in floating point addition. *BIT*, 5:37–50 and 251–255, 1965.
69. W. Kahan. A survey of error ananlysis. In *Proc. IFIP congress, Ljubljana, Information Processing*, volume 71, pages 1214–1239, Amsterdam, 1972. North-Holland.
70. W. Kahan. Implementation of algorithms. Technical Report Tech. Rep. 20, Dept. Comp. Science, Univ. California, Berkeley, 1973.
71. M. Jankowski, A. Smoktunowicz, and H. Wosniaskowski. A note on floating-point summation of very many terms. *J. Inform. Process. Cybern.*, 19:435–440, 1983.
72. M. Jankowski and H. Wosniaskowski. The accurate solution of certain continuous problems using only single precision arithmetic. *BIT*, 25:635–651, 1985.
73. A. Kielbasinski. Summation algorithm with corrections and some of its applications. *Math. Stos.*, 1:22–41, 1973.
74. A. Neumaier. Rundungsfehlerananlyse einiger Verfahren zur Summation endlicher Summen. *Z. Angew. Math. Mech.*, 54:39–51, 1974.
75. D. Levesque and W. T. Ashurst. Long time behavior of the velocity autocorrelation function for a fluid of soft repulsive particles. *Phys. Rev. Lett.*, 33:277–280, 1974.
76. C. Scovel. On symplectic lattice maps. *Phys. Lett. A*, 159:396–400, 1991.
77. D. J. Earn and S. Tremaine. Exact numerical studies of Hamiltonian maps: iterating without round off error. *Physica D*, 56:1–22, 1992.
78. D. Levesque and L. Verlet. Molecular dynamics and time reversibility. *J. Stat. Phys.*, 72:519–537, 1993.
79. R. D. Skeel. Symplectic integration with floating-point arithmetic and other approximations. *Appl. Numer. Math.*, 29:3–18, 1999.
80. B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics*. Cambridge University Press, Cambridge, 2004.
81. 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.
82. H. Yoshida. Construction of higher order symplectic integrators. *Phys. Lett. A*, 150:262–268, 1990.

83. H. Yoshida. Recent progress in the theory and application of symplectic integrators. *Celestial Mech. Dynam. Astronom.*, 56:27–43, 1993.
84. M. Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. *Phys. Lett. A*, 146:319–323, 1990.
85. M. Suzuki. General theory of higher-order decomposition of exponential operators and symplectic integrators. *Phys. Lett. A*, 165:387–395, 1992.
86. W. Kahan and R.-C. Li. Composition constants for raising the orders of unconventional schemes for ordinary differential equations. *Math. Comput.*, 66:1089–1099, 1997.
87. R. McLachlan. On the numerical integration of ordinary differential equations by symmetric composition methods. *SIAM J. Sci. Comp.*, 16:151–168, 1995.