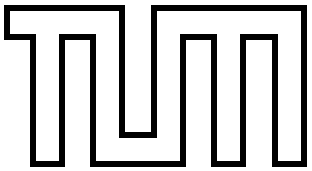


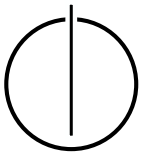
SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

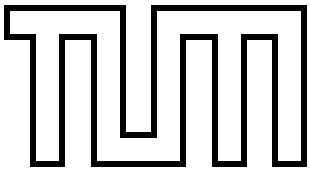
TECHNICAL UNIVERSITY OF MUNICH

Interdisciplinary Project in Mathematics

Evaluation of the Multiple Time-Stepping Method for 3-Body Molecular Dynamics Simulations

David Martin





SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - INFORMATICS

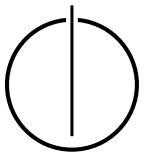
TECHNICAL UNIVERSITY OF MUNICH

Interdisciplinary Project in Mathematics

Evaluation of the Multiple Time-Stepping Method for 3-Body Molecular Dynamics Simulations

Evaluation der Multi-Zeitschrittmethode für Dreikörperwechselwirkungen in Molekulardynamik Simulationen

Author:	David Martin
Supervisor:	Univ.-Prof. Dr. Hans-Joachim Bungartz
Advisor:	Samuel Newcome M.Sc., Markus Mühlhäußer M.Sc.
Date:	06.06.2024



I confirm that this interdisciplinary project is my own work and I have documented all sources and material used.

Munich, 06.06.2024 David Martin

Abstract

Understanding the complex behavior of molecular systems is fundamental to fields such as chemistry, physics, materials science, and biology. Molecular dynamics (MD) simulations are crucial tools for studying atomic-level dynamics. This work focuses on improving the efficiency of MD simulations involving two-body and three-body interactions. Traditional two-body potentials often can not fully capture the complexity of molecular systems, making the inclusion of three-body interactions necessary. However, these interactions have a computational complexity of $\mathcal{O}(n^3)$, compared to $\mathcal{O}(n^2)$ for two-body interactions. High Performance Computing (HPC) offers solutions for parallelizing these calculations to reduce computational time. Additionally, using Multiple Time-Stepping methods like the r-RESPA algorithm can further enhance efficiency by integrating certain interactions at different time steps. This study investigates the feasibility of using r-RESPA to integrate three-body interactions with a larger time step than two-body interactions, aiming to reduce computational cost while maintaining accuracy. The results and methods are discussed, providing insights into potential advancements in MD simulation efficiency.

Contents

Part I.

Introduction and Background

1. Introduction

Analyzing the behavior of molecular systems is crucial to numerous disciplines, including chemistry, physics, materials science, and biology. Molecular dynamics (MD) simulations have emerged as essential tools for studying the dynamics of molecular systems. These simulations allow researchers to observe and analyze the motions and interactions of atoms and molecules over time. Throughout this thesis we use the term *particle* to denote a molecule or an atom.

Crucial to the accuracy of MD simulations is the comprehensive modeling of interatomic interactions. Traditionally, two-body potentials such as the Lennard-Jones and Coulomb potentials are used to describe the pairwise interactions between atoms. However, many molecular systems exhibit complex behavior that cannot be adequately captured by pairwise interactions alone. In particular, three-body interactions, where the motion of one atom is influenced by the presence of two others, play a crucial role in determining the structural stability, phase transitions and dynamic properties of different materials like noble gases for example [MS99].

However, three-body interactions are very computationally expensive [May23] and are in a complexity class of $\mathcal{O}(n^3)$ compared to two-body interactions, which are in a complexity class of $\mathcal{O}(n^2)$ with n particles. Accordingly, there is interest in High Performance Computing (HPC) in calculating these interactions as efficiently as possible. Algorithms that efficiently parallelize the calculation of three-body interactions and distribute them across several compute nodes, for example, offer possibilities for reducing the calculation time.

Another aspect that makes the calculation of MD simulations so expensive is that, depending on the system to be simulated, we have to use very small time steps in the femtosecond range for the approximation algorithms for time integration in order to minimize numerical errors. Accordingly, it can take hours, days or weeks to calculate such simulations over a longer time period. [LZS06]

Also here there are possibilities to reduce the calculation time. These include, for example, time integration algorithms such as the r-RESPA (reversible Reference System Propagator Algorithm) method [GKZ07, LM15], which integrate certain interactions between particles with different time steps. Such algorithms are also known as Multiple Time-Stepping methods. For example, forces that have a very high frequency can be integrated with smaller time steps than forces that have a lower frequency.

In this work, we address the question of whether we can use such Multiple Time-Stepping methods, specifically the r-RESPA method, to integrate two-body and three-body interactions with different time steps. The underlying idea is that in some scenarios the very computationally expensive three-body interactions rather act as a corrector for the two-body

interactions and thus in the equilibrium of the MD system the vibrations are largely determined by the two-body interactions. In this work, we investigate whether we can integrate the very computationally expensive three-body interactions with larger time steps than the two-body interactions.

The paper is organized as follows: In chapter 2 we give a rough overview of required theoretical background, a literature review on numerical time integration and Multiple Time-Stepping methods. Chapter 3 deals with the required technical background for this report. In chapter 4, we give an outline of which methods have been used so far to reduce the computational cost of three-body interactions. Chapter 5 briefly describes which methods we have chosen to perform our measurements and how we have implemented them. In chapter 6 we discuss the results of our work. In the last chapter, we then give a brief outlook on how this work can be continued.

2. Theoretical Background

2.1. Molecular Dynamic Simulations

To simulate the motion of particles over a certain time interval $I = [t_{start}, t_{end}]$, we need to solve the equations of motion from classical mechanics. They are given, for example, by Newton's equation of motion

$$F = m \cdot a \tag{2.1}$$

where m describes the mass and a the acceleration of a particle.

However, we can not directly solve these equations analytically. In order to be able to simulate the motion of particles with computers, the corresponding time interval I is discretized into time steps of the size δt. In each of these time steps, the equations of motion from classical mechanics are solved using numerical methods. Details on numerical integration can be found in section 2.2.

To calculate the acceleration a from Formula 2.1, we have to calculate the force acting on each individual particle at each time step. With the acceleration, we can calculate the velocity, with which we can finally update the position of the particles. This force can be calculated using the negative gradient of the potential:

$$F = -\nabla U \tag{2.2}$$

In order to approximate the force acting on the particles as accurately as possible, there are various potential functions that are used in MD simulations to calculate different types of interactions. These can be divided into short-range and long-range potentials, with short-range potentials decreasing rapidly with increasing distance between the interacting particles, while long-range potentials are designed to calculate interactions with large distances between the particles. In this work we only use short-range potentials. Furthermore, potential functions can be classified according to the number of particles they use to calculate the interactions. The most common potential functions consider pairwise interactions between two particles.

To calculate the exact force acting on a particle i we have to add up the sum of all potentials from single-body to N-body potential and calculate the negative gradient:

$$F_i = -\nabla \left[\phi_1(i) + \sum_j \phi_2(i,j) + \sum_{j,k} \phi_3(i,j,k) + \dots \right] \tag{2.3}$$

As the number of particles involved in a potential function increases, the contribution to the overall potential decreases, but the calculation effort increases. In practice, it would therefore be too time-consuming to calculate all N-body potentials and is often limited to two-body potentials.

2.1.1. Lennard Jones 12-6 Potential

The Lennard Jones 12-6 potential is a common short range potential used in MD simulations for two-body interactions. It is defined as follows:

$$\phi_2(i,j) = 4\epsilon_{LJ}\left[\left(\frac{\sigma_{LJ}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{LJ}}{r_{ij}}\right)^{6}\right] \tag{2.4}$$

Here, r_{ij} is the distance between particles i and j. The parameters ϵ_{LJ} and σ_{LJ} depend on the material to be simulated. The potential has two terms, where the first term (r_{ij}^{-12}) describes the attractive, the second term (r_{ij}^{-6}) describes the repulsive van der Waals interaction.

2.1.2. Axilrod-Teller-Muto Potential

The Axilrod-Teller-Muto potential is a three-body potential that is often used in combination with the Lennard Jones 12-6 potential. It can be used, for example, to simulate of the vapor-liquid transition of noble gases with more accuracy [MS99].

The Axilrod-Teller-Muto potential is defined as follows:

$$\phi_3(i,j,k) = \nu_{ATM} * \left[\frac{1 + 3 * \cos(\theta_i) * \cos(\theta_j) * \cos(\theta_k)}{(r_{ij} * r_{ik} * r_{jk})^3}\right] \tag{2.5}$$

Where r_{ij} is the distance between particle i and j, θ_i is the angle between $\vec{ij}$ and $\vec{ik}$. The positive coefficient ν_{ATM} depends on the particles being simulated [AT43].

2.1.3. Cutoff Distance

In contrast to a direct force calculation in which we consider all possible interactions between all particles, a cutoff distance is often used for short-range potentials in MD simulations to save calculation time. Since short-range potentials quickly converge to zero as the distance between the particles increases, we can ignore the interactions from a fixed distance onwards, as they only make a minor contribution. This brings a significant performance advantage, but understandably also leads to a further approximation.

In contrast to two-body interactions, in which the cutoff criterion is simply the distance between the two particles, there are different criteria for three-body interactions. A common method for a cutoff criterion is that all three pairwise distances between the particles involved must be within the specified cutoff distance. For more cutoff criteria for three-body interactions and a visual representation, see [Hab24].

2.1.4. Newton3 Optimization

Newton's third law states that the force $F_{B \leftarrow A}$ exerted by one particle A on another particle B is also exerted in the other direction with the same magnitude. $F_{B \leftarrow A} = -F_{A \leftarrow B}$. In the context of MD simulations and high performance computing, we can exploit this law to save computation time as we only need to evaluate the force between two particles once.

This law can also be applied to three-body interactions [Mar01].

2.2. Numerical Integration

In this section, we use the Hamiltonian formalism to describe MD systems. This is another way, in addition to the equations presented in section 2.1, to represent the motion equations from classical mechanics. The following paragraphs refer to the mathematical formulation of [DHL+99, GKZ07].

The Hamilton function for a particle with mass m moving with a potential U is given as:

$$\mathcal{H}(t, q, p) = K(p) + U(q) = \frac{p^2}{2m} + U(q) \tag{2.6}$$

The Hamilton function is made up of the kinetic energy, which is described by the first summand, and the potential energy, which corresponds to the second summand.

In MD simulations with N particles we have to numerically solve the Hamiltonian equations of motion:

$$\dot{q}_i = \frac{\partial \mathcal{H}}{\partial p_i} \tag{2.7}$$

$$\dot{p}_i = -\frac{\partial \mathcal{H}}{\partial q_i} \tag{2.8}$$

Here q_i refers to the position of particle i, p_i refers to the momentum of particle i, $\dot{q}_i$ is the change in positions over time and therefore the velocities. $\dot{p}_i$ are is the change in momentum over time.

We can therefore rewrite the formulas 2.7 and 2.8 as:

$$\dot{q}_i = \frac{\partial \mathcal{H}}{\partial p_i} = \frac{p}{m} \tag{2.9}$$

$$\dot{p}_i = -\frac{\partial \mathcal{H}}{\partial q_i} = -\frac{\partial U}{\partial q} \tag{2.10}$$

As already mentioned, computers are discrete machines and we cannot directly solve these equations analytically. There are various methods for time integration like the Euler Method, Midpoint or Runge Kutta methods or the Störmer Verlet Algorithm that provide an approximation for the time integration. The Euler method, which is a first-order method, degenerates very quickly if the time steps are too large. Other methods like the Midpoint or Runge Kutta methods provide better results, by including several initial conditions [HLW06]. In this paper, we focus on the Störmer Verlet algorithm, which is discussed in more detail in Section 2.2.1.

Ultimately, however, all time integration algorithms are always an approximation to the reality. Especially in simulations with a large number of time steps, the resulting positions of particles towards the end of the simulation can deviate significantly from the exact result. However, it has been found that in MD simulations it is often more important to obtain other properties than the exact position of the molecules at the end. For example statistical properties like the total energy can be obtained quite accurately. [GKZ07]

Beside the stability and accuracy, also the computational effort plays an important role. Given that force calculations occupy up to 95 percent of the time in molecular dynamics simulations, as noted by Deuflhard et. el. [DHL+99], we should employ time integration schemes that minimize the number of these calculations. For example, we should favor the Störmer Verlet method over the Midpoint method, since the Midpoint method needs two force computations to get from t_n to $t_{(n+1)}$. In addition, there are other ways in which performance can be improved with the help of time integration algorithms. For example, higher order schemes can be used, which make it possible to use larger time steps and still maintain numerical stability. There are also time integration algorithms that integrate different components of the MD system at different time steps, which can also optimize the calculation time in suitable cases. These algorithms are also called Multiple Time-Stepping Methods, which we deal with in this report.

2.2.1. Symplectic Integrators

A special quantity, that is conserved in Hamiltonian systems, is the volume in phase space [GKZ07]. Time integration algorithms that preserve the volume in phase space are also called symplectic time integration methods. The symplectic property, is crucial for capturing the intrinsic conservation laws that govern many physical phenomena. By utilizing this property, symplectic time integrators can exhibit better long-term behavior compared to conventional integration schemes like the Euler integrator.

In this report, we focus on the Störmer-Verlet algorithm, as it is one of the most commonly used symplectic integrators. The key advantage of this method is that it is a first-order method and therefore requires only one force evaluation per iteration, but offers second-order accuracy. It can be formulated as follows [HLW06]:

$$\begin{aligned} q_{n+1/2} &= q_n + \frac{\Delta t}{2}\mathcal{H}_q(p_n, q_{n+1/2}) \\ p_{n+1} &= p_n - \frac{\Delta t}{2}(\mathcal{H}_q(p_n, q_{n+1/2}) + \mathcal{H}_q(p_{n+1}, q_{n+1/2})) \\ q_{n+1} &= q_{n+1/2} + \frac{\Delta t}{2}\mathcal{H}_q(p_{n+1}, q_{n+1/2}) \end{aligned} \tag{2.11}$$

A concrete implementation of this method can be found, for example, in [GKZ07]. In our work, we use the Störmer-Verlet algorithm as the basis for the r-RESPA algorithm. The lines 10 to line 11 in algorithm 1 correspond to the classic Störmer-Verlet algorithm.

2.2.2. Multiple Time-Stepping (MTS) Methods

In classical time integration schemes, such as Störmer Verlet, the time step to be selected is based on the highest frequency in the system. These are, for example, bond, angle and

torsion forces. However, other forces, such as non-bonded forces, can have a much lower frequency and the time steps could be chosen larger without significantly losing accuracy [GKZ07]. This is where the Multiple Time-Stepping methods come into play, with which we integrate different forces at different time scales and allows us to save calculation time.

In this section, we briefly present the theoretical background and the mathematical formulation to the Multiple Time-Stepping methods. The following paragraphs refer to the mathematical formulation of [DHL+99].

Liouville's Theorem is a fundamental result in Hamiltonian mechanics. This theorem is concerned with the conservation of phase space volume under the time evolution of a classical mechanical system. Liouville's Theorem states that as the system evolves over time according to Hamilton's equations of motion, the volume of the region in phase space occupied by a collection of such points remains constant. By formulating the time evolution as a Liouvillian, it is quickly apparent that it is easy to generate reversible and symplectic integrators. In particular, by splitting the Liouvillian into two components, we can apply the symmetric Trotter factorization, which is the base to obtain the MTS algorithms.

First, we formulate the problem as Liouvillian. The following formulation only considers a one-dimensional system for simplicity.

$$iL = \{\bullet, \mathcal{H}\} = \dot{q}\frac{\partial}{\partial q} + F_s\frac{\partial}{\partial p} + F_l\frac{\partial}{\partial p} \tag{2.12}$$

Where F_s is the Force that should be integrated with smaller time steps and F_l the force that should be integrated with larger time steps, $\dot{q}$ is the velocity and p is the momentum.
With that we can formulate the operator:

$$G(t) \equiv e^{iLt} \tag{2.13}$$

which is the propagator of motion for time t. The operator

$$G(t) \equiv e^{iL\delta t} \tag{2.14}$$

therefore represents the mapping from one time step to the next.

If we now divide iL into $iL = iL_1 + iL_2$, e.g. by splitting the two forces into L_1 and L_2, we obtain

$$G(t) \equiv e^{(iL_1+iL_2)\delta t} \tag{2.15}$$

Now we can use the reversible trotter factorization to approximate the operator as follows

$$e^{(iL_1+iL_2)\delta t} \approx e^{iL_1\frac{\delta t}{2}} iL_2 e^{\delta t} e^{iL_1\frac{\delta t}{2}} \tag{2.16}$$

In this way, we have divided the system into two components, integrating one component at the beginning and end and the second in the middle. If we now divide the middle component into several time steps, we obtain

$$e^{(iL_1+iL_2)\delta t} \approx e^{iL_1\frac{\delta t}{2}}\left[e^{iL_2\frac{\delta t}{r}}\right]^r e^{iL_1\frac{\delta t}{2}} \tag{2.17}$$

Accordingly, we can integrate the inner part into any number of smaller time steps $\delta t/r$.

In this way, we can also decompose the Hamiltonian system into several parts and use the above procedure to develop a symplectic and reversible time propagator operator.

For example, we can decompose $\mathcal{H}$ into

$$\mathcal{H} = U_1(q) + (U_2(p) + K(p)) \tag{2.18}$$

and integrate U_1 to larger time steps and U_2 to smaller time steps using the above scheme. The kinetic energy is integrated together with U_2 to the smaller time steps.

2.2.3. The r-RESPA Method

The r-RESPA (reversible Reference System Propagation Algorithm) method or impulse method is a concrete implementation of MTS. In this algorithm, different time steps are used for the different potentials to be calculated. This method is described and used for different purposes in [GHWS91, TBM91, TBM92, TBR91]. These purposes are also described in more detail in section 4. The algorithm can not only be used for two different time-scales, but can also be extended to an arbitrary number of separations. This can be easily implemented with a nested loop, where the innermost loop integrates the force that requires the smallest time-step and the outermost loop integrates the force that requires the longest time-step. An exemplary pseudo algorithm that we used as a reference can be found in [Lea01] and [GKZ07]. The specific implementation that we use for our work can be found in Section 5.

The flaw of the r-RESPA method

Although the r-RESPA algorithm can contribute to a significant reduction in computational effort, it must be used with caution. A wrong choice for a time step can lead to so-called resonances. In [LM15] in particular, the problem of resonances is illustrated using an analytically solvable example with a harmonic oscillator.

To mitigate the problem of resonances, other MTS algorithms have been developed, such as the Mollified Multiple Time-Stepping algorithm [GASSS98]. In this algorithm, the potential that is integrated with longer time steps is modified by filtering the original potential. This filtering essentially smooths out the high-frequency oscillations in the potential energy function, ensuring numerical stability while still accurately capturing the dynamics of the system. There are various methods of filtering the potential function. One method often mentioned in the literature is B-spline averaging [IRS99]. In this paper, however, we have not dealt with the Mollified Multiple Time-Stepping algorithm in detail, as we first want to examine the results with the r-RESPA algorithm.

3. Technical Background

3.1. MD Simulations on HPC Systems

HPC systems are powerful computational resources designed to perform complex calculations at high speeds, utilizing parallel processing and large-scale memory capabilities. These systems consist of numerous interconnected processors working simultaneously to solve large problems. MD simulations require the computation of interactions between thousands to millions of atoms over longer time periods. HPC systems address these challenges by distributing the computational load across multiple processors. This parallel processing capability drastically reduces the time required to perform simulations, enabling researchers to run more detailed and longer simulations than would be possible on standard computers.

Message Passing Interface: The Message Passing Interface (MPI)[1] is a standardized and portable message-passing system designed to function on a variety of parallel computing architectures. It enables multiple processes running on a distributed memory system to communicate with each other, making it fundamental for HPC applications. Since we use a *DirectSum* algorithm in our work, which is implemented as a distributed memory application for several processors, we use MPI for communication between the processors. Details on the implementation can be found in Section 5 and in particular in the references given there.

3.2. Direct Sum Calculation

In a Direct Sum calculation, all interactions between all particles are calculated without cutoff distance. To calculate all forces between all pairs of particles using the newton3 optimization, we need to calculate $\binom{N+2-1}{2}$ pairs in a system with N particles. Similarly, we need to calculate $\binom{N+3-1}{3}$ triplets. Although this is very computationally intensive, it provides very accurate results. In the context of our project, this is particularly necessary in order to be able to investigate the effects of different step-size-factors for the three-body interactions. Due to the high computational effort, it is advisable to use a distributed memory implementation in order to utilize the capacities of an HPC system. The concrete implementation of the algorithm can be found in section 5.

3.3. AutoPas

Autopas is a software library for efficient short-range particle simulations with a cutoff distance, such as MD simulations, which acts as a black box and automatically selects the best possible algorithm by autotuning during the simulation. The software is designed for

[1] `https://www.mpi-forum.org/`

the node-level of HPC systems. As of May 2024, support for three-body interactions is being integrated into Autopas.

Autopas uses so-called containers that hold the particles and enable efficient traversing of particle pairs and particle triplets for calculating the particle interactions.

In the following, we briefly introduce one of the containers in AutoPas that is relevant for this work.

Linked Cells: In the linked cells container, the simulation domain is divided into cells, with each cell containing the particles whose area it covers. This container is particularly advantageous for simulations with a cutoff distance, as starting from a fixed particle, only the particles in neighboring cells that lie in a cell covered by the cutoff radius need to be checked for their distance. In this way, a lot of calculation time can be saved for the distance checks.

Traversals: Although there are several traversals for the linked cells container that also support the newton3 optimization, we use a very simple traversal in our work, which we refer to as LCC01. As of March 2024, when we performed our measurements, only this traversal was supported for three-body interactions for the linked cells container. Figure 3.1 shows a schematic representation of the container for 2D. It should be noted that this traversal does not support the newton3 optimization.

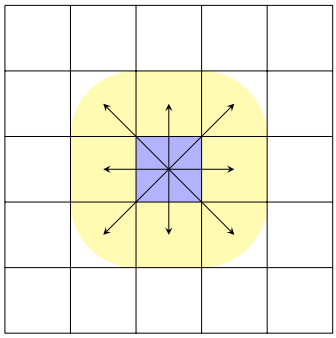


Figure 3.1.: Linked Cells C01 Traversal: In this traversal, all interactions with particles in neighboring cells that lie within the cutoff radius are calculated starting from the base cell (blue). From a base cell, only particles in neighboring cells that are covered by the cutoff radius need to be checked.

Part II.

Related Work

4. Related Work

This section presents previous work using Multiple Time-Stepping (MTS) methods, their use cases, and other approaches to reduce the computational time for three-body interactions in molecular dynamics (MD) simulations.

In general, MTS methods are employed when processes in a system occur at different frequencies. This subdivision can be based on various criteria.

Deuflhard et al. [DHL+99] distinguishes between two basic properties that can be used to divide time integration: Fast and slow degrees of freedom, which are then further subdivided. For example, different forces can be divided into fast and slow degrees of freedom, particularly bonded and non-bonded interactions. As noted by Griebel et al. [GKZ07], inter-molecular interactions such as bond, angle, and torsion forces have higher frequencies than non-bonded interactions like Lennard-Jones or Coulomb forces and can thus be integrated at different time scales.

Another way of dividing into fast and slow degrees of freedom is to classify the mass of the particles. Light particles, which move faster, require smaller time steps for integration. In contrast, heavier particles with lower velocities can be integrated with longer time steps [DHL+99, TBR91].

Apart from fast and slow degrees of freedom, Deuflhard et al. [DHL+99] distinguishes between long-range and short-range forces. Short-range forces, having a stronger influence on particle movement and a higher frequency, can be limited to a smaller radius using cutoff algorithms, thus reducing computational effort. Long-range forces, being more computationally expensive but having less impact on particle movement, can be integrated with larger time steps. Cutoff functions can be used to split the Lennard-Jones and Coulomb forces into short and long-range parts, optimizing the integration process [GKZ07].

Nakano et al. [NVK93] utilize the MTS method to minimize communication overhead in a distributed memory setup for MD simulations with two- and three-body interactions. They introduce an intermediate cutoff r_a between 0 and r_c to distinguish between interactions. All three-body interactions are considered within the range 0 to r_a, and only Lennard-Jones interactions are considered between the whole cutoff radius from 0 to r_c. The three-body interactions are computed at each time step, while interactions between r_a and r_c are only computed every few time-steps according to the chosen step-size-factor. This approach contrasts with our work, where two- and three-body interactions are separated and the three-body interactions are integrated with longer time steps.

Grubmüller et al. [GT98] compare different MTS methods to reduce calculation time for long-range forces. They evaluate a simple cutoff algorithm against distance class algorithms,

which classify particles based on distance and apply the MTS scheme accordingly. Their results indicate that all MTS methods perform relatively poorly compared to a cutoff algorithm in terms of energy drift.

Another approach to reduce the calculation time of three-body interactions is the Adaptive Resolution Scheme, AdResS. AdResS allows for a seamless transition between different force types in certain regions within a simulation, enabling for example the reduction of computation time for expensive three-body interactions in areas where they are not necessary. One work that uses AdResS for two-body and three-body forces is that of M. Mayr [May23]. In this work, the simulation domain was divided along the z-axis into a part in which only three-body interactions were calculated, a transition part and an area in which only two-body interactions were calculated. With a division of 20/10/70, where the numbers correspond to the percentage range for the three-body area, interface layer and two-body area respectively, a speedup of about 3 was measured. However, AdResS subdivides the simulation domain spatially, which means that the high-resolution calculations are confined to designated areas. This approach does not allow for three-body interactions to be computed everywhere in the simulation domain while still reducing computation time. In contrast, MTS methods can be applied globally across the simulation, integrating different interactions at varied time steps regardless of their spatial location.

Part III.

Implementation and Results

5. Implementation

This chapter briefly describes how the r-RESPA method is implemented. To analyze the results, one algorithm for calculating all possible particle pairs and particle triplets from Koanantakool and Yelick [KY14] is implemented. This will be called *DirectSum* throughout this report. On the other hand, a proxy application of an existing implementation of algorithms for calculating interactions with a cutoff called AutoPas [GSBN22], is extended by the r-RESPA method. This is referred to as *Cutoff*.

5.1. Algorithms

5.1.1. DirectSum Algorithm

For the *DirectSum* implementation, the Embedded Algorithm [KY14] was implemented as a distributed memory implementation using MPI. In contrast to the other algorithms in the paper by P. Koanantakool and K. Yelick, the embedded algorithm saves some buffer shifts by integrating these interactions in other rounds. Based the literature review in [Mar22], we concluded that this algorithm is currently a state-of-the-art algorithm for the calculation of three-body interactions on HPC clusters. In addition, this algorithm was chosen because, unlike many other algorithms, the calculation of two-body forces can be seamlessly integrated into the calculation of three-body interactions, and it makes use of the newton3 optimization.

Inclusion of Two-Body Interactions: Since, for a given number of p processors, there are significantly more buffer interactions between three buffers for the three-body interactions than buffer interactions between two buffers for the two-body interactions, we only need to calculate pairwise interactions in the first $s_{pair} = \binom{p+2-1}{2}/p$ shift rounds of the algorithm. In our implementation, we add the buffer interactions (b_1, b_1) and (b_0, b_1) to the three-body buffer interactions in lines 11 to 13 of the pseudo algorithm for the embedded algorithm in [KY14]. In line 18, we add the interaction (b_0, b_2). We do this until we have reached the required number s_{pair} of pairwise buffer actions. This scheme ensures that we calculate all necessary, but no redundant buffer interactions. It should be noted that, analogous to the special case when $p \,\%\, 3 == 0$ for three-body interactions, also for the case $p \,\%\, 2 == 0$ an additional shift round must take place.

For further details on how one of these algorithms from P. Koanantakool and K. Yelick is implemented for a distributed memory environment using MPI and a comparison of algorithms for three-body interactions see [Mar22]. Although the embedded algorithm was not implemented in the given reference, but the All Unique Triplets Algorithm, the Embedded Algorithm can be implemented analogously and a representative pseudo algorithm is given in [KY14].

5.1.2. Cutoff Algorithm

The proxy application MD-flexible[1], which is supplied with AutoPas as an example application, was extended in the main simulation loop by the r-RESPA algorithm as in the pseudo algorithm 1.

MD-flexible uses the Störmer-Verlet algorithm from [GKZ07] for the time integration. Since this algorithm works with the forces from time step t^{i-1} and t^i to update the velocities, the data structure for the particles has two force buffers in which the calculated force from time step t^{i-1} and t^i is stored for each particle. In order to be able to integrate the three-body forces separately, we have added another temporary buffer for the forces to the particle data structure. This allows us to use the two-body force, which is calculated in the pseudo algorithm in line 10 and is required in the next iteration in line 11 to calculate the velocities, store them temporarily and use the main force buffer for the three-body force.

5.2. r-RESPA Implementation

The r-RESPA method was implemented in the same way for both applications. In contrast to the pseudo algorithm given in the literature [GKZ07], which is implemented using a double nested loop, we only use a simple loop in which each iteration corresponds to a time step. In each time step, the two-body force is integrated and every step-size-factor time steps, the three-body force is integrated. The implementations ensure that the total number of iterations is a multiple of the step-size-factor. Algorithm 1 gives a rough overview of how r-RESPA time integration is implemented in our case.

Algorithm 1: r-RESPA implementation for two-body and three-body interactions. Note: Only an iteration that is divisible by the step-size-factor can be seen as a full iteration.

Input: Initial particle positions x^0 and velocities v^0
Output: Particle positions and velocities updated in place.

1 *calculate* $F_{2b}(x^0)$
2 *calculate* $F_{3b}(x^0)$
3 $s \leftarrow stepSizeFactor$
4 **for** $i \leftarrow 0$ **to** *numIterations - 1* **do**
5 $\quad IsRespaIteration \leftarrow (i \,\%\, s) == 0$
6 $\quad nextIterationIsRespa \leftarrow ((i+1) \,\%\, s) == 0$
7 $\quad$ **if** $IsRespaIteration$ **then**
8 $\quad\quad v^i \leftarrow v^i + \frac{\delta t\, s}{2m} F_{3b}(x^i)$
9 $\quad x^{i+1} \leftarrow x^i + \delta t\, v^i + \frac{\delta t^2}{2m} F_{2b}(x^i)$
10 $\quad$ *calculate* $F_{2b}(x^{i+1})$
11 $\quad v^{i+1} \leftarrow v^i + \frac{\delta t}{2m}(F_{2b}(x^i) + F_{2b}(x^{i+1}))$
12 $\quad$ **if** $nextIterationIsRespa$ **then**
13 $\quad\quad$ *calculate* $F_{3b}(x^{i+1})$
14 $\quad\quad v^{i+1} \leftarrow v^{i+1} + \frac{\delta t\, s}{2m} F_{3b}(x^{i+1})$

[1] https://github.com/AutoPas/AutoPas/tree/master/examples/md-flexible

6. Results

In this section, we investigate the r-RESPA method for simulations with two- and three-body interactions, where the two-body interactions are integrated at every time step and the three-body interactions only every few time steps according to the chosen step-size-factor. We investigate different step-size-factors in comparison to the classical Störmer-Verlet algorithm, where both the two-body and three-body forces are integrated at each time step.

We use the Lennard-Jones 12-6 (LJ) potential as the two-body potential and the Axilrod-Teller-Muto (ATM) potential as the three-body potential, since these are two frequently used potentials in MD simulations.

On the one hand, we carry out measurements with our implementation for direct force calculation which consider interactions between all pairs or triples (*DirectSum*). This is especially used to investigate the effect on the resulting total energy of the MD-system.

On the other hand, we perform measurements with our cutoff implementation, called *Cutoff* in the following, to analyze how different step-size-factors for the three-body force behave when an approximation given by the cutoff is already present.

It should be noted that certain properties can be better analyzed with the *DirectSum* method, others better with *Cutoff* implementation. For example, the influences on the total energy of different step-size-factors can be better analyzed with *DirectSum*. Structural properties such as the radial distribution function or the pressure can in our case be better analyzed with the *Cutoff* implementation, since the *DirectSum* measurements are missing some aspects that would be needed to measure these quantities. Details can be found in the respective sections.

We use three different scenarios for our measurements:

- **Pseudo scenario:** This scenario has no relation to a real physical simulation, but is used to investigate the effect of different ν_{ATM} parameters for the three-body interactions in combination with different step-size-factors. Since in many simulations the parameters are made dimensionless and the Lennard Jones parameters ϵ_{LJ} and σ_{LJ} are set to 1 and all other parameters, such as time step δt or the ATM parameter ν_{ATM}, are adjusted accordingly, this scenario provides a good overview of the possible effects of the r-RESPA method with different materials.
- **Argon scenario:** We have used the values from [AS22] in order to carry out our measurements with parameters for a noble gas.
- **Aluminium scenario:** In contrast to the noble gas, we used the values for aluminum from [BC21] to carry out the measurements for a solid.

The following bullet points briefly explain the metrics that are measured. We focus primarily on the accuracy of the simulations with different step-size-factors.

- **Relative Variation in True Energy (RVITE):** The RVITE is calculated as

$$RVITE = \frac{1}{KJ} \sum_{i=1}^{J} |e(i) - \overline{e}| \tag{6.1}$$

  where $\overline{e}$ is the average total energy, $e(i)$ is the total energy of iteration i. J is the number of time steps. K is the average kinetic energy for the simulation. According to [IRS99], the RVITE is a good indicator for the accuracy of symplectic integrators. The smaller this value is, the smaller the RVITE and the more accurate the time integration is accordingly. It should be noted that we only use this metric for our measurements with the *DirectSum* algorithm, as our results have shown that the cutoff implementation, being an approximation, already has a strong influence on the total energy in itself, which is also described in [DO20, FPR$^+$96]. This makes the effects of different step-size-factors of the r-RESPA algorithm difficult to measure for an algorithm that uses a cutoff distance.

- **Energy Deviation:** Here we compare the values for the total energy with different step-size-factors relative to the total energy with Störmer Verlet. To do this, we measure the maximum absolute deviation in the total energy $maxAbsDev$ of all step-size-factors across all iterations relative to Störmer Verlet. The range $2 * maxAbsDev$ is then divided into $numBins$ bins. For each step-size-factor and each iteration, which represents a complete iteration for each step-size-factor, the deviations are distributed to bins, which in the end gives an overview of the distribution of the deviation over the entire simulation. This measurement was also only carried out with the *DirectSum* algorithm.

- **Two-Body and Three-Body Comparison:** In order to better illustrate the results of the measurements with different step-size-factors for the three-body interactions compared to a pure two-body simulation, the differences between a pure two-body simulation and the results of the three-body simulation with r-RESPA are also compared. This comparison was also only performed with the *DirectSum* algorithm.

- **Radial distribution function (RDF):** The RDF can be used to examine and compare structural properties of the MD system. Starting from a fixed particle, surrounding particles are categorized into $numBins$ distance classes that all together cover a spherical area with radius R around an arbitrary but fixed particle. The j^{th} distance class, with $j \in \{0, numBins - 1\}$, is represented by the spherical shell between $r = j\delta$ and $r + \delta$ with $\delta = \frac{R}{numBins}$. Each distance class stores the number of particles that are contained within the spherical shell around the fixed particle. This procedure is carried out for each relevant particle in the MD system. The number in the distance classes is normalized at the end over the number of particles, the volume of the simulation domain and over time. Accordingly, the RDF gives us the probability of finding a neighboring particle starting from a fixed particle at a certain distance. Our implementation follows the approach of guard areas, which is described in section

3 and 3.3 in [LS18]. The whole formula for one time step is given by

$$RDF(r) = \sum_{i=1}^{N} \frac{\psi_i(r)/N}{(N-1)(\frac{dV_r}{V})} \tag{6.2}$$

where N is the total number of particles in the relevant area, $\psi_i(r)$ is the number of particles which are located in a distance from r to $r + \delta$ from particle i. V is the volume of interest for which we perform the measurement. dV_r is the volume of the spherical shell between the radii r and $r + \delta$. It should also be noted here that we only use this metric for the cutoff implementation. Further details on why this metric is not used for the *DirectSum* algorithm can be found in the corresponding sections.

- **Pressure:** The pressure is also an indicator of the structural properties of an MD system. Our results were calculated using the method described in section 3 of [KNA14]. This property is also only measured for the cutoff implementation, as the *DirectSum* implementation has no simulation boundaries. Further details can be found in the corresponding sections.
- **Performance:** To measure the performance we calculate the speedup based on the total simulation time compared to a step-size-factor of 1 which corresponds to the Störmer-Verlet algorithm.

In all our simulations, the parameters are made dimensionless as described in section 3.7.3 of [GKZ07]. The LJ parameters ϵ_{LJ}, σ_{LJ} and the mass m are normalized to 1. All other parameters, such as the ATM parameter ν_{ATM}, the time step δt or the temperature T are scaled accordingly. In all our measurements, the MD systems were first equilibrated before measuring the quantities.

In most of our experiments, we use the step-size-factors 1, 2, 3, 4, 6 and 12. A step-size-factor of one corresponds to the classic Störmer-Verlet algorithm, in which both the two-body and three-body forces are calculated at each time step. The reason for choosing these step-size-factors is that for each step-size-factor every 12th iteration represents a completed time step. This allows us to compare values of aligned iterations. For the calculation of the RVITE values, all values from 1 to 12 were used for the step-size-factor, because the RVITE is not calculated dependent on values from other simulations.

All our experiments were carried out on the HSUper supercomputer of the Helmut Schmidt University in Hamburg [1]. This HPC system has 571 compute nodes, each with 2 Intel(R) Xeon (R) Platinum 8360Y processors with 36 cores each. Thus 72 cores per node.

6.1. Two-Body and Three-Body Force Impact

First, we examine the differences between the two-body force and the three-body force using two scenarios. We have simulated argon and aluminum using our cutoff implementation. The values used are the same as those in section 6.3, with the exception that we only calculate 1000 iterations.

[1] `https://www.hsu-hh.de/hpc/en/hsuper/`

We then calculated and compared the average magnitude of the force vectors for the two body force and the three body force in each iteration. This comparison can be seen in Figures 6.1 and 6.2.

As we can see, the effect of the twobody force is significantly higher compared to the threebody force, which is a justification for us to test and evaluate the r-RESPA algorithm for three-body interactions.

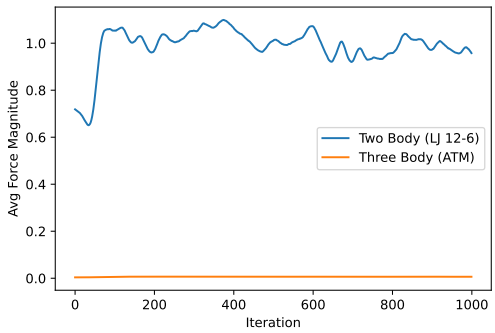


Figure 6.1.: Argon Simulation

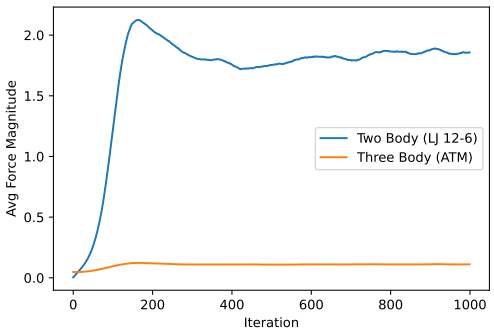


Figure 6.2.: Aluminium Simulation

In a further experiment, we investigated the two-body and three-body forces around the equilibrium for three particles. We used the parameters for aluminum from section 6.3. At a distance of 2.8, we obtained a force magnitude of 0.1127 with the *LJ* potential and a force magnitude of -0.0502 for a distance of 2.9. On the other hand, we obtained a force magnitude of -0.0117 for the *ATM* potential at a distance of 2.8 and a force magnitude of -0.0079 at 2.9. From this we conclude that around the equilibrium the attraction and repulsion is largely driven by the *LJ* potential, which we see as a further reason to test and evaluate the r-RESPA algorithm for three-body interactions.

6.2. Direct Measurements

In this section we investigate the effect of different step-size-factors using the *DirectSum* implementation. This allows a detailed analysis of the effect of different step-size-factors on quantities, such as the total energy.

It should also be noted that we have omitted boundary conditions in all *DirectSum* experiments. Our experiments have shown that the implementation of reflective boundary conditions leads to influences on the total energy that would distort our results. Periodic boundary conditions were also omitted, as this is difficult to implement for a *DirectSum* implementation in conjunction with three-body forces [Mar01]. A thermostat was also omitted, as our implementation of the thermostat directly scales the velocities of the particles and thus has a direct influence on the kinetic energy.

In all experiments with the *DirectSum* implementation, 5 compute nodes with 72 cores each, i.e. a total of 360 cores, were used.

6.2.1. Influence of Different ATM Parameters and Step-Size-Factors

In this experiment we investigate the effect of different values for the ATM parameter ν_{ATM} from 0.05 to 1.0 in combination with different step-size-factors from 1 to 12 on the Relative Variation in True Energy. The parameters for the ATM potential were chosen to cover a wide range from gaseous to solid materials. The following simulation parameters were used:

Property / Parameter	Value
Domain Size	10x10x10
Number of Particles	675
Number of Iterations	24000
δt	0.001
m	1
ϵ_{LJ}	1
σ_{LJ}	1

Table 6.1.: Parameters for the pseudo scenario.

Note: for the step-size-factors of 7, 9 and 11 we have used 24003, 24003, 24002 iterations respectively, otherwise the number of iterations would not be a multiple of the step-size-factor.

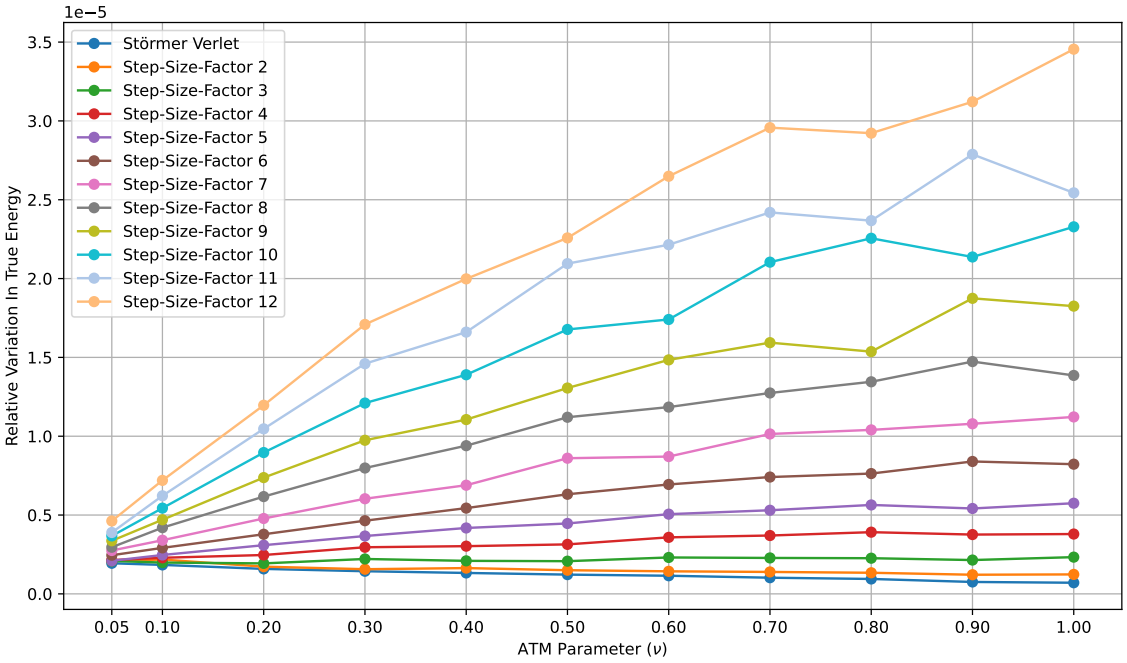


Figure 6.3.: Comparison of the Relative Variation In True Energy for different step-size-factors and different values for the ν_{ATM} parameter. Each line represents one step-size-factor. From bottom to top, this corresponds to a step-size-factor of 1 (Störmer Verlet) up to 12.

As we can see in Figure 6.3, we have a decreasing trend for a step-size-factor of 1 and 2, with an increasing ν_{ATM} parameter. According to our observations, the inclusion of three-body interactions provides a stabilization of the potential and kinetic energy, whereas a pure two-body simulation showed much higher fluctuations in the energies. For this reason, we assume that for these step-size-factors, increasing the ν_{ATM} parameter has a stronger influence on the overall simulation. For a step-size-factor of 3, we see a similar value for all ν_{ATM} parameters. For the higher step-size-factors from 4 to 12, we then see an increase in the RVITE value as the ν_{ATM} parameter increases and thus a loss of accuracy of the r-RESPA algorithm.

We can therefore conclude from this experiment that step-size-factors up to around 4 achieve reasonably good results.

6.2.2. Argon Parameters

In this scenario we investigate how different step-size-factors affect a simulation performed with parameters for an argon simulation.

The following simulation parameters were used:

Property / Parameter	Dimensionless Normalized Value	Actual Value
Domain Size	20x20x20	20x20x20
Number of Particles	4995	4995
Number of Iterations	24000	24000
δt	0.000474628	1 fs
m	1	39.948u
ϵ_{LJ}	1	$1.653 \cdot 10^{-21} J$
σ_{LJ}	1	3.403 $\mathring{A}$
ν_{ATM}	0.073	$7.35 \cdot 10^{-18} J\mathring{A}^9$

Table 6.2.: Parameters for the argon scenario.

Also here for the step-size-factors of 7, 9 and 11 we have used 24003, 24003, 24002 iterations respectively.

Relative Variation In True Energy and Speedup

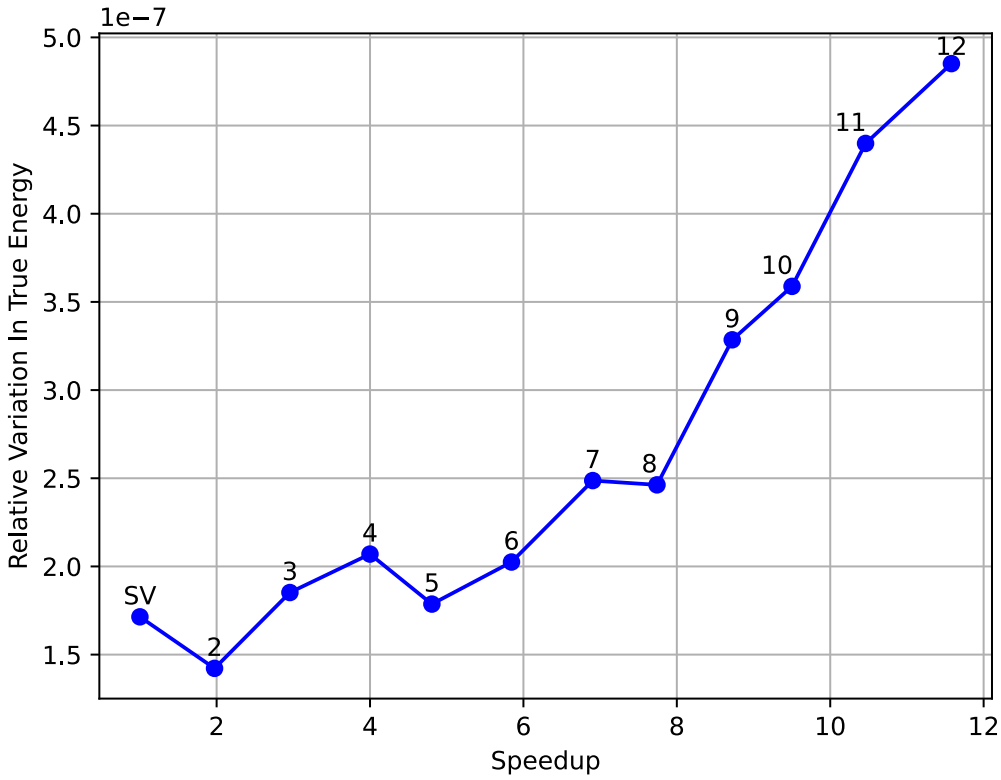


Figure 6.4.: Relative Variation In True Energy on the y-axis for different step-size-factors shown as labels at the respective measurements points. The **SV** label corresponds to Störmer-Verlet. The speedup against Störmer-Verlet is plotted on the x-axis. This Figure refers to the values for our argon simulation.

As we can see in Figure 6.4, calculation time is reduced by almost half with a step-size-factor of 2 compared to Störmer Verlet. This trend continues for the other step-size-factors. The result was to be expected, since the three-body interactions take up most of the calculation time compared to the two-body interactions [May23].

In general, we see an increasing trend with increasing step-size-factor, but we also see unexpected values for example at step-size-factor 2, 5, 6 and 7. Similar to the previous experiment, where we observed a decrease in the RVITE value at low parameters for ATM potential, we assume that the unexpected values are due to the more fluctuating dominant two-body force.

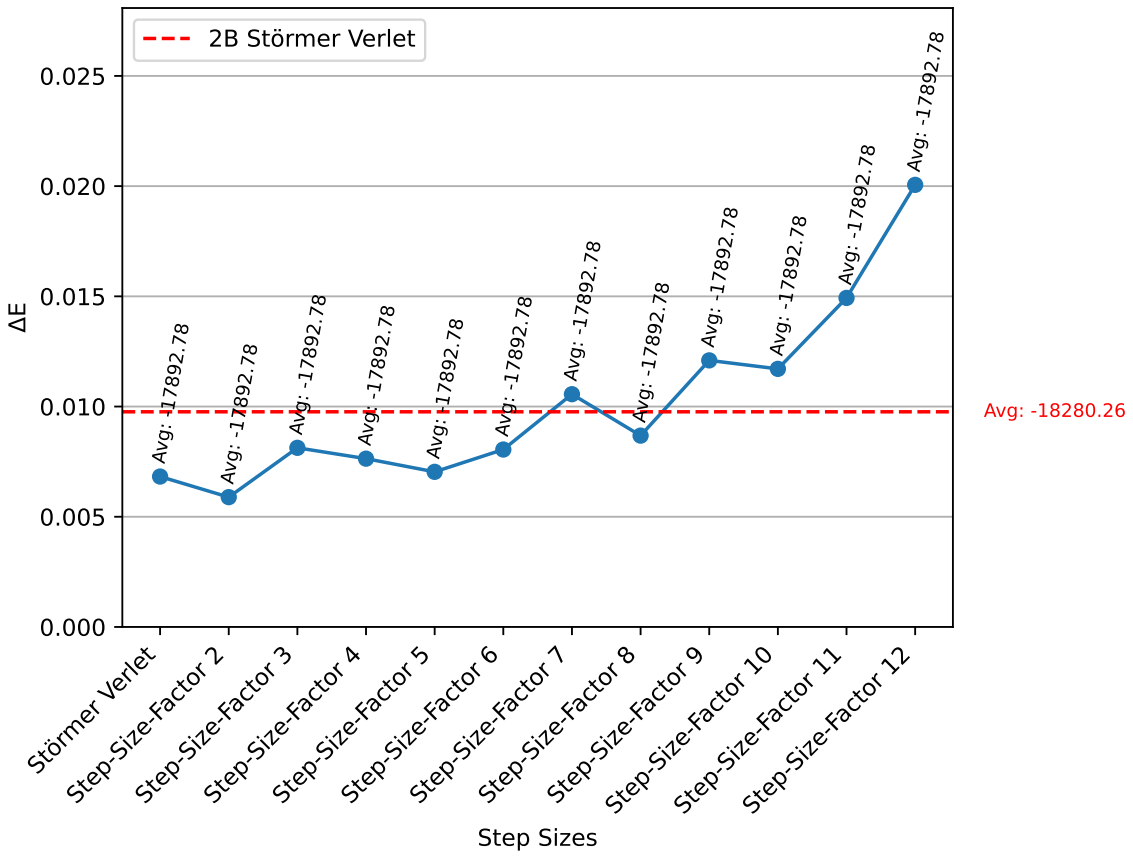


Figure 6.5.: Comparison of the minimum and maximum values of the total energy for different step-size-factors of our argon simulation compared to a pure two-body simulation. ΔE is the difference between the minimum and maximum measured total energy for each simulation. The value above each bar is the average value in the total energy.

In this experiment, we used the same parameters for σ_{LJ} and ϵ_{LJ} for both the pure LJ simulation and the $LJ{+}ATM$ simulation. In our literature research, we concluded that for pure LJ simulations for argon approximately the same parameters for ϵ_{LJ} and σ_{LJ} are used as for the $LJ{+}ATM$ parameters, which we obtained from [AS22]. For example, in the work by White [Whi99], 3.405 $\AA$ is used for σ_{LJ} and 119.8 K for ϵ_{LJ}/k_B for a pure LJ simulation. Therefore, we concluded that we can compare the values for a pure LJ and a $LJ{+}ATM$ simulation.

As we can see in Figure 6.5, the average total energy is identical for all step-size-factors up to the second decimal place. We also see that the average total energy for the two-body simulation differs considerably from that with three-body interactions. Again, we conclude that the inclusion of three-body interactions definitely has an influence on the simulation, but the oscillations in the total energy up to a step-size-factor of 6 do not exceed those

of a pure two-body simulation. On the one hand this confirms our observation that the three-body interactions stabilize the simulation, on the other hand we also see that up to the step-size-factor of 6 we get quite good results regarding the values for ΔE.

Energy Deviation

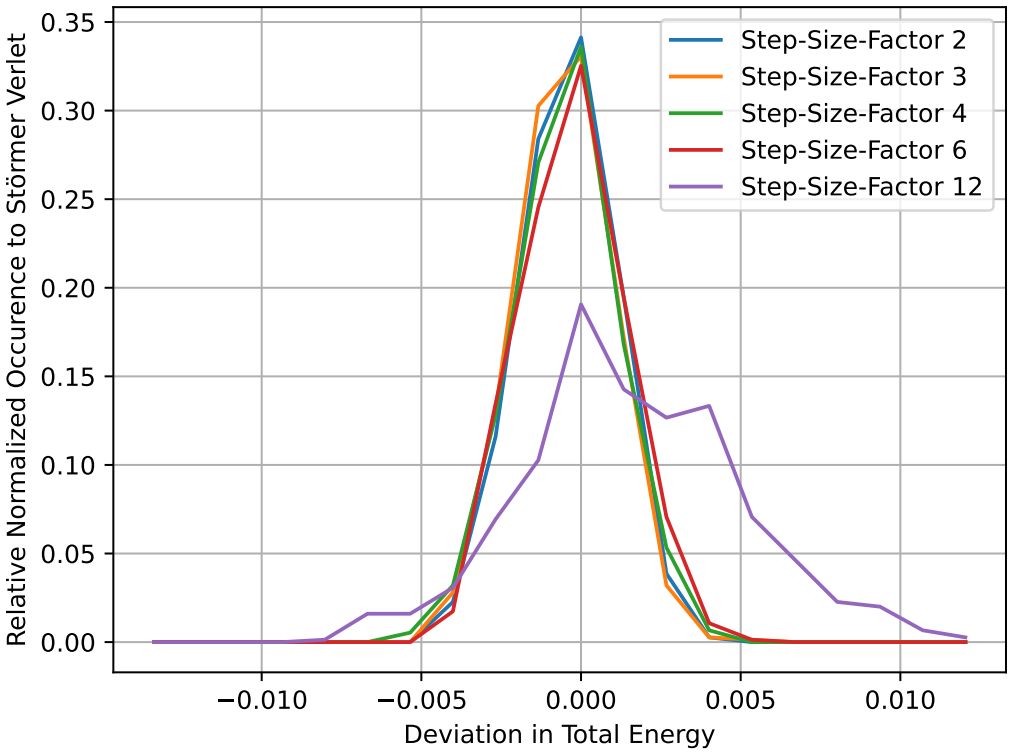


Figure 6.6.: Values of total energy relative to Störmer Verlet. Here, the range between the minimum and maximum total energy across all step-size-factors was divided into 50 bins and the values of the respective simulations of each iteration were distributed across these bins. The line for Störmer Verlet was intentionally not shown here. It would be a vertical line at 0.0 with a height of 1.

Figure 6.6 compares the resulting total energy from our argon simulation with different step-size-factors relative to Störmer Verlet. We divided the range between the maximum deviation of the total energy in positive and negative direction into bins. For each step-size-factor, the values of the total energy over all iterations after reaching the equilibrium were then divided into bins.

As can be seen, for all step-size-factors except 12 the peak is at around zero deviation in energy. However, it can also be seen that there is a clear shift in energy at a step-size-factor of 12.

6.2.3. Aluminium Parameters

In this scenario we investigate how different step-size-factors affect a simulation performed with parameters for an aluminium simulation. The following simulation parameters were used:

Property / Parameter	Dimensionless Normalized Value	Actual Value
Domain Size	20x20x20	20x20x20
Number of Particles	4995	4995
Number of Iterations	24000	24000
δt	0.00304	1 fs
m	1	26.982u
ϵ_{LJ}	1	$2.688 \cdot 10^{-20} J$
σ_{LJ}	1	2.5487 $\mathring{A}$
ν_{ATM}	0.3095	$3.776 \cdot 10^{-17} J\mathring{A}^9$

Table 6.3.: Parameters for the aluminium scenario.

Relative Variation In True Energy and Time

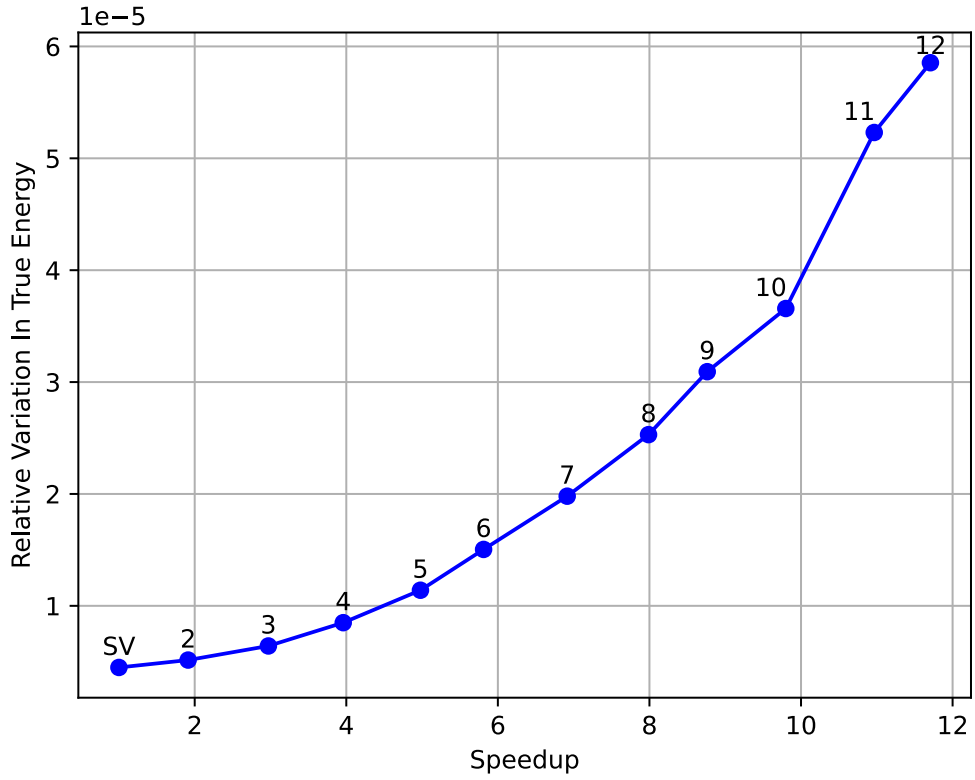


Figure 6.7.: Relative Variation In True Energy on the y-axis for different step-size-factors shown as labels at the respective measurements points. The **SV** label corresponds to Störmer-Verlet. The speedup against Störmer-Verlet is plotted on the x-axis. This Figure refers to the values for our aluminium simulation.

Analogous to the argon experiment, we see the same behavior for the runtime for the aluminum simulation in Figure 6.7. In contrast to the argon simulation, however, we see the more expected values for the RVITE with these parameters. As already mentioned, we assume that this is related to the stabilization of the simulation due to the stronger three-body forces.

TwoBody vs. ThreeBody Energy Impact

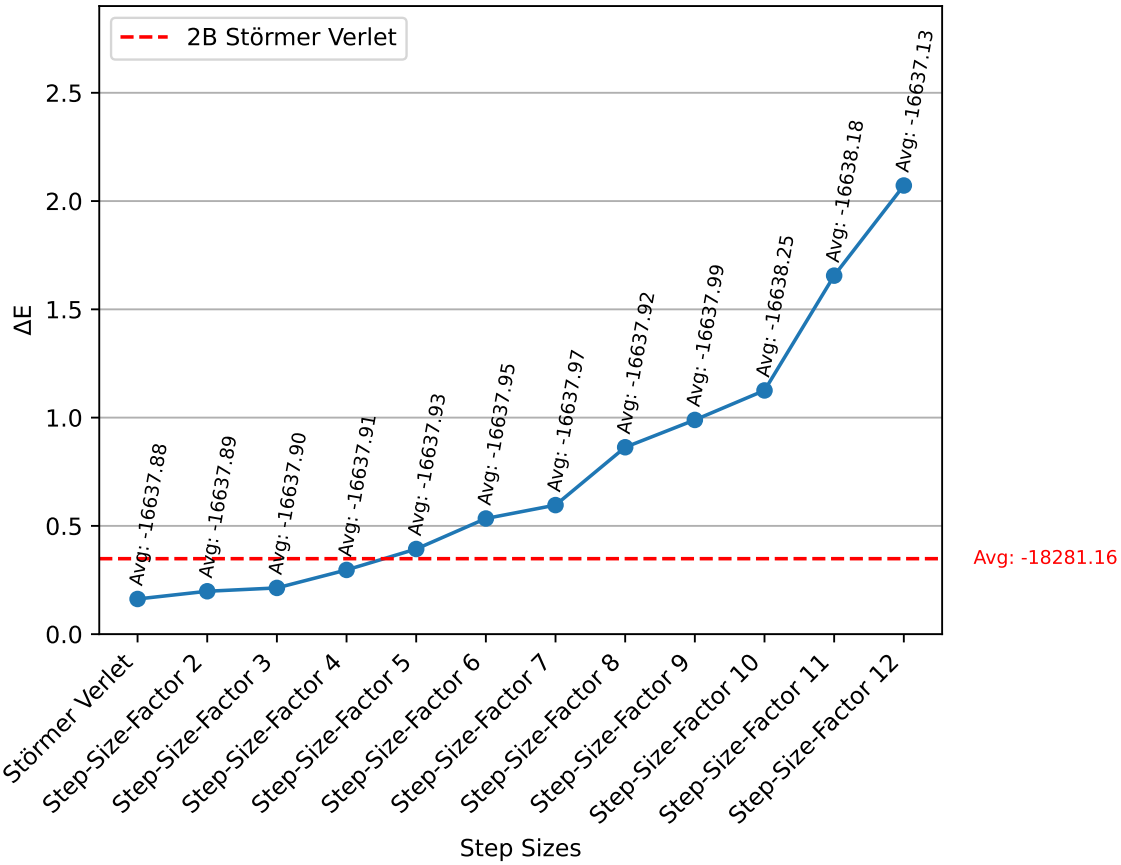


Figure 6.8.: Comparison of the minimum and maximum values of the total energy for different step-size-factors of our aluminium simulation compared to a pure two-body simulation. ΔE is the difference between the minimum and maximum measured total energy for each simulation. The value above each bar is the average value in the total energy.

Very similar to the RVITE values, we see an increasing trend in ΔE with increasing step-size-factor in Figure 6.8. Here, too, the average total energy is very similar across all step-size-factors and the value for the pure two-body simulation deviates considerably from the simulations with three-body interactions. Up to a step-size-factor of 4, we get quite acceptable results here.

Compared to the measurements for the argon parameters, however, we see stronger oscillations with increasing step-size-factor. This is due to the fact that the ATM parameter is higher here.

Energy Deviation

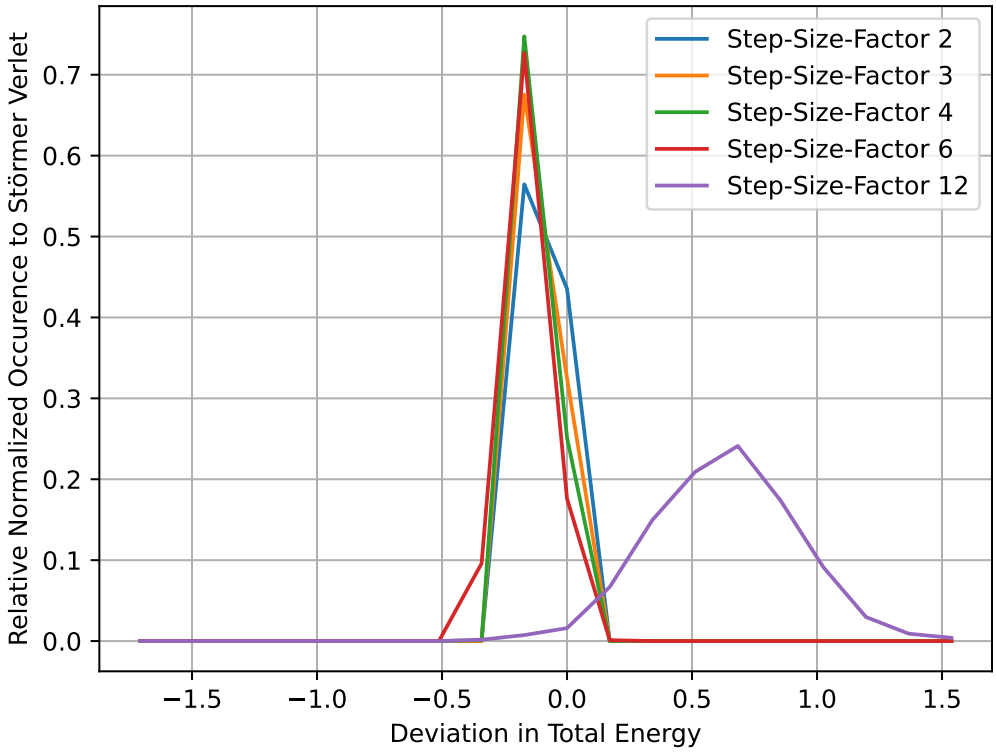


Figure 6.9.: Values of total energy relative to Störmer Verlet. Here, the range between the minimum and maximum total energy across all step-size-factors was divided into 50 bins and the values of the respective simulations of each iteration were distributed across these bins. The line for Störmer Verlet was intentionally not shown here. It would be a vertical line at 0.0 with a height of 1.

In this experiment, a clearer offset of the energy values can be seen than in the simulation with the argon parameters in Figure 6.9. Especially for step-size-factors 6 and 12, the total energy measured by Störmer Verlet shifts. We also attribute this to the higher ATM parameter, which causes the three-body interactions to have a stronger influence.

6.3. Cutoff Measurements

In this section we examine the effect of different step-size-factors using the *Cutoff* implementation. Here, in contrast to the energy values we examined with *DirectSum*, we examine the structural properties of the MD systems for argon and aluminum. The reason for this is that we can more easily implement periodic boundaries here. In addition, we use a thermostat to perform a physically correct simulation. We investigate the structural properties on the one

side using the radial distribution function and on the other side using the pressure of the MD system.

The guard area approach, as mentioned in the introduction of this section, was used to calculate the radial distribution function. Here, it is not the entire part of the particles within the simulation domain that is used to classify the distance to neighboring particles, but only an inner part. The space between the inner part and the boundaries of the MD system is called guard area. Particles in this area are considered as neighbors, but they do not serve as reference particles. In our measurements, a guard area of 3 was used. Accordingly, the radial distribution function refers to the inner part of the 20x20x20 simulation domain with a size of 14x14x14.

In all experiments with the *Cutoff* implementation, one compute node with 72 cores was used.

For both the argon and the aluminium simulation we use a cutoff distance of 2.5 for all our measurements.

Runtime: As we can see in Figure 6.10 and 6.11, we do not achieve as much speedup here as for the measurements with the *DirectSum* algorithm. We assume that the *Cutoff* implementation calculates significantly fewer triplets in relation to the pairwise interactions than the *DirectSum* implementation. This is partly due to the criterion we use in the cutoff implementation to form triplets. All three pairwise distances must lie within the cutoff distance for a triplet to be classified as valid. For the measurement with the aluminum parameters, we see that we achieve a higher speedup than with the argon simulation. This is due to the fact that in the aluminum simulation the particles are denser in the equilibrium and accordingly more triplets are formed for calculation within the cutoff distance.

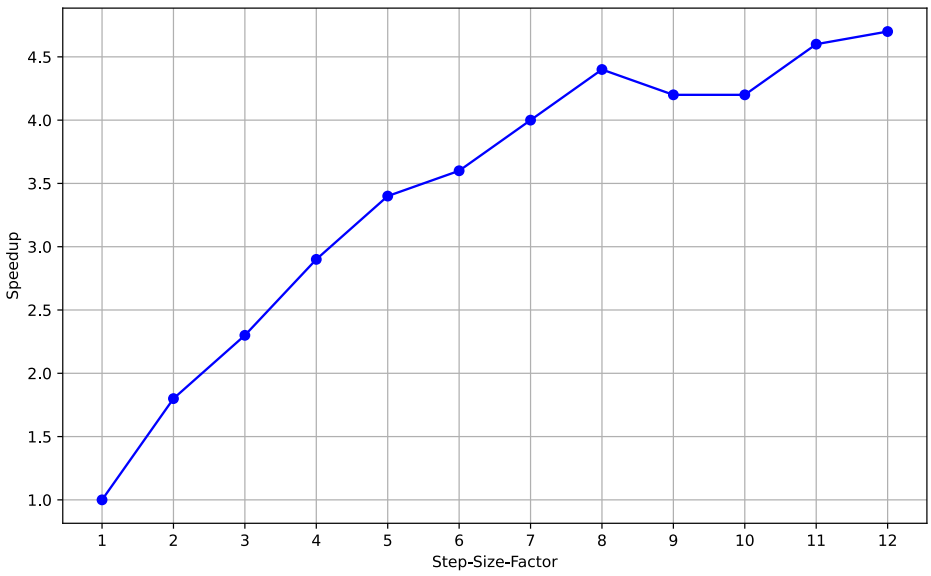


Figure 6.10.: Speedup for the argon simulation with different step-size-factors.

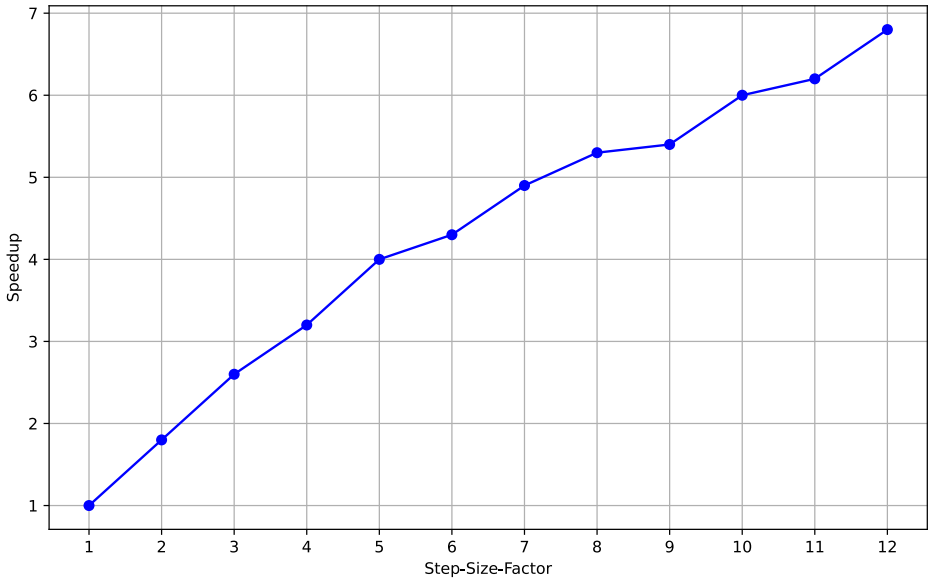


Figure 6.11.: Speedup for the aluminium simulation with different step-size-factors.

6.3.1. Argon Parameters

We use the same parameters for the argon simulation as for the *DirectSum* measurements. In addition, however, we use a thermostat that equilibrates the MD system at 300 K and, as already mentioned, periodic boundary conditions. Furthermore, we could analyze the simulation over a longer period of time due to the shorter calculation time. 120000 time steps were performed.

Radial Distribution Function

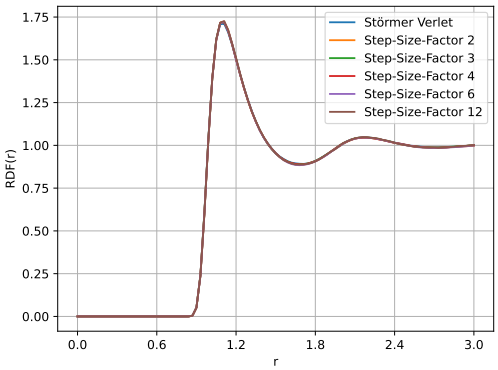


Figure 6.12.: RDF for the argon simulation.

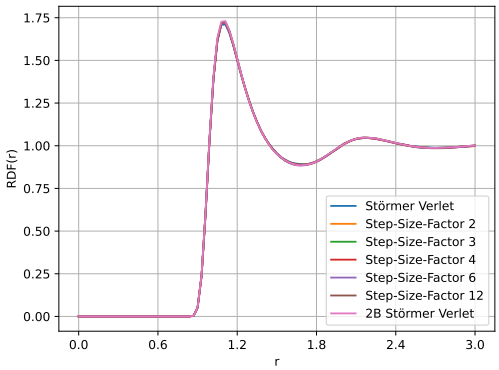


Figure 6.13.: RDF for the argon simulation including a pure two-body simulation.

For the argon simulation, we see no significant deviations in the absolute values for the

RDF in Figure 6.12. We attribute this to the fact that the three-body potential only has a relatively small influence here. This is also confirmed in Figure 6.13, which shows the values for a pure two-body simulation in addition to the simulation with two-body and three-body potential. On closer inspection, there is only a small difference in the first peak, which we will look at in more detail in the next paragraph.

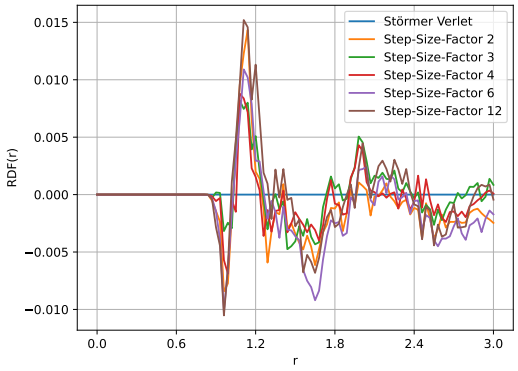


Figure 6.14.: Values of the RDF of argon relative to Störmer Verlet.

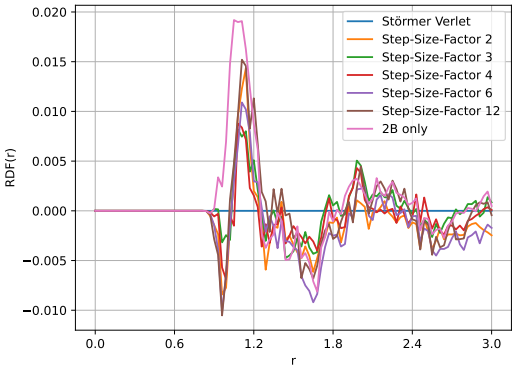


Figure 6.15.: Values of the RDF of argon relative to Störmer Verlet including a pure two-body simulation.

To examine the effects in more detail, we look at the deviations relative to Störmer Verlet in Figure 6.14 and 6.15. In Figure 6.15 in particular, we can see that there is a small shift between two-body and three-body in the first peak. Since this peak is at approximately the same level for all step-size-factors with three-body interactions, we conclude that the structural change caused by the three-body interactions remains approximately the same for all step-size-factors.

Pressure

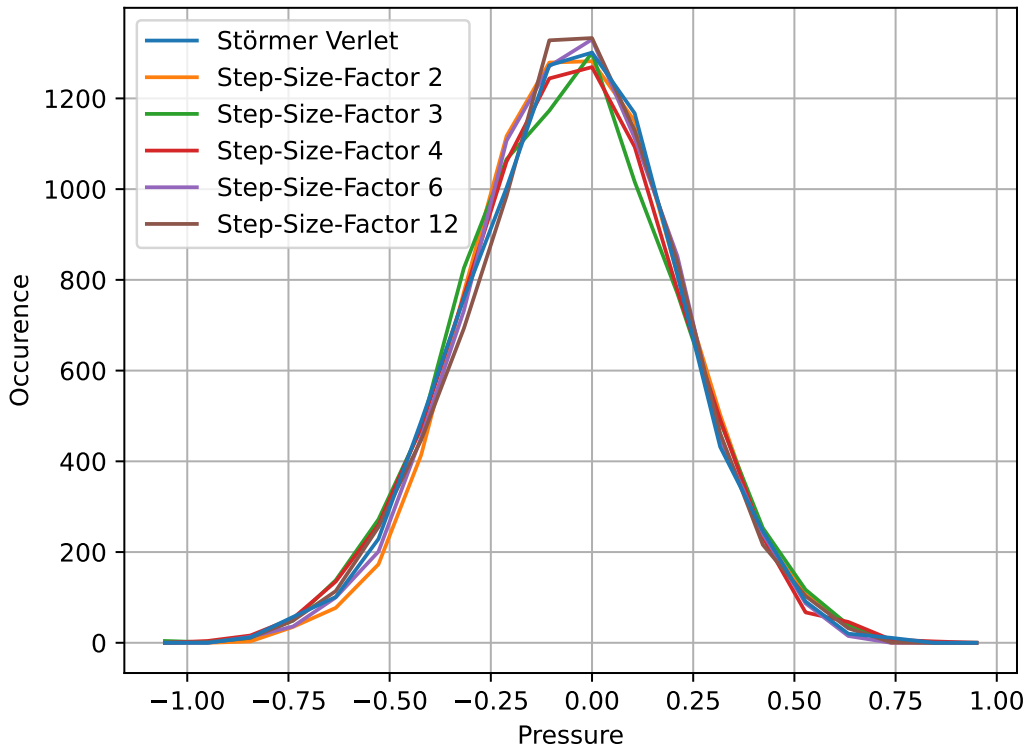


Figure 6.16.: Comparison of the absolute values for the pressure of the MD system with different step-size-factors. The values of all iterations for the respective step-size-factors were distributed over 20 bins. The pressure values were calculated here over 96000 iterations after reaching the equilibrium.

In this section, we investigate the pressure within the MD system. To do so, we use the procedure mentioned in the introduction of this section. Since the pressure in our measurements oscillates around a certain value after reaching the equilibrium, we again use the approach of classifying the values for each step-size-factor and each iteration in bins. This makes it easier for us to visually determine whether there are deviations or shifts in the pressure.

For the argon parameters, we cannot detect any significant deviations with all step-size-factors in Figure 6.16. We attribute this again to the fact that the three-body potential only has a minor influence on the simulation here.

6.3.2. Aluminium Parameters

We also use the same parameters for the aluminum simulation as for the *DirectSum* implementation. In addition to the *DirectSum* implementation, periodic boundaries are also

used and the system is equilibrated at 300K with a thermostat. 120000 time steps were performed.

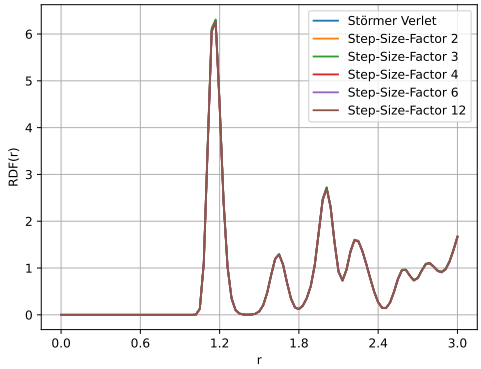


Figure 6.17.: RDF for the aluminium simulation.

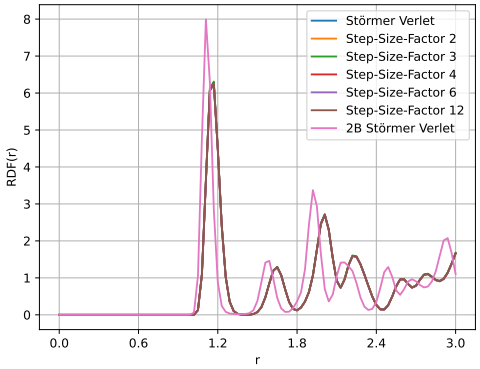


Figure 6.18.: RDF for the aluminium simulation including a pure two-body simulation.

For the aluminum simulation, we also see no significant differences with the two- and three-body simulation with different step-size-factors. However, when comparing these simulations with a pure two-body simulation, we see that there are clear differences in the radial distribution function. We conclude that the inclusion of three-body interactions has a noticeable influence on the structure of the system. However, the deviations using different step-size-factors for the integration of the three-body interactions are relatively small.

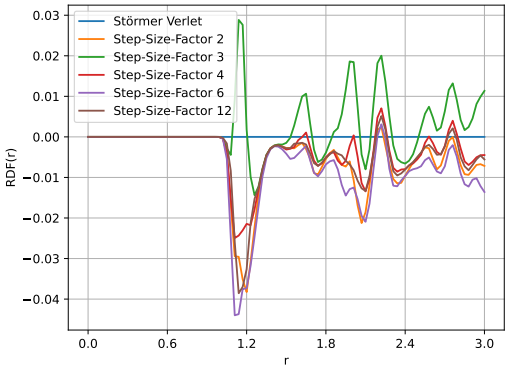


Figure 6.19.: Values of the RDF for the aluminium simulation relative to Störmer Verlet.

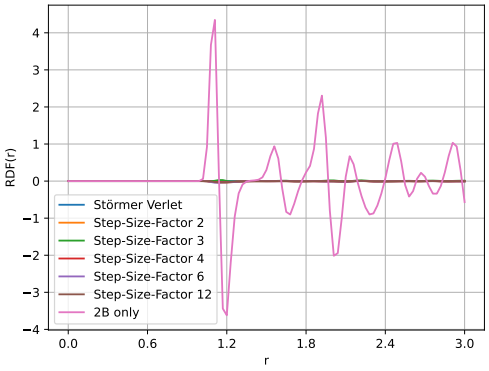


Figure 6.20.: Values of the RDF of the aluminium simulation relative to Störmer Verlet including a pure two-body simulation.

If we analyze the relative differences, we see a deviation for a step-size-factor of 3 compared to other step-size-factors in this experiment. However, we have not investigated this further

in this work. Again, the relative comparison in which the pure two-body simulation is included is interesting. Because here we see again that the inclusion of the three-body interactions has a clear influence on the structure of the system, but the relative deviation with different step-size-factors is comparatively small.

Pressure

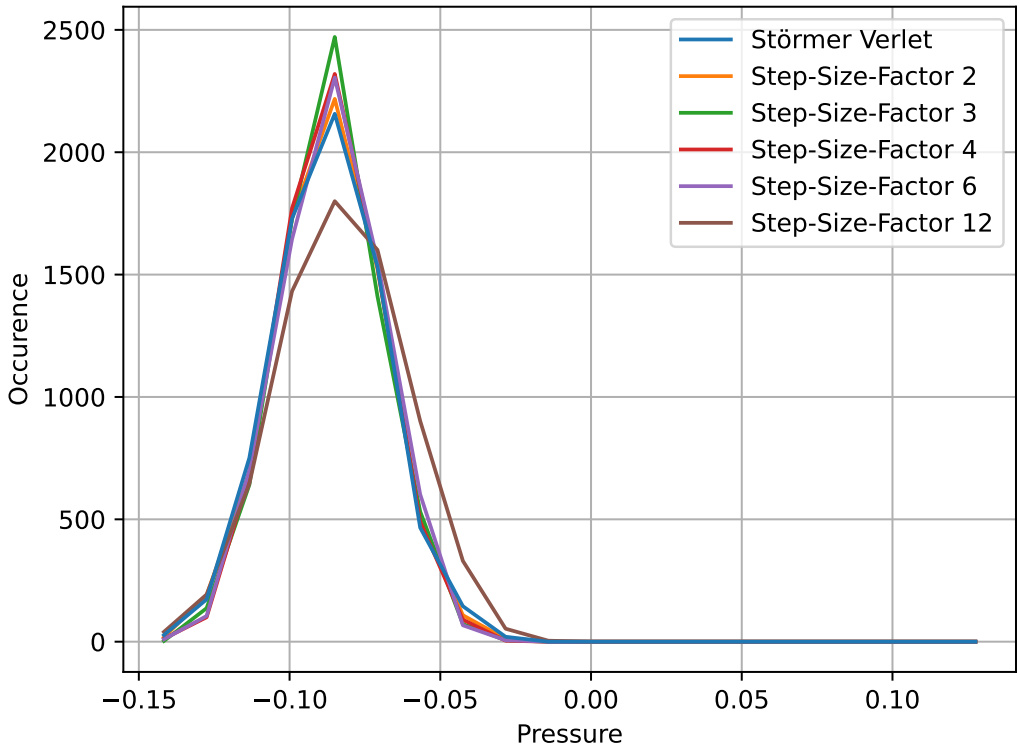


Figure 6.21.: Comparison of the absolute values for the pressure of the MD system with different step-size-factors. The values of all iterations for the respective step-size-factors were distributed over 20 bins. The pressure values were calculated here over 84000 iterations after reaching the equilibrium.

In Figure 6.21 we have examined the pressure for the aluminum simulation analogous to the argon parameters. For step-size-factors 2 to 6, we see no significant shift in the values compared to Störmer Verlet. In contrast to the argon simulation, however, a shift in pressure with a step-size-factor of 12 can be seen here. However, the peak remains at the same level as for the other step-size-factors.

Part IV.

Future Work and Conclusion

7. Future Work

In our work we mainly used the step-size-factors 1, 2, 3, 4, 6 and 12, as they are all divisors of 12 and thus we could make the measured quantities comparable. However, it would be interesting to observe even larger step-size-factors to see whether the trend of increasing oscillations in the total energy continues or whether this is just a coincidence and larger step-size-factors might even bring an improvement again. If this is the case, it could possibly also help to better understand the phenomenon of resonances.

For our measurements, we always used the same initial conditions for each scenario, i.e. the same positions and velocities of the particles. However, it would be interesting to change either the positions or the velocities slightly in order to achieve a different simulation process. This would allow us to investigate how the measured values change and to what extent the differences between the values for the different step-size factors can be attributed to the chaotic behavior of the MD system and which deviations are actually due to the different step-size-factors.

Ultimately, it would be interesting to examine the concept of different distance classes. This is addressed and used for example in [GT98, NVK93], for example. Here, the cutoff distance is divided into several intermediate cutoffs and the interactions are integrated based on the distance with different step-size-factors. Interactions that lie in inner spherical shells are integrated more often and interactions that lie further away are integrated with larger time steps. In this way, one could investigate to what extent the calculation time of three-body interactions can be further optimized while maintaining the measured quantities of the MD system.

8. Conclusion

In our work, we have investigated whether we can use the r-RESPA method to optimize the calculation time of three-body interactions in simulations with two-body and three-body interactions by integrating the three-body interactions with larger time steps.

We have chosen three scenarios for this purpose, the first one being a pseudo-scenario that has no relation to a real physical simulation. However, this scenario uses a range of parameters that often occur in real scenarios and thus this experiment provides a good overview of the effects of different step-size factors. The other two scenarios refer to parameters for the real scenarios argon and aluminum. We examined all three scenarios with our *DirectSum* implementation and different step-size factors for deviations in the total energy. The last two scenarios were also analyzed with our *Cutoff* implementation for structural properties such as the radial distribution function and the pressure.

Our measurements have shown that the deviations up to a step-size factor of 4 are sufficiently low for the method to be applicable. Above this step-size, deviations become increasingly noticeable. It should be noted, however, that a step-size factor of 4 already results in an almost 4-times speedup in the *DirectSum* calculation and approximately a speedup of 3 in the *Cutoff* calculation.

To summarize, we can state that the r-RESPA method can be used for MD simulations with two-body and three-body interactions, whereby the three-body interactions are integrated at larger time steps than the two-body interactions up to certain step-size factors in our scenarios in order to reduce the calculation time.

List of Figures

List of Tables

Bibliography

[AS22] B.P. Akhouri and J.R. Solana. Thermodynamic properties of Ar, Kr and Xe from a Monte Carlo-based perturbation theory with an effective two-body Lennard-Jones potential. *Physica A: Statistical Mechanics and its Applications*, 608:128280, December 2022.

[AT43] BM Axilrod and Ei Teller. Interaction of the van der waals type between three atoms. *The Journal of Chemical Physics*, 11(6):299–300, 1943.

[BC21] Danilo De Camargo Branco and Gary J. Cheng. Employing Hybrid Lennard-Jones and Axilrod-Teller Potentials to Parametrize Force Fields for the Simulation of Materials' Properties. *Materials*, 14(21):6352, October 2021.

[DHL+99] Peter Deuflhard, Jan Hermans, Benedict Leimkuhler, Alan E. Mark, Sebastian Reich, Robert D. Skeel, M. Griebel, D. E. Keyes, R. M. Nieminen, D. Roose, and T. Schlick, editors. *Computational Molecular Dynamics: Challenges, Methods, Ideas: Proceedings of the 2nd International Symposium on Algorithms for Macromolecular Modelling, Berlin, May 21–24, 1997*, volume 4 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[DO20] Matthias Diem and Chris Oostenbrink. The effect of different cutoff schemes in molecular simulations of proteins. *Journal of Computational Chemistry*, 41(32):2740–2749, December 2020.

[FPR+96] Scott E. Feller, Richard W. Pastor, Atipat Rojnuckarin, Stephen Bogusz, and Bernard R. Brooks. Effect of Electrostatic Force Truncation on Interfacial and Transport Properties of Water. *The Journal of Physical Chemistry*, 100(42):17011–17020, January 1996.

[GASSS98] B. García-Archilla, J. M. Sanz-Serna, and R. D. Skeel. Long-Time-Step Methods for Oscillatory Differential Equations. *SIAM Journal on Scientific Computing*, 20(3):930–963, January 1998.

[GHWS91] H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten. Generalized Verlet Algorithm for Efficient Molecular Dynamics Simulations with Long-range Interactions. *Molecular Simulation*, 6(1-3):121–142, March 1991.

[GKZ07] Michael Griebel, Stephan Knapek, and Gerhard Zumbusch. *Numerical simulation in molecular dynamics: numerics, algorithms, parallelization, applications ; with 63 tables*. Number 5 in Texts in computational science and engineering. Springer, Berlin Heidelberg, 2007.

[GSBN22] Fabio Alexander Gratl, Steffen Seckler, Hans-Joachim Bungartz, and Philipp Neumann. N ways to simulate short-range particle systems: Automated algorithm selection with the node-level library AutoPas. *Computer Physics Communications*, 273:108262, April 2022.

[GT98] Helmut Grubmueller and Paul Tavan. Multiple time step algorithms for molecular dynamics simulations of proteins: How good are they? *Journal of Computational Chemistry*, 19(13):1534–1552, October 1998.

[Hab24] Alexander Haberl. Implementation of verlet lists for 3-body interactions in autopas. Master's thesis, Technical University of Munich, Apr 2024.

[HLW06] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*. Number 31 in Springer series in computational mathematics. Springer, Berlin, 2006.

[IRS99] Jesús A Izaguirre, Sebastian Reich, and Robert D Skeel. Longer time steps for molecular dynamics. *The Journal of chemical physics*, 110(20):9853–9864, 1999.

[KNA14] S M Hossein Karimian, Hamed R Najafi, and Masoud Arabghahestani. Details about pressure calculation in molecular dynamic analysis. 2014.

[KY14] Penporn Koanantakool and Katherine Yelick. A Computation- and Communication-Optimal Parallel Direct 3-Body Algorithm. In *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 363–374, New Orleans, LA, USA, November 2014. IEEE.

[Lea01] Andrew R. Leach. *Molecular modelling: principles and applications*. Prentice Hall, Harlow, England ; New York, 2nd ed edition, 2001.

[LM15] Ben Leimkuhler and Charles Matthews. *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*, volume 39 of *Interdisciplinary Applied Mathematics*. Springer International Publishing, Cham, 2015.

[LS18] Michael L. Larsen and Raymond A. Shaw. A method for computing the three-dimensional radial distribution function of cloud particles from holographic images. *Atmospheric Measurement Techniques*, 11(7):4261–4272, July 2018.

[LZS06] Jianhui Li, Zhongwu Zhou, and Richard J Sadus. Parallelization algorithms for three-body interactions in molecular dynamics simulation. In *International Symposium on Parallel and Distributed Processing and Applications*, pages 374–382. Springer, 2006.

[Mar01] Gianluca Marcelli. *The role of three-body interactions on the equilibrium and non-equilibrium properties of fluids from molecular simulation*. PhD thesis, University of Kent (United Kingdom), 2001.

[Mar22] David Martin. A comparison of three-body algorithms for molecular dynamics simulations. Master's thesis, Technical University of Munich, Nov 2022.

[May23] Maximilian Mayr. Integrating three-body interactions for molecular dynamics simulation into simplemd. Master's thesis, Technical University of Munich, Aug 2023.

[MS99] Gianluca Marcelli and Richard J. Sadus. Molecular simulation of the phase behavior of noble gases using accurate two-body and three-body intermolecular potentials. *The Journal of Chemical Physics*, 111(4):1533–1540, July 1999.

[NVK93] Aiichiro Nakano, Priya Vashishta, and Rajiv K. Kalia. Parallel multiple-time-step molecular dynamics with three-body interaction. *Computer Physics Communications*, 77(3):303–312, November 1993.

[TBM91] Mark E. Tuckerman, Bruce J. Berne, and Glenn J. Martyna. Molecular dynamics algorithm for multiple time scales: Systems with long range forces. *The Journal of Chemical Physics*, 94(10):6811–6815, May 1991.

[TBM92] M. Tuckerman, B. J. Berne, and G. J. Martyna. Reversible multiple time scale molecular dynamics. *The Journal of Chemical Physics*, 97(3):1990–2001, August 1992.

[TBR91] Mark E. Tuckerman, Bruce J. Berne, and Angelo Rossi. Molecular dynamics algorithm for multiple time scales: Systems with disparate masses. *The Journal of Chemical Physics*, 94(2):1465–1469, January 1991.

[Whi99] John A. White. Lennard-Jones as a model for argon and test of extended renormalization group calculations. *The Journal of Chemical Physics*, 111(20):9352–9356, November 1999.