Semi-Analytical Integration in Molecular Dynamics

Zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

von der KIT-Fakultät für Mathematik des
Karlsruher Instituts für Technologie (KIT)
genehmigte

Dissertation

von

Tobias Kliesch

Tag der mündlichen Prüfung: 26. September 2023
Erster Gutachter: PD Dr. Volker Grimm
Zweite Gutachterin: Prof. Dr. Marlis Hochbruck

Acknowledgements

I gratefully acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) — Project-ID 389000774 — RTG 2450.

This work would not have been possible without the support and guidance of the people in my surroundings. I want to express my sincere gratitude to each person that has stood by my side over the past years.

First and foremost, I thank my supervisor PD Dr. Volker Grimm. His door was always open to openly discuss the challenges that I faced in my research and I am grateful for his assistance in bringing this thesis to its final form. I thoroughly enjoyed our close cooperation in creating and conducting lectures over the past years.

Further, I thank Prof. Dr. Marlis Hochbruck for creating an excellent work environment and making this research possible. Undoubtedly, her valuable notes enhanced the quality of this thesis.

During the isolated Corona times at the start of my journey, the connection I made in the RTG 'Tailored Scale-Bridging Approaches to Computational Nanoscience' were immensely valuable. The interdisciplinary discussions as well as the personal chats with people from different fields and backgrounds motivated me greatly. I want to especially thank Mohamed Tarek Elewa for the countless hours of collaborative work and the unquestioned help he provided for my research.

I thank the research group 'numerical analysis' in all its different configurations over the years for the excellent environment and for the countless productive and entertaining exchanges. From the joint coffee in the morning to the occasional beer in the evening, the time spent together has truly been enjoyable. My gratitude goes to Constantin Carle who read through and improved early versions of this thesis.

I especially want to express my gratitude to my family for their unwavering support in everything I do and every decision I make.

It is hard to put in words how much your commitment, Christina, to push me through the lowest lows and to celebrate the highest highs with me has helped me in achieving my goals. I dedicate this work to you.

Contents

List of Figures

List of Symbols

Mathematical Symbols

∇U Gradient of the function U
$\nabla_x U$ Vector containing the partial derivatives of U with respect to x
H_U Hessian matrix of the function U
$\mathbb{I}$ Identity matrix
$m_{\mathbf{max}}$ Maximal dimension of the Krylov subspace
p Degree of the Chebyshev polynomial in the Leapfrog-Chebyshev scheme
$\langle \cdot\,,\ \cdot \rangle$ Euclidean scalar product
$\| \cdot \|$ Euclidean norm
$\otimes$ Dyadic product

Potential Parameters

r_{ij} Vector pointing from particle i to particle j
ρ_{ij} Distance between particle i and j
ϵ Lennard-Jones energy parameter
σ Lennard-Jones distance parameter
$\bar{\rho}_{LJ}$ Lennard-Jones equilibrium distances
$\rho_{\mathbf{cut}}$ Cutoff radius of the Lennard-Jones potential
$\bar{\rho}_B$ Bond equilibrium value
k_B Bond constant
θ_{ijk} Angle between the particles i, j and k
$\bar{\theta}$ Angle equilibrium value
k_A Angle constant
ϕ_{ijkl} Dihedral angle between the planes spanned by the particles i, j, k and l
$\bar{\phi}$ Dihedral angle equilibrium value
k_T Torsion constant

Sets and Spaces

$\mathcal{B}$ Index set of all particles sharing a bond
$\mathcal{A}$ Index set of all particles sharing an angle
$\mathcal{T}$ Index set of all quadruplets sharing a dihedral angle
$\mathcal{K}_m(A, b)$ Krylov space w.r.t. the matrix A and vector b with dimension m

1 Introduction

Molecular dynamics simulation is a computational technique that has long proven to be an essential tool in today's scientific world [18]. Scientists from computational biology, chemistry, and physics are able to explain, validate, and expand on classic experiments using molecular dynamics simulations. The software that governs these simulations as well as the hardware that runs them have improved consistently over the past decades. A detailed summary of the history of molecular simulations is given in [29]. As a result of these improvements, molecular dynamics simulations achieve time scales that begin to be long enough to be compatible with biological processes and have moved from the analysis of single structures to the analysis of conformational ensembles [49].

This work aims to play a role in this process of continuous improvement from the viewpoint of numerical mathematics. From the late 1990s to the early 2000s, molecular dynamics was a much discussed topic in the mathematical community. Despite many advances in numerical time integration in molecular dynamics simulation during that time [43, 55, 72, 83], the most used integrator in this field remains to be the Verlet method. While mathematicians have laid out the theoretical basis on how to improve numerical integration in molecular dynamics, the widespread application of these ideas in software commonly used by computational scientists is lacking.

Recent applications of numerical time integration techniques involve the implementation of the mollified impulse method by L. Fath et al. [21] in the widely used molecular dynamics software package LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) [81], which controls numerical resonances of the impulse method proposed by M. Tuckerman et al. [83]. An important step for introducing semi-analytical integrators to applications in molecular dynamics simulation has been made by D. L. Michels and M. Desbrun [59]. They considered exponential integrators in a test environment with a large speedup compared to the standard Verlet integrator. This work is inspired by these results and aims to provide their massive speedup to a broad community of computational scientists. For that, we implement a similar approach in LAMMPS and thoroughly test it.

The implementation of integrators that follow a semi-analytical approach in molecular dynamics comes with a wide range of challenges, from mathematics, physics, and computer science. As a consequence, the work we present here is an interdisciplinary effort that combines the knowledge of, but also aims to reach, scientists from these fields.

In molecular dynamics simulation the classical equations of motions, stemming from Newton's second law of motion, are solved to describe the evolution of a system with many atoms over time. While most interesting phenomena are observed over multiple nanoseconds, the fastest motions require a femtosecond resolution. This contrast of necessary resolution and considered timeframe makes molecular dynamics simulations computationally expensive.

In order to understand where these contrasting scales come from and how we are able to overcome the smallest scales, we present in **chapter 2** the basics of Hamiltonian systems in the context of molecular dynamics. We discuss the known numerical techniques to integrate such systems and define the semi-analytical approach that we follow in this work. We apply numerical integrators that follow the semi-analytical approach to a well known test problem which is motivated by a numerical experiment conducted by E. Fermi, J. Pasta, S. Ulam and M. Tsingou [22].

The use of such integrators in molecular dynamics is not a trivial task. In **chapter 3** we discuss the technical details of semi-analytical integration in molecular dynamics and present the algorithms that we study throughout this work. We apply these integrators to the system introduced in chapter 2.

As we will see in chapter 2 and further discuss in chapter 3, we transform the nonlinear problem into a semi-linear problem by adding and subtracting a specific Hessian matrix. The construction and discussion of this Hessian matrix is the content of **chapter 4**.

With the mathematical and physical foundation established we switch the focus to important aspects of the efficient implementation. In **chapter 5**, we introduce the LAMMPS software package. LAMMPS is one of many packages that handles large scale molecular dynamics simulations and provides through its modular implementation a convenient way for modification. We discuss some important details of the software that is used for large scale molecular dynamics simulations and demonstrate the use of LAMMPS on a large scale model problem.

In **chapter 6**, we describe our implementation of the semi-analytical integrators in LAMMPS, the heart of this work. We thereby give special attention to the parallel implementation and use different molecular systems to motivate heuristic parameter choices and to illustrate the properties of the new integration scheme.

Finally, we explore the behavior of the integrator over long time periods in **chapter 7**. For that we apply it to various systems with different physical properties. We further validate the integrator with multiple realistic molecular dynamics simulations from computational physics, ranging from solid structures over liquids to vaporous systems. In this context, we also present the results of a close collaboration with computational physicists of the Research Training Group (RTG) 2450 — Tailored Scale-Bridging Approaches to Computational Nanoscience.

2 | Background

In this chapter we present the theoretical basics of molecular dynamics simulation from a mathematical and physical perspective. We approach this in four steps. First, we introduce *Hamiltonian* systems, which describe a large variety of dynamical systems, see for example [54, Chapter 1]. After discussing important properties of these systems, we specify it in a second step to the setting of this work, namely molecular dynamics. It is in general not possible to give an exact solution of these systems. We therefore introduce the most common numerical integration techniques that are used in this context as well as the semi-analytical approach that we will follow throughout this thesis. For detailed information on the topic of molecular dynamics simulation we refer to [6, 66, 70]. Finally, we compare the properties of a semi-analytical integrator with the commonly used Verlet integrator by applying both to a well-known test system, the Fermi-Pasta-Ulam-Tsingou problem [22].

Notation For a function $y : \mathbb{R} \to \mathbb{R}^n$ of time t, we use the dot notation to denote the derivative with respect to the time, i.e., $\dot{y} = \frac{dy}{dt}$. For a continuously differentiable function $U : \mathbb{R}^n \to \mathbb{R}$, we denote the gradient of U at x as the column vector $\nabla U(x)$. Further, for a function $U(x, y) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, $\nabla_x U(x, y)$ denotes the column vector containing the partial derivatives of U with respect to x, and analogously for y. With $\langle \cdot\,,\ \cdot \rangle$ we denote the Euclidean scalar product and with $\| \cdot \|$ the Euclidean norm.

2.1 Hamiltonian Systems

We start by introducing the general form of a Hamiltonian system. Let $q : \mathbb{R} \to \mathbb{R}^n$ and $p : \mathbb{R} \to \mathbb{R}^n$ be time-dependent functions. Given the Hamiltonian

$$H : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, \quad (q, p) \mapsto H(q, p), \tag{2.1}$$

the evolution of the functions q, p over time $t \in \mathbb{R}$ starting from initial values $q(t_0) = q^0$ and $p(t_0) = p^0$ is described by the Hamiltonian system

$$\begin{aligned} \dot{q} &= \nabla_p H(q, p), \\ \dot{p} &= -\nabla_q H(q, p), \end{aligned} \qquad \text{with} \quad \begin{bmatrix} q(t_0) \\ p(t_0) \end{bmatrix} = \begin{bmatrix} q^0 \\ p^0 \end{bmatrix}. \tag{2.2}$$

For a detailed introduction to this topic we refer to [54, Chapter 3]. With

$$Y = \begin{bmatrix} q \\ p \end{bmatrix} \in \mathbb{R}^{2n}, \qquad J = \begin{bmatrix} 0 & \mathbb{I} \\ -\mathbb{I} & 0 \end{bmatrix} \in \mathbb{R}^{2n \times 2n}, \quad \mathbb{I}, 0 \in \mathbb{R}^{n \times n}, \tag{2.3}$$

where $\mathbb{I}$ is the identity matrix, we are able to write (2.2) in a shorter and in the course of this chapter more useful formulation

$$\dot{Y} = J \nabla H(Y), \qquad Y(t_0) = Y^0. \tag{2.4}$$

In most cases, and in the remainder of this work, q refers to the position vector and p to the momentum vector respectively. Further, the Hamiltonian corresponds to the total energy in the system.

A fundamental law of physics is the *law of conservation of energy*. It is well known that Hamiltonian systems adhere to that law. This can be seen by computing the time-derivative of $H(Y)$

$$\dot{H}(Y) = (\nabla H(Y))^T \dot{Y} = (\nabla H(Y))^T J \nabla H(Y) = \left(\nabla_q H\right)^T \nabla_p H + \left(\nabla_p H\right)^T \left(-\nabla_q H\right) = 0.$$

It follows that the Hamiltonian is constant along the solution of (2.4) over time.

In order to get a better understanding of Hamiltonian systems, we collect two important properties. We revisit these properties in section 2.3 to discuss their importance in the context of numerical integration.

2.1.1 Symplecticity

While the word ‘symplectic’ was introduced by H. Weyl in 1946 [86, chapter IV] (derived from the Greek word meaning ‘complex’), the current viewpoint of symplectic geometry can be traced back to W. R. Hamilton’s formulation of Lagrangian mechanics [39], in 1835.

Symplecticity is a geometric property of the flow of the Hamiltonian system. The flow of the Hamiltonian system (2.4) is the mapping

$$\varphi_t : \mathbb{R}^{2n} \to \mathbb{R}^{2n}, \qquad \varphi_t \left(\begin{bmatrix} q^0 \\ p^0 \end{bmatrix} \right) = \begin{bmatrix} q(t) \\ p(t) \end{bmatrix}, \tag{2.5}$$

where $\varphi_t \left(\begin{bmatrix} q^0 \\ p^0 \end{bmatrix} \right)$ denotes the solution at time t with initial values $q(0) = q^0$ and $p(0) = p^0$.

Definition 1 (Hairer et al. [38, Def. VI.2.2]). *Let $U \subset \mathbb{R}^{2n}$ be an open set. A differential map $g : U \to \mathbb{R}^{2n}$ is called* symplectic *if the Jacobian matrix $g'(q,p)$ is symplectic for all $\begin{bmatrix} q \\ p \end{bmatrix} \in U$, i.e., if*

$$g'(q,p)^T J g'(q,p) = J,$$

where J is the matrix in (2.3).

For $n = 1$, symplecticity implies the area preservation of the flow φ_t. Poincaré showed in 1899 that the flow of the Hamiltonian system (2.4) is symplectic.

Theorem 1 (Poincaré [65]). *Let the Hamiltonian (2.1) be a twice continuously differentiable function on an open subset $U \subset \mathbb{R}^{2n}$. Then, for each fixed t, the flow φ_t is a symplectic transformation wherever it is defined.*

Furthermore, the following theorem shows that symplecticity of the flow is a characteristic property for Hamiltonian systems.

Theorem 2 (Hairer et al. [38, Thm. VI.2.6]). *Let $f : U \to \mathbb{R}^{2n}$ be continuously differentiable. Then, $\dot{y} = f(y)$ is locally Hamiltonian if and only if its flow φ_t is symplectic for all $y \in U$ and for all sufficiently small t.*

2.1.2 Reversibility

It is intuitive that a physical motion, like the undamped swing of a pendulum, follows the same trajectory backwards if the time is reversed. Hence, the motion is symmetric with respect to time. The following is a more general definition of this concept.

Definition 2 (Hairer et al. [38, Def. V.1.1]). *Let ϱ be an invertible linear transformation in the phase space[1] of $\dot{y} = f(y)$. This differential equation and the vector field $f(y)$ are called* ϱ-reversible *if*

$$\varrho f(y) = -f(\varrho y) \qquad \text{for all } y.$$

[1] The phase space is the set of all possible states of the system.

The specific transformation $\varrho(q,p) = (q,-p)$ plays an important role in this work. An interpretation of a system described by a differential equation that is ϱ-reversible under this transformation is the following: Starting from an initial state (q^0,p^0), the system follows the same trajectory whether one goes forward in time or first changes the sign of the momentum and then goes backward in time. Changing the sign of the momentum does not change the trajectory it only inverts the direction of motion. For differential equations that are reversible under this specific transformation one often omits explicitly naming the transformation and calls them *reversible*. In the following we consider Hamiltonian systems with Hamiltonian function satisfying $H(q,p) = H(q,-p)$. Such systems are reversible.

Before we turn our attention to molecular dynamics simulations we take a look at a simple, but for this work important Hamiltonian system, namely the harmonic oscillator.

2.1.3 The Harmonic Oscillator

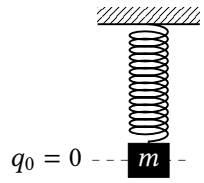


Figure 2.1. Representation of the harmonic oscillator as a mass-spring system (in equilibrium).

A common representation of the harmonic oscillator is an undamped *mass-spring* system with a single mass $m > 0$ at position q connected to a spring as shown in figure 2.1. The energy in the system is given by the Hamiltonian $H(q,p) = T(p) + V(q)$ with the functions

$$T(p) = \frac{1}{2m}p^2, \qquad V(q) = \frac{1}{2}kq^2, \tag{2.6}$$

where $k > 0$ is called the spring constant. Note that $V(q)$ has a global minimum at $q = 0$ with $V(0) = 0$ so that its gradient is zero. If in this case $p = 0$ we have $\nabla_p H(q,p) = \nabla_q H(q,p) = 0$ and the state (q,p) of the particle stays constant. We say the system is in its equilibrium and refer to the position at this state as the equilibrium position q^e.

With initial values q^0, p^0 at $t_0 = 0$, we obtain the Hamiltonian system

$$\dot{q} = m^{-1}p, \qquad q(0) = q^0, \tag{2.7a}$$

$$\dot{p} = -kq, \qquad p(0) = p^0. \tag{2.7b}$$

By differentiating (2.7a) with respect to time, we are able to rewrite the system as a second-order initial value problem

$$\ddot{q} = -\frac{k}{m}q, \qquad q(0) = q^0, \quad \dot{q}(0) = m^{-1}p^0. \tag{2.8}$$

The solution of this problem is given by

$$q(t) = \cos(t\omega)q^0 + \frac{1}{\omega}\sin(t\omega)p^0 \qquad \text{with} \quad \omega = \sqrt{\frac{k}{m}}. \tag{2.9}$$

The potential function in (2.6) is a quadratic function of the position. As a result, the right-hand side of (2.8) is a linear term. From the solution (2.9) of the initial value problem (2.8)

we get the trajectory of the mass m over time. This trajectory follows a harmonic oscillation with frequency ω. For this reason we refer to quadratic potentials as harmonic potentials. Harmonic potentials play a crucial role in this work. To understand in which form they appear in molecular dynamics, we present the Hamiltonian system governing molecular dynamics simulation.

2.2 Molecular Dynamics

In molecular dynamics the behavior of N particles in d dimensions is described by a Hamiltonian system with a *separable* Hamiltonian function

$$H(q,p) = T(p) + V(q) \qquad \text{with} \quad T(p) = \frac{1}{2} p^T M^{-1} p. \tag{2.10}$$

The first part of $H(q,p)$ in (2.10) is the quadratic kinetic energy $T(p)$ with a positive-definite diagonal matrix M that contains the masses of the particles. Note that Hamiltonian systems with Hamiltonians of the form (2.10) are $\varrho-$reversible (definition 2) under the transformation $\varrho(q,p) = (q,-p)$, i.e., they are reversible. The second part is the potential energy $V(q)$, which is specific to the problem at hand. In most cases it is a sum of different potential energies that model different interactions. For a detailed introduction in this topic we refer to [31, Chapter 1].

In the following we assign every particle a unique index between 1 and N, and refer to the d-dimensional position and momentum vector of a single particle with index i as q_i and p_i, respectively. The position vector q and the momentum vector p are then given by

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{bmatrix} \in \mathbb{R}^{dN}, \qquad p = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{bmatrix} \in \mathbb{R}^{dN}. \tag{2.11}$$

The simplest types of interactions between particles are pair-wise interactions. These interactions only depend on the distance between two particles.

Definition 3. *Let $q_i \in \mathbb{R}^d$ and $q_j \in \mathbb{R}^d$ be the positions of the particles with indices i and j. We define the vector that points from particle i to particle j as*

$$r_{ij} := q_j - q_i \in \mathbb{R}^d$$

and denote the distance between the particles with indices i and j by $\rho_{ij} := \|r_{ij}\|$.

A prominent potential that models such a pair-wise interaction is the *Lennard-Jones* potential [52, 53] which describes the attraction and repulsion of two uncharged, non-bonded particles. The energy parameter ϵ specifies the rate of attraction and repulsion,

and the distance parameter σ determines the equilibrium distance $\bar{\rho}_{LJ}$, i.e., the distance for which the gradient of the Lennard-Jones potential is zero. The most commonly used expression is

$$V_{LJ}(\rho_{ij}) := 4\epsilon \left[\left(\frac{\sigma}{\rho_{ij}} \right)^{12} - \left(\frac{\sigma}{\rho_{ij}} \right)^{6} \right], \tag{2.12}$$

where the equilibrium distance is given by the relation $\bar{\rho}_{LJ} = \sqrt[6]{2}\sigma$. Due to the exponents of the repulsive and attractive term in (2.12), this specific formulation is also referred to as the *Lennard-Jones-(12-6)* potential.

Another important pair-wise potential is the *Coulomb* potential. As the name suggests, it is an implementation of Coulomb's law [16] and describes the pair-wise, electrostatic interaction of charged particles. It is given by

$$V_C(\rho_{ij}) := \frac{1}{4\pi\varepsilon_0} \frac{e_i e_j}{\rho_{ij}}, \tag{2.13}$$

with the dielectric constant ε_0 and the charges e_i, e_j of particle i and j, respectively.

A second important type of interaction is between particles that are linked by bonds. Since a group of bonded particles forms a molecule, these interactions are also called intramolecular interactions. In contrast, the pair-wise interactions between different molecules are referred to as intermolecular interactions. The potentials governing these interactions are named accordingly, see [31, Section 5.2] for more details on intramolecular potentials.

The length of the bond between two particles i and j is given by the distance of the particles, i.e., ρ_{ij}. We further define the angle between three particles as follows.

Definition 4. *Let $q_i, q_j \in \mathbb{R}^d$ and $q_k \in \mathbb{R}^d$ be the positions of three successive particles with indices i, j and k. With the vectors*

$$r_{ij} = q_j - q_i, \qquad r_{kj} = q_j - q_k \in \mathbb{R}^d,$$

the angle determined by the particles i, j and k as shown in figure 2.2 is given by

$$\theta_{ijk} := \arccos\left(\frac{\langle r_{ij}, r_{kj} \rangle}{\|r_{ij}\| \|r_{kj}\|} \right).$$

The third parameter is the dihedral angle between the planes spanned by four successive particles as presented in figure 2.2.

Definition 5. *Let $q_i, q_j, q_k \in \mathbb{R}^d$ and $q_l \in \mathbb{R}^d$ be the positions of the four successive particles with indices i, j, k and l. With the vectors*

$$r_{ij} = q_j - q_i, \qquad r_{jk} = q_k - q_j, \qquad r_{kl} = q_l - q_k \in \mathbb{R}^d,$$

the dihedral angle between the two intersecting planes spanned by the four particles i, j, k and l, as displayed in figure 2.2 reads

$$\begin{aligned} \phi_{ijkl} &:= sign(\langle r_{ij}, n\rangle) \arccos\left(\langle \frac{m}{\|m\|}, \frac{n}{\|n\|}\rangle\right), \\ m &:= r_{ij} \times r_{jk}, \\ n &:= r_{jk} \times r_{kl}. \end{aligned}$$

This definition of the dihedral angle corresponds to the IUPAC convention [46]. In this convention the molecule is in the *cis* conformation when $\phi_{ijkl} = 0$. The cis conformation means that all particles lie in one plane and i and l are at the same side of the line through j and k. For $\phi_{ijkl} \neq 0$, the sign of the scalar product $\langle r_{ij}, n\rangle$ determines on which side of the plane through j, k, and l the particle i lies.

Each potential that we consider has an equilibrium value, i.e., a value at which its gradient is zero. We mark the equilibrium values for the bond potential as $\bar{\rho}_B$, for the angle potentials as $\bar{\theta}$, and for the torsional potential as $\bar{\phi}$. The rate at which the particles are pulled towards this equilibrium values is determined by the constant k_B, k_A, and k_T for the bond, angle and torsional potential, respectively. The standard formulations of the intramolecular potentials for a single bond, angle, and dihedral angle are:

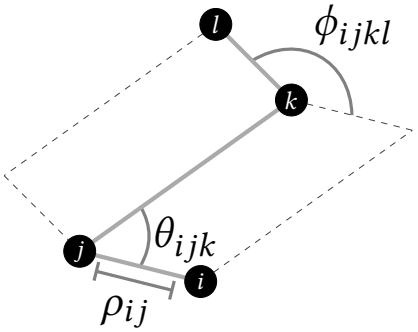


Figure 2.2. Illustration of the bond, angle, and dihedral angle parameters.

Bond potential

$$V_B(\rho_{ij}) := \frac{k_B}{2}(\rho_{ij} - \bar{\rho}_B)^2, \tag{2.14}$$

Angle potential

$$V_A(\theta_{ijk}) := \frac{k_A}{2}(\theta_{ijk} - \bar{\theta})^2, \tag{2.15}$$

Torsional potential

$$V_T(\phi_{ijkl}) := k_T\big(1 - \cos(\phi_{ijkl} - \bar{\phi})\big). \tag{2.16}$$

For small deviations from the equilibrium angle $\bar{\phi}$ one may approximate the torsional potential by $\widetilde{V}_T(\phi_{ijkl}) = \frac{k_T}{2}(\phi_{ijkl} - \bar{\phi})^2$. In this case the three intramolecular potentials are harmonic potentials with respect to the bond length ρ_{ij}, the angle θ_{ijk} and dihedral angle ϕ_{ijkl} respectively.

Definition 6. *The sets of indices of particles that interact with the bond, angle and torsional potential are given by*

$$\mathcal{B} = \{(i,j) \mid \textit{the particles with indices } i \textit{ and } j \textit{ share a bond}\},$$
$$\mathcal{A} = \{(i,j,k) \mid \textit{the particles with indices } i, j \textit{ and } k \textit{ share an angle}\},$$
$$\mathcal{T} = \{(i,j,k,l) \mid \textit{the particles with indices } i, j, k \textit{ and } l \textit{ share a dihedral angle}\},$$

where there is exactly one element in the set for each bond, angle and dihedral angle respectively, i.e., the sets contain no element that is a permutation of another element in the same set.

The systems that we consider in this work may contain a combination of these inter- and intramolecular potentials. The full potential energy for these systems thus takes the form

$$V = V_{\text{Intra}} + V_{\text{Inter}}, \tag{2.17a}$$

with

$$V_{\text{Intra}} = \sum_{(i,j)\in\mathcal{B}} V_B(\rho_{ij}) + \sum_{(i,j,k)\in\mathcal{A}} V_A(\theta_{ijk}) + \sum_{(i,j,k,l)\in\mathcal{T}} V_T(\phi_{ijkl}), \tag{2.17b}$$

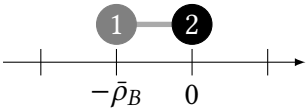
$$V_{\text{Inter}} = \sum_{i=2}^{N}\sum_{j=1}^{i-1} V_{LJ}(\rho_{ij}) + \sum_{i=2}^{N}\sum_{j=1}^{i-1} V_C(\rho_{ij}). \tag{2.17c}$$

The individual potentials describe the attraction and repulsion of particles. They differ, however, in the rate of which the particles are pulled towards and pushed away from each other. Broadly speaking, the motion imposed by the intramolecular potentials is faster than those imposed by the intermolecular potentials. This motivates a further modification for the separable Hamiltonian in (2.10), namely

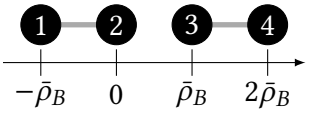
$$H(q,p) = T(p) + W(q) + U(q), \tag{2.18}$$

where we sum up the potentials introducing the fastest motions in $W(q)$ and the remainder of the potential $V(q)$ in $U(q)$. From now on, we refer to the potentials summed in $W(q)$ as fast and those in $U(q)$ as slow potentials. The distinction in fast and slow potentials is dependent on the underlying system. In many cases this means that we collect the intramolecular potentials in $W(q)$ and the intermolecular potentials in $U(q)$. Note that the frequency of the harmonic oscillations, i.e., the rate of attraction and repulsion, is determined by the constant k that is multiplied to the quadratic term (cf. (2.9)). In molecular dynamics simulations it commonly holds that $k_B > k_A > k_T$. As a consequence, the oscillations around the bond equilibrium have a higher frequency than those around the angle and dihedral angle equilibrium.

A second difference of the potential functions is the number of interactions each potential imposes in relation to the number of particles in the system. Looking at (2.17b), one can



(a) Equilibrium state of the two particle system, the position of the gray particle is fixed.



(b) Equilibrium state of the four particle system.

Figure 2.3. First alterations to the mass-spring model seen in figure 2.1.

see that the intramolecular potentials act between a fixed number of particles. A particle interacts through the bond potential with one, through the angle potential with two and through the torsional potential with three different particles. As a consequence, the number of intramolecular interactions grows at most linearly with the number of particles in the system. In contrast, a particle interacts through the intermolecular potentials with every other particle in the system. The number of intermolecular interactions thus grows quadratically with the number of particles[2]. In a large scale simulation the intramolecular potentials are thus significantly cheaper to compute than the intermolecular potentials.

To further motivate this separation of potentials and its consequences for numerical integration we extend the mass-spring system introduced in section 2.1.3.

2.2.1 The Harmonic Oscillator Revisited

We take a second look at the harmonic oscillator introduced in section 2.1.3, now in the context of molecular dynamics. In order to describe the system with the standard molecular dynamics potentials, we consider two particles with position $q_1, q_2 \in \mathbb{R}$ that are connected with a bond and fix the position of the particle with index 1 over time. We use the bond potential V_B (see (2.14)) with equilibrium value $\bar{\rho}_B$ and constant $k_B = 1200$ and set $q_1 = -\bar{\rho}_B$. We further assume that the position of the free moving particle stays on the right-hand side of the fixed particle, i.e., $q_2(t) > -\bar{\rho}_B$ for all t. Note that this is not a restriction as long as the initial positions fulfill $q_2(0) = q_2^0 > q_1^0 = q_1(0)$. As a consequence, the equilibrium position of the second particle is given by $q_2^e = 0$, and it holds that

$$\rho_{12} = |q_2 - q_1| = |q_2 + \bar{\rho}_B| = q_2 + \bar{\rho}_B. \tag{2.19}$$

An illustration of this system can be seen in figure 2.3a. Inserting (2.19) in (2.14) we get the formulation

$$V_B(\rho_{12}) = \frac{k_B}{2}(\rho_{12} - \bar{\rho}_B)^2 = \frac{k_B}{2}q_2^2. \tag{2.20}$$

In this case the potential (2.20) coincides with the potential function of the mass-spring system (2.6).

We extend this system by introducing a second pair of bonded particles, with the same bond parameters, on the right-hand side of the original pair and unfix the fixed particle. The non-bonded particles interact with the Lennard-Jones potential with energy parameter $\epsilon = 0.5$

[2] We discuss the techniques to compute these interactions in more detail in section 5.1.1

and distance parameter $\sigma = \bar{\rho}_B / \sqrt[6]{2}$ so that the equilibrium distance of the Lennard-Jones potential is given by $\bar{\rho}_{LJ} = \sqrt[6]{2}\sigma = \bar{\rho}_B$. We end up with a system with four particles and equilibrium position vector $q_e = \begin{bmatrix} -\bar{\rho}_B & 0 & \bar{\rho}_B & 2\bar{\rho}_B \end{bmatrix}^T$. We enumerate the particles from left to right and show the full system in figure 2.3b. The set of indices pairs of particles that interact with the bond potential is thus given by $\mathcal{B} = \{(1,2),(3,4)\}$. Since it is common that directly bonded particles do not interact with the Lennard-Jones potential, the set of indices pairs of particles that interact with the Lennard-Jones potential is $\mathcal{L} = \{(1,3),(1,4),(2,3),(2,4)\}$ and the Hamiltonian reads

$$H(q,p) = T(p) + \sum_{(i,j)\in\mathcal{B}} V_B(\rho_{ij}) + \sum_{(i,j)\in\mathcal{L}} V_{LJ}(\rho_{ij}).$$

While we can not give an exact solution to the resulting Hamiltonian system with general initial values q^0, p^0, we use this example to further motivate the different quantitative properties of the potentials and the separation of potentials in (2.18).

We first consider the number of interactions with the Lennard-Jones and bond potentials, i.e., the number of elements in the sets $\mathcal{L}$ and $\mathcal{B}$, in relation to the bonded pairs in the system. If we were to add a third pair of bonded particles, the cardinality of $\mathcal{L}$ would increase from four to 12, whereas the cardinality of $\mathcal{B}$ would only increase by one. For a general system with N_P bonded pairs of particles, we get

$$\begin{aligned} |\mathcal{L}| &= 4\sum_{k=1}^{N_P-1}(N_P-k) = 4\left(\frac{1}{2}(N_P-1)N_P\right) = 2(N_P-1)N_P, \\ |\mathcal{B}| &= N_P. \end{aligned}$$

As mentioned before we see that the number of interactions with the Lennard-Jones potential grows quadratically, while the number of interactions with the bond potential only grows linearly with the number of pair of particles in the system.

The values chosen for the parameters ϵ and k_B that describe the rate of attraction and repulsion of the particles, are chosen to mimic realistic molecular dynamics. The rate of attraction and repulsion of the Lennard-Jones potential depends on ϵ. Since $k_B \gg \epsilon$, the motions imposed by the bond potential is significantly faster.

Since we cannot solve this system analytically we turn our attention to numerical time integration and will revisit this example.

2.3 Numerical Time Integration

In numerical time integration one iteratively computes an approximation to the solution at given times of a time-dependent differential equation. For a detailed introduction on techniques for solving ordinary differential equations, we refer to [37]. We first discretize in time with a fixed (time)stepsize $\tau > 0$ and denote by $t_n = t_0 + n\tau$ the timestep after $n \in \mathbb{N}$ steps with stepsize τ. A specific algorithm, referred to as a *numerical integrator*,

computes from an initial state $q^0 = q(t_0), p^0 = p(t_0)$ an approximation to the state at t_1, denoted by $q^1 \approx q(t_1), p^1 \approx p(t_1)$. The simplest class of such numerical integrators only uses the currently known state to compute the state at the next timestep. Such integrators are called *one-step* methods.

Analogously to the continuous case (cf. (2.5)), we consider in the context of autonomous ordinary differential equations the mapping

$$\Phi_\tau : \mathbb{R}^{2dN} \to \mathbb{R}^{2dN}, \qquad \Phi_\tau \left(\begin{bmatrix} q^0 \\ p^0 \end{bmatrix} \right) = \left(\begin{bmatrix} q^1 \\ p^1 \end{bmatrix} \right),$$

where $\Phi_\tau \left(\begin{bmatrix} q^0 \\ p^0 \end{bmatrix} \right)$ denotes the approximation after one step of the one-step method with stepsize τ and initial values q^0 and p^0. In analogy to the flow φ_t of a continuous system, Φ_τ is called the *numerical flow*. We further denote by $\Phi_{-\tau}$ one step of the numerical integrator with stepsize $-\tau$.

Recall, that the Hamiltonian systems in molecular dynamics with Hamiltonian (2.10) have certain structural properties, namely they are symplectic (cf. definition 1) and reversible (cf. definition 2). A natural requirement for the numerical integrator is to preserve these properties, see [38, Chapter V, VI] for an in-depth discussion.

We first introduce the concept of symplecticity to numerical integration.

Definition 7 (Hairer et al. [38, Def. VI.3.1])**.** *A numerical one-step method is called* symplectic *if the numerical flow Φ_τ is symplectic (cf. definition 1) whenever the method is applied to a sufficiently smooth Hamiltonian system.*

We have seen in theorem 2 that symplecticity is a characteristic property of Hamiltonian systems. For that reason a lot of effort has been put into finding numerical integrators that conserve the symplectic structure. We refer to [38, Chapter VI] for an overview and to [54, Section 4.2] for a detailed discussion on the construction of symplectic integrators. It has been shown that symplectic numerical integrators almost conserve the total energy over an exponentially long period of time $t = \exp(c/\tau\omega)$ [5, 67], where ω is the frequency of the fastest oscillation in the system. This result requires that the product $\tau\omega$ is sufficiently small.

The second property of the Hamiltonian system in molecular dynamics is reversibility, i.e., it is $\varrho-$reversible with $\varrho(q, p) = (-q, p)$ (cf. definition 2). We introduce a similar concept to numerical one-step methods.

Definition 8 (Hairer et al. [38, Def. V.1.4])**.** *A numerical one-step method Φ_τ is called* symmetric *or* time-reversible *if it satisfies*

$$\Phi_\tau \circ \Phi_{-\tau} = id.$$

This means that after one step of a symmetric one-step method with stepsize τ and a second step with stepsize $-\tau$ one arrives back at the initial state. The flow of a symmetric one-step method applied to a reversible differential equation is reversible [38, Chapter V]. This means that a symmetric one-step method conserves the reversibility of the Hamiltonian system with Hamiltonian (2.10).

An important measure for the quality of a numerical integration of a Hamiltonian system over long time periods is the conservation of the energy, i.e., the Hamiltonian. E. Hairer and C. Lubich [35] showed that in the case of numerical long-time integration of oscillatory differential equations, symmetry and not symplecticity is the key to long-time energy conservation. More recently, E. Hairer [34] showed that there are even cases, where the symplectic structure and the total energy can not be preserved at the same time. For this reason we consider in this work numerical integrators that are symmetric but not necessarily symplectic.

For the sake of readability we set the mass of all particles in the remainder of this chapter to one. Hence, we have $M = \mathbb{I}$ and discuss the case of a general, diagonal mass matrix in section 3.1.

2.3.1 The Verlet Method

For the derivation of the Verlet method we rewrite the system of ordinary first-order differential equations (2.2) with the separable Hamiltonian (2.10) as the second-order differential equation

$$\ddot{q} = -\nabla V(q), \tag{2.21}$$

with initial values $q(t_0) = q^0$ and $\dot{q}(t_0) = \dot{q}^0 = p^0$.

A discretization of (2.21) is obtained by approximating the second time derivative on the right-hand side with the central second-order difference quotient

$$\ddot{q}(t_n) \approx \frac{q(t_{n+1}) - 2q(t_n) + q(t_{n-1})}{\tau^2}. \tag{2.22}$$

Inserting (2.22) in the left-hand side of (2.21) and replacing the exact function evaluation at time t by its approximation $q_n \approx q(t_n)$ on the right-hand side one gets

$$q^{n+1} - 2q^n + q^{n-1} = -\tau^2 \nabla V(q^n). \tag{2.23}$$

With the position at the previous step q^{n-1} and the position at the current step q^n we are able to compute q^{n+1}. In contrast to one-step methods, methods that use information of the past two steps are called *two-step* methods. Since several scientists from different branches of science proposed the integrator (2.23), or closely related variations, it is known by many names (see [36, p.402]). We refer to it as the *Verlet* method in reference to L. Verlet [84], who proposed this method for computations in molecular dynamics. As a future reference for section 2.3.3, we note that in the context of partial differential equations, the method is also called *leapfrog* method.

A one-step formulation, that gives an additional approximation to the momentum p^{n+1}, can be derived from (2.23) by setting

$$p^{n+1/2} := \frac{q^{n+1} - q^n}{\tau},\ n = 0, 1, \dots, \qquad p^n := \frac{1}{2}(p^{n+1/2} + p^{n-1/2}) = \frac{q^{n+1} - q^{n-1}}{2\tau},\ n = 1, 2, \dots .$$

The resulting integration scheme is known as the *Velocity-Verlet* scheme and reads

$$\begin{aligned} p^{n+1/2} &= p^n - \tfrac{\tau}{2}\nabla V(q^n), \\ q^{n+1} &= q^n + \tau p^{n+1/2}, \\ p^{n+1} &= p^{n+1/2} - \tfrac{\tau}{2}\nabla V(q^{n+1}). \end{aligned} \tag{2.24}$$

Note that the evaluation of $\nabla V(q^{n+1})$ can be reused in the next step. We discuss the details of an efficient implementation in section 3.3.

Despite or more likely because of its simple nature, the Verlet scheme is the most used numerical integrator in molecular dynamics (see [33] for more details on its application in molecular dynamics). This is not only due to its easy implementation but also thanks to its structure preserving properties.

Theorem 3. *The one-step formulation of the Verlet scheme (2.24) is a symplectic and symmetric method of order 2.*

Proof. The proof is a combination of [38, Theorem. VI.3.4] and [38, Theorem V.2.5]. □

For more details on the Verlet integrator in the context of geometric numerical integration we refer to [36].

Numerical Solution to the Four Particle System

We use the one-step formulation of the Verlet integrator to approximate the evolution of the 4−particle system defined in section 2.2.1 with $\bar{\rho}_B = 1$ and the initial values

$$q(0) = q^0 = \begin{bmatrix} -0.99 & 0 & 2.55 & 3.55 \end{bmatrix}^T, \qquad p(0) = p^0 = 0 \in \mathbb{R}^4.$$

We integrate the system with the stepsize $\tau = 0.002$ over 37 200 timesteps and plot the trajectory of particle 2 in figure 2.4.

On the top left of figure 2.4, we display the position of the second particle over the whole simulation period. In this timescale we can see the initial attraction to and later repulsion from the particle pair $(3, 4)$ due to the Lennard-Jones potential. In general, we are more interested in this behavior over long time periods. We also plot the trajectory in a smaller scale at different times of the simulation. In this finer timescale we see the fast harmonic oscillations with frequency $\omega = \sqrt{k_B} = \sqrt{1200} \approx 34.64$ that are due to the bonded interactions with particle 1. The fast intramolecular interactions are superimposed on the slow intermolecular interactions.

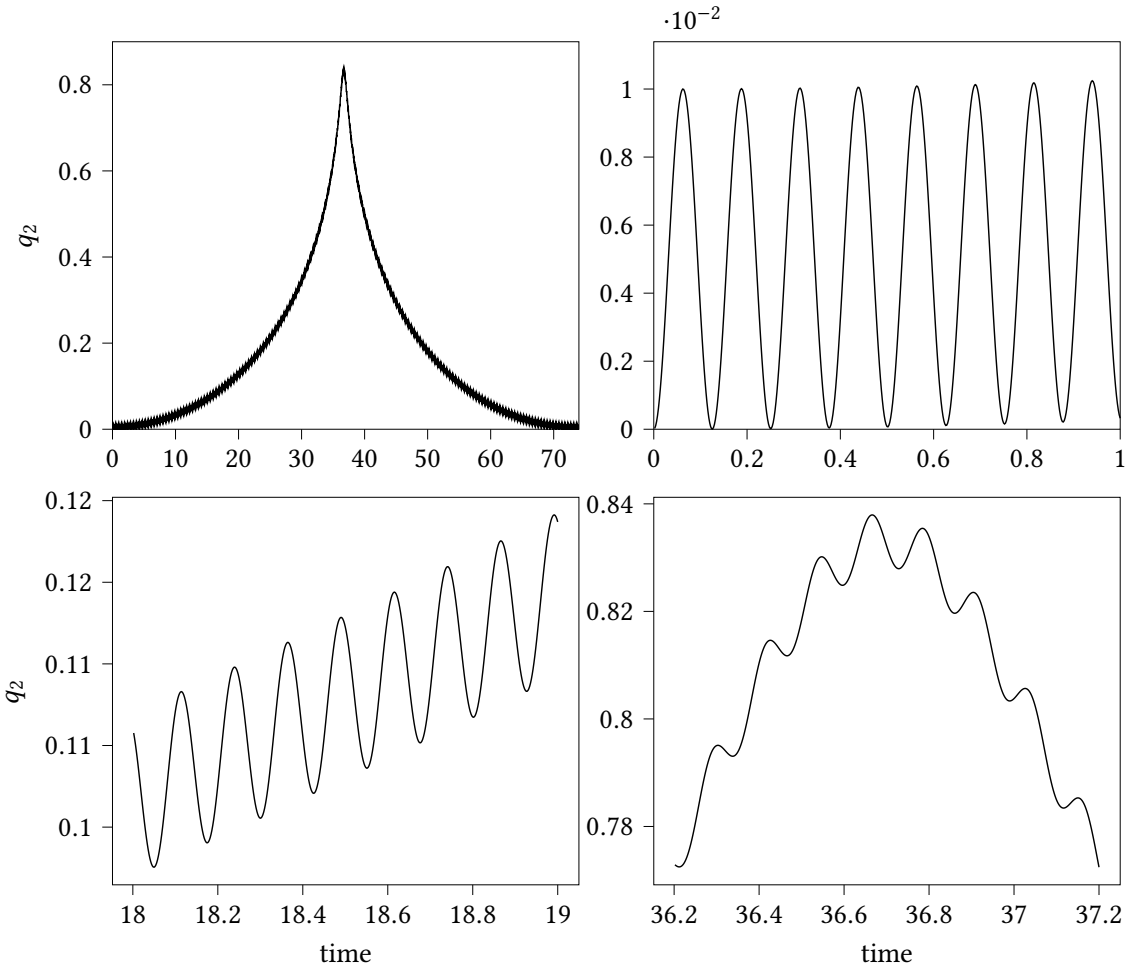


Figure 2.4. Position q_2 of the second particle of the four-particle system shown in figure 2.3 as a function of time, over the whole simulation time (top left) and zoomed in at various moments of the simulation. The trajectory is generated with the Verlet integrator and the stepsize $\tau = 0.002$.

We want to compute accurate approximations of the long-time behavior as shown on the top left of figure 2.4 in as little time as possible. Each step of the numerical integration contains the evaluation of the gradients of the potentials as well as the scalar operations in the integrator (2.24). The larger we choose the stepsize τ, the fewer steps and consequently computations are needed. However, we can not choose the stepsize arbitrarily large. To demonstrate this we consider again the harmonic oscillator discussed in section 2.1.3.

Lemma 1. *The approximations q^n, p^n to the solution of (2.7) with arbitrary initial values q^0, p^0 obtained from the Verlet integrator (2.24) are only bounded independent of the time interval when $\tau\omega < 2$, where $\omega = \sqrt{k} > 0$.*

Proof. A proof of this well-known property can for example be seen in [36, Example 3.4]. □

The stepsize is thus restricted by the highest frequency in the system. In the case of the 4−particle system the highest frequency stems from the harmonic bond potential with $k_B = 1200$ which results in a stepsize restriction of

$$\tau < 2/\sqrt{1200} \approx 0.0577.$$

In general, the stepsize in molecular dynamics simulation is often restricted by the cheap but fast potentials $W(q)$, whereas the main computational effort stems from the expensive but slow potentials $U(q)$ (cf (2.18)). A method that incorporates the underlying structure of Hamiltonian systems with Hamiltonian (2.18) is the *impulse method*.

2.3.2 The Impulse Method

In order to derive the impulse method, we first consider a different interpretation of the Verlet integrator. Note that the first and third equation of the one-step formulation (2.24) only depend on the potential energy in the system, whereas the second equation is solely dependent on the kinetic energy, since $\nabla T(p^n) = p^n$. We remind ourselves that this separation coincides with the two separated functions of the separable Hamiltonian (2.10) and write the Hamiltonian system (2.2) as two separate Hamiltonian systems, i.e.,

$$\begin{aligned} \dot{p} &= -\nabla_q V(q), & \dot{p} &= 0, \\ \dot{q} &= 0, & \dot{q} &= \nabla_p T(p), \end{aligned}$$

where each system can be solved exactly. This allows the interpretation of the Verlet method as the composition

$$\varphi^V_{\tau/2} \circ \varphi^T_\tau \circ \varphi^V_{\tau/2}, \tag{2.25}$$

where φ^V_t is the exact flow of the first and φ^T_t of the second subsystem. The idea of splitting a problem into two parts and expressing the total flow as a composition of the flows of the subproblems goes back to H. F. Trotter [82]. Numerical methods of this kind are called *splitting methods* and the specific splitting in (2.25) is called *Strang splitting* [77]. The Hamiltonian (2.18), where the potential function $V(q)$ is separated in a fast potential $W(q)$ and slow potential $U(q)$ motivates splitting the problem up differently.

To derive the impulse method we consider the Hamiltonian $H(q,p) = [T(p) + W(q)] + U(q)$ (cf. (2.18)) and write the Hamiltonian system as the two systems

$$\begin{aligned} \dot{p} &= -\nabla_q U(q), & \dot{p} &= -\nabla_q W(q), \\ \dot{q} &= 0, & \dot{q} &= \nabla_p T(p), \end{aligned}$$

We approximate the flow of the system by the composition

$$\varphi^U_{\tau/2} \circ \varphi^{(T+W)}_\tau \circ \varphi^U_{\tau/2},$$

where φ^U_t is the flow of the first and $\varphi^{(T+W)}_t$ the flow of the second system. In 1992 M. Tuckerman, B. J. Berne and G. J. Martyna [83] proposed this method under the name

r-RESPA (Reversible-Reference System Propagator Algorithm) in the context of molecular dynamics. One step of the method is given by

1. kick	$p_+^n \leftarrow p^n - \frac{\tau}{2}\nabla U(q^n)$
2. oscillate	advance by τ on the flow of the system with reduced Hamiltonian $\widetilde{H} = T + W$ and initial values (q^n, p_+^n) to get (q^{n+1}, p_-^{n+1})
3. kick	$p^{n+1} \leftarrow p_-^{n+1} - \frac{\tau}{2}\nabla U(q^{n+1})$,

After an initial impulse (`kick`) to the momentum given by the Hamiltonian system with the slow potential, the system is propagated along the solution of the Hamiltonian system with Hamiltonian $\widetilde{H}(q,p) = T(p) + W(q)$ (`oscillate`). The impulse from the Hamiltonian system with the slow potential is then applied to the momentum for a second time (`kick`). This interpretation of the scheme motivates the name impulse method, see [26]. In general, it is not possible to solve the second system exactly. In the implementation of the r-RESPA scheme the solution of the reduced system is approximated by the Verlet method with a smaller stepsize. Note that the fast potential $W(q)$ may be split up further into a sum of fast and faster potentials. In this case the `oscillate` step can be expanded to an additional r-RESPA step, with even smaller inner stepsize.

The method is symplectic as a composition of symplectic transformations and symmetric as a symmetric composition of symmetric steps (see [38, XIII.1.3]). The outer step-size restriction of the impulse method is now determined by the slow potential U, provided that the `oscillate` step is performed with sufficient precision.

At the cost of additional computations of the fast interactions, we are able to use larger stepsizes for the slow interactions. This gives a computational advantage if the computation of the slow potentials is more expensive than those of the fast potentials. We have discussed in section 2.2, and illustrated with the example in section 2.2.1 that this is the case in molecular dynamics.

However, this method comes with the challenge of numerical resonances. Numerical resonance can be described as a corruption of a system's dynamics due to the integrator (cf. [71, p.26]). In our case they may appear when the slow potential gets evaluated at a rate that is larger than half the period of the fast oscillations. If, in this case, the evaluation rate aligns with the frequency of the fast oscillation, the position of the particle does not properly reflect the oscillatory behavior. B. García-Archilla, J. M. Sanz-Serna and R. D. Skeel [26] analyzed numerical resonances for linear model problems when the product of the stepsize and the eigenfrequency of the fast potential is near an integral multiple of π. They also provide a solution to this problem: Instead of using the exact position after the `kick` step one evaluates the slow potential at an averaged position. This leads to the mollified impulse method. L. Fath, M. Hochbruck and C. V. Singh [21] provide more details in the context of molecular dynamics and a specific implementation of this method in LAMMPS.

In the four-particle example presented in section 2.2.1, the Verlet integration in the `oscillate` step of the r-RESPA method needs very small stepsizes in order to give an accurate approximation to the harmonic oscillation of the bond. However, the exact solution of the harmonic oscillator is known for any time with arbitrary initial values (see section 2.1.3).

In this work we focus on integrators that combine the exact solution of the linear problem with the numerical approximation of the nonlinear problem and refer to this approach as the *semi-analytical* approach. The idea for this approach can be traced back to J. Hersch [40].

2.3.3 The Semi-Analytical Approach

In the following we consider the Hamiltonian

$$H(q,p) = T(p) + W(q) + U(q) \qquad \text{with} \quad W(q) = \frac{1}{2} q^T A q, \tag{2.26}$$

where we now assume that the fast potential $W(q)$, is a quadratic function with a real, symmetric, positive semi-definite matrix A. In section 3.1.1, we discuss in more detail how the nonlinear, separable Hamiltonian (2.18) is rewritten to fit this formulation. We derive the approach again from the second-order differential equations (cf. (2.21)). With the Hamiltonian (2.26) it reads

$$\ddot{q} = -Aq + g(q), \tag{2.27}$$

with $g(q) = -\nabla U(q)$. We refer to the system of differential equations (2.27) with the initial values $q(t_0) = q^0, \dot{q}(t_0) = p(t_0) = p^0$ as a *semilinear* initial value problem.

Gautschi-type Integrator

Semi-analytical integrators are derived from the variation-of-constant formula

$$q(t+\tau) = \cos(\tau\Omega)q(t) + \Omega^{-1}\sin(\tau\Omega)\dot{q}(t) + \int_0^\tau \Omega^{-1}\sin((\tau-s)\Omega)g(q(t+s))\mathrm{d}s, \tag{2.28}$$

with $\Omega = A^{1/2}$, by an appropriate approximation of the integral. We present two different ideas to derive numerical integrators from (2.28). Note that the trigonometric functions are matrix-valued functions. We give a general introduction to matrix functions in appendix A.3 and discuss their implementation in section 3.2.

(G) For constant nonlinearity $g(q) \equiv const$ the integral in (2.28) can be solved analytically. This motivates a scheme that was proposed in 1961 by W. Gautschi [27]. Its two-step formulation reads

$$q^{n+1} - 2\cos(\tau\Omega)q^n + q^{n-1} = \tau^2 \mathrm{sinc}^2(\tfrac{1}{2}\tau\Omega)g^n, \tag{2.29}$$

with $g^n = g(q^n)$ and $\operatorname{sinc}(\xi) = \frac{\sin(\xi)}{\xi}$. An approximation of the momentum is then given by

$$2\tau \operatorname{sinc}(\tau\Omega) p^n = q^{n+1} - q^{n-1}.$$

With the starting values

$$q^0, \qquad q^1 = \cos(\tau\Omega) q^0 + \Omega^{-1} \sin(\tau\Omega) \dot{q}^0 + \frac{1}{2}\tau^2 \operatorname{sinc}^2(\tfrac{1}{2}\tau\Omega) g^0, \tag{2.30}$$

the method solves the semilinear initial value problem (2.27) exactly if $g(q)$ is constant.

(D) P. Deuflhard [17] proposed a scheme that uses the trapezoidal rule to approximate the integral. It reads in the two-step formulation

$$q^{n+1} - 2\cos(\tau\Omega) q^n + q^{n-1} = \tau^2 \operatorname{sinc}(\tau\Omega) g^n$$

and a suitable one-step formulation is given by

$$\begin{bmatrix} q^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \cos(\tau\Omega) & \Omega^{-1} \sin\tau\Omega \\ -\Omega \sin\tau\Omega & \cos(\tau\Omega) \end{bmatrix} \begin{bmatrix} q^n \\ p^n \end{bmatrix} + \frac{\tau}{2} \begin{bmatrix} \tau \operatorname{sinc}(\tau\Omega) g^n \\ \cos(\tau\Omega) g^n + g^{n+1} \end{bmatrix}. \tag{2.31}$$

From this formulation we get an interesting connection to the previously discussed impulse method when applied to a Hamiltonian system with Hamiltonian (2.26). In fact, the one-step formulation of the *Deuflhard* method (2.31) and the impulse method, with exact solution of the `oscillate` step

$$\begin{bmatrix} q^{n+1} \\ p_-^{n+1} \end{bmatrix} = \begin{bmatrix} \cos(\tau\Omega) & \Omega^{-1} \sin\tau\Omega \\ -\Omega \sin\tau\Omega & \cos(\tau\Omega) \end{bmatrix} \begin{bmatrix} q^n \\ p_+^n \end{bmatrix},$$

coincide.

It is consequently no surprise that both, the Gautschi **(G)** and the Deuflhard **(D)** method, suffer from similar numerical resonances as the impulse method. The resonances can be avoided with the same approach as for the impulse method, i.e., by evaluating the gradient of the slow nonlinear potential at an averaged position. More specifically, B. García-Archilla, J. M. Sanz-Serna and R. D. Skeel [26] proposed to use the time average of the solution of an auxiliary initial value problem

$$\ddot{y} = -Ay, \qquad y(0) = q, \quad \dot{y}(0) = 0,$$

given by

$$a(q) = \frac{1}{\tau} \int_0^\tau y(s) ds = \operatorname{sinc}(\tau\Omega) q.$$

We refer to [26, Section 5] for more details.

This motivates schemes that set

$$g^n = g(\Phi q^n), \quad \text{with} \quad \Phi = \phi(\tau\Omega), \tag{2.32}$$

where ϕ is an even function satisfying $\phi(0) = 1$ that vanishes at integral multiples of π. The function ϕ is commonly referred to as the *inner filter function*, as it filters out the numerical resonances.

The two schemes (2.29) and (2.31) belong to a general class of trigonometric integrators, for more details see [34, Chapter XIII]. Trigonometric integrators have been studied extensively in the past. Important contributions are given in [32, 35, 44] and for a closer look at their relation with splitting methods we refer to [12]. These integrators belong to an even larger class of exponential integrators. A detailed summary of exponential integrators can be found in [45].

For the remainder of this work we will consider methods of the form (2.29) as derived in **(G)** and refer to them as Gautschi-type integrators. With the trigonometric identity $\cos(2\alpha) = 1 - 2\sin^2(\alpha)$ we reformulate (2.29) and obtain

$$q^{n+1} - 2q^n + q^{n-1} = \tau^2 \operatorname{sinc}^2(\tfrac{1}{2}\tau\Omega)(-Aq^n + g^n). \tag{2.33}$$

This formulation allows a similar reformulation to a one-step method as the Verlet integrator (cf. (2.24)), see [44] for more details. It reads

$$\begin{aligned} p^{n+1/2} &= p^n + \tfrac{\tau}{2}\operatorname{sinc}^2(\tfrac{1}{2}\tau\Omega)(-Aq^n + g^n), \\ q^{n+1} &= q^n + \tau p^{n+1/2}, \\ p^{n+1} &= p^{n+1/2} + \tfrac{\tau}{2}\operatorname{sinc}^2(\tfrac{1}{2}\tau\Omega)(-Aq^{n+1} + g^{n+1}). \end{aligned} \tag{2.34}$$

With initial values

$$q^0, \qquad p^0 = \operatorname{sinc}(\tau\Omega)\dot{q}^0, \tag{2.35}$$

the one-step method is equivalent to (2.33) with (2.30). In our implementation of the scheme, we set the initial values to

$$q^0, \qquad p^0 = \dot{q}^0, \tag{2.36}$$

so that equivalence no longer holds and the approximation of the momenta is no longer exact in the linear case $g(q) \equiv const.$ However, the introduced error is small and in general $g(q)$ is not constant. The one-step method (2.34) is symmetric but not symplectic. In [36, Secction XIII.2.2] one can find conditions under which the general one-step formulation is symmetric and symplectic.

Leapfrog-Chebyshev Integrator

Recently an approach closely related to Gautschi-type integrators has been proposed by C. Carle, M. Hochbruck and A. Sturm [14]. Instead of trigonometric functions, they use the polynomials

$$P(z) = P_p(z) = 2 - \frac{2}{T_p(v)} T_p\left(v - \frac{z}{\alpha_p}\right), \qquad \alpha_p = 2\frac{T_p'(v)}{T_p(v)}, \tag{2.37}$$

where T_p denotes the pth Chebyshev polynomial of the first kind and $\nu \geq 1$ is a stabilization parameter. They propose the scheme

$$q^{n+1} - 2q^n + q^{n-1} = -P(\tau^2 A)q^n + \tau^2 g^n,$$

for semilinear problems of the form (2.27) with initial value

$$q^1 = (I - \frac{1}{2}P(\tau^2 A))q^0 + \tau P'(\tau^2 A)\dot{q}^0 + \frac{1}{2}\tau^2 g^0.$$

Since

$$P(\tau^2 A) \approx 2\left(1 - \cos(\tau\Omega)\right),$$

the scheme can be interpreted as a 'particular implementation' of Gautschi-type methods in the sense that the trigonometric functions are approximated by a fixed matrix polynomial. The stabilization parameter ν is designed to overcome linear instabilities of this method. It is important to note that this stabilization approach is different from the inner filter functions discussed in the previous section. We will compare the two approaches throughout this thesis.

C. Carle proposes multiple variations of the scheme in his dissertation [13]. Of particular interest for us is a one-step formulation where the polynomial is applied to the complete right-hand side and thus is of the same structure as the Gautschi-type integrator in the one-step formulation (2.34). It reads

$$\begin{aligned} p^{n+1/2} &= p^n + \frac{\tau}{2}\widehat{P}(\tau^2 A)(-Aq^n + g^n), \\ q^{n+1} &= q^n + \tau p^{n+1/2}, \\ p^{n+1} &= p^{n+1/2} + \frac{\tau}{2}\widehat{P}(\tau^2 A)(-Aq^{n+1} + g^{n+1}), \end{aligned} \tag{2.38}$$

where $\widehat{P}(z) = z^{-1}P(z) \approx \operatorname{sinc}^2\left(\frac{1}{2}\sqrt{z}\right)$. The one-step formulation is symmetric but not symplectic [13, Lemma 3.8].

In the next chapter we discuss the specific implementation of the Gautschi-type and the Leapfrog-Chebyshev integrator in the context of molecular dynamics in more detail.

2.4 The Fermi-Pasta-Ulam-Tsingou Problem

We close this chapter by presenting a well-known test problem for the aforementioned integrators. One of the first numerical experiments was performed by E. Fermi, J. Pasta, S. Ulam and M. Tsingou [22]. They consider a one-dimensional system of 64 particles with nonlinear forces acting between neighboring particles. The interest lied in the long-time behavior of the system with the aim to establish the rate of approach to the equipartition of energy among the various degrees of freedom. To the surprise of the authors, the energy did not evenly distribute over the modes. Instead, the energy distribution in the system showed periodic behavior. In 1992, L. Galgani, A. Giorgilli, A. Martinoli and S. Vanzini [24] proposed a modified version of the experiment. They consider $2m$ particles in a fixed line with alternating stiff, linear and soft, nonlinear springs between the particles. This system has become a useful model problem for nonlinear dynamic systems.

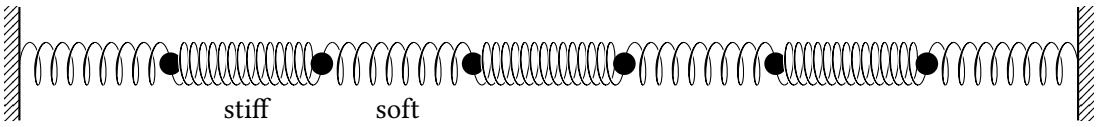


Figure 2.5. Visualization of the modified Fermi-Pasta-Ulam-Tsingou problem with six particles connected by alternating stiff and soft springs.

2.4.1 A First Modification to the Fermi-Pasta-Ulam-Tsingou Problem

We consider the displacements $q_1, \dots, q_{2m}$ ($q_0 = q_{2m+1} = 0$) of the particles with their respective velocities $p_1, \dots, p_{2m}$. The motion of the particle chain is described by a Hamiltonian system with the Hamiltonian

$$H(q,p) = \frac{1}{2}\sum_{i=1}^{m}(p_{2i-1}^2 + p_{2i}^2) + \frac{\omega^2}{4}\sum_{i=1}^{m}(q_{2i} - q_{2i-1})^2 + \sum_{i=0}^{m}(q_{2i+1} - q_{2i})^4.$$

We follow the parametrization in [38, Chapter I.5.1] and study a Hamiltonian system with respect to the scaled displacements $x_{0,i}$ and expansions $x_{1,i}$ and velocities $y_{0,i}$ and $y_{1,i}$ of the ith stiff spring

$$\begin{aligned} x_{0,i} &= \frac{1}{\sqrt{2}}(q_{2i} + q_{2i-1}), & x_{1,i} &= \frac{1}{\sqrt{2}}(q_{2i} - q_{2i-1}),\\ y_{0,i} &= \frac{1}{\sqrt{2}}(p_{2i} + p_{2i-1}), & y_{1,i} &= \frac{1}{\sqrt{2}}(p_{2i} - p_{2i-1}). \end{aligned}$$

The Hamiltonian is then given by

$$\begin{aligned} H(x,y) = \frac{1}{2}\sum_{i=1}^{m}\left(y_{0,i}^2 + y_{1,i}^2\right) + \frac{\omega^2}{2}\sum_{i=1}^{m}x_{1,i}^2 \\ + \frac{1}{4}\left((x_{0,1} - x_{1,1})^4 + \sum_{i=1}^{m-1}(x_{0,i+1} - x_{1,i+1} - x_{0,i} - x_{1,i})^4 + (x_{0,m} + x_{1,m})^4\right). \end{aligned}$$

The parameter ω determines the frequency of the linear oscillation.

While this system offers a range of interesting properties, we focus on the global error of different integration schemes over short time periods. We construct the system with $m = 3$, $\omega = 50$ and the initial data

$$x_{0,1}^0 = 1, \quad y_{0,1}^0 = 1, \quad x_{1,1}^0 = \omega^{-1}, \quad y_{1,1}^0 = 1, \tag{2.39}$$

with zero for the remaining values. A visual representation of the system can be seen in figure 2.5. We compute the error at $t = 1$ with the Verlet and the Gautschi-type integrator. To illustrate the concept of the inner filter function, we consider the cases $\phi(\xi) = 1$ and $\phi(\xi) = \operatorname{sinc}\xi$. Due to a lack of an exact solution, we compute a reference solution with the

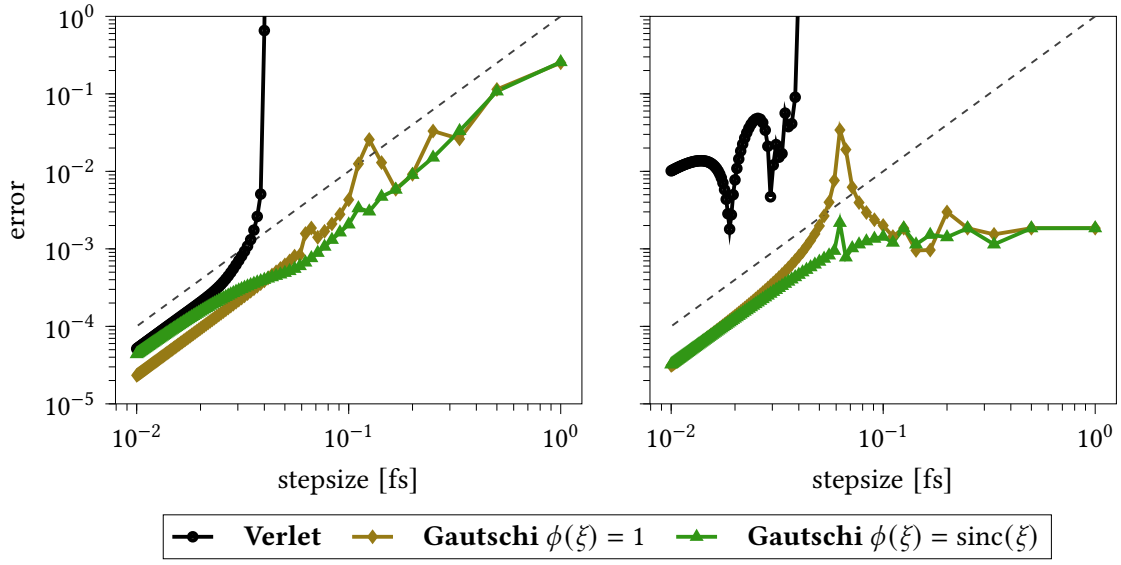


Figure 2.6. Global error at $t = 1$ for the Verlet and Gautschi-type method when applied to the system with initial data (2.39) in x_0 (left) and x_1 (right) as function of the stepsize τ and a gray dashed reference line with slope 2.

Verlet integrator and a small stepsize $\tau = 10^{-6}$. We express the error in displacement x_0 and expansion x_1 towards that reference solution as a function over the stepsize. For easier readability of the information it is custom to display the plot in double logarithmic scales. An error plot with slope m then translates to a convergence with order m. As a reference, we plot a gray dashed reference line with slope 2 since all considered schemes are of order 2. We follow this procedure for every error-plot in this work.

Figure 2.6 shows that the Gautschi-type integrator is stable for larger stepsizes than the Verlet method. We can further observe a very distinct peak for the Gautschi-type integrator with $\phi = 1$ around the stepsize $\tau = \frac{\pi}{\omega} \approx 0.06$. The peak is especially visible for the expansion parameter x_1 and is a result of resonances in the system as described in the context of the impulse method in section 2.3.2. With the inner filter function $\phi(\xi) = \mathrm{sinc}\xi$, we average over the harmonic oscillation and do not observe a peak.

As in section 2.2.1, we transfer these results to the molecular dynamics setting. Before we do so in section 3.4, we discuss in chapter 3 important details of the implementation of the semi-analytical approach in molecular dynamics.

3 Technical Details for Semi-Analytical Integration in Molecular Dynamics

In section 2.3.3 we introduced and discussed the semi-analytical approach, which makes use of the separable Hamiltonian (2.26) with a quadratic fast potential $W(q)$. However, the potentials in molecular dynamics are not quadratic with respect to the position q (cf. section 2.2). In order to apply the semi-analytical approach we therefore introduce a quadratic term that approximates the fast potentials in the separable Hamiltonian (2.10). We have briefly touched on a second challenge for the Gautschi-type integrator in section 2.3.3, namely the approximation of the matrix functions applied to some vectors. In this context we also introduced the Leapfrog-Chebyshev scheme. We discuss the implementation of both approaches in more detail. We then collect the pseudocodes of the integration schemes that are used in the rest of this work and apply them to a molecular dynamics system that is motivated by the Fermi-Pasta-Ulam-Tsingou problem presented in section 2.4.

Notation For a twice continuously differentiable function $U : \mathbb{R}^n \to \mathbb{R}$, $H_U(x)$ denotes the Hessian of U at x.

3.1 Reformulation of the Hamiltonian system

We recall the separable Hamiltonian (see (2.10) and (2.18))

$$H(q,p) = T(p) + V(q) = T(p) + W(q) + U(q), \quad \text{with} \quad T(p) = \frac{1}{2} p^T M^{-1} p, \tag{3.1}$$

where $W(q)$ contains the fast and $U(q)$ the slow potentials. With the Hamiltonian equations (2.2) this results in the second-order differential equation (cf. (2.21))

$$M\ddot{q} = -\nabla\big(W(q) + U(q)\big). \tag{3.2}$$

From now on the particles may have general positive masses $m_i > 0$, which leads to a positive definite, diagonal mass matrix M. In this section we present how we construct a semilinear problem with a real, symmetric, positive semi-definite matrix A (cf. (2.27)) of the form

$$\ddot{q} = -Aq + g(q), \tag{3.3}$$

that is equivalent to the second-order differential equation (3.2) obtained from the Hamiltonian system with Hamiltonian (3.1).

3.1.1 Construction of the Semilinear Formulation

We motivated the semi-analytical approach with the fact that the fast intramolecular potentials are quadratic functions with respect to the specific value of the potential, i.e., the bond length, angle, and dihedral angle (cf. section 2.2). In the following we assume that the potential functions are well approximated by a quadratic function of the position and illustrate the quality of the approximation in chapter 4. The quadratic function is given by a second-order *Taylor expansion* around a suitable expansion point $a \in \mathbb{R}^{dN}$.

$$\widetilde{W}^a(q) := W(a) + \nabla W(a)(q-a) + \frac{1}{2}(q-a)^T H_W(a)(q-a), \tag{3.4}$$

with the real, symmetric, positive semi-definite Hessian matrix H_W of the fast potential W evaluated at a. The approximation is exact if $W(q)$ is a quadratic function of the position q. We introduce this quadratic approximation to the fast potential by adding and subtracting it in (3.1) and obtain the Hamiltonian

$$H(q,p) = T(p) + \widetilde{W}^a(q) + \Big(W(q) - \widetilde{W}^a(q)\Big) + U(q) = T(p) + \widetilde{W}^a(q) + \widetilde{U}(q), \tag{3.5}$$

with the fast, quadratic potential $\widetilde{W}^a(q)$ and the potential

$$\widetilde{U}(q) := \Big(W(q) - \widetilde{W}^a(q)\Big) + U(q).$$

Note that the motions imposed by the potential function $\widetilde{U}(q)$ are slow if the term $W(q) - \widetilde{W}^a(q)$ is small, i.e., if the quadratic function $\widetilde{W}^a(q)$ is a good approximation of the fast potential $W(q)$.

We obtain from the Hamiltonian equations (2.2) with the Hamiltonian (3.5) the second-order differential equation

$$\begin{aligned} M\ddot{q} &= -\nabla\left(\widetilde{W}^a(q) + \widetilde{U}(q)\right) = -\nabla\left(\widetilde{W}^a(q)\right) - \nabla\left(\widetilde{U}(q)\right) \\ &= -\nabla W(a)a - H_W(a)(q-a) - \Big(\nabla\big(W(q)+U(q)\big) - \nabla W(a)a - H_W(a)(q-a)\Big) \\ &= -H_W(a)(q-a) - \Big(\nabla\big(W(q)+U(q)\big) - H_W(a)(q-a)\Big) \\ &= -H_W(a)q - \Big(\nabla V(q) - H_W(a)q\Big), \end{aligned}$$

which we reformulate to

$$M\ddot{q} = -A^a q + g^a(q), \tag{3.6a}$$

with

$$A^a := H_W(a), \qquad g^a(q) := -\nabla V(q) + A^a q. \tag{3.6b}$$

In many cases, the best choice for the expansion point is the equilibrium position q^e. For the remainder of this chapter, we therefore consider the constant expansion point $a = q^e$ and omit the superscripts. Since the exact equilibrium position might not always be available, we discuss details for the choice of a suitable expansion point in section 4.5.

A straightforward way to bring (3.6a) in the desired form (3.3) would be to multiply (3.6a) with the inverse of the mass matrix M^{-1} from the left. This would, however, result in the linear term $M^{-1}Aq$, where the matrix $M^{-1}A$ is in general not symmetric. Note that the matrix is symmetric with respect to the M-inner product $\langle\cdot,\cdot\rangle_M$ given by

$$\langle\cdot,\cdot\rangle_M = x^T M y,$$

so that one could use this inner product and induced norms to maintain a symmetric formulation. However, M is a positive definite diagonal matrix so that taking the inverse poses no problem. We therefore follow [13, 59] and consider the following approach.

3.1.2 Introduction of a General Mass Matrix

We reformulate (3.6a) using the modified coordinates $z(t) := M^{1/2}q(t)$ and get

$$M^{1/2}\ddot{z} = -AM^{-1/2}z + g\big(M^{-1/2}z\big).$$

Note that M is a diagonal matrix so that the evaluation of the root and inverse of M is a trivial task. By multiplying with $M^{-1/2}$ from the left we obtain

$$\begin{aligned} \ddot{z} &= -M^{-1/2}AM^{-1/2}z + M^{-1/2}g\big(M^{-1/2}z\big) \\ &= -A_M z + g_M(z), \end{aligned}$$

with

$$A_M := M^{-1/2}AM^{-1/2} \qquad g_M(\zeta) := M^{-1/2}g\big(M^{-1/2}\zeta\big).$$

The matrix A_M is symmetric for an arbitrary, symmetric positive definite matrix M so that we have constructed a semilinear differential equation (cf. (3.3)) that is equivalent to the original Hamiltonian system with Hamiltonian (3.1).

With $z^n = M^{1/2}q^n$ we get the two-step formulation of the Gautschi-type integrator (cf. (2.33))

$$z^{n+1} - 2z^n + z^{n-1} = \tau^2 \operatorname{sinc}^2(\tfrac{1}{2}\tau\Omega_M)\left(-A_M z^n + g_M(\tilde{z}^n)\right), \tag{3.7}$$

where $\Omega_M = A_M^{1/2}$ and $\tilde{z}^n$ is an averaged, modified position. We elaborate on the details of how the averaged position $\tilde{z}^n$ is obtained.

3.1.3 Discussion of the Inner Filter Function

As discussed in section 2.3.3 we filter out resonant frequencies in the Gautschi-type integrator by evaluating the nonlinearity at an averaged position. The approach presented in (2.32) applies an inner filter function ϕ to the value at which the nonlinearity is evaluated. The function ϕ satisfies $\phi(0) = 1$ and $\phi(k\pi) = 0$ for $k \in \mathbb{N}$. In the context of the Fermi-Pasta-Ulam-Tsingou problem (see section 2.4) this means that the scaled displacements and expansions $x_{0,i}, x_{1,i}$ of the springs are averaged around their equilibrium value zero. In order to get a suitable time average of the particle positions in molecular dynamics we consider the following approach.

We apply the filter function to the displacement from its equilibrium position, i.e., to $(q^n - q^e)$ and add the averaged value to the equilibrium position q^e. The averaged position $\tilde{q}^n$ is thus given by

$$\tilde{q}^n = q^e + \phi(\tau\Omega)(q^n - q^e). \tag{3.8}$$

The averaged position is equal to the equilibrium position q^e whenever ϕ is zero, and it is equal to the original position q^n when ϕ is one. The average of the modified position is directly given by $\tilde{z}^n = M^{1/2}\tilde{q}^n$.

3.1.4 The Complete Integration Scheme

We combine the results of the three previous sections (sections 3.1.1 to 3.1.3) to formulate the final integration scheme.

We first take a closer look at the term

$$-A_M z^n + g_M(\tilde{z}^n) = -A_M M^{1/2} q^n + M^{-1/2} g(\tilde{q}^n)$$

in (3.7), with the averaged position $\tilde{q}^n$ (cf. (3.8)). With (3.6b) it holds that

$$\begin{aligned} -A_M M^{1/2} q^n + M^{-1/2} g(\tilde{q}^n) &= -A_M M^{1/2} q^n - M^{-1/2}\nabla V(\tilde{q}^n) + M^{-1/2} A\tilde{q}^n \\ &= -A_M M^{1/2} q^n - M^{-1/2}\nabla V(\tilde{q}^n) + A_M M^{1/2}\tilde{q}^n \\ &= A_M M^{1/2}(\tilde{q}^n - q^n) - M^{-1/2}\nabla V(\tilde{q}^n). \end{aligned}$$

Note that for the case $\phi \equiv 1$, i.e., $\tilde{q}^n = q^n$, the linear term with A_M vanishes.

We use this integrator in the context of molecular dynamics simulations. During these simulations one is often interested in the original position q^n in each step and a suitable formulation is obtained from (3.7) by

$$\begin{aligned} q^{n+1} &= M^{-1/2} z^{n+1} \\ &= 2M^{-1/2} z^n - M^{-1/2} z^{n-1} + M^{-1/2} \tau^2 \operatorname{sinc}^2(\tfrac{1}{2}\tau\Omega_M)\,(-A_M z^n + g_M(\tilde{z}^n)) \\ &= 2q^n - q^{n-1} + \tau^2 M^{-1/2} \operatorname{sinc}^2(\tfrac{1}{2}\tau\Omega_M) \left(-A_M M^{1/2} q^n + M^{-1/2} g(\tilde{q}^n)\right) \\ &= 2q^n - q^{n-1} + \tau^2 M^{-1/2} \operatorname{sinc}^2(\tfrac{1}{2}\tau\Omega_M) \left(A_M M^{1/2} (\tilde{q}^n - q^n) - M^{-1/2} \nabla V(\tilde{q}^n)\right). \end{aligned}$$

From this formulation we derive the one-step formulation as before (cf. (2.34)) and get

$$\begin{aligned} p^{n+1/2} &= p^n + \tfrac{\tau}{2} M^{-1/2} \operatorname{sinc}^2(\tfrac{1}{2}\tau\Omega_M) \left(A_M M^{1/2} (\tilde{q}^n - q^n) - M^{-1/2} \nabla V(\tilde{q}^n)\right), \\ q^{n+1} &= q^n + \tau p^{n+1/2}, \\ p^{n+1} &= p^{n+1/2} + \tfrac{\tau}{2} M^{-1/2} \operatorname{sinc}^2(\tfrac{1}{2}\tau\Omega_M) \left(A_M M^{1/2} (\tilde{q}^{n+1} - q^{n+1}) - M^{-1/2} \nabla V(\tilde{q}^{n+1})\right), \end{aligned} \tag{3.9}$$

that gives direct access to the position q^n and momentum p^n at every step.

3.2 Details on Efficient Matrix Function Evaluation

In each step of the integrator defined in (3.9), the products of the matrix functions $\phi(\tau\Omega_M)$ and $\operatorname{sinc}^2(\frac{\tau}{2}\Omega_M)$ with some vectors have to be computed. The matrix Ω_M is of dimension $dN \times dN$, where d is the space dimension and N the number of particles in the system. Direct evaluation of the matrix function is very expensive for large systems. Instead of the full matrix function evaluation, we only consider the action of the matrix function on a given vector. We use the fact that the $d \times d$ submatrix Ω_{ij} of Ω_M corresponding to the particles with index i and j has only entries if $i = j$ or if the particles i and j have fast interactions that are approximated by $\widetilde{W}_{q^e}$. We have discussed in section 2.2 that the fast potentials summed in $W(q)$ are short ranged potentials. As a consequence, Ω_M is sparse and matrix-vector products with Ω_M are cheap (see appendix A.1). We present details of the two approaches discussed in section 2.3.3 that are based on matrix vector products and thus make use of this property.

3.2.1 Matrix Function Approximation with Krylov Subspaces

Our implementation of the Gautschi-type integrator approximates the product of a matrix function $f(A) \in \mathbb{C}^{n\times n}$ with a vector $b \in \mathbb{C}^n$ with polynomial Krylov subspaces. For more details we refer to chapter 13 of 'Functions of Matrices: Theory and Computation' by N. J. Higham [41] as well as to Y. Saad [68] and E. Gallopoulos and Y. Saad [25].

The polynomial Krylov subspace $\mathcal{K}_m(A, b)$ for the matrix $A \in \mathbb{C}^{n\times n}$ and the vector $b \in \mathbb{C}^n$ is defined as

$$\mathcal{K}_m(A, b) := \operatorname{span}\left\{b, Ab, \ldots, A^{m-1}b\right\} = \{p(A)b \mid p \text{ is a polynomial of degree} \leq m-1\}.$$

The Krylov subspaces are nested up to index $L = \deg(p_{A,b})$, where $p_{A,b}$ is the *minimal polynomial* of A with respect to b (see appendix A.2), i.e.,

$$\mathcal{K}_1(A, b) \subsetneq \mathcal{K}_2(A, b) \subsetneq \cdots \subsetneq \mathcal{K}_L(A, b) = \mathcal{K}_{L+1}(A, b). \tag{3.10}$$

Since the Lth Krylov subspace is A-invariant, the index L is referred to as the invariance index. The product $f(A)b$ is an element of $\mathcal{K}_L(A, b)$ (see appendix A.2).

The Arnoldi process iteratively computes an orthonormal basis $V_m = \left(v_1 | \ldots | v_m\right) \in \mathbb{C}^{n\times m}$ of the Krylov subspace $\mathcal{K}_m(A, b)$ and gives rise to the *Hessenberg reduction* $V_m^H A V_m = H_m$, where $H_m \in \mathbb{C}^{m\times m}$ is an upper *Hessenberg* matrix.

With the Hessenberg reduction after m iterations of the Arnoldi process an approximation to $f(A)b$ is obtained by the Krylov subspace approximation

$$f(A)b \approx \|b\| V_m f(H_m) e_1. \tag{3.11}$$

Throughout this work we only consider matrix functions of real, symmetric matrices. In this case the Arnoldi algorithm is equivalent to the symmetric Lanczos algorithm (see e.g., [69, Section 6.3]) that uses a three-term recurrence. The symmetric Lanczos algorithm is given in algorithm 1.

Algorithm 1 Symmetric Lanczos Algorithm

Input Symmetric matrix $A \in \mathbb{R}^{n\times n}$, vector $b \in \mathbb{R}^n$, $b \neq 0$

$\beta_1 \leftarrow 0, v_0 \leftarrow 0$
$v_1 \leftarrow b/\|b\|$
for m = 1,2,... **do**
  $\alpha_m \leftarrow v_m^T A v_m$
  $\tilde{v}_{m+1} \leftarrow A v_m - \beta_m v_{m-1} - \alpha_m v_m$
  $\beta_{m+1} \leftarrow \|\tilde{v}_{m+1}\|$
  if $\beta_{m+1} = 0$ **then**
    STOP
  end if
  $v_{m+1} \leftarrow \tilde{v}_{m+1}/\beta_{m+1}$
end for

Output Hessenberg matrix H_L, Orthonormal basis V_L

Applying the function to the small, real and symmetric tridiagonal matrix H_m in (3.11) is cheap in comparison to the application to the large matrix A. We use the eigendecomposition $H_m = Q\Lambda Q^{-1}$ of the symmetric tridiagonal matrix H_m with the diagonal matrix Λ and compute $f(H_m) = Q f(\Lambda) Q^{-1}$. The decomposition is obtained by the QR algorithm with implicit shifts as described by J. H. Wilkinson et al. in [9]. We use the optimized LAPACK

[1] routine '*dsteqr*' that is based on the algorithm proposed by A. Dubrulle [19]. For more details on the topic of the QR decomposition and related iterative algorithms, we refer to chapter 4 of 'The Matrix Eigenvalue Problem' by D. S. Watkins [85].

The Lanczos iteration terminates when the invariance index L (cf. (3.10)) has been reached, i.e., when the Krylov space is A-invariant and thus $\beta_{m+1} = \|\tilde{v}_{m+1}\| = 0$ in line 7 of algorithm 1. In most cases we have $L = n$, but it is not necessary to compute that many iterations, since the convergence to the exact matrix function evaluation, is superlinear after a certain number of iterations. This number depends on the field of values of the matrix. In this work we consider methods that involve the evaluation of trigonometric matrix functions of a real symmetric positive semi-definite matrix $\tau\Omega$ (cf (2.29) and (2.30)). The following Lemma gives a relation between the error in the Lanczos approximation for the sine and cosine of a real symmetric matrix applied to some vector b and the number of Lanczos iterations.

Lemma 2. *Let A be a real symmetric matrix with real eigenvalues in an interval of length 4ρ and b a vector with $\|b\| = 1$. Then the error in the Lanczos approximation of $f(\tau A)b$, i.e., $\varepsilon_m^c := \|f(\tau A)b - V_m f(\tau H_m)e_1\|$, for $f(z) = \cos(z)$ and $f(z) = \sin(z)$, is bounded in the following way:*

$$\varepsilon_m \leq 12e^{-(\rho\tau)^2/m}\left(\frac{e\rho\tau}{m}\right)^m, \qquad m \geq 2\rho\tau.$$

Proof. We have

$$\cos(\tau A) = \frac{1}{2}\left(e^{i\tau A} + e^{-i\tau A}\right),$$

so that

$$\begin{aligned}
\varepsilon_m &= \|\cos(\tau A)b - V_m \cos(\tau H_m)e_1\| \\
&= \left\|\frac{1}{2}\left(e^{i\tau A} + e^{-i\tau A}\right)b - V_m\left(e^{i\tau H_m} + e^{-i\tau H_m}\right)e_1\right\| \\
&= \frac{1}{2}\left\|\left(e^{i\tau A}b - V_m e^{i\tau H_m}e_1\right) + \left(e^{-i\tau A}b - V_m e^{-i\tau H_m}e_1\right)\right\| \\
&\leq \frac{1}{2}\left(\left\|e^{i\tau A}b - V_m e^{i\tau H_m}e_1\right\| + \left\|e^{-i\tau A}b - V_m e^{-i\tau H_m}e_1\right\|\right). \qquad (3.12)
\end{aligned}$$

The matrices $\pm iA$ are skew-Hermitian matrices with eigenvalues in an interval on the imaginary axis of length 4ρ. Applying Theorem 4 in [42] to the error in the Lanczos approximations of $e^{\pm i\tau A}b$ leads to the desired error bound.

Analogously we get an error bound for the error in the Arnoldi approximation of $\sin(\tau A)b$. □

This means that for a general real symmetric matrix one gets superlinear convergence to the exact trigonometric matrix function after $2\rho\tau$ steps. In a recent preprint, M. Botchev, L. Knizhnerman, and M. Schweitzer [8] consider related approximations and provide convergence results for some special cases. We study a suitable choice for the number of Arnoldi iterations in the context of molecular dynamics more closely in section 6.3.

3.2.2 Matrix Function Approximation with Chebyshev Polynomials

In section 2.3.3 we also introduced the Leapfrog-Chebyshev integrator where the trigonometric functions are approximated by polynomials. What makes the scheme interesting from a computational perspective is the fact that the polynomials $\widehat{P}_p$ satisfy a three-term recursion.

Corollary 1 (Carle [13, Corollary 4.16]). *The polynomials* $\widehat{P}_{k,p}(z) : \mathbb{R} \to \mathbb{R}$,

$$\widehat{P}_{k,p}(z) = \frac{1}{z}\left(2 - \frac{2}{T_k(\nu)} T_k\left(\nu - \tfrac{z}{\alpha_p}\right)\right)$$

satisfy the linear recurrence relation

$$\begin{aligned}
\widehat{P}_{0,p}(z) &= 0, \\
\widehat{P}_{1,p}(z) &= \frac{2}{\alpha_p \nu}, \\
T_k(\nu)\widehat{P}_{k,p}(z) &= 2\nu T_{k-1}(\nu)\widehat{P}_{k-1,p}(z) \\
&\quad + \frac{2}{\alpha_p} T_{k-1}(\nu)\left(2 - z\widehat{P}_{k-1,p}(z)\right) - T_{k-2}(\nu)\widehat{P}_{k-2,p}(z),
\end{aligned}$$

for $k = 2, \ldots, p$.

It holds by definition that $\widehat{P}_p = \widehat{P}_{p,p}$, where $\widehat{P}_p(z) = \frac{P_p(z)}{z}$ with P_p defined in (2.37). This three-term recursion allows for an efficient and straightforward way to evaluate $\widehat{P}(\tau^2 A)\nabla V(q)$ in (2.38). We present the pseudocode of the recursion when applied to a matrix $A \in \mathbb{R}^{n\times n}$ and multiplied with a vector $b \in \mathbb{R}^n$ in algorithm 2, where $w_k = \widehat{P}_{k,p}(\tau^2 A)b$.

Algorithm 2 Chebyshev Polynomials Recursion [13, Algorithm 4.2]

Input $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, $p \in \mathbb{N}$, $\nu, \tau, \alpha_p, T_0(\nu), \ldots, T_p(\nu) \in \mathbb{R}$,

1: $w_0 \leftarrow 0, w_1 \leftarrow \frac{2}{\alpha_p \nu} b$
2: **for** $k = 2, \ldots, p$ **do**
3: $\quad w_k \leftarrow 2\frac{T_{k-1}(\nu)}{T_k(\nu)}\left(\nu w_{k-1} + \frac{1}{\alpha_p}\left(2b - \tau^2 A w_{k-1}\right)\right) - \frac{T_{k-2}(\nu)}{T_k(\nu)} w_{k-2}$
4: **end for**

Output $w_p = \widehat{P}_p(\tau^2 A)b$

The parameters α_p as well as the evaluations of the Chebyshev polynomials $T_0(\nu), \ldots, T_p(\nu)$ are computed once at the initialization of the integrator. We need $p - 1$ matrix-vector products for the evaluation of the polynomial $\widehat{P}_p$ with degree $p - 1$. Again, we only use

the matrix A in the context of matrix-vector products. In the following we use a specific choice of the stabilization parameter proposed in [13, (4.3)] and consider

$$\nu = \nu_{p,\eta} = 1 + \frac{\eta^2}{2p^2}, \tag{3.13}$$

with a second stabilization parameter $\eta \geq 0$. Note that for $\eta = 0$ the stabilization parameter ν is equal to one. In this case the polynomials are unstabilized. For an in-depth discussion of the stabilization parameters and the Leapfrog-Chebyshev scheme in general we refer to [13, Chapter 3 & 4].

3.3 Collection of the Considered Integrators

We collect the previous considerations and present the formulation of the integrators that we consider in the remainder of this work. The integrators approximate the solution of Hamiltonian systems with Hamiltonian (3.1) with initial values $q^0, p^0 \in \mathbb{R}^{dN}$ at $t_K = K \cdot \tau$, where K is the number of integration steps.

As a reference for the semi-analytical integrators, we use the Verlet integrator in the one-step formulation that is also known as the Velocity-Verlet method (cf. (2.24)). The algorithm is given in algorithm 3.

Algorithm 3 Verlet

Input $M^{-1} \in \mathbb{R}^{dN \times dN}$, $q^0, p^0 \in \mathbb{R}^{dN}$, $\tau \in \mathbb{R}$, $K \in \mathbb{N}$

$g^0 \leftarrow -M^{-1}\nabla V(q^0)$
for $n = 0, \dots, K-1$ **do**
  $p^{n+1/2} \leftarrow p^n + \frac{\tau}{2} g^n$
  $q^{n+1} \leftarrow q^n + \tau p^{n+1/2}$
  $g^{n+1} \leftarrow -M^{-1}\nabla V(q^{n+1})$
  $p^{n+1} \leftarrow p^{n+1/2} + \frac{\tau}{2} g^{n+1}$
end for

Output $q^K, p^K \in \mathbb{R}^{dN}$

From the formulation in algorithm 3 it is clear that we only need one evaluation of the force vector per step. In the following we write the integrators following the semi-analytical approach in a similar form.

We study the Gautschi-type integrator without an inner filter function, i.e., with $\phi \equiv 1$, and with the inner filter function $\phi(\zeta) = \operatorname{sinc}(\zeta)$, see (3.9). The matrix functions are evaluated according to the discussion in section 3.2.1 with the dimension of the Krylov subspace given by $m_{\max}$. The choice of $m_{\max}$ will be discussed in section 6.3. We refer to the two one-step formulations as *unfiltered* and *filtered* respectively and present the algorithms in algorithms 4 and 5.

Algorithm 4 Gautschi-type *unfiltered*

Input $H_W(q^e), M^{-1/2} \in \mathbb{R}^{dN\times dN}$, $q^0, p^0 \in \mathbb{R}^{dN}$, $\tau \in \mathbb{R}$, $K, m_{\max} \in \mathbb{N}$

$A_M \leftarrow M^{-1/2} H_W(q^e) M^{-1/2}$

$g^0 \leftarrow -M^{-1/2}\operatorname{sinc}^2(\frac{1}{2}\tau A_M^{1/2}) M^{-1/2}\nabla V(q^0)$

for $n = 0, \dots, K-1$ **do**

$\quad p^{n+1/2} \leftarrow p^n + \frac{\tau}{2} g^n$

$\quad q^{n+1} \leftarrow q^n + \tau p^{n+1/2}$

$\quad g^{n+1} \leftarrow -M^{-1/2}\operatorname{sinc}^2(\frac{1}{2}\tau A_M^{1/2}) M^{-1/2}\nabla V(q^{n+1})$

$\quad p^{n+1} \leftarrow p^{n+1/2} + \frac{\tau}{2} g^{n+1}$

end for

Output $q^K, p^K \in \mathbb{R}^{dN}$

For the filtered variant we need in addition to the initial values $q^0, p^0 \in \mathbb{R}^{dN}$ access to a suitable equilibrium position $q^e \in \mathbb{R}^{dN}$. Note that in general, this position may not be exact but rather an approximation to the equilibrium. The pseudocode of the filtered Gautschi-type integrator is given in algorithm 5.

Algorithm 5 Gautschi-type *filtered*

Input $H_W(q^e), M^{-1/2} \in \mathbb{R}^{dN\times dN}$, $q^0, p^0, q^e \in \mathbb{R}^{dN}$, $\tau \in \mathbb{R}$, $K, m_{\max} \in \mathbb{N}$

$A_M \leftarrow M^{-1/2} H_W(q^e) M^{-1/2}$

$\tilde{q}^0 \leftarrow q^e + \operatorname{sinc}(\tau A_M^{1/2})(q^0 - q^e)$

$g^0 \leftarrow M^{-1/2}\operatorname{sinc}^2(\frac{1}{2}\tau A_M^{1/2}) \left(A_M M^{1/2}(\tilde{q}^0 - q^0) - M^{-1/2}\nabla V(\tilde{q}^0)\right)$

for $n = 0, \dots, K-1$ **do**

$\quad p^{n+1/2} \leftarrow p^n + \frac{\tau}{2} g^n$

$\quad q^{n+1} \leftarrow q^n + \tau p^{n+1/2}$

$\quad \tilde{q}^{n+1} \leftarrow q^e + \operatorname{sinc}(\tau A_M^{1/2})(q^{n+1} - q^e)$

$\quad g^{n+1} \leftarrow M^{-1/2}\operatorname{sinc}^2(\frac{1}{2}\tau A_M^{1/2}) \left(A_M M^{1/2}(\tilde{q}^{n+1} - q^{n+1}) - M^{-1/2}\nabla V(\tilde{q}^{n+1})\right)$

$\quad p^{n+1} \leftarrow p^{n+1/2} + \frac{\tau}{2} g^{n+1}$

end for

Output $q^K, p^K \in \mathbb{R}^{dN}$

The matrix functions in the unfiltered and filtered Gautschi-type integrators are approximated with the Krylov approximation (3.11) after the basis of the Krylov subspace and its corresponding Hessenberg matrix have been computed with $m_{\max}$ steps of the symmetric Lanczos process (cf. algorithm 1).

The fourth integrator that we study is the Leapfrog-Chebyshev integrator with polynomial degree p and stabilization parameter $\nu_{p,\eta}$ (cf. (3.13)). The necessary equivalent reformulations of the original equation discussed for the Gautschi-type integrator in

sections 3.1.1 to 3.1.4 apply analogously to the Leapfrog-Chebyshev integrator. The algorithm is given in algorithm 6. We follow the nomenclature of C. Carle in [13] and call the Leapfrog-Chebyshev integrator with $\eta = 0$ unstabilized and with $\eta \geq 0$ stabilized.

Algorithm 6 Leapfrog-Chebyshev

Input $H_W(q^e), M^{-1/2} \in \mathbb{R}^{dN \times dN}$, $q^0, p^0, q^e \in \mathbb{R}^{dN}$, $\tau, \eta \in \mathbb{R}$, $K, p, \in \mathbb{N}$

1: $A_M \leftarrow M^{-1/2} H_W(q^e) M^{-1/2}$
2: $\nu_{p,\eta} \leftarrow 1 + \frac{\eta^2}{2p^2}$
3: Calculate parameters $T_0(\nu_{p,\eta}), \ldots, T_p(\nu_{p,\eta}), T_0'(\nu_{p,\eta}), \ldots, T_p'(\nu_{p,\eta})$
4: $\alpha_p \leftarrow 2\frac{T_p'(\nu_{p,\eta})}{T_p(\nu_{p,\eta})}$
5: $g^0 \leftarrow -M^{-1/2} \widehat{P}_p(\tau^2 A_M) M^{-1/2} \nabla V(q^0)$
6: **for** $n = 0, \ldots, K-1$ **do**
7: $\quad p^{n+1/2} \leftarrow p^n + \frac{\tau}{2} g^n$
8: $\quad q^{n+1} \leftarrow q^n + \tau p^{n+1/2}$
9: $\quad g^{n+1} \leftarrow -M^{-1/2} \widehat{P}_p(\tau^2 A_M) M^{-1/2} \nabla V(q^{n+1})$
10: $\quad p^{n+1} \leftarrow p^{n+1/2} + \frac{\tau}{2} g^{n+1}$
11: **end for**

Output $q^K, p^K \in \mathbb{R}^{dN}$

We evaluate the polynomials in line 5 and 9 with the recursion given in algorithm 2.

When comparing the four algorithms, algorithms 3 to 6, one notices that the propagation from q^n, p^n to q^{n+1}, p^{n+1} follows the same steps where only the calculation of the term g^{n+1} differs. This term is computed once per iteration. We will see in chapter 6 that we can use this fact to our advantage for the efficient implementation of the semi-analytical integrators in existing software packages.

3.4 From Fermi-Pasta-Ulam-Tsingou to Molecular Dynamics

As mentioned in the end of section 2.4, we construct and study a molecular dynamic system that is motivated by the Fermi-Pasta-Ulam-Tsingou problem. We do this by adding a third bonded particle pair to the system presented in section 2.2.1 as well as two single particles with fixed position on the left and right end of the chain. This means that we consider a total of eight particles in one dimension. We illustrate the system in figure 3.1. The bonded particles interact through the bond potential V_B (cf. (2.14)) and the non-bonded particles through the Lennard-Jones potential V_{LJ} (cf. (2.12)). The parameters of the potentials are the same as before and read

$$k_B = 1200, \qquad \bar{\rho}_B = 1,$$
$$\epsilon = 0.5, \qquad \sigma = 1/\sqrt[6]{2}.$$

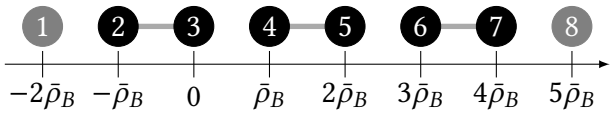


Figure 3.1. Equilibrium state of the eight particle system.

The units of these parameters follow the *real* unit style of LAMMPS. This means that the distances are given in Angstrom [Å] ($1\,\text{Å} = 10^{-10}\,\text{m}$) and the time in femtoseconds [fs] ($1\,\text{fs} = 10^{-15}\,\text{s}$). In this setting, one time tick is equal to 48.888 21 fs ($1\,\text{fs} \approx 0.02$ time units).

We set the mass of all particles to one, enumerate the particle from left to right and set the set of indices of particle that share a bond to $\mathcal{B} = \{(2,3),(4,5),(6,7)\}$. Similar to the soft, nonlinear springs in the Fermi-Pasta-Ulam-Tsingou problem, the Lennard-Jones potential is only applied to directly neighboring particles that do not share a bond. This means the set of indices of particles that interact with the Lennard-Jones potential is given by $\mathcal{L} = \{(1,2),(3,4),(5,6),(7,8)\}$.

In order to apply the semi-analytical integrators we need to compute the Hessian matrix of the bond potential. We present the detailed calculations for the general three-dimensional case in section 4.1. In one dimension the Hessian of the bond potential (2.20) evaluated at the equilibrium position is given by

$$H_{V_B}(q^e) = k_B \begin{bmatrix} 0 & & & \cdots & & & & 0 \\ & 1 & -1 & & & & & \\ & -1 & 1 & & & & & \\ \vdots & & & 1 & -1 & & & \vdots \\ & & & -1 & 1 & & & \\ & & & & & 1 & -1 & \\ & & & & & -1 & 1 & \\ 0 & & & \cdots & & & & 0 \end{bmatrix}.$$

With the initial data

$$q_4^0 = q_4^e - 0.1, \quad q_5^0 = q_5^e + 0.1, \qquad q_i^0 = q_i^e \quad \forall i \notin \{4,5\}, \tag{3.14}$$

and starting momentum $p = 0 \in \mathbb{R}^8$ we compute the error of the integrators collected in section 3.3 at $t = 50\,\text{fs}$ and plot it as a function of the stepsize in figure 3.2. We consider the Leapfrog-Chebyshev integrator with $p = 6$ and stabilization parameters $\eta = 0$, i.e., unstabilized, and with $\eta = 1$.

We can see on the top of figure 3.2 that we can recreate the core properties of the Gautschi-type integrator as demonstrated with the Fermi-Pasta-Ulam-Tsingou problem in figure 2.5 before. However, the resonances in this example are much more dominant. We do not only experience a small peak but the simulation crashes completely at multiple stepsizes without the inner filter function. With the inner filter function present, this example shows promising results. We also consider the unstabilized and stabilized Leapfrog-Chebyshev

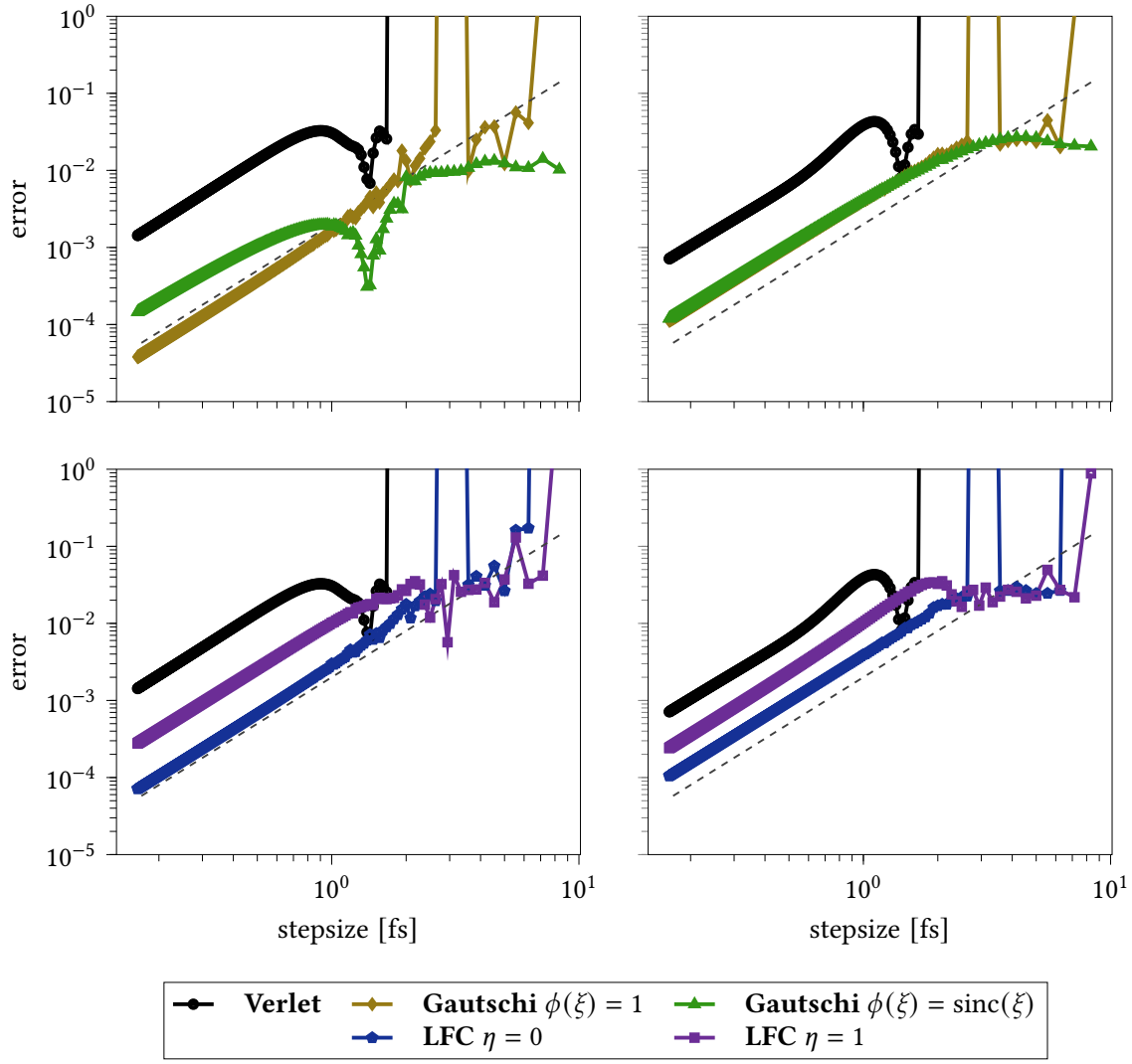


Figure 3.2. Global error at $t = 50$ fs for the Verlet and Gautschi-type method (top) and Leapfrog-Chebyshev method (bottom) when applied to the system with initial data (3.14) in q (left) and p (right) as function of the stepsize τ in [fs] and a gray dashed reference line with slope 2.

integrator. The plots on the bottom of figure 3.2 show that the unstabilized version has resonances for the same stepsizes as the unfiltered Gautschi-type integrator. In this example the Leapfrog-Chebyshev method with stabilization parameter $\eta = 1$ stays stable for these stepsizes.
In order to explore these results in the context of larger systems we present the expressions for the Hessian matrix of the bond, angle, and torsional potential in chapter 4.

4 | Details for the Approximation of the Intramolecular Potentials

In the previous chapter, in section 3.1.1, we motivated the semilinear equation (3.6) with linear term $A^a q$, where $A^a = H_W(a)$ is the Hessian of the fast potential $W(q)$ evaluated at the expansion point a of the Taylor expansion (cf. (3.4)). We therefore need access to the Hessian H_W of the fast potential. We consider the three intramolecular potentials for the bond, angle and dihedral angle discussed in section 2.2. Computing the Hessian of these potential, is a straightforward but tedious task. We briefly summarize the steps for the first derivatives of the potential functions, which are needed for the force evaluations, before calculating the second derivatives. We did not find any references to the second derivates of these potentials in the literature. Especially for the angle and torsional potential, the calculation are quite technical. As mentioned in section 2.2, the potentials are harmonic with respect to the bond, angle, and dihedral angle respectively, but not with respect to the particle positions.

To get an idea of the quality of the linear approximation to the non-linear potentials, we construct simple systems, where the distinct effects of the bond, angle, and torsional potential can be observed. All particles in this chapter have the mass $m = 1$, and we have access to the equilibrium position q^e which we use as the expansion point of the Taylor expansion. We compare the fast force $-\nabla W(q)$ in these systems to the linear approximation $-A^a(q - q^e)$. To get a first impression of the behavior of the semi-analytical approach with this linearization of the fast force, we apply the unfiltered Gautschi-type integrator (cf. algorithm 4) and compare the error for different stepsizes with the error of the Verlet integrator. Since we do not have access to an exact solution, we consider the error in relation to a reference solution, which is computed with the Verlet integrator with stepsize $\tau = 10^{-4}$ fs. We will further see that in some cases it is necessary to reevaluate the Hessian, i.e., using different expansion points, over time. We discuss the nature of those cases and elaborate how we implement the reevaluation process.

Notation We denote by $x \otimes y = xy^T$ the dyadic product of two vectors $x, y \in \mathbb{R}^n$. For a twice continuously differentiable function $U : \mathbb{R}^{dN} \to \mathbb{R}$ and a vector $x = (x_1, \dots x_N)^T \in \mathbb{R}^{dN}$ with $x_k = (x_{k,1}, \dots, x_{k,d})^T \in \mathbb{R}^d$, we denote by

$\nabla^2_{x_i x_j} U = \nabla_{x_i}\left(\nabla_{x_j} U\right)$ the $d \times d$ submatrix of the Hessian matrix $H_U(x)$ corresponding to the vectors x_i and x_j, i.e.,

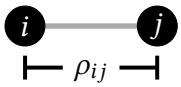
$$\nabla^2_{x_i x_j} U = \begin{bmatrix} \frac{\partial^2}{\partial x_{i,1} \partial x_{j,1}} & \cdots & \frac{\partial^2}{\partial x_{i,d} \partial x_{j,1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_{i,1} \partial x_{j,d}} & \cdots & \frac{\partial^2}{\partial x_{i,d} \partial x_{j,d}} \end{bmatrix}.$$

Note that we consider the Hessian as the Jacobian of the Gradient.

4.1 Derivatives of the Bond Potential

With the distance between two particles i and j, defined as in definition 3, i.e.,

$$\rho_{ij} = \|r_{ij}\| = \|q_j - q_i\|,$$



the harmonic bond potential reads (cf. (2.14))

$$V_B(\rho_{ij}) = \frac{k_B}{2}(\rho_{ij} - \bar{\rho}_B)^2. \tag{4.1}$$

Lemma 3. *The partial derivates of the bond potential (4.1) are*

$$\nabla_{q_i} V_B(\rho_{ij}) = -k_B \frac{\rho_{ij} - \bar{\rho}_B}{\rho_{ij}} r_{ij} \tag{4.2a}$$

$$\nabla_{q_j} V_B(\rho_{ij}) = \quad k_B \frac{\rho_{ij} - \bar{\rho}_B}{\rho_{ij}} r_{ij}. \tag{4.2b}$$

Proof. Using

$$\nabla_{q_i} \rho_{ij} = \nabla_{q_i} \|r_{ij}\| = -\frac{1}{\|r_{ij}\|} r_{ij} = -\frac{1}{\rho_{ij}} r_{ij}$$

and the chain rule, the partial derivative of $V_B(\rho_{ij})$ with respect to q_i is given by

$$\nabla_{q_i} V_B(\rho_{ij}) = \frac{dV_B}{d\rho_{ij}} \nabla_{q_i} \rho_{ij} = k_B(\rho_{ij} - \bar{\rho}_B)\left(-\frac{1}{\rho_{ij}} r_{ij}\right) = -k_B \frac{\rho_{ij} - \bar{\rho}_B}{\rho_{ij}} r_{ij}.$$

Similarly, we get the partial derivative of $V_B(\rho_{ij})$ with respect to q_j. □

Note that $-\nabla_{q_i} V_B(\rho_{ij})$ is the force applied to the particle i due to the bond with particle j. The relation $\nabla_{q_i} V_B(\rho_{ij}) = -\nabla_{q_j} V_B(\rho_{ij})$, i.e., the fact that the particles apply forces to one another that are equal in magnitude and opposite in direction corresponds with Newton's third law.

Lemma 4. *The Hessian matrix of the bond potential (4.1) reads*

$$H_{V_B}(\rho_{ij}) = \begin{bmatrix} h_{ii} & -h_{ii} \\ -h_{ii} & h_{ii} \end{bmatrix},$$

with

$$h_{ii} = \nabla^2_{q_i q_i} V_B(\rho_{ij}) = k_B \left(\frac{\bar{\rho}_B}{\rho_{ij}^3} r_{ij} \otimes r_{ij} + \frac{\rho_{ij} - \bar{\rho}_B}{\rho_{ij}} \mathbb{I} \right).$$

Proof. With the first partial derivative (4.2a) we obtain the second partial derivative of the bond potential by

$$\begin{aligned} \nabla^2_{q_i q_i} V_B(\rho_{ij}) = \nabla_{q_i} \left(\nabla_{q_i} V_B(\rho_{ij}) \right) &= -k_B \left(r_{ij} \otimes \nabla_{q_i} \left(\frac{\rho_{ij} - \bar{\rho}_B}{\rho_{ij}} \right) + \frac{\rho_{ij} - \bar{\rho}_B}{\rho_{ij}} \nabla_{q_i} r_{ij} \right) \\ &= -k_B \left(-r_{ij} \otimes \nabla_{q_i} \left(\frac{\bar{\rho}_B}{\rho_{ij}} \right) - \frac{\rho_{ij} - \bar{\rho}_B}{\rho_{ij}} \mathbb{I} \right) \\ &= k_B \left(\frac{\bar{\rho}_B}{\rho_{ij}^3} r_{ij} \otimes r_{ij} + \frac{\rho_{ij} - \bar{\rho}_B}{\rho_{ij}} \mathbb{I} \right). \end{aligned}$$

To obtain the full Hessian matrix of the bond potential for a single bond, we use Newton's third law and the symmetry of the partial derivatives

$$\nabla^2_{q_i q_i} V_B = \nabla_{q_i} \left(\nabla_{q_i} V_B \right) = \nabla_{q_i} \left(-\nabla_{q_j} V_B \right) = -\nabla^2_{q_i q_j} V_B = -\nabla^2_{q_j q_i} V_B = -\nabla_{q_j} \left(-\nabla_{q_j} V_B \right) = \nabla^2_{q_j q_j} V_B.$$

□

4.1.1 Accuracy of the Harmonic Approximation to the Bond Potential illustrated on Two Particles with One Bond

In order to get an idea how well the linear approximation

$$-\nabla \widetilde{W}^{q^e}(q) = -H_{V_B}(q^e)(q - q^e),$$

approximates the force due to the bond potential, we consider two particles in one dimension with equilibrium positions $q^e = \begin{bmatrix} 1 & 2 \end{bmatrix}^T$. The particles share a bond with bond constant $k_B = 1000$ and equilibrium length $\bar{\rho}_B = 1$. With the initial values

$$q^0 = \begin{bmatrix} 0.975 & 2.025 \end{bmatrix}^T \qquad p^0 = \begin{bmatrix} 0.01 & -0.01 \end{bmatrix}^T \tag{4.3}$$

the center of mass of the system remains the same. The initial potential energy $V_B(q^0)$ due to the deflection along the direction of the bond as well as the kinetic energy $T(p^0)$ result in a harmonic oscillation around the equilibrium bond length $\bar{\rho}_B$.

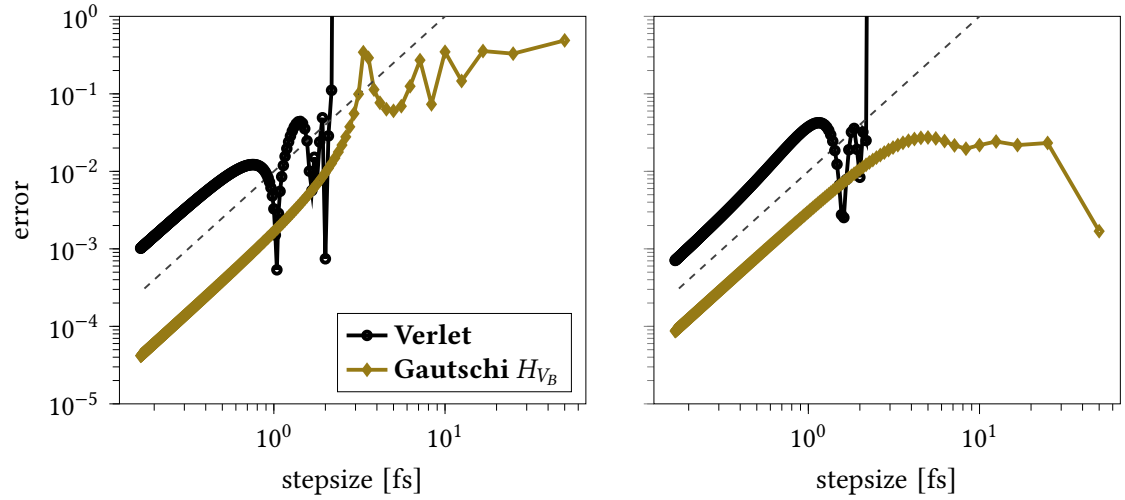


Figure 4.1. Global error at $t = 50$ fs for the Verlet and unfiltered Gautschi-type method when applied to the system with initial data (4.3) in position (left) and momentum (right) as function of the stepsize τ[fs] and a gray dashed reference line with slope 2.

The force due to the bond at the start of the simulation is

$$-\nabla V_B(\rho_{ij}^0) = k_B \frac{\rho_{ij}^0 - \bar{\rho}_B}{\rho_{ij}^0} r_{ij}^0 = 1000 \frac{1.05 - 1}{1.05} \begin{bmatrix} 1.05 \\ -1.05 \end{bmatrix} = \begin{bmatrix} 50 \\ -50 \end{bmatrix}.$$

In contrast, the force due to the harmonic approximation of the bond potential is given by,

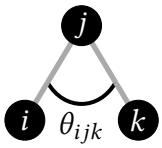
$$-\nabla \widetilde{W}^{q^e}(q^0) = -H_{V_B}(q^e)(q^0 - q^e) = -1000 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} -0.025 \\ 0.025 \end{bmatrix} = \begin{bmatrix} 50 \\ -50 \end{bmatrix}.$$

Evidently, the approximation is exact in the one dimensional case. As shown in section 2.2.1, the bond potential is in this case equal to the quadratic potential of the harmonic oscillator (see section 2.1.3). The integration with the Gautschi-type integrator (2.34) with (2.35) would thus be exact, as discussed in section 2.3.3. Due to the initial momenta and the fact that we use the initial values (2.36), we get second order convergence. We compute a reference solution with the Verlet integrator and stepsize $\tau = 10^{-4}$ fs and plot the error of the Verlet and unfiltered Gautschi-type integrator as a function of the stepsize in figure 4.1. As expected, we can observe the second order convergence with larger stability region. We further see that the Gautschi-type integrator has a smaller error constant compared to the Verlet method.

4.2 Derivatives of the Angle Potential

The angle spanned by three bonded particles i, j and k, as displayed below, is defined in definition 4 by

$$\theta_{ijk} = \arccos\left(\frac{\langle r_{ij}, r_{kj}\rangle}{\|r_{ij}\|\|r_{kj}\|}\right).$$



The corresponding angle potential reads (cf. (2.15))

$$V_A(\theta_{ijk}) = \frac{k_A}{2}(\theta_{ijk} - \bar{\theta})^2. \tag{4.4}$$

The calculation for the first partial derivatives have been taken from 'Numerical Simulation in Molecular Dynamics' by M. Griebel, S. Knapek and G. Zumbusch [31].

Lemma 5. *The partial derivative of the angle potential (4.4) with respect to q_i and q_k are*

$$\begin{aligned}
\nabla_{q_i} V_A(\theta_{ijk}) &= \quad k_A \frac{\theta_{ijk} - \bar{\theta}}{\sin(\theta_{ijk})} \frac{1}{D}\left(r_{kj} - \frac{S\|r_{kj}\|^2}{D^2} r_{ij}\right), \\
\nabla_{q_k} V_A(\theta_{ijk}) &= -k_A \frac{\theta_{ijk} - \bar{\theta}}{\sin(\theta_{ijk})} \frac{1}{D}\left(r_{ij} - \frac{S\|r_{ij}\|^2}{D^2} r_{kj}\right),
\end{aligned}$$

with

$$S := \langle r_{ij}, r_{kj}\rangle \qquad \text{and} \qquad D := \|r_{ij}\|\|r_{kj}\|. \tag{4.5}$$

The partial derivative with respect to q_j follow from the relation

$$\nabla_{q_j} V_A = -\nabla_{q_i} V_A - \nabla_{q_k} V_A. \tag{4.6}$$

Proof. Taking the gradient of the angle potential with respect to q_i yields

$$\begin{aligned}
\nabla_{q_i} V_A(\theta_{ijk}) &= \frac{dV_A}{d\theta_{ijk}} \nabla_{q_i} \theta_{ijk} \\
&= \frac{dV_A}{d\theta_{ijk}} \left(-\frac{1}{\sqrt{1 - \cos^2(\theta_{ijk})}}\right) \nabla_{q_i} \cos(\theta_{ijk}) \\
&= k_A(\theta_{ijk} - \bar{\theta}) \left(-\frac{1}{\sin(\theta_{ijk})}\right) \nabla_{q_i} \cos(\theta_{ijk}) \\
&= -k_A \frac{\theta_{ijk} - \bar{\theta}}{\sin(\theta_{ijk})} \nabla_{q_i} \cos(\theta_{ijk}).
\end{aligned} \tag{4.7}$$

The term $\nabla_{q_i} \cos(\theta_{ijk})$ can be compactly written down as $\cos(\theta_{ijk}) = \frac{S}{D}$ by using (4.5). With that one gets the expression

$$\nabla_{q_i} \cos(\theta_{ijk}) = -\frac{1}{D}\left(r_{kj} - \frac{S\|r_{kj}\|^2}{D^2} r_{ij}\right). \tag{4.8}$$

The gradient thus reads

$$\nabla_{q_i} V_A(\theta_{ijk}) = k_A \frac{\theta_{ijk} - \bar{\theta}}{\sin(\theta_{ijk})} \frac{1}{D} \left(r_{kj} - \frac{S\|r_{kj}\|^2}{D^2} r_{ij} \right).$$

The gradient with respect to q_k is computed analogously and the gradient with respect to q_j is obtained by Newton's third law of motion and the resulting relation

$$\nabla_{q_j} V_A = -\nabla_{q_i} V_A - \nabla_{q_k} V_A. \qquad \square$$

Lemma 6. *The Hessian matrix of the angle potential (4.4) reads*

$$H_{V_A}(\theta_{ijk}) = \begin{bmatrix} h_{ii} & -h_{ii} - h_{ik} & h_{ik} \\ -h_{ii} - h_{ki} & h_{ii} + h_{ki} + h_{ik} + h_{kk} & -h_{ik} - h_{kk} \\ h_{ki} & -h_{ki} - h_{kk} & h_{kk} \end{bmatrix}$$

with

$$\begin{aligned}
h_{ii} &= \nabla^2_{q_i q_i} V_A(\theta_{ijk}) \\
&= k_A \left(\frac{\sin(\theta_{ijk}) - \frac{S}{D}(\theta_{ijk} - \bar{\theta})}{\sin^3(\theta_{ijk})} \left(\nabla_{q_i} \cos(\theta_{ijk}) \otimes \nabla_{q_i} \cos(\theta_{ijk}) \right) - \frac{\theta_{ijk} - \bar{\theta}}{\sin(\theta_{ijk})} \nabla^2_{q_i q_i} \cos(\theta_{ijk}) \right), \\
h_{kk} &= \nabla^2_{q_k q_k} V_A(\theta_{ijk}) \\
&= k_A \left(\frac{\sin(\theta_{ijk}) - \frac{S}{D}(\theta_{ijk} - \bar{\theta})}{\sin^3(\theta_{ijk})} \left(\nabla_{q_k} \cos(\theta_{ijk}) \otimes \nabla_{q_k} \cos(\theta_{ijk}) \right) - \frac{\theta_{ijk} - \bar{\theta}}{\sin(\theta_{ijk})} \nabla^2_{q_k q_k} \cos(\theta_{ijk}) \right), \\
h_{ik} &= \nabla^2_{q_k q_i} V_A(\theta_{ijk}) \\
&= k_A \left(\frac{\sin(\theta_{ijk}) - \frac{S}{D}(\theta_{ijk} - \bar{\theta})}{\sin^3(\theta_{ijk})} \left(\nabla_{q_i} \cos(\theta_{ijk}) \otimes \nabla_{q_k} \cos(\theta_{ijk}) \right) - \frac{\theta_{ijk} - \bar{\theta}}{\sin(\theta_{ijk})} \nabla^2_{q_k q_i} \cos(\theta_{ijk}) \right), \\
h_{ki} &= \nabla^2_{q_i q_k} V_A(\theta_{ijk}) = h_{ik}^T
\end{aligned}$$

where the second derivatives of $\cos(\theta_{ijk})$ *are*

$$\begin{aligned}
\nabla^2_{q_i q_i} \cos(\theta_{ijk}) &= -\frac{\|r_{kj}\|^2}{D^3} \left(r_{kj} \otimes r_{ij} + r_{ij} \otimes r_{kj} + S\mathbb{I} - 3\frac{S}{\|r_{ij}\|^2} r_{ij} \otimes r_{ij} \right), \\
\nabla^2_{q_k q_k} \cos(\theta_{ijk}) &= -\frac{\|r_{ij}\|^2}{D^3} \left(r_{ij} \otimes r_{kj} + r_{kj} \otimes r_{ij} + S\mathbb{I} - 3\frac{S}{\|r_{kj}\|^2} r_{kj} \otimes r_{kj} \right), \\
\nabla^2_{q_k q_i} \cos(\theta_{ijk}) &= -\frac{1}{D} \left(\frac{1}{\|r_{kj}\|^2} r_{kj} \otimes r_{kj} + \frac{1}{\|r_{ij}\|^2} r_{ij} \otimes r_{ij} - \frac{S}{D^2} r_{kj} \otimes r_{ij} - \mathbb{I} \right).
\end{aligned}$$

Proof. From (4.7) we obtain the second derivative with respect to q_i by

$$\nabla^2_{q_i q_i} V_A(\theta_{ijk}) = \nabla_{q_i} \left(-k_A \frac{\theta_{ijk} - \bar{\theta}}{\sin(\theta_{ijk})} \nabla_{q_i} \cos(\theta_{ijk}) \right). \tag{4.9}$$

With

$$\nabla_{q_i}\theta_{ijk} = -\sin^{-1}(\theta_{ijk})\nabla_{q_i}\cos(\theta_{ijk}) \quad \text{and} \quad \nabla_{q_i}\sin(\theta_{ijk}) = -\frac{\cos(\theta_{ijk})\nabla_{q_i}\cos(\theta_{ijk})}{\sin(\theta_{ijk})},$$

it holds that

$$\nabla_{q_i}\frac{\theta_{ijk}-\bar{\theta}}{\sin(\theta_{ijk})} = -\left(\frac{\sin(\theta_{ijk})-\cos(\theta_{ijk})(\theta_{ijk}-\bar{\theta})}{\sin^3(\theta_{ijk})}\right)\nabla_{q_i}\cos(\theta_{ijk}). \tag{4.10}$$

We further get with

$$\nabla_{q_i}S = -r_{kj}, \qquad \nabla_{q_i}D = -\frac{\|r_{kj}\|}{\|r_{ij}\|}r_{ij}, \qquad \nabla_{q_i}\frac{S}{D^2} = \frac{-D^2 r_{kj} + 2SD\frac{\|r_{kj}\|}{\|r_{ij}\|}r_{ij}}{D^4},$$

and the product rule that

$$\nabla_{q_i}\frac{S}{D^2}r_{ij} = \frac{-D^2 r_{kj}\otimes r_{ij} + 2SD\frac{\|r_{kj}\|}{\|r_{ij}\|}r_{ij}\otimes r_{ij}}{D^4} - \frac{S}{D^2}\mathbb{I} = \frac{-1}{D^2}\left(r_{kj}\otimes r_{ij} - \frac{S}{\|r_{ij}\|^2}r_{ij}\otimes r_{ij} + S\mathbb{I}\right).$$

With this result we obtain the second derivative of the cosine of θ_{ijk} with respect to q_i by

$$\begin{aligned}\nabla^2_{q_iq_i}\cos(\theta_{ijk}) &= \nabla_{q_i}\left(-\frac{1}{D}\left(r_{kj} - \frac{S}{D^2}r_{ij}\|r_{kj}\|^2\right)\right) \\ &= -\frac{\|r_{kj}\|^2}{D^3}\left(r_{kj}\otimes r_{ij} + r_{ij}\otimes r_{kj} + S\mathbb{I} - 3\frac{S}{\|r_{ij}\|^2}r_{ij}\otimes r_{ij}\right).\end{aligned} \tag{4.11}$$

Using the product rule yet again in (4.9) together with (4.10) and $\cos(\theta_{ijk}) = \frac{S}{D}$ yields

$$\begin{aligned}\nabla^2_{q_iq_i}V_A(\theta_{ijk}) = k_A\Bigg(&\frac{\sin(\theta_{ijk}) - \frac{S}{D}(\theta_{ijk}-\bar{\theta})}{\sin^3(\theta_{ijk})}\left[\nabla_{q_i}\cos(\theta_{ijk})\otimes\nabla_{q_i}\cos(\theta_{ijk})\right] \\ &- \frac{(\theta_{ijk}-\bar{\theta})}{\sin(\theta_{ijk})}\nabla^2_{q_iq_i}\cos(\theta_{ijk})\Bigg),\end{aligned}$$

with the derivatives $\nabla_{q_i}\cos(\theta_{ijk})$ and $\nabla^2_{q_iq_i}\cos(\theta_{ijk})$ given by (4.8) and (4.11) respectively. The computations for $\nabla^2_{q_kq_k}V_A(\theta_{ijk})$ are performed analogously so that we are left with computing $\nabla^2_{q_kq_i}V_A(\theta_{ijk})$. In addition to the considerations above we compute

$$\begin{aligned}\nabla^2_{q_kq_i}\cos(\theta_{ijk}) &= \nabla_{q_k}\left(-\frac{1}{D}\left(r_{kj} - \frac{S}{D^2}r_{ij}\|r_{kj}\|^2\right)\right) \\ &= -\frac{1}{D}\left(\frac{1}{\|r_{kj}\|^2}r_{kj}\otimes r_{kj} + \frac{1}{\|r_{ij}\|^2}r_{ij}\otimes r_{ij} - \frac{S}{D^2}r_{kj}\otimes r_{ij} - \mathbb{I}\right),\end{aligned} \tag{4.12}$$

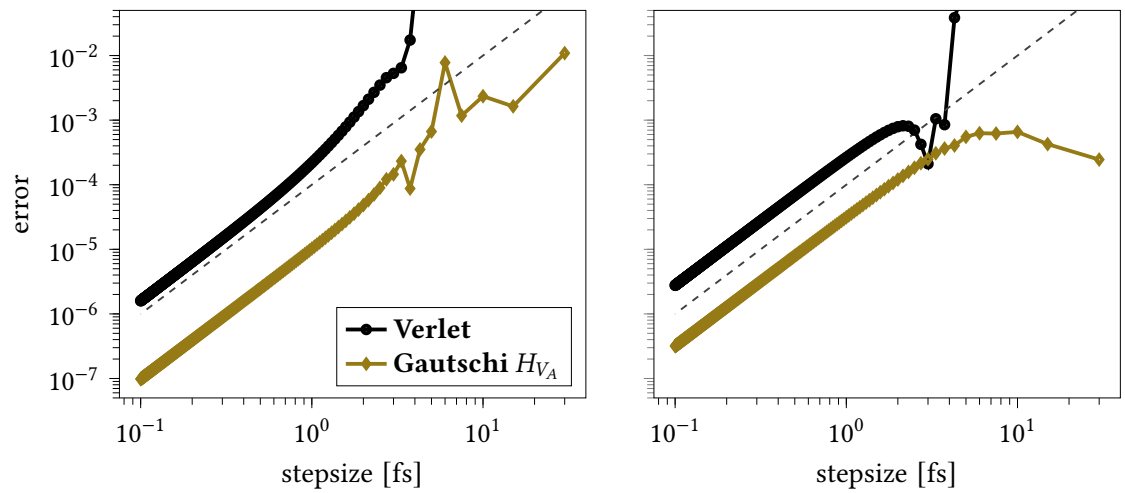


Figure 4.2. Global error at $t = 30$ fs for the Verlet and unfiltered Gautschi-type method when applied to the system with initial data (4.13) in position (left) and momentum (right) as function of the stepsize τ[fs] and a gray dashed reference line with slope 2.

analogously. As before we then get with the product rule

$$\nabla^2_{q_k q_i} V_A(\theta_{ijk}) = k_A \Bigg(\frac{\sin(\theta_{ijk}) - \frac{S}{D}(\theta_{ijk} - \bar{\theta})}{\sin^3(\theta_{ijk})} \left[\nabla_{q_k} \cos(\theta_{ijk}) \otimes \nabla_{q_i} \cos(\theta_{ijk}) \right] - \frac{(\theta_{ijk} - \bar{\theta})}{\sin(\theta_{ijk})} \nabla^2_{q_k q_i} \cos(\theta_{ijk}) \Bigg),$$

with the partial derivative $\nabla^2_{q_k q_i} \cos(\theta_{ijk})$ given by (4.12).

For the computation of the full Hessian matrix we repeatedly make use of the relation (4.6) to get

$$\begin{aligned}
\nabla^2_{q_i q_j} V_A(\theta_{ijk}) &= -\nabla^2_{q_i q_i} V_A(\theta_{ijk}) - \nabla^2_{q_i q_k} V_A(\theta_{ijk}), \\
\nabla^2_{q_j q_j} V_A(\theta_{ijk}) &= -\nabla^2_{q_j q_i} V_A(\theta_{ijk}) - \nabla^2_{q_j q_k} V_A(\theta_{ijk}), \\
\nabla^2_{q_k q_j} V_A(\theta_{ijk}) &= -\nabla^2_{q_k q_i} V_A(\theta_{ijk}) - \nabla^2_{q_k q_k} V_A(\theta_{ijk}),
\end{aligned}$$

combined with the fact that the Hessian matrix is symmetric, i.e.,

$$\nabla^2_{q_j q_i} V_A = \left(\nabla^2_{q_i q_j} V_A \right)^T, \quad \nabla^2_{q_i q_k} V_A = \left(\nabla^2_{q_k q_i} V_A \right)^T, \quad \nabla^2_{q_j q_k} V_A = \left(\nabla^2_{q_k q_j} V_A \right)^T. \qquad \square$$

4.2.1 Accuracy of the Harmonic Approximation to the Angle Potential illustrated on Three Particles with One Angle

In order to get an idea how well the linear approximation

$$-\nabla \widetilde{W}^{q^e}(q) = -H_{V_A}(q^e)(q - q^e),$$

approximates the force due to the angle potential, we consider three particles in two dimensions with equilibrium positions

$$q_1^e = \begin{bmatrix} 0.816 \\ 0.577 \end{bmatrix}, \qquad q_2^e = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad q_3^e = \begin{bmatrix} -0.816 \\ 0.577 \end{bmatrix}.$$

The particles share an angle with angle constant $k_A = 150$ and equilibrium angle $\bar{\theta} = 109.5°$. We set the initial values to

$$q_1^0 = \begin{bmatrix} 0.8290 \\ 0.5529 \end{bmatrix}, \qquad q_2^0 = q_2^e, \qquad q_3^0 = \begin{bmatrix} -0.8290 \\ 0.5529 \end{bmatrix}, \tag{4.13}$$

and $p^0 = 0$. The introduced potential energy $V_A(q^0)$ results in a harmonic oscillation around the equilibrium angle $\bar{\theta}$. Rounded to two digits, the force due to the angle at the start of the simulation is

$$-\nabla V_A(\theta_{ijk}^0) \approx \begin{bmatrix} -4.1 & 6.07 & -0.03 & -12.18 & 4.07 & 6.11 \end{bmatrix}^T,$$

and the force due to the harmonic approximation reads

$$-\nabla \widetilde{W}^{q^e}(q) = -H_{V_A}(q^e)\,(q - q^e) \approx \begin{bmatrix} -4.27 & 6.03 & 0 & -12.06 & 4.27 & 6.03 \end{bmatrix}^T.$$

Evidently, the approximation is not exact for the angle potential and the norm of the approximation error is

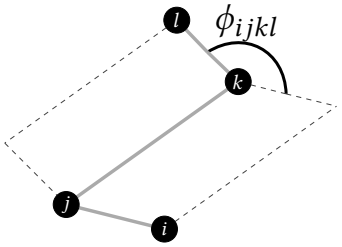
$$\left\| -\nabla V_A(\theta_{ijk}^0) + H_{V_A}(q^e)\,(q - q^e) \right\| \approx \left\| \begin{bmatrix} 0.17 & 0.04 & -0.03 & -0.12 & -0.2 & 0.08 \end{bmatrix}^T \right\| \approx 0.3. \tag{4.14}$$

While the angle potential is harmonic with respect to the angle, the angle is not a harmonic function of the position. However, the approximation is sufficiently accurate, as the norm in (4.14) is over 98% smaller compared to the norm of the fast, angular potential $\| -\nabla V_A(\theta_{ijk}^0) \| \approx 16$. We compute a reference solution with the Verlet integrator and stepsize $\tau = 10^{-4}$ fs and plot the error of the Verlet and the unfiltered Gautschi-type integrator as a function of the stepsize in figure 4.2. As before, we see that the Gautschi-type integrator has the expected second order convergence with smaller error constant compared to the Verlet scheme. In addition, the Gautschi-type method is stable for larger timesteps.

4.3 Derivatives of the Torsional Potential

We defined the dihedral angle between the two intersecting planes spanned by the four particles i, j, k and l, as displayed below, in definition 5 by

$$\phi_{ijkl} := \operatorname{sign}(\langle r_{ij}, n\rangle) \arccos\left(\langle \frac{m}{\|m\|}, \frac{n}{\|n\|}\rangle\right)$$
$$m := r_{ij} \times r_{jk},$$
$$n := r_{jk} \times r_{kl}.$$



The corresponding torsional potential reads (cf. (2.16))

$$V_T(\phi_{ijkl}) = k_T\big(1 - \cos(\phi_{ijkl} - \bar{\phi})\big). \tag{4.15}$$

Lemma 7. *The partial derivatives of the torsional potential (4.15) are*

$$\begin{aligned}
\nabla_{q_i} V_T(\phi_{ijkl}) &= -\frac{dV_T}{d\phi_{ijkl}} \|r_{jk}\| \frac{m}{\|m\|^2}, \\
\nabla_{q_j} V_T(\phi_{ijkl}) &= \frac{dV_T}{d\phi_{ijkl}} \left[\frac{\langle (r_{ij} + r_{jk}), r_{jk}\rangle m}{\|r_{jk}\| \|m\|^2} - \frac{\langle r_{kl}, r_{jk}\rangle n}{\|r_{jk}\| \|n\|^2} \right], \\
\nabla_{q_k} V_T(\phi_{ijkl}) &= -\frac{dV_T}{d\phi_{ijkl}} \left[\frac{\langle (r_{jk} + r_{kl}), r_{jk}\rangle n}{\|r_{jk}\| \|n\|^2} - \frac{\langle r_{ij}, r_{jk}\rangle m}{\|r_{jk}\| \|m\|^2} \right], \\
\nabla_{q_l} V_T(\phi_{ijkl}) &= \frac{dV_T}{d\phi_{ijkl}} \|r_{jk}\| \frac{n}{\|n\|^2}.
\end{aligned}$$

The equations above yield the relations

$$\nabla_{q_j} V_T(\phi_{ijkl}) = -\nabla_{q_i} V_T(\phi_{ijkl}) - \frac{\langle r_{ij}, r_{jk}\rangle}{\|r_{jk}\|^2} \nabla_{q_i} V_T(\phi_{ijkl}) + \frac{\langle r_{kl}, r_{jk}\rangle}{\|r_{jk}\|^2} \nabla_{q_l} V_T(\phi_{ijkl}), \tag{4.16}$$

$$\nabla_{q_k} V_T(\phi_{ijkl}) = -\nabla_{q_l} V_T(\phi_{ijkl}) - \frac{\langle r_{kl}, r_{jk}\rangle}{\|r_{jk}\|^2} \nabla_{q_l} V_T(\phi_{ijkl}) + \frac{\langle r_{ij}, r_{jk}\rangle}{\|r_{jk}\|^2} \nabla_{q_i} V_T(\phi_{ijkl}). \tag{4.17}$$

Proof. We sketch the steps for the first derivative and refer to H. Bekker, H. J. C. Berendsen, and W. F. Gunsteren [4] for more details. With the vector identity

$$A \times (B \times C) = B\langle C, A\rangle - C\langle B, A\rangle \tag{4.18}$$

and the fact that $n \times m = -\frac{\|m\|\|n\|}{\|r_{jk}\|}\sin(\phi_{ijkl})r_{jk}$, the partial derivative of the torsional potential energy function (4.15) with respect to q_i is given by

$$\begin{aligned}
\nabla_{q_i} V_T(\phi_{ijkl}) &= \frac{dV_T}{d\phi_{ijkl}} \frac{1}{\sin(\phi_{ijkl})} \left[r_{jk} \times \left(\frac{n}{\|m\|\|n\|} - \cos(\phi_{ijkl}) \frac{m}{\|m\|^2} \right) \right] \\
&= \frac{dV_T}{d\phi_{ijkl}} \frac{1}{\sin(\phi_{ijkl})} r_{jk} \times \left[\frac{n\|m\|^2 - m\langle m, n\rangle}{\|m\|^3\|n\|} \right] \\
&= \frac{dV_T}{d\phi_{ijkl}} \frac{1}{\sin(\phi_{ijkl})} r_{jk} \times \left[\frac{m \times (n \times m)}{\|m\|^3\|n\|} \right] \\
&= \frac{dV_T}{d\phi_{ijkl}} \frac{1}{\sin(\phi_{ijkl})} r_{jk} \times \left[\frac{m \times (-r_{jk})\|m\|\|n\| \sin(\phi_{ijkl})}{\|r_{jk}\|\|m\|^3\|n\|} \right] \\
&= -\frac{dV_T}{d\phi_{ijkl}} \frac{r_{jk} \times (m \times r_{jk})}{\|r_{jk}\|\|m\|^2} \\
&= -\frac{dV_T}{d\phi_{ijkl}} \|r_{jk}\| \frac{m}{\|m\|^2}. \qquad (4.19)
\end{aligned}$$

Similarly, one gets

$$\nabla_{q_l} V_T(\phi_{ijkl}) = -\frac{dV_T}{d\phi_{ijkl}} \|r_{jk}\| \frac{n}{\|n\|^2}.$$

We further get, following the same steps as before, that

$$\begin{aligned}
\nabla_{q_j} V_T(\phi_{ijkl}) &= \frac{dV_T}{d\phi_{ijkl}} \frac{-1}{\sin(\phi_{ijkl})} \left[r_{ik} \times \left(\frac{n}{\|n\|\|m\|} - \cos(\phi_{ijkl}) \frac{m}{\|m\|^2} \right) - \right. \\
&\qquad \left. r_{kl} \times \left(\frac{m}{\|n\|\|m\|} - \cos(\phi_{ijkl}) \frac{n}{\|n\|^2} \right) \right] \\
&= \frac{dV_T}{d\phi_{ijkl}} \left[\frac{r_{ik} \times (m \times r_{jk})}{\|r_{jk}\|\|m\|^2} - \frac{r_{kl} \times (n \times r_{jk})}{\|r_{jk}\|\|n\|^2} \right] \\
&= \frac{dV_T}{d\phi_{ijkl}} \left[\frac{\langle r_{ik}, r_{jk}\rangle m}{\|r_{jk}\|\|m\|^2} + \frac{\langle r_{kl}, r_{jk}\rangle n}{\|r_{jk}\|\|n\|^2} \right] \\
&= \frac{dV_T}{d\phi_{ijkl}} \left[\frac{\langle (r_{ij} + r_{jk}), r_{jk}\rangle m}{\|r_{jk}\|\|m\|^2} + \frac{\langle r_{kl}, r_{jk}\rangle n}{\|r_{jk}\|\|n\|^2} \right].
\end{aligned}$$

We finally get the relation

$$\begin{aligned}
\nabla_{q_j} V_T(\phi_{ijkl}) &= \frac{dV_T}{d\phi_{ijkl}} \left[\frac{\langle (r_{ij} + r_{jk}), r_{jk}\rangle m}{\|r_{jk}\|\|m\|^2} + \frac{\langle r_{kl}, r_{jk}\rangle n}{\|r_{jk}\|\|n\|^2} \right] \\
&= \frac{dV_T}{d\phi_{ijkl}} \left[\frac{\|r_{jk}\| m}{\|m\|^2} + \frac{\langle r_{ij}, r_{jk}\rangle m}{\|r_{jk}\|\|m\|^2} + \frac{\langle r_{kl}, r_{jk}\rangle n}{\|r_{jk}\|\|n\|^2} \right] \\
&= -\nabla_{q_i} V_T(\phi_{ijkl}) - \frac{\langle r_{ij}, r_{jk}\rangle}{\|r_{jk}\|^2} \nabla_{q_i} V_T(\phi_{ijkl}) + \frac{\langle r_{kl}, r_{jk}\rangle}{\|r_{jk}\|^2} \nabla_{q_l} V_T(\phi_{ijkl}).
\end{aligned}$$

The partial derivative $\nabla_{q_k} V_T(\phi_{ijkl})$ and corresponding relation can be computed analogously. □

Lemma 8. *The Hessian matrix of the torsional potential (4.15) reads*

$$H_{V_T}(\phi_{ijkl}) = \begin{bmatrix} h_{ii} & h_{ij} & h_{ik} & h_{il} \\ h_{ji} & h_{jj} & h_{jk} & h_{jl} \\ h_{ki} & h_{kj} & h_{kk} & h_{kl} \\ h_{li} & h_{lj} & h_{lk} & h_{ll} \end{bmatrix},$$

where $h_{\alpha\beta}$ denotes the 3×3 submatrix of the second-order partial derivative $\nabla^2_{q_\beta q_\alpha} V_T$. The derivates are

$$\begin{aligned}
\nabla^2_{q_i q_i} V_T &= \frac{\|r_{jk}\|}{\|m\|^4} \left[\frac{d^2 V_T}{d\phi^2_{ijkl}} \|r_{jk}\| (m \otimes m) - \frac{dV_T}{d\phi_{ijkl}} \left((r_{jk} \times m) \otimes m + m \otimes (r_{jk} \times m) \right) \right], \\
\nabla^2_{q_l q_l} V_T &= \frac{\|r_{jk}\|}{\|n\|^4} \left[\frac{d^2 V_T}{d\phi^2_{ijkl}} \|r_{jk}\| (n \otimes n) + \frac{dV_T}{d\phi_{ijkl}} \left((r_{jk} \times n) \otimes n + n \otimes (r_{jk} \times n) \right) \right], \\
\nabla^2_{q_l q_i} V_T &= -\frac{d^2 V_T}{d\phi^2_{ijkl}} \frac{\|r_{jk}\|^2}{\|m\|^2 \|n\|^2} m \otimes n,
\end{aligned}$$

and the remaining derivatives follow with the relations

$$\begin{aligned}
\nabla^2_{q_i q_j} V_T &= -\frac{\langle r_{ij} + r_{jk}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_i q_i} V_T + \frac{\langle r_{kl}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_i q_l} V_T + \nabla_{q_i} V_T \otimes \frac{r_{jk}}{\|r_{jk}\|^2}, \\
\nabla^2_{q_l q_j} V_T &= -\frac{\langle r_{ij} + r_{jk}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_l q_i} V_T + \frac{\langle r_{kl}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_l q_l} V_T + \nabla_{q_i} V_T \otimes \frac{r_{jk}}{\|r_{jk}\|^2}, \\
\nabla^2_{q_i q_k} V_T &= -\frac{\langle r_{kl} + r_{jk}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_i q_l} V_T + \frac{\langle r_{ij}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_i q_j} V_T - \nabla_{q_j} V_T \otimes \frac{r_{jk}}{\|r_{jk}\|^2}, \\
\nabla^2_{q_l q_k} V_T &= -\frac{\langle r_{kl} + r_{jk}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_l q_l} V_T + \frac{\langle r_{ij}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_l q_j} V_T - \nabla_{q_j} V_T \otimes \frac{r_{jk}}{\|r_{jk}\|^2}, \\
\nabla^2_{q_j q_j} V_T &= -\frac{\langle r_{ij} + r_{jk}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_j q_i} V_T + \frac{\langle r_{kl}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_j q_l} V_T \\
&\quad + \nabla_{q_l} V_T \otimes \nabla_{q_j} \frac{\langle r_{kl}, r_{jk} \rangle}{\|r_{jk}\|^2} - \nabla_{q_i} V_T \otimes \nabla_{q_j} \frac{\langle r_{ij}, r_{jk} \rangle}{\|r_{jk}\|^2}, \\
\nabla^2_{q_j q_k} V_T &= -\frac{\langle r_{kl} + r_{jk}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_j q_l} V_T + \frac{\langle r_{ij}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_j q_j} V_T \\
&\quad + \nabla_{q_j} V_T \otimes \nabla_{q_j} \frac{\langle r_{ij}, r_{jk} \rangle}{\|r_{jk}\|^2} - \nabla_{q_l} V_T \otimes \nabla_{q_j} \frac{\langle r_{kl}, r_{jk} \rangle}{\|r_{jk}\|^2}, \\
\nabla^2_{q_k q_k} V_T &= -\frac{\langle r_{kl} + r_{jk}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_k q_l} V_T + \frac{\langle r_{ij}, r_{jk} \rangle}{\|r_{jk}\|^2} \nabla^2_{q_k q_j} V_T \\
&\quad + \nabla_{q_j} V_T \otimes \nabla_{q_k} \frac{\langle r_{ij}, r_{jk} \rangle}{\|r_{jk}\|^2} - \nabla_{q_l} V_T \otimes \nabla_{q_k} \frac{\langle r_{kl}, r_{jk} \rangle}{\|r_{jk}\|^2},
\end{aligned}$$

where

$$\nabla_{q_j}\frac{\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^2}=\frac{(r_{jk}-r_{ij})\|r_{jk}\|^2+2r_{jk}\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^4},\quad \nabla_{q_j}\frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2}=\frac{-r_{kl}\|r_{jk}\|^2+2r_{jk}\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^4},$$
$$\nabla_{q_k}\frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2}=\frac{(r_{kl}-r_{jk})\|r_{jk}\|^2-2r_{jk}\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^4},\quad \nabla_{q_k}\frac{\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^2}=\frac{r_{ij}\|r_{jk}\|^2-2r_{jk}\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^4}.$$

The remaining derivatives are given by the symmetry of the derivatives, i.e.,

$$h_{li}=h_{il}^T,\quad h_{ij}=h_{ji}^T,\quad h_{lj}=h_{jl}^T,\quad h_{ik}=h_{ki}^T,\quad h_{jk}=h_{kj}^T,\quad h_{lk}=h_{kl}^T.$$

Proof. We compute the second-order partial derivative with respect to q_i using (4.19) twice. We get

$$\begin{aligned}
\nabla^2_{q_iq_i}V_T &= \nabla_{q_i}\left(-\frac{dV_T}{d\phi_{ijkl}}\frac{\|r_{jk}\|}{\|m\|^2}m\right)\\
&= -\|r_{jk}\|\left[\frac{m}{\|m\|^2}\otimes\left(\nabla_{q_i}\frac{dV_T}{d\phi_{ijkl}}\right)+\frac{dV_T}{d\phi_{ijkl}}\nabla_{q_i}\frac{m}{\|m\|^2}\right]\\
&= -\|r_{jk}\|\left[\frac{m}{\|m\|^2}\otimes\left(-\frac{d^2V_T}{d\phi_{ijkl}^2}\frac{\|r_{jk}\|}{\|m\|^2}m\right)+\frac{dV_T}{d\phi_{ijkl}}\nabla_{q_i}\frac{m}{\|m\|^2}\right]\\
&= -\|r_{jk}\|\left[-\frac{d^2V_T}{d\phi_{ijkl}^2}\frac{\|r_{jk}\|}{\|m\|^4}(m\otimes m)+\frac{dV_T}{d\phi_{ijkl}}\frac{\|m\|^2\nabla_{q_i}m-2\big(m\otimes(m\times r_{jk})\big)}{\|m\|^4}\right]\\
&= \frac{\|r_{jk}\|}{\|m\|^4}\left[\frac{d^2V_T}{d\phi_{ijkl}^2}\|r_{jk}\|(m\otimes m)-\frac{dV_T}{d\phi_{ijkl}}\Big(\|m\|^2\nabla_{q_i}m-2\big(m\otimes(m\times r_{jk})\big)\Big)\right],
\end{aligned}\tag{4.20}$$

with

$$\nabla_{q_i}m=\nabla_{q_l}n=\begin{bmatrix}0 & -r_{jk}^{(3)} & r_{jk}^{(2)}\\ r_{jk}^{(3)} & 0 & -r_{jk}^{(1)}\\ -r_{jk}^{(2)} & r_{jk}^{(1)} & 0\end{bmatrix},\tag{4.21}$$

where $r_{jk}^{(\ell)}$ denotes the ℓ-th component of r_{jk}. Note that multiplying (4.21) with a vector $v\in\mathbb{R}^3$ from the left, is the same as computing the cross product $r_{jk}\times v$. This motivates the notation

$$[r_{jk}]_\times := \nabla_{q_i}m=\nabla_{q_l}n.$$

Since (4.21) is skew-symmetric, it is not obvious that (4.20) is symmetric. We therefore consider the matrix

$$\begin{aligned}
B &= \frac{\|r_{jk}\|}{\|m\|^4}\Big(\|m\|^2\nabla_{q_i}m-2m\otimes(m\times r_{jk})\Big)=\frac{\|r_{jk}\|}{\|m\|^4}\Big(m^Tm[r_{jk}]_\times+2m\otimes\big([r_{jk}]_\times m\big)\Big)\\
&= \frac{\|r_{jk}\|}{\|m\|^4}\Big(m^Tm[r_{jk}]_\times+2m\otimes m[r_{jk}]_\times^T\Big),
\end{aligned}$$

which is symmetric if and only if its transpose is symmetric. We show the symmetricity of

$$A = B^T = \frac{\|r_{jk}\|}{\|m\|^4}\Big([r_{jk}]_\times(2m \otimes m - m^T m\mathbb{I})\Big),$$

by studying its eigendecomposition. In the following we use the notations

$$\hat{r}_{jk} = \frac{1}{\|r_{jk}\|} r_{jk}, \qquad \tilde{m} = r_{ij} \times \hat{r}_{jk},$$

and consider the scaled matrix A which reads

$$A = \frac{1}{\|\tilde{m}\|^4}\Big([\hat{r}_{jk}]_\times(2\tilde{m} \otimes \tilde{m} - \tilde{m}^T \tilde{m}\mathbb{I})\Big).$$

Since we get with (4.18)

$$\hat{r}_{jk} \times \tilde{m} = [\hat{r}_{jk}]_\times \tilde{m} = \hat{r}_{jk} \times (r_{ij} \times \hat{r}_{jk}) = r_{ij} - \hat{r}_{jk}^T r_{ij} \hat{r}_{jk}$$

and further $r_{ij} \perp \tilde{m}, \hat{r}_{jk} \perp \tilde{m}$ we get

$$\tilde{m}^T(\hat{r}_{jk} \times \tilde{m}) = 0.$$

With

$$\begin{aligned}
A\hat{r}_{jk} &= 0, \\
A\tilde{m} &= \frac{1}{\|\tilde{m}\|^4}[\hat{r}_{jk}]_\times(2\|\tilde{m}\|^2\tilde{m} - \|\tilde{m}\|^2\tilde{m}) = \frac{1}{\|\tilde{m}\|^2}\hat{r}_{jk} \times \tilde{m}, \\
A(\hat{r}_{jk} \times \tilde{m}) &= \frac{1}{\|\tilde{m}\|^4}[\hat{r}_{jk}]_\times\big(2\tilde{m} \otimes \tilde{m}(\hat{r}_{jk} \times \tilde{m}) - \|\tilde{m}\|^2(\hat{r}_{jk} \times \tilde{m})\big) \\
&= -\frac{1}{\|\tilde{m}\|^2}[\hat{r}_{jk}]_\times(r_{ij} - \hat{r}_{jk}^T r_{ij}\hat{r}_{jk}) = -\frac{1}{\|\tilde{m}\|^2}\hat{r}_{jk} \times r_{ij} = \frac{1}{\|\tilde{m}\|^2}\tilde{m},
\end{aligned}$$

we have three vectors that are perpendicular to each other and are not zero whenever m is not zero, and we get the decomposition

$$AQ = Q\frac{1}{\|\tilde{m}\|^2}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad Q = \begin{bmatrix} \frac{\hat{r}_{jk} \times \tilde{m}}{\|\hat{r}_{jk} \times \tilde{m}\|} & \frac{\tilde{m}}{\|\tilde{m}\|} & \hat{r}_{jk} \end{bmatrix}$$

where Q is an orthonormal matrix. Due to the eigendecomposition

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

the change of basis to the orthogonal eigendecomposition of A is given by

$$AQ\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = Q\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \frac{1}{\|\tilde{m}\|^2} & 0 & 0 \\ 0 & -\frac{1}{\|\tilde{m}\|^2} & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

With the orthonormal eigenvector

$$v_1 = \frac{1}{\sqrt{2}}\left(\frac{\hat{r}_{jk}\times\tilde{m}}{\|\hat{r}_{jk}\times\tilde{m}\|}+\frac{\tilde{m}}{\|\tilde{m}\|}\right),\quad v_2 = \frac{1}{\sqrt{2}}\left(\frac{\hat{r}_{jk}\times\tilde{m}}{\|\hat{r}_{jk}\times\tilde{m}\|}-\frac{\tilde{m}}{\|\tilde{m}\|}\right),\quad v_3 = \hat{r}_{jk},$$

we may write A as an expansion of its eigenvectors, i.e.,

$$A = \frac{1}{\|\tilde{m}\|^2}(v_1\otimes v_1 - v_2\otimes v_2), \tag{4.22}$$

which is symmetric. We derive a representation of A by undoing the scaling and inserting

$$v_1 = \frac{1}{\sqrt{2}\|m\|}\left(\frac{r_{jk}\times m}{\|r_{jk}\|}+m\right),\qquad v_2 = \frac{1}{\sqrt{2}\|m\|}\left(\frac{r_{jk}\times m}{\|r_{jk}\|}-m\right),$$

into (4.22) to get

$$\begin{aligned}A &= \frac{\|r_{jk}\|^2}{\|m\|^4}\frac{1}{2}\left(\left(\frac{r_{jk}\times m}{\|r_{jk}\|}+m\right)\otimes\left(\frac{r_{jk}\times m}{\|r_{jk}\|}+m\right)-\left(\frac{r_{jk}\times m}{\|r_{jk}\|}-m\right)\otimes\left(\frac{r_{jk}\times m}{\|r_{jk}\|}-m\right)\right)\\ &= \frac{\|r_{jk}\|}{\|m\|^4}\left((r_{jk}\times m)\otimes m + m\otimes(r_{jk}\times m)\right).\end{aligned}$$

Inserting $B = A^T = A$ in (4.20) we get the now obviously symmetric representation

$$\nabla^2_{q_iq_i}V_T = \frac{\|r_{jk}\|}{\|m\|^4}\left[\frac{d^2V_T}{d\phi^2_{ijkl}}\|r_{jk}\|(m\otimes m) - \frac{dV_T}{d\phi_{ijkl}}\left((r_{jk}\times m)\otimes m + m\otimes(r_{jk}\times m)\right)\right].$$

Analogously, we compute the expression

$$\begin{aligned}\nabla^2_{q_lq_l}V_T &= \nabla_{q_l}\left(\frac{dV_T}{d\phi_{ijkl}}\frac{\|r_{jk}\|}{\|n\|^2}n\right)\\ &= \frac{\|r_{jk}\|}{\|n\|^4}\left[\frac{d^2V_T}{d\phi^2_{ijkl}}\|r_{jk}\|(n\otimes n)+\frac{dV_T}{d\phi_{ijkl}}\left(\|n\|^2\nabla_{q_l}n - 2\left(n\otimes(n\times r_{jk})\right)\right)\right]\\ &= \frac{\|r_{jk}\|}{\|n\|^4}\left[\frac{d^2V_T}{d\phi^2_{ijkl}}\|r_{jk}\|(n\otimes n)+\frac{dV_T}{d\phi_{ijkl}}\left((r_{jk}\times n)\otimes n + n\otimes(r_{jk}\times n)\right)\right].\end{aligned}$$

Finally, we compute

$$\nabla^2_{q_lq_i}V_T = -\frac{\|r_{jk}\|}{\|m\|^2}m\left(\nabla_{q_l}\frac{dV_T}{d\phi_{ijkl}}\right) = -\frac{d^2V_T}{d\phi^2_{ijkl}}\frac{\|r_{jk}\|^2}{\|m\|^2\|n\|^2}m\otimes n.$$

Using the symmetry of the Hessian we have $\nabla^2_{q_iq_l}V_T = \left(\nabla^2_{q_lq_i}V_T\right)^T$ and we compute the remaining derivatives with the identity (4.16)

$$\nabla_{q_j}V_T = -\nabla_{q_i}V_T - \frac{\langle r_{ij}, r_{jk}\rangle}{\|r_{jk}\|^2}\nabla_{q_i}V_T + \frac{\langle r_{kl}, r_{jk}\rangle}{\|r_{jk}\|^2}\nabla_{q_l}V_T,$$

and get

$$\begin{aligned}\nabla^2_{q_iq_j}V_T &= -\nabla^2_{q_iq_i}V_T - \left(\frac{\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_iq_i}V_T + \nabla_{q_i}V_T \otimes \nabla_{q_i}\frac{\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^2}\right) + \frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_iq_l}V_T \\ &= -\nabla^2_{q_iq_i}V_T - \left(\frac{\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_iq_i}V_T + \nabla_{q_i}V_T \otimes \frac{-r_{jk}}{\|r_{jk}\|^2}\right) + \frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_iq_l}V_T \\ &= -\frac{\langle r_{ij}+r_{jk},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_iq_i}V_T + \frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_iq_l}V_T + \nabla_{q_i}V_T \otimes \frac{r_{jk}}{\|r_{jk}\|^2},\end{aligned}$$

as well as

$$\begin{aligned}\nabla^2_{q_lq_j}V_T &= -\nabla^2_{q_lq_i}V_T - \frac{\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_lq_i}V_T + \left(\frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_lq_l}V_T + \nabla_{q_i}V_T \otimes \nabla_{q_l}\frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2}\right) \\ &= -\nabla^2_{q_lq_i}V_T - \frac{\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_lq_i}V_T + \frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_lq_l}V_T + \nabla_{q_i}V_T \otimes \frac{r_{jk}}{\|r_{jk}\|^2} \\ &= -\frac{\langle r_{ij}+r_{jk},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_lq_i}V_T + \frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_lq_l}V_T + \nabla_{q_i}V_T \otimes \frac{r_{jk}}{\|r_{jk}\|^2},\end{aligned}$$

and

$$\begin{aligned}\nabla^2_{q_jq_j}V_T = -\nabla^2_{q_jq_i}V_T &- \left(\frac{\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_jq_i}V_T + \nabla_{q_i}V_T \otimes \nabla_{q_j}\frac{\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^2}\right) \\ &+ \left(\frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_jq_l}V_T + \nabla_{q_l}V_T \otimes \nabla_{q_j}\frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2}\right) \\ = -\frac{\langle r_{ij}+r_{jk},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_jq_i}V_T &+ \frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla^2_{q_jq_l}V_T \\ - \nabla_{q_i}V_T \otimes \nabla_{q_j}\frac{\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^2} &+ \nabla_{q_l}V_T \otimes \nabla_{q_j}\frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2},\end{aligned}$$

with the partial derivatives

$$\nabla_{q_j}\frac{\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^2} = \frac{(r_{jk}-r_{ij})\|r_{jk}\|^2 + 2r_{jk}\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^4}, \nabla_{q_j}\frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2} = \frac{-r_{kl}\|r_{jk}\|^2 + 2r_{jk}\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^4}.$$

The second-order partial derivates $\nabla^2_{q_iq_k}V_T$, $\nabla^2_{q_jq_k}V_T$, $\nabla^2_{q_kq_k}V_T$ and $\nabla^2_{q_lq_k}V_T$ can be computed along the same steps, using the identity (4.17)

$$\nabla_{q_k}V_T = -\nabla_{q_l}V_T + \frac{\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla_{q_j}V_T - \frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2}\nabla_{q_l}V_T$$

and

$$\nabla_{q_k}\frac{\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^2} = \frac{(r_{kl}-r_{jk})\|r_{jk}\|^2 - 2r_{jk}\langle r_{kl},r_{jk}\rangle}{\|r_{jk}\|^4}, \nabla_{q_k}\frac{\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^2} = \frac{r_{ij}\|r_{jk}\|^2 - 2r_{jk}\langle r_{ij},r_{jk}\rangle}{\|r_{jk}\|^4}.$$

□

4.3.1 Accuracy of the Harmonic Approximation to the Torsional Potential illustrated on Four Particles with One Dihedral Angle

In order to get an idea how well the linear approximation

$$-\nabla \widetilde{W}^{q^e}(q) = -H_{V_T}(q^e)(q - q^e),$$

approximates the force due to the torsional potential, we consider four particles in three dimensions with equilibrium positions

$$q_1^e = \begin{bmatrix} 0.6662 \\ 1.9426 \\ 0 \end{bmatrix}, \qquad q_2^e = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \qquad q_3^e = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}, \qquad q_4^e = \begin{bmatrix} 2.3338 \\ 0.0574 \\ 0 \end{bmatrix}.$$

The particles share a dihedral angle with energy constant $k_T = 80$. We set the initial values to

$$q_1^0 = \begin{bmatrix} 0.6662 \\ 1.9426 \\ 0.1 \end{bmatrix}, \qquad q_2^0 = q_2^e, \qquad q_3^0 = q_3^e, \qquad q_4^0 = \begin{bmatrix} 2.3338 \\ 0.0574 \\ 0.1 \end{bmatrix} \tag{4.23}$$

and $p^0 = 0$ in order to introduce a harmonic oscillation of the dihedral angle. Rounded to the last two digits, the force due to the torsional potential at the start of the simulation is

$$-\nabla V_T(\phi_{ijkl}^0) \approx [0\ 1.87\ {-17.67} \mid 0\ -3.12\ 17.67 \mid 0\ 3.12\ 17.67 \mid 0\ -1.87\ -17.67]^T$$

and the force due to the harmonic approximation reads

$$-H_{V_T}(q^e)(q^0 - q^e) \approx [0\ 0\ -18.01 \mid 0\ 0\ 18.01 \mid 0\ 0\ 18.01 \mid 0\ 0\ -18.01]^T.$$

The approximation to the torsional potential is not exact for the torsional potential and

$$\begin{aligned} \| -\nabla V_T(\phi_{ijkl}^0) + H_{V_T}(q^e)(q^0 - q^e)\| \\ &\approx \left\|[0\ 1.87\ 0.34 \mid 0\ -3.12\ -0.34 \mid 0\ 3.12\ -0.34 \mid 0\ -1.87\ 0.34]^T\right\| \\ &\approx 5.2. \end{aligned} \tag{4.24}$$

As with the angle potential, the torsional potential is harmonic with respect to the dihedral angle, but the dihedral angle is not a harmonic function of the position. Nevertheless, the norm of (4.24) is still over 85% smaller than that of the fast, torsional potential $\|\nabla V_T(\phi_{ijkl}^0)\| \approx 35.71$. The integration with the Verlet method and the unfiltered Gautschi-type integrator results in the error plot shown in figure 4.3. The Gautschi-type integrator stays stable for larger stepsizes compared to the Verlet method. However, the difference in the error constant is smaller than in the previous examples in section 4.1.1 and section 4.2.1.

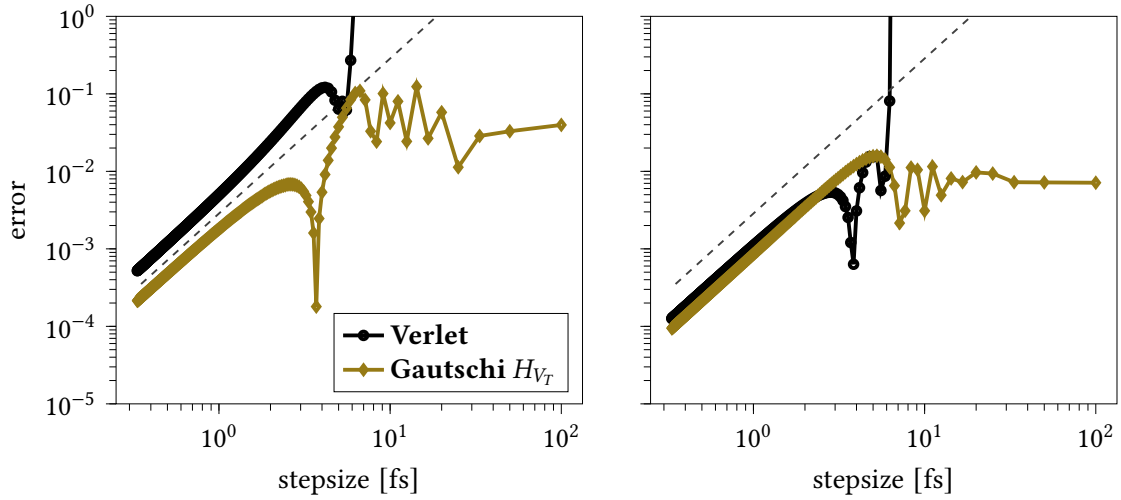


Figure 4.3. Global error at $t = 100$ fs for the Verlet and unfiltered Gautschi-type method when applied to the system with initial data (4.23) in position (left) and momentum (right) as function of the stepsize τ[fs] and a gray dashed reference line with slope 2.

4.4 Combination of the Individual Harmonic Approximations

In the examples above, (cf. section 4.1.1, section 4.2.1, section 4.3.1), we isolated the effects of the individual potentials to get an idea of the quality of their corresponding harmonic approximations. In molecular dynamics simulation multiple potentials are present at the same time. To get a first impression on the quality of the linear approximation in a more complex system, we combine the three previously discussed potentials (bond, angle, and torsional potential) in one system. The linearization of a sum of potentials is the sum of their linearizations, i.e.,

$$H_{V_B+V_A+V_T} = H_{V_B} + H_{V_A} + H_{V_T}.$$

We use the particle configuration from section 4.3.1 this time with bond, angle, and torsional potentials. We keep the parametrization from the chapters before and activate all modes at once by setting the initial position to

$$q_1^0 = \begin{bmatrix} 0.6662 \\ 1.9426 \\ 0.05 \end{bmatrix}, \qquad q_2^0 = \begin{bmatrix} 0.95 \\ 1 \\ -0.05 \end{bmatrix}, \qquad q_3^0 = \begin{bmatrix} 2.05 \\ 1 \\ -0.05 \end{bmatrix}, \qquad q_4^0 = \begin{bmatrix} 2.3338 \\ 0.0574 \\ 0.05 \end{bmatrix}$$

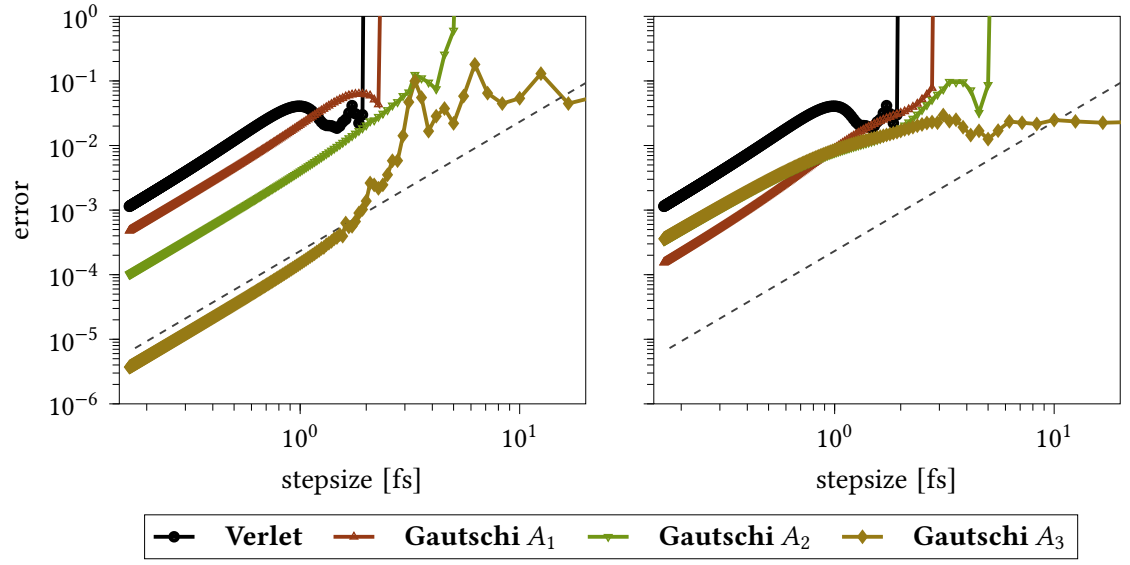


Figure 4.4. Global error at $t = 50$ fs for the Verlet and Gautschi-type method with different linearization matrices (see (4.25)), unfiltered (left) and filtered(right), when applied to the system with initial data (4.23) as function of the stepsize τ[fs] and a gray dashed reference line with slope 2.

and $p^0 = 0$. To see the effect of the Hessian of each potential function, we employ the unfiltered and filtered Gautschi-type integrator with the three different linearization matrices

$$\begin{aligned} A_1 &= -H_{V_B}(q^e), \\ A_2 &= -H_{V_B}(q^e) - H_{V_A}(q^e), \\ A_3 &= -H_{V_B}(q^e) - H_{V_A}(q^e) - H_{V_T}(q^e). \end{aligned} \tag{4.25}$$

From the error plot in figure 4.4 on the left we deduce that with each added Hessian to the linear approximation matrix A, the Gautschi-type integrator has a smaller error constant and is stable for larger stepsizes. We can further see peaks, especially for the Gautschi-type integrator with the matrix A_3. Since there are no further potentials in the system, these are resonances stemming from the remaining nonlinear term $-\nabla V(q) - A_3(q - q^e)$. With the inner filter function, as described in section 3.1.3, the peaks get filtered out, as seen on the right plot in figure 4.4. This comes with the cost of a larger error constant, which is not a strong limitation, since the main interest lies in achieving the largest possible timestep.

4.5 Update Procedure of the Harmonic Approximation

In all examples of this chapter, we evaluated the Hessian at the equilibrium position and kept it constant for the complete integration. This strategy works for problems where the particles do not deviate much from their equilibrium position. However, in most molecular

dynamics simulations molecules are translating and rotating through space. While the linear approximation is invariant regarding translation, it is not invariant with respect to rotations. There are different ways to deal with this issue.

One approach, is to decompose the molecular structure into disjunct clusters of smaller size and construct suitable rotation matrices. The Hessian is calculated for the initial orientation of these clusters. In each step a matrix is calculated that rotates each cluster from its current orientation to the initial orientation. The Hessian is applied to the cluster in its initial orientation, before the cluster is rotated back to its current orientation. The challenge in this approach lies in the construction of suitable clusters and the efficient calculation of the rotation matrices in each step. D. Michels and M. Desbrun [59] successfully used this technique in a test environment. Further, L. Fath, M. Hochbruck, and C. V. Singh [21] follow a similar approach in their implementation of the mollified impulse method in LAMMPS. Due to the details of the implementation presented by L. Fath et al. the method can only be applied to molecules with certain structures.

In this work we follow a different approach. We reevaluate the Hessian in every step of the simulation. We refer to the reevaluation of the Hessian as *updating* the matrix. This approach can be applied to any system with arbitrary topology.

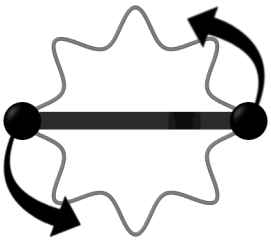
To underline the issue with a constant linear approximation matrix and the effect of an updated matrix we consider the example from section 4.1.1 in two dimensions with initial data

$$\begin{aligned} q_1^0 &= \begin{bmatrix} -0.6 \\ 0 \end{bmatrix} \qquad p_1^0 = \begin{bmatrix} 0 \\ -0.48 \end{bmatrix} \\ q_2^0 &= \begin{bmatrix} 0.6 \\ 0 \end{bmatrix} \qquad p_2^0 = \begin{bmatrix} 0 \\ 0.48 \end{bmatrix}. \end{aligned} \tag{4.26}$$

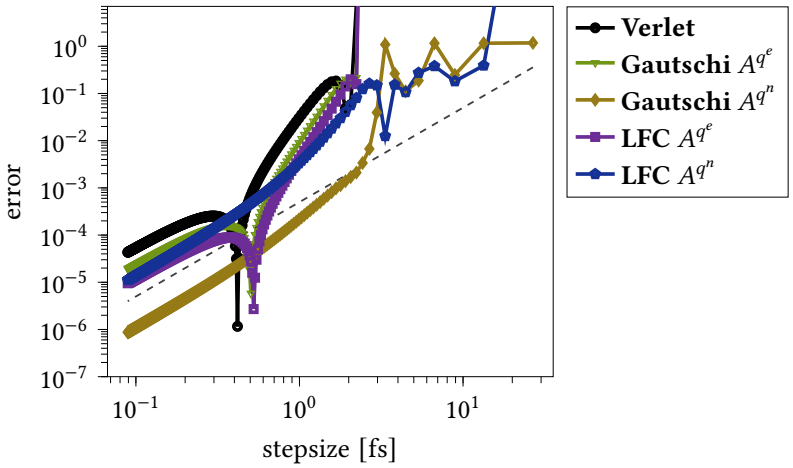
In addition to stretching the bond in x-direction, the particles have initial velocities in opposite y-direction. This introduces a rotation to the system, as shown by the trajectory in figure 4.5a. After 26.75 fs half a revolution is completed and the particles have switched places. We integrate the system with the unfiltered Gautschi-type integrator with the constant Hessian $A^{q^e} = -H_{V_B}(q^e)$ and with continuous reevaluations of the Hessian at the current position, i.e., $A^{q^n} = -H_{V_B}(q^n)$.

We can see in figure 4.5b that without an update, the unfiltered Gautschi-type integrator has a slightly better error constant with a similar maximal stepsize compared to the Verlet integrator. The constant linear approximation is not able to resolve the rotation in the system. With the updates, the approximation with the unfiltered Gautschi-type integrator has a significantly lower error constant and stays stable for large stepsizes.

Note that we cannot employ our averaging approach as discussed in section 3.1.3 to systems with rotating particles. This is not due to the quality of the updated Hessian but rather the fact that we have no longer access to a suitable equilibrium position which is needed to calculate the filtered position $\tilde{q}^n = q^e + \Phi(\tau\Omega)(q^n - q^e)$. As a consequence, our averaging approach is limited to problems that exhibit little rotational movement.



(a) Trajectory of two rotating bonded particles.



(b) Global error of the position at $t = 26.75\,\mathrm{fs}$ for the Verlet, unfiltered Gautschi-type and stabilized Leapfrog-Chebyshev integrators with and without updated Hessian when applied to the system with initial data (4.26) as function of the stepsize $\tau[\mathrm{fs}]$ and a gray dashed reference line with slope 2.

Figure 4.5. Application of the unfiltered Gautschi-type integrator to a rotating, bonded pairs of particles. Continuous reevaluations allow for an accurate approximation of the harmonic motion in the rotating system.

The stabilized Leapfrog-Chebyshev integrator, on the other hand, does not require the equilibrium position. While the error constant for small stepsizes is larger compared to the unfiltered Gautschi-type integrator, the stabilized Leapfrog-Chebyshev integrator with updated Hessian is able to overcome the numerical resonances in this system as the plot in figure 4.5b shows.

Note that due to rounding errors, the updated matrix may have small negative eigenvalues. Long-time simulations of large systems (see chapter 7) have shown that this does not pose a problem for either the Gautschi-type or the Leapfrog-Chebyshev integrator. To avoid complex values when taking the square root of the negative eigenvalues of the Hessenberg matrix H_m associated with the mth Krylov subspace we consider the absolute value of such eigenvalues.

5 Details on Molecular Dynamics Software

The fundamental concepts of molecular dynamics are easy to understand and implementing rudimentary software to simulate a small system does not pose a big challenge. However, understanding, building, and maintaining a large-scale massively parallel simulator for molecular systems used by industry and academia alike is no easy undertaking. There is a range of different molecular dynamics software packages that accomplish just that, such as AMBER [15], CHARMM [11], GROMACS [2], LAMMPS [81], [3], NAMD [64]. We implemented the integrators in the open source software package LAMMPS (*Large-scale Atomic/Molecular Massively Parallel Simulator*) for multiple reasons. The code can be extended and modified in a straightforward way and modifications are easy accessible due to its open source nature. As a consequence, related contributions for example by L. Fath et al. [21] have also been made in LAMMPS. Further, our collaborators are using LAMMPS for their simulations. We present some joint results in section 7.4.

We use LAMMPS-specific terminology throughout this chapter, even though the concepts discussed at the start of this chapter are not unique to LAMMPS. The following description of the software package has no intention of being complete. It serves as a short introduction with the aim to illustrate concepts that are important for our modification presented in chapter 6.

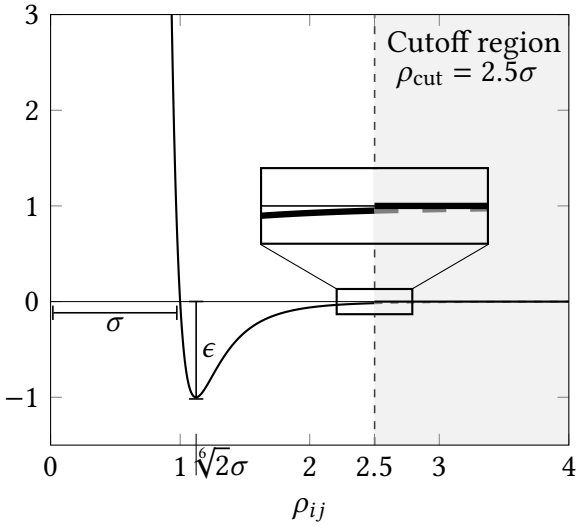


Figure 5.1. Plot of the Lennard-Jones potential ($\epsilon = \sigma = 1$) with (black solid line) and without (gray dotted line) cutoff as a function of the distance ρ_{ij} of two particles i and j. The zoomed in region shows the discontinuity imposed by the cutoff.

5.1 Scales in Molecular Dynamics

Simulating and understanding materials on the macroscopic scale solely with molecular dynamics is not possible with current technology. The reason is twofold. On one hand, the number of atoms exceeds, even in a few grams of matter, the scope of imagination. For example, one gram of hydrogen consists of roughly $6 \cdot 10^{23}$ atoms. The second factor is the considered timescale. As discussed in section 2.2, the fastest oscillations occur at the femtosecond regime. Simulating a second would thus entail a numerical integration with about 10^{15} steps. Since there is neither enough storage for that amount of atoms nor enough computational power to integrate them over quadrillions of steps, molecular dynamics is limited to a certain size and timescale. Most molecular dynamics simulations consider systems containing up to 100 000 particles over a couple of nanoseconds (1 ns = 10^6 fs). In the applied sciences, and in the RTG 2450 'Tailored Scale-Bridging Approaches to Computational Nanoscience' especially, multiscale approaches link this scale with finer scales, like quantum mechanics, as well as with coarser scales, such as fluid dynamics.

5.1.1 Efficient Calculation of Intermolecular Potentials in Large Scale Simulations

By definition, intermolecular pair potentials, like the Lennard-Jones (2.12) or Coulomb (2.13) potential, act between every pair of particles in the simulation. For a simulation with N particles, this implies $N(N-1)/2$ interactions (cf. section 2.2.1). The computational effort thus grows quadratically with the number of particles in the system. In face of this quadratic growth of computational cost it is crucial to avoid the computation of all

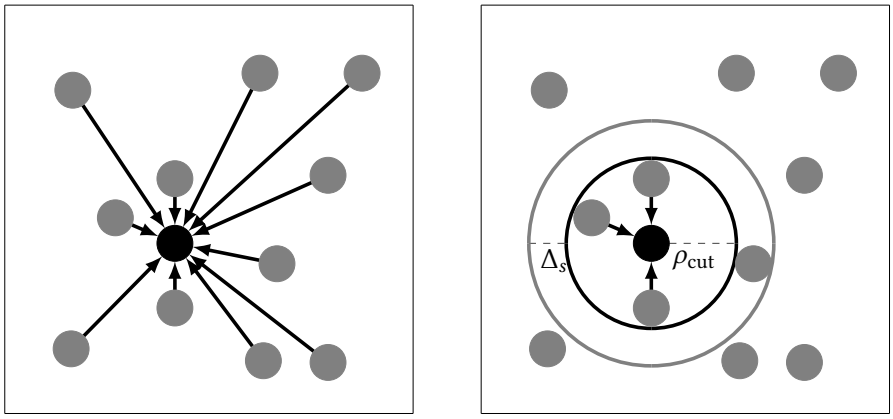


Figure 5.2. Pairwise interactions for one particle (black) without a cutoff (left) and with a cutoff (right). With the cutoff, only particles with a distance smaller than ρ_{cut} to the black particle have interactions. An additional skin distance Δ_s is added for computational efficiency.

pair-wise interactions. For long-range potentials, as for example the Coulomb potential, the interactions with distant particles can not simply be neglected. There are various ways to treat the long-range computations of these potentials. The most established method is based on Ewald [20], where short-range interactions are treated differently from the long-range interactions. More recently the 'fast multipole method' developed by Greengard and Rokhlin [30] has become popular. For more details on these algorithms and a comparison we refer to [74].

If, in comparison, the potential decays faster than ρ^{-d}, where d is the space dimension and ρ the distance of the interacting particles, the long-range interactions can simply be truncated (see [31, Section 3.5]). The idea is to introduce a cutoff radius ρ_{cut} and neglect all interactions between particles that are farther apart than ρ_{cut}. The Lennard-Jones potential falls in this class of fast decaying potentials (see figure 5.1) and its cut version might be given by

$$V_{LJ_{\text{cut}}}(\rho_{ij}) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{\rho_{ij}}\right)^{12} - \left(\frac{\sigma}{\rho_{ij}}\right)^{6} \right], & \rho_{ij} \leq \rho_{\text{cut}} \\ 0, & \rho_{ij} > \rho_{\text{cut}}, \end{cases}$$

where a common choice of the cutoff distance is $\rho_{\text{cut}} = 2.5\sigma$. This formulation has a discontinuity (see figure 5.1) that can lead to unwanted behavior near the cutoff distance. There are different ways to approach this issue, Stoddard and Ford [76], for example, propose a version of the Lennard-Jones potential that continuously decays to the cutoff distance. It reads

$$\begin{aligned} V_{LJ_{\text{SF}}}(\rho_{ij}) = 4\epsilon \Bigg(& \left[\left(\frac{\sigma}{\rho_{ij}}\right)^{12} - \left(\frac{\sigma}{\rho_{ij}}\right)^{6} \right] \\ & + \left[6\left(\frac{\sigma}{\rho_{\text{cut}}}\right)^{12} - 3\left(\frac{\sigma}{\rho_{\text{cut}}}\right)^{6} \right] \left(\frac{\rho_{ij}}{\rho_{\text{cut}}}\right)^{2} - 7\left(\frac{\sigma}{\rho_{\text{cut}}}\right)^{12} + 4\left(\frac{\sigma}{\rho_{\text{cut}}}\right)^{6} \Bigg), \end{aligned}$$

for $\rho_{ij} \leq \rho_{\text{cut}}$, and $V_{LJ_{\text{SF}}}(\rho_{ij}) = 0$ for $\rho_{ij} \geq \rho_{\text{cut}}$.

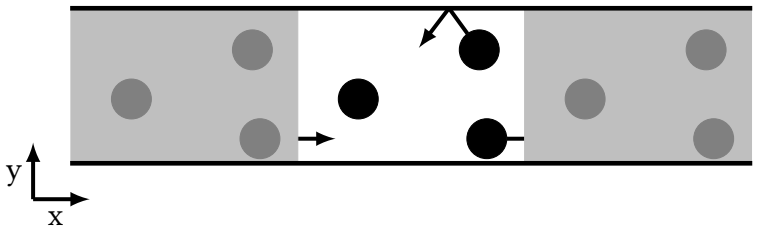


Figure 5.3. A simulation cell (white) with particles, with non-periodic, reflecting boundaries in y direction and periodic boundaries in x direction. The periodically extended unit cell in x direction is shown in gray.

The cutoff radius ensures that each particle only interacts with other particles in a small neighborhood. A neighbor list stores and enumerates all interactions between particles that have a distance smaller than the cutoff distance. In general, the neighbor list of a particle is constantly changing, as the particles move towards and away from each other. Determining which pairs of particles have a distance smaller than the cutoff distance is again a problem with quadratic complexity. In order to avoid frequent updates of the neighbor list, the neighborhood is extended with particles in a radius

$$\rho_{\text{nei}} = \rho_{\text{cut}} + \Delta_s, \tag{5.1}$$

where Δ_s is a preset *skin depth*, see figure 5.2. In our examples, the skin depth is set to the default value $\Delta_s = 2\,\text{Å}$. This method allows to keep the neighbor list for longer time intervals. The neighbor list is in most cases updated when a particle has moved half the skin distance.

5.2 Structure of the Computations

Not only the systems considered by molecular software such as LAMMPS are massive, but also the software governing the simulation is immense. For a successful application of the software, both the particle and implementation structure have to be clearly organized.

5.2.1 Particle Structure

Storing, using, and altering the state of this many particles is a non-trivial task. We first introduce the approach followed by LAMMPS for a single thread and then discuss how it is extended to allow for parallel computations on multiple threads.

Serial Approach

At the beginning of each simulation the properties of the simulation cell, i.e., the space in which the particles exist, have to be defined. The simulation cell is given by a finite domain $\mathcal{D} \subset \mathbb{R}^d$ together with suitable boundary conditions. The boundary conditions

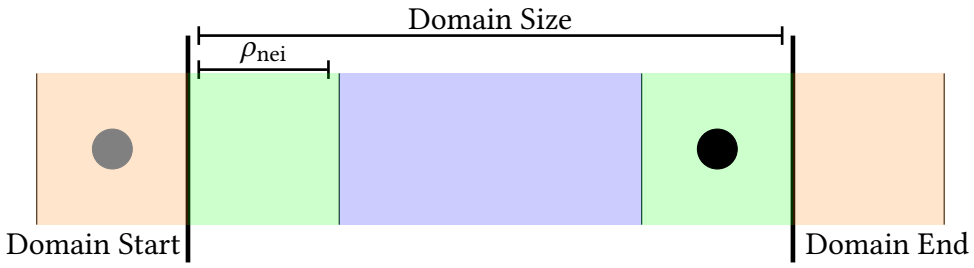


Figure 5.4. A simulation cell (green and blue) is extended by two ghost regions (orange). The gray particle in the ghost region mirrors the black particle in the border region, with position $q_{\text{gray}} = (q_{\text{black}} - \text{Domain Size})$.

describe the behavior of particles that reach the boundary surface of $\mathcal{D}$ and thus specify the surroundings of the simulation cell. There are two main kinds of boundaries.

Non-periodic boundaries imply that no interactions over the boundary of the domain occur. As a result, particles that are approaching a non-periodic boundary either reflect back into the domain or leave the cell and are deleted from the simulation. This setting may for example be used to simulate a vacuum or a small closed system.

Periodic boundaries extend the finite domain, that is in this context also referred to as the *unit cell*, periodically to create a larger simulation cell and therefore allow for long-range interactions over large distances. While only the particles inside the unit cell are stored, one gets access to the behavior of a much larger system. The quality of the approximation to the behavior of the large system depends on the size of the unit cell and the structure of the problem. Periodic boundary conditions are the most common type in molecular dynamics. Applications are for example the generation of solutions that fill up a large space and surround a biomolecule or the study of the flow of a liquid.

We can employ both of these boundary conditions at the same time as shown in figure 5.3. The choice of boundary conditions strongly influences the behavior of the simulation. In a simulation cell with periodic boundaries, particles that are close to the boundary surface interact with particles on the other side of the unit cell. We call the region in which these particles are situated the *border region*. The width of this region is given by ρ_{nei} (cf. (5.1)). To get efficient access to an accurate representation of the distance of particles on opposing sides of the unit cell, the simulation domain is extended by a *ghost region* with the same width. The particles in this region are called *ghost particles* and mirror the properties of the particles on the opposite side of the domain but are positioned according to the boundary condition. A visualization of this concept is given in figure 5.4. As a result, the simulation domain is made up of three distinct regions as shown in figure 5.5. The particles in the inner blue region have no interactions across the periodic boundary. The border region of the domain, displayed in green, contains the particles that may interact with particles on the other side of the boundary surface. Copies of the particles in this region are stored in the orange ghost region with modified positions. For example in figure 5.5, the particles in the green subcell with a star '$*$' are mirrored in the starred orange subcell outside the thickly outlined unit cell.

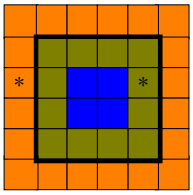


Figure 5.5. Sketch of a domain with ghost regions.

In addition to the numbering of the particles introduced in (2.11) the ghost particles are indexed from N to $N + N_G$, where N_G is the number of ghost particles in the simulation. In the following we refer to the non-ghost particles as *local* particles. LAMMPS provides a unique mapping from the index of a ghost particle to its local counterpart.

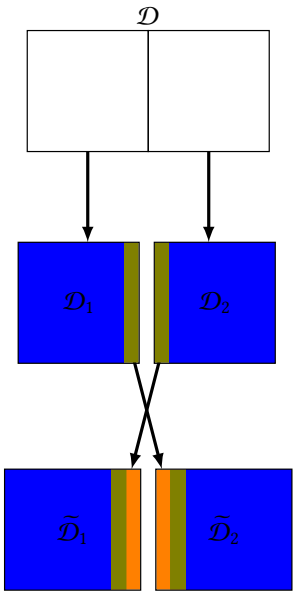
Information about molecular structures, i.e., bonds, angle, and dihedral angle, and associated intramolecular potential parameters are stored in arrays, containing the identifiers of the involved particles as well as the type of the intramolecular connection. Since these arrays contain information of the topological structure, we refer to them as topological lists. LAMMPS loops over these lists in order to calculate the corresponding forces.

Parallel Approach

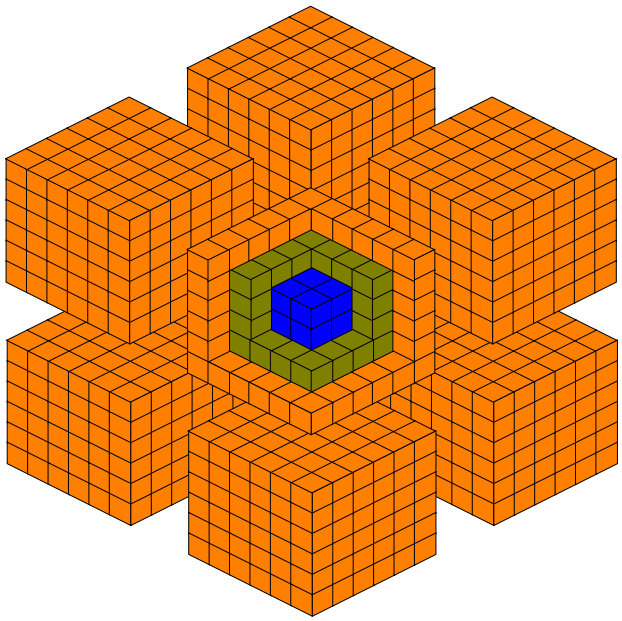
In order to distribute the computational effort on multiple threads, LAMMPS decomposes the simulation domain $\mathcal{D}$ in k subdomains $\mathcal{D}_1, \ldots, \mathcal{D}_k$. Each subdomain gets assigned to a unique thread. Every thread handles its subdomain as in the serial case. Subdomains at the boundary of the domain keep the original boundary conditions. Similar to the serial case, particles with a distance smaller than ρ_{nei} to the boundary of the subdomain may interact with particles on the other side of the boundary, i.e., with particles handled by another thread. For an efficient calculation of the forces between particles in different subdomains the positions and types of particles in the outer layer of each subdomain are mirrored in the ghost region of the neighboring subdomain. This means that each thread also stores the particles from neighboring border domains, we denote this extended subdomain as $\widetilde{\mathcal{D}}_i$ for $i = 1, \ldots, k$ and sketch the concept for a one dimensional communication between two subdomains in figure 5.6a.

Again, the subdomain is composed of a region with particles that have only interactions with other local particles (blue) and a border region with particles that may have interactions with particles with the neighboring subdomain (green). In every step of the time integration, the information associated with the particles in the border region is shared with the adjacent thread. The information is stored in the ghost region (orange) of this thread. Every thread has then all the information needed to calculate the forces in its subdomain. We illustrate the decomposition in three dimensions in figure 5.6b, where now each box represents the extended simulation subdomain of each thread.

LAMMPS differentiates between two ways of computing these forces. The default option, *'Newton On'*, uses Newton's third law in order to reduce the necessary force computations. This means that for example the force contribution due to a bond between two particles is only computed once and then added to one and subtracted from the other particle. However, as a result an interaction between particles in two subdomains is only stored and computed in one of the threads. Thus, a second communication step is necessary after the force has been calculated to add the force to the particle in the neighboring thread. The second option is to turn *'Newton Off'*. In this case every interaction is calculated for every particle. This leads to many redundant computations but saves an extra communication



(a) Domain decomposition in two subdomains with orange ghost regions for communication in x direction.



(b) Domain decomposition in eight subdomains for parallel computations in a $2 \times 2 \times 2$ grid.

Figure 5.6. Sketch for the domain decomposition for communication in two direction (left) and three direction (right). The particles in the blue inner region only interact with particles inside the subdomain. Particles in the green border region interact with particles from neighboring subdomains. Their per-atom information is mirrored in the orange ghost region of the neighboring subdomain.

step. Which of these two options is faster depends on the problem size, the potentials and their cutoff length as well as hardware specifications.

As in the serial case, LAMMPS updates the neighbor list when the reneighboring criterion is fulfilled. In most cases this means that the neighbor list, which governs the particles that are stored as ghost particles in the neighboring threads, is updated when a particle has moved half the skin distance.

5.2.2 Implementation Structure

The core code of LAMMPS is written in C++ and its functionalities are spread over numerous classes. To satisfy the complexity of molecular dynamics simulation, LAMMPS consists of many classes organized in 14 main strands. The focus for us lies on the parts relevant for numerical integration in general and the here presented integrators specifically.

The heart of the numerical integration is the `Update` class. The `Update` class holds an instance of the `Integrate` class which initiates either the standard Verlet integrator (see section 2.3.1) or the r-RESPA integrator (see section 2.3.2). As such, the integrator class performs the integration steps, controlling when and in which way the position and velocity of each particle changes. Necessary for that are the force calculations in every time step. The force is computed in the sup-classes of the `Force` class. The `Force` class

holds instances of the inter- and intramolecular forces present in the system, as discussed in section 2.2.

The final building block needed for this work is the `Modify` class. As the name suggest it allows for modifications of the procedure. There are two main ways to extend the code without altering the existing implementation. The `Compute` modifications calculate properties of the whole system, a local region around atoms, or a specific atom. A `Fix` is applied to a group of atoms during time stepping or minimization. With a `Fix` one is able to alter the properties, i.e., position, velocity, force, and so on, of this group of particles. For example, the two-step Verlet formulation is in itself a fix that executes after the force calculations and at the end of each time step. One can further modify the attributes that are associated with the atoms by defining an *atom_style*.

One of the main reasons for choosing LAMMPS for this work was the easy modifiability. Due to its modular design, LAMMPS recognizes files that are added to the build directory as long as we abide by some basic rules. In this way we can change, or add to, almost all functionality of LAMMPS. In this work we take advantage of this by adding an atom style, making small changes to the angle and dihedral potential and most importantly, adding the Gautschi-type and Leapfrog-Chebyshev integrator as fixes to LAMMPS.

5.3 Standard Workflow

Setting up, running, and understanding a molecular dynamics simulation are three distinct stages. We give a brief glimpse into the required steps and potential challenges in each of these stages.

At first the topology, i.e., positions of and connections between particles, is defined. While this is relatively easy for simulations of material with well known properties like a graphene sheet, it can get arbitrarily difficult for simulations of a mixture of different molecules and structures that interact with each other. The pentane simulations that we consider in chapter 7 were constructed with the help of Moltemplate [51]. Moltemplate is a tool that uses small building blocks in order to build larger systems. We also used VMD [50], a molecular visualization program. VMD can be used to build different nanostructures with given lattice structure, such as the graphene sheet. Finding the appropriate potentials with correct parametrization is a second challenge at this stage. For most cases, a suitable parametrization and selection of potentials is given by so-called force fields, such as AMBER [15] or CHARMM [11]. The particles finally have to be set into a domain with boundary conditions, as discussed above.

As an optional, intermediate step, steepest descent, conjugated gradient or comparable methods minimize the energy in the system, before the system is integrated in time. This procedure is often employed to resolve non-physical constellations, such as overlapping particles.

A simulation is performed with a certain goal in mind. Different goals require different simulation techniques. The simulations we consider in this thesis follow the same structure.

First the system undergoes a relaxation process. After this, the system is in a stationary state. This procedure is called equilibration. Depending on the problem we then consider the system under different thermodynamic ensembles. The three mostly used thermodynamic ensembles are the microcanonical ensemble (NVE), the canonical ensemble (NVT) and the Isothermal-Isobaric ensemble (NPT). The abbreviations, NVE, NVT, and NPT describe the properties of the system that stay constant. The number of particles in the system is denoted by N, the energy by E, the volume by V, the pressure by P and the temperature by T.

The NVE ensemble describes a system without energy transfer from or to the outside world, i.e., the Hamiltonian stays constant. The canonical ensemble conserves the temperature in the system. A straight forward way of achieving this is to rescale the velocities of all particles in the system every couple steps. The Berendsen method [7] is a popular method that follows this approach. Conceptually, the system is coupled to an external heat bath with fixed temperature T_0. The strength of the coupling determines the rate of temperature change and is denoted by γ. This means that the velocities are scaled with the factor

$$\lambda^2 = 1 + \frac{\tau}{\gamma}\left(\frac{T_0}{T} - 1\right),$$

where T is the current temperature. While this procedure allows for a very efficient temperature control, it can in general not be used to generate a canonical ensemble [61]. Instead, a more elaborate approach may be applied. The Hamiltonian equations are extended by a fictitious dynamical variable ζ that slows down or accelerates particles to reach the desired temperature. This procedure is called thermostating and different variable constellations are referred to as thermostats. The default thermostat of LAMMPS, and in general the most common thermostat, is the Nosé-Hoover thermostat [48, 63]. The extended equations of motions, with target temperature T_0, are given by

$$\begin{aligned} M\ddot{q} &= -\nabla V(q) - \zeta p, \\ \dot{\zeta} &= \frac{1}{Q}\left[T(p) - N_d T_0\right], \end{aligned}$$

where N_d are the degrees of freedom and Q is a coupling parameter. The momentum vector p, is not fixed to a specific value, as with the Berendsen method but instead ζ damps or raises p if the temperature in the system is higher or lower than T_0.

After the simulation is complete, the data that LAMMPS produced during the simulation can be post processed. While we use Python for most of our tasks, we also use OVITO [78] an **O**pen **VI**sualization **TO**ol to visualize the simulation.

5.4 Construction of a Large Scale Model System

With the tools and techniques discussed above, we construct a large scale model system that is motivated by the system introduced in section 2.2.1. We define a molecule template

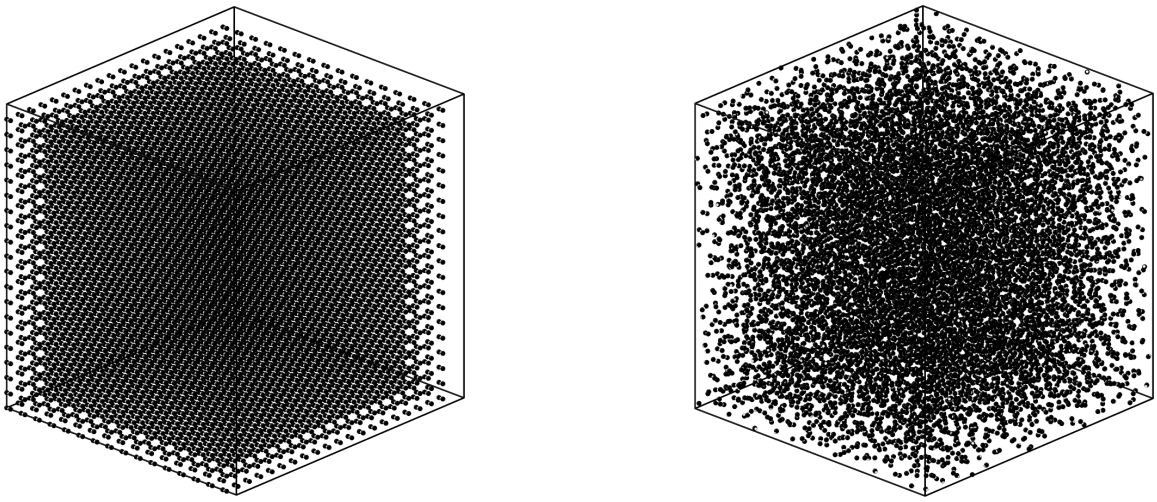


Figure 5.7. Domain with 6400 pairs of particles in a 59 Å × 62 Å × 62 Å simulation cell with periodic boundary conditions, before equilibration (left) and after (right).

for the pair of bonded particles and use Moltemplate to generate a grid of $16 \times 20 \times 20$ particle pairs and place them inside a 59 Å × 62 Å × 62 Å simulation domain with periodic boundaries in all directions. The initial position of the particles can be seen on the left-hand side of figure 5.7.

We use the harmonic bond potential and the Lennard-Jones potential with the parameters

$$\bar{\rho}_B = 1.00\,\text{Å}, \qquad k_B = 1200\frac{\text{kcal}}{\text{mol}},$$
$$\sigma = 2.97\,\text{Å}, \qquad \epsilon = 0.5\,\frac{\text{kcal}}{\text{mol}}.$$

After minimizing the energy of this 12 800 particle system, we set the temperature in the system to 10 Kelvin by changing the velocity of all particles accordingly. With the help of a Berendsen thermostat we equilibrate the system over 25 picoseconds with a set temperature of 10 Kelvin. Note that the choice of this extremely low temperature has been made deliberately to avoid unwanted near collision events due to fast moving particles. For this reason, the system can be viewed as an academic example which is constructed to show important properties of the integrators, and we refer to it in the following as our model system. In chapter 7 we will consider a range of realistic molecular dynamic systems with varying temperatures. The equilibrated system is shown on the right-hand side of figure 5.7, as rendered by OVITO. In the following chapter, we will use this system to evaluate the properties of the semi-analytical approach in molecular dynamics simulation.

6 | Implementation Details

In this chapter we discuss details of the implementation of the semi-analytical integrators as introduced in algorithms 4 to 6 inside LAMMPS. We first present the serial approach before expanding it to the parallel case. Besides the technical details of the implementation, we motivate our choice of the maximal dimension of the Krylov space for the Gautschi-type integrator and the degree of the polynomials of the Leapfrog-Chebyshev integrator. We recall the Gautschi-type integrator in the one-step formulation with a diagonal mass matrix M (cf. (3.9))

$$\begin{aligned} p^{n+1/2} &= p^n + \tfrac{\tau}{2} M^{-1/2} \operatorname{sinc}^2(\tfrac{1}{2}\tau\Omega_M^a) \left(A_M^a M^{1/2} (\tilde{q}^n - q^n) - M^{-1/2} \nabla V(\tilde{q}^n) \right), \\ q^{n+1} &= q^n + \tau p^{n+1/2}, \\ p^{n+1} &= p^{n+1/2} + \tfrac{\tau}{2} M^{-1/2} \operatorname{sinc}^2(\tfrac{1}{2}\tau\Omega_M^a) \left(A_M^a M^{1/2} (\tilde{q}^{n+1} - q^{n+1}) - M^{-1/2} \nabla V(\tilde{q}^{n+1}) \right), \end{aligned} \tag{6.1}$$

where Ω_M^a is the square root of the symmetric positive semi-definite matrix $A_M^a = M^{-1/2} A^a M^{-1/2}$. Further, the matrix $A^a = H_W(a)$ is the Hessian of the fast potentials (cf. chapter 4) evaluated at the expansion point a and $\tilde{q}^n$ the averaged position given by $\tilde{q}^n = q^e + \Phi(q^n - q^e)$ (cf. (3.8)). As discussed in section 2.3.3, we consider the choice $\Phi \equiv 1$, i.e., the force evaluation at an unfiltered position, and the choice $\Phi = \operatorname{sinc}(\tau\Omega_M^a)$.

We explain the implementation details of the one-step formulation of the Gautschi-type integrator (6.1). The implementation of the Leapfrog-Chebyshev integrator follows the same steps. Since LAMMPS is a highly optimized software package, the goal of our implementation is to use as much of the existing code as possible. We will show with suitable examples that our implementation is indeed able to not only allow for larger stepsizes but is also computationally competitive with the well established Verlet integrator. This is due to the efficient implementation of the integrators presented in this chapter.

6.1 Extending LAMMPS with Fixes

As mentioned in section 5.2.2, LAMMPS can be modified with fixes, which are C++ files with corresponding header files. The header file contains the name of the fix by which it is referred to in the LAMMPS input file. The functions defined by the fix are executed during the run routine of the Integrate class. We have already interpreted the approach (6.1) as an adaptation to the Verlet integrator in section 3.3. We thus use the implementation of the Verlet integrator in LAMMPS and add the changes to the force vector as a fix to the run routine. In order to use a fix the line

`fix <myfix>` *<mygroup>* *<fixName>* $\texttt{arg}_1, \texttt{arg}_2, \ldots$

is added to the input script. Here, *<myfix>* is a name chosen by the user and *<mygroup>* specifies the particles the fix gets applied to. The *<fixName>* determines which fix is employed. The integrators presented in this work, are implemented in four different fixes

- `trigonometric`,
- `trigonometricFull`,
- `LFC`,
- `LFCFull`.

The `trigonometric(Full)` fixes are implementations of the Gautschi-type integrator and the `LFC(Full)` fixes are implementations of the Leapfrog-Chebyshev integrator. However, the fixes do not perform the time integration by themselves. They only work in combination with the implementation of the Verlet integrator given by the LAMMPS fix `nve`, or any of the related fixes, e.g., `nvt`, `nvp`.

The suffix `Full` indicates that the full Hessian, i.e., the Hessian including the bond, angle, and dihedral angle, as well as the inner filter function of the Gautschi-type integrator can be used. For the `Full` implementation we need to introduce further modifications to LAMMPS. To offer a lightweight, parallel implementation that only relies on a single fix we introduce the non-`Full` variants, `trigonometric` and `LFC`. We implement the parallel approach for the Hessian of the bond potential in the non-`Full` variant and discuss it in section 6.4. As mentioned in section 3.2.1, we make use of the LAPACK routine '*dsteqr*' to calculate the eigendecompostion of the small Hessenberg matrix H_m. For this reason, LAMMPS has to be linked to a LAPACK library.

The user can submit arguments $\text{arg}_1, \text{arg}_2, \ldots$, to specify the input. For the lightweight, parallel implementations, `trigonometric` and `LFC`, the user can specify the maximal dimension of the Krylov subspace and the degree of the Chebyshev polynomial as well as the stabilization parameter η by adding it to the line, i.e.,

`fix myGautschi` `all` `trigonometric` $m_{\max}$,
`fix myLFC` `all` `LFC` p η.

If no arguments are submitted, the parameters are set to their default values $m_{\max} = 5$, $p = 6$, and $\eta = 1$. We discuss the motivation for these default values in section 6.3.

For the `Full` variants, the user can additionally choose which combination of the three linearization matrices presented in chapter 4 are used in the semi-analytical approach. This is done by adding the letter 'b', 'a' and 't' to include the bond, angle, and torsional potential, respectively. Further, the user can control whether the linearization gets updated over time by adding the letter 'u'. For the Gautschi-type integrator, the user may also decide if the position at which the force is calculated gets averaged with the inner filter function $\Phi = \mathrm{sinc}(\tau\Omega)$ by adding the letter 'f'. The full input lines, with all options enabled, read

```
fix myGautschi          all   trigonometricFull   b a t u f m_max,
fix myLFC               all   LFCFull             b a t u p η.
```

Options that are not desired can be turned off by replacing the respective character with the letter 'n'. The dimension of the Krylov space as well as the polynomial degree of the Chebyshev polynomials are again set to their default values if not further specified. The default values are discussed in section 6.3.3 and are given in table 6.1.

If, for example, one wants to use the Gautschi-type integrator with the updated Hessian of the bond and angle potential without an inner filter and maximal Krylov subspace dimension $m_{\max} = 8$, the input reads

```
fix myGautschi          all   trigonometricFull   b a n u n 8.
```

In order to calculate the averaged position $\tilde{q}^n = q^e + \Phi(q^n - q^e)$ the integrator needs access to a suitable equilibrium position q^e. We therefore extend the existing atom structure of LAMMPS with the atom_style `trigo`, which associates each atom with its equilibrium position q^e in addition to the standard attributes given by the atom_style `full`. We further use a fix called `trigoInit` to read in the equilibrium position of the system. The workflow of the input file might then read as follows.

```
include             "equilibrium data"

fix myInit          all   trigoInit

run 1                                 % run one step without integration to fill q^e

% load new, non equilibrium data or manipulate existing positions

fix myGautschi      all   trigonometricFull   b a n u f 8

fix integration     all   nve

run K                                 % start integration run
```

In order to get the necessary information from the harmonic angle and harmonic torsional potential we made some additional, but small changes to the respective C++ files.

6.1.1 Workflow

In the implementation of the fix we specify at which points of the run routine changes are made. For the formulation of (6.1) we first average the position at which the force is evaluated and then apply the matrix function to the force vector. Thus, we access the run routine directly before and directly after the force calculations. This is done by adding `PRE_FORCE` and `POST_FORCE` to the internal mask variable. In the `PRE_FORCE` routine we compute the averaged position vector $\tilde{q}^n = q^e + \Phi(q^n - q^e)$. LAMMPS uses this position to calculate the force. After the force calculation the `POST_FORCE` routine is executed where we set the position vector back to its original, unfiltered value and apply the matrix function $\operatorname{sinc}^2(\frac{1}{2}\tau\Omega_M^a)$ to the force vector. The rest of the integration procedure, i.e., the Verlet step with the modified force vector $\operatorname{sinc}^2(\frac{\tau}{2}\Omega_M^a)F(\tilde{q}^n)$, is then carried out by the given LAMMPS routines.

We also have to keep track of the particle ordering. Whenever a particle crosses the border of its domain, either due to periodic boundary conditions or domain decomposition from parallelization (cf. section 5.2.1), the local indices of all particles in the domain may change. Since the number of particles in the domain might also have changed, we need to reassess the memory reserved for the Hessian matrix. This is done by adding `POST_NEIGHBOR` to the mask variable. The corresponding routine is executed after LAMMPS internally decides to update the neighbor structure.

We sketch the workflow for an integration with K steps with the fix `trigonometricFull` in a flowchart and mark our contributions in red. The flowchart with the fix `LFCFull` has the same structure but omits the averaging step in the `PRE_FORCE` routine. Note that we were able to use many of the optimized LAMMPS routines for our integration. Despite the fact that we consider a different integration approach, the implementation makes only very small changes to the core code of LAMMPS. For this reason, our implementation conserves the easily modifiable structure of LAMMPS and can be used for many applications.

6.2 Efficient Implementation of the Matrix-Vector Product

The computational bottleneck of the semi analytical approach is the approximation of matrix valued functions $f(A)$ applied to a vector b. As we have discussed in section 3.2, the core of an efficient computation lies in an efficient matrix-vector product evaluation. In order to achieve this we make use of the structure of the Hessian matrix. We use the Hessian of the bond potential to explain the details of our approach. The approach is, however, not limited to the bond potential. We followed the same idea for the Hessian matrices of the angle and dihedral angle in the implementation of the `Full` variants.

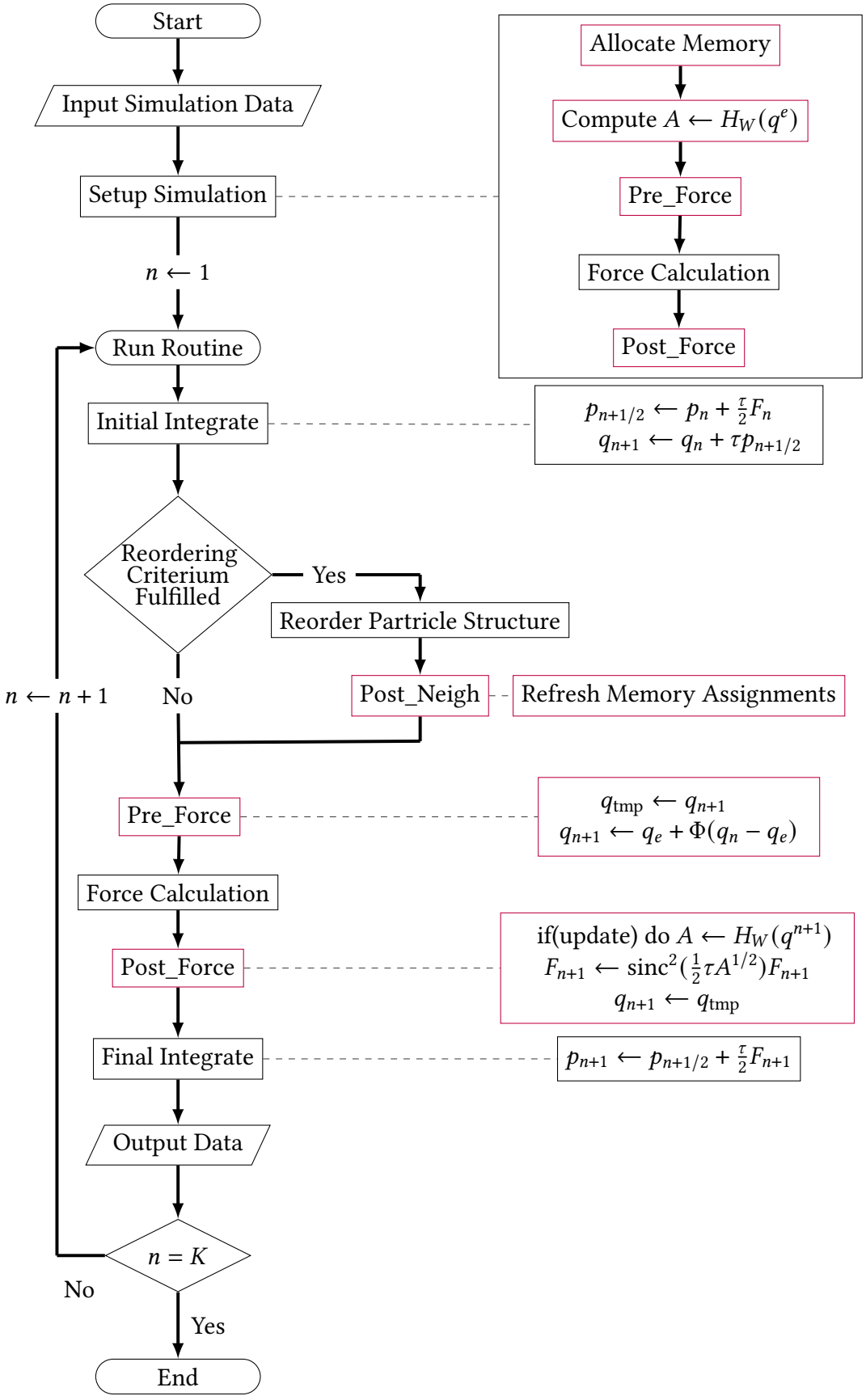

Start
Input Simulation Data
Setup Simulation
$n \leftarrow 1$
Run Routine
Initial Integrate
Reordering Criterium Fulfilled
Yes
Reorder Partricle Structure
Post_Neigh
Refresh Memory Assignments
No
$n \leftarrow n+1$
Pre_Force
Force Calculation
Post_Force
Final Integrate
Output Data
$n = K$
No
Yes
End
Allocate Memory
Compute $A \leftarrow H_W(q^e)$
Pre_Force
Force Calculation
Post_Force
$p_{n+1/2} \leftarrow p_n + \frac{\tau}{2} F_n$
$q_{n+1} \leftarrow q_n + \tau p_{n+1/2}$
$q_{\text{tmp}} \leftarrow q_{n+1}$
$q_{n+1} \leftarrow q_e + \Phi(q_n - q_e)$
if(update) do $A \leftarrow H_W(q^{n+1})$
$F_{n+1} \leftarrow \text{sinc}^2(\frac{1}{2}\tau A^{1/2}) F_{n+1}$
$q_{n+1} \leftarrow q_{\text{tmp}}$
$p_{n+1} \leftarrow p_{n+1/2} + \frac{\tau}{2} F_{n+1}$

Figure 6.1. One-dimensional system with three bonded particles.

In one dimension, the Hessian matrix H_{V_B} of the bond potential has only nonzero entries at index-pairs (i, j) corresponding with particles i and j that share a bond and at (i, i) for $i = 1, \ldots, N$. As a result it is sparse, i.e., most of its entries are zero. We can save significant storage by not storing the nonzero entries. A common approach is to store the nonzero entries (nnz) in a contiguous memory location and access a specific entry by storing the order of appearance in their respective row. Storing the $n \times n$ matrix explicitly in such a sparse matrix format means that we only need to store $2\text{nnz} + n + 1$ instead of the original n^2 entries (see appendix A.1 for more details). There are two downsides to this approach. For one, we need an explicit ordering of all rows and columns, i.e., of all particles in the system. Since the particles are regularly changing their identification number when moving from one domain to another, this would force us to reconstruct the complete sparse matrix. A second downside is the fact that we need an indirect addressing step for every scalar operation of the matrix-vector product. For these reasons, we follow an alternative approach that is closely linked to the topology of the molecular system.

We do not store the structure and entries of the complete Hessian matrix explicitly. Instead, we store the entries of the $2d \times 2d$ matrix belonging to a bond and perform the matrix-vector product bond wise. In that sense, the full matrix is stored implicitly and is not linked to an explicit particle order. To illustrate this we consider a one-dimensional line of three bonded particles in one dimension as shown in figure 6.1. For this 3-particle system, the full Hessian matrix has the structure

$$H_{V_B} = \begin{bmatrix} \triangle & -\triangle & 0 \\ -\triangle & \triangle + \triangledown & -\triangledown \\ 0 & -\triangledown & \triangledown \end{bmatrix} \in \mathbb{R}^{3\times 3}, \qquad \text{with} \quad \begin{aligned} \triangle &= \nabla^2_{q_1 q_1} V_{B_1} \in \mathbb{R}, \\ \triangledown &= \nabla^2_{q_2 q_2} V_{B_2} \in \mathbb{R}, \end{aligned}$$

where the entries $\triangle$ come from the first and the entries $\triangledown$ come from the second bond. Instead of storing the full matrix above, we only store the values $\triangle$ and $\triangledown$. This allows us to compute the matrix-vector product for an N particle system with N_B bonds as shown for one dimension in algorithm 7.

This procedure can be extended for general dimensions and to allow for potentials containing more than two particles, such as the angle or torsional potential by looping over the respective topology lists. Computing the matrix-vector product in this way offers several advantages. We only need to consider the topology list which is maintained by LAMMPS for the regular force computation. For each bond we need a constant number of computations. Since the number of bonds in the system is always less or equal to the number of particles N, the computation has a complexity of $O(N)$. This is a significant improvement to the complexity of the full matrix-vector product, $O(N^2)$. We will later see that this formulation also allows for easy parallelization.

Algorithm 7 Implicit matrix-vector product

Input vector $v \in \mathbb{R}^N$; bond-wise Hessian entries $h_B \in \mathbb{R}^{N_B}$

out $\leftarrow 0 \in \mathbb{R}^N$
for each bond $b \in \mathcal{B}$ **do**
    i $\leftarrow$ index of first particle in bond b
    j $\leftarrow$ index of second particle in bond b
    out$[i]$ $\leftarrow$ out$[i]+h_B[b]v[i]$
    out$[i]$ $\leftarrow$ out$[i]-h_B[b]v[j]$
    out$[j]$ $\leftarrow$ out$[j]-h_B[b]v[i]$
    out$[j]$ $\leftarrow$ out$[j]+h_B[b]v[j]$
end for

Output out $= H_{V_B} v$

6.3 Application of the Semi-Analytical Approach in LAMMPS

Now that we have discussed the technical details needed for constructing and implementing the semi-analytical approach in LAMMPS, we perform the first set of experiments. The experiments are designed to visualize the possible increase in stepsize of our approach and to further motivate some choices that we have made to optimize the costs of the integration. We first revisit the large scale model problem introduced in section 5.4 and then consider a graphene sheet under deformation.

6.3.1 Application to the Model Problem

In section 3.2, we have seen that the convergence of the exponential matrix function is slow for general skew-Hermitian matrices. Choosing the stopping criterion $m = 2\rho\tau$ (cf. lemma 2) would lead to many iterations and would make the matrix function approximation too expensive. During the extensive testing of the integration scheme we were able to determine a suitable choice for our specific application. Note that this default choice has proven sufficiently good for all considered application. However, we provide the user with the option to override the default choice with a different value that might be determined with a suitable error estimator. We motivate our default choice with the academic example introduced in section 5.4.

We use a fraction of the 12 800-particle system described in section 5.4 with 200 particles in a 14.75 Å × 15.5 Å × 15.5 Å simulation domain, as shown on the left of figure 6.2, to motivate the use of a much smaller $m_{\max}$. Since the particles are translating and rotating freely, we consider the updated Hessian matrix and cannot use our averaging approach (cf. section 4.5). After computing a reference solution with the Verlet integrator and stepsize $\tau = 10^{-4}$ fs, we integrate the system for 100 fs with the unfiltered Gautschi-type integrator

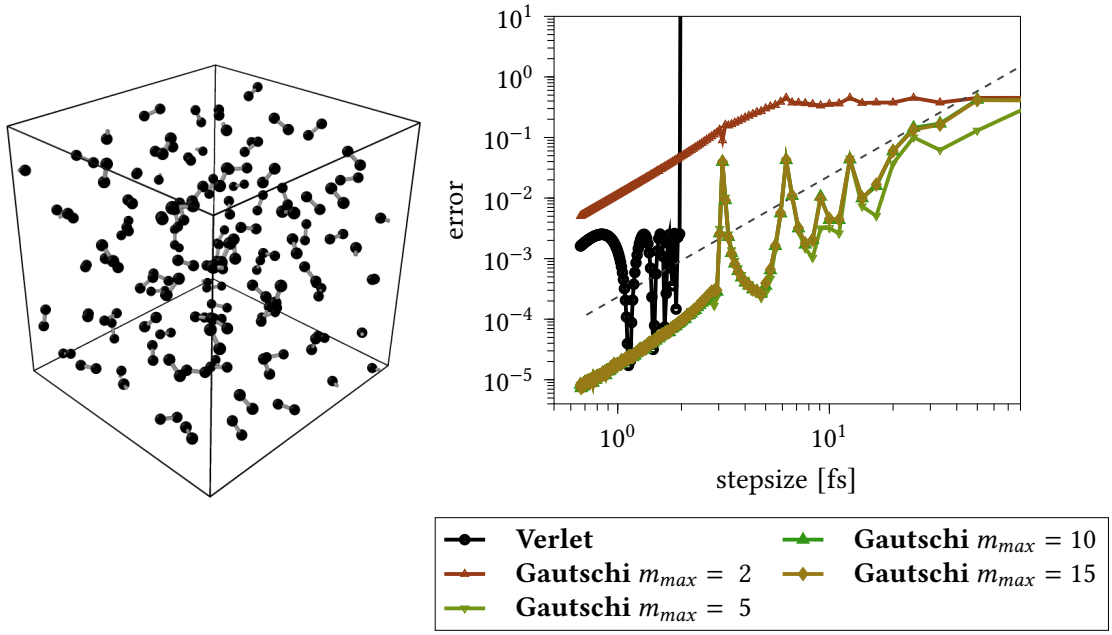


Figure 6.2. Global error at $t = 100\,\text{fs}$ for the Verlet and unfiltered Gautschi-type integrator with varying Krylov subspace dimensions when applied to the small equilibrated model system (left) as a function of the stepsize τ[fs] (right) with a gray dashed reference line with slope 2.

(cf. algorithm 4) with updated Hessian and vary the maximal dimension of the Krylov subspace. On the right-hand side of figure 6.2 we plot the error with respect to the reference solution as a function of the stepsize. While the Gautschi-type integrator with $m_{\text{max}} = 5$ has a significantly smaller error constant compared to the integrator with $m_{\text{max}} = 2$, there is only a small difference between the error plots of the Gautschi-type integrator with maximal Krylov dimensions larger than $m_{\text{max}} = 5$. This observation has been confirmed for all systems that we considered with the unfiltered Gautschi-type integrator and the regularly updated Hessian of the bond potential. For this reason, we use for this integrator the Krylov subspace with maximal dimension $m_{\text{max}} = 5$ as the default value.

In order to get an estimate on the needed polynomial degree p of the Chebyshev polynomials in the Leapfrog-Chebyshev scheme, we integrate the 200-particle system as shown on the left of figure 6.2 for 100 fs with the Leapfrog-Chebyshev scheme. We use different polynomial degrees for the Leapfrog-Chebyshev integrator with $\eta = 0$ and $\eta = 1$. We plot the error as a function of the stepsize in figure 6.3 together with the error of the unfiltered Gautschi-type integrator. For polynomial degree $p > 4$ and $\tau < 2$, the Leapfrog-Chebyshev schemes perform with similar accuracy. We can see that this changes for larger stepsizes. The stability region of the integrator depends on the degree of the Chebyshev polynomials. As expected, the error plot of the unstabilized Leapfrog-Chebyshev scheme, i.e., $\eta = 0$, gets closer to the error plot of the Gautschi-type integrator with increasing degree. However, even with polynomial degree $p = 16$, the Leapfrog-Chebyshev scheme has a lower maximal stepsize. These considerations motivated the default value $p = 6$ for the Leapfrog-Chebyshev with the Hessian of the bond potential. On the right-hand side of figure 6.3, we can see the effect of the stabilization term. While the Gautschi-type integrator has

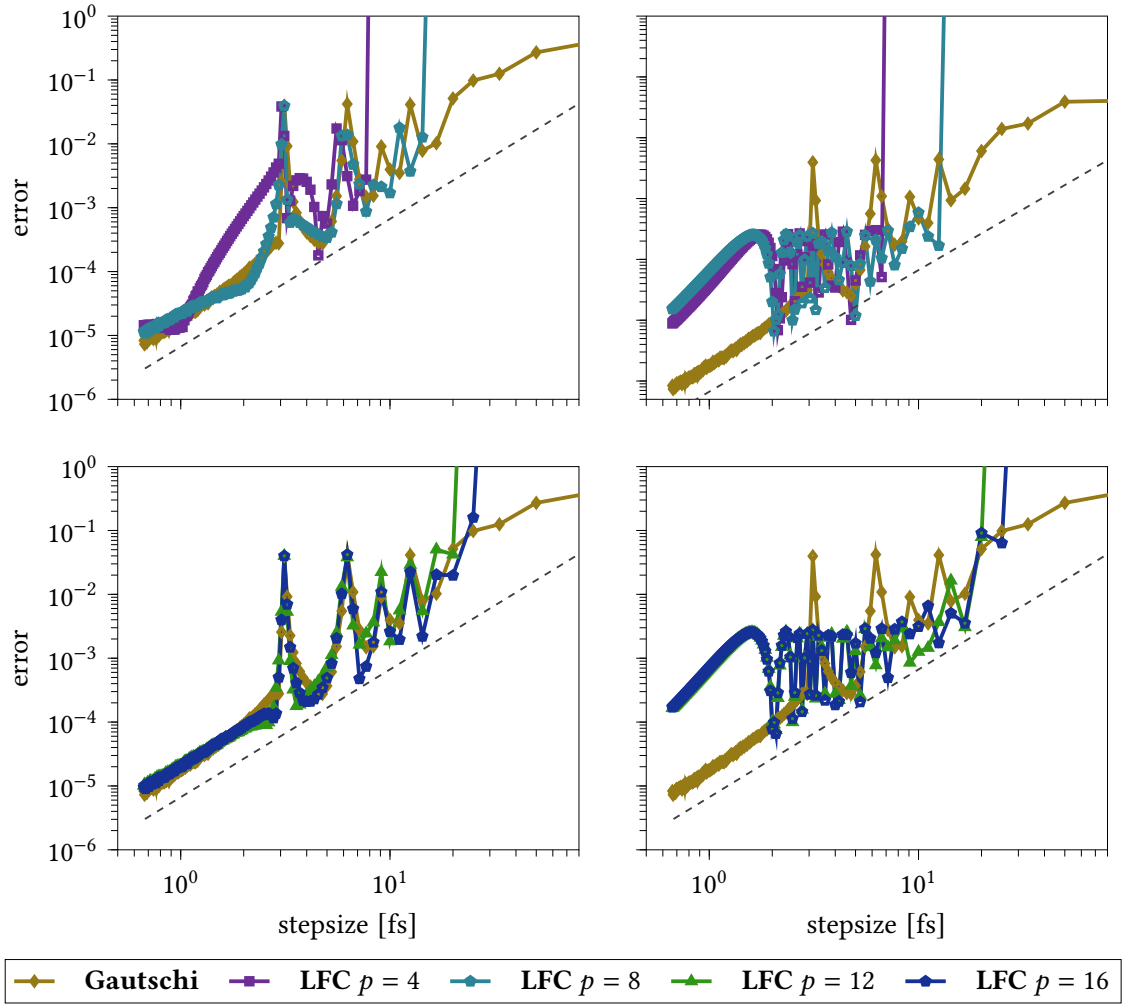


Figure 6.3. Global error at $t = 100$ fs for the unfiltered Gautschi-type and Leapfrog-Chebyshev integrators with varying polynomial degrees ($p = 4, 8$ on the top and $p = 12, 16$ at the bottom) when applied to the equilibrated model system (cf. section 5.4), with $\eta = 0$ (left) and $\eta = 1$ (right), as a function of the stepsize τ[fs]. All four plots contain a gray dashed reference line with slope 2.

three peaks due to numerical resonances, the error of the stabilized Leapfrog-Chebyshev integrator stays bounded. We will come back to this phenomenon in section 7.1.

To get an idea of the efficiency of the semi-analytical approach we apply the Gautschi-type and unstabilized Leapfrog-Chebyshev integrators with the default values $m_{\max} = 5, p = 6$ to the full 12 800-particle system described in section 5.4 and compare them to the Verlet scheme over 40 fs. Since the system is too large to compute the global error of all particles, we pick a random but fixed particle and calculate the error in its position. We compute a reference solution with the Verlet integrator and the stepsize $\tau = 10^{-4}$ fs and plot the errors of the integrators with respect to this reference solution in figure 6.4.

One can see that the semi-analytical approach is computational very competitive. This is due to two reasons. First, the main effort, with over 99% of the computational time, in the standard Verlet scheme are long-range Lennard-Jones calculations. This means that the

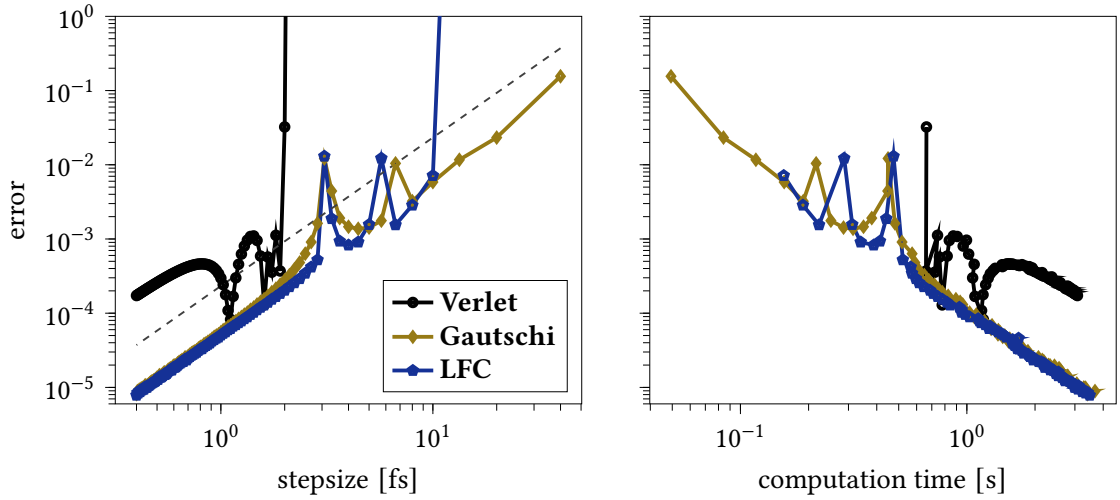


Figure 6.4. Global error at $t = 40\,\mathrm{fs}$ for the Verlet, unfiltered Gautschi-type and Leapfrog-Chebyshev integrators when applied to the equilibrated model system (cf. section 5.4) as a function of the stepsize τ[fs] with a gray dashed reference line with slope 2 (left) and computational time in seconds (right)

potential function $V(q)$ is well separated in fast, cheap potential $W(q)$ and slow, expensive potential $U(q)$. The second reason is the efficient implementation of the semi-analytical approach. In the simulation only roughly 1.8% of the total computation time is spent on additional computations. As a result, for the same stepsize, both the Gautschi-type and the Leapfrog-Chebyshev integrators, are only around 1.8% slower than the Verlet method, while having a significantly higher accuracy. The semi-analytical approach additionally provides a stable solution with larger stepsizes than the Verlet method. There is almost no difference in the computational time between the Gautschi-type integrator and the Leapfrog-Chebyshev integrator. The main effort, namely the calculation of the linearization and the matrix vector product, coincide for the two schemes. We expand on these results in section 7.1.

The results presented have to be taken with a grain of salt. The performance of the semi-analytical approach, as we will later see, is heavily dependent on the underlying system. We can also observe numerical resonances in figure 6.4. The effect of these resonances gets stronger for longer simulations. While we have already discussed how to handle these numerical resonances for the Gautschi-type integrator, we have no access to a suitable equilibrium position q^e. Since this position is needed for our inner filter approach, the stepsize is limited by these numerical resonances.

6.3.2 Application to a Graphene Sheet

We consider the deformation of a graphene sheet as a second example. A graphene sheet is a single layer of carbon atoms arranged in a hexagonal lattice. We model the interactions

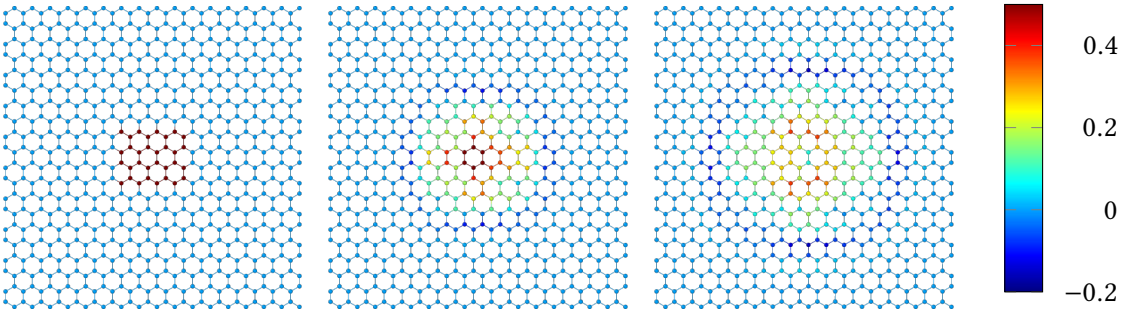


Figure 6.5. Propagation of an initial impulse in the middle of a graphene sheet with 680 carbon atoms at 0 fs (left), 50 fs (middle) and 100 fs (right) as simulated by the Verlet integrator with $\tau = 10^{-4}$ fs and visualized with OVITO.

between the bonded particles with the three harmonic intramolecular potentials $V_B(\rho_{ij})$, $V_A(\theta_{ijk})$, and $V_T(\phi_{ijkl})$ discussed in section 2.2 with the parameters

$$\begin{aligned} \bar{\rho}_B &= 1.418, & \bar{\theta} &= 120, & \bar{\phi} &= 0, \\ k_B &= 938, & k_A &= 126, & k_T &= 3.625. \end{aligned}$$

We arrange 680 carbon particles in a hexagonal lattice with dimension 40.5 Å × 41.12 Å. The particles are in their equilibrium position q^e, meaning that the force in the system is, up to round-off errors, zero. We introduce an initial impulse to the system by deflecting 32 particles in the middle in z-direction by 0.5 Å. A visualization of the initial system, made with OVITO [78], is given in figure 6.5.

We generate a reference solution with the Verlet integrator with stepsize $\tau = 10^{-4}$ fs and the plot the simulation after 0 fs, 50 fs, and 100 fs in figure 6.5. The initial impulse is propagated through the graphene sheet over 100 fs. While the individual particles are oscillating freely around the bond, angle and dihedral angle, the overall stability of the graphene sheet ensures that the particles keep their original orientation. As a consequence, we are able to use a constant linearization of the fast potentials and have access to the equilibrium position at every step which allows us to use the filtered Gautschi-type integrator.

We integrate the system with the Verlet method, the unfiltered Gautschi-type and filtered Gautschi-type integrator, as well as with the stabilized and unstabilized Leapfrog-Chebyshev method. Similar to the example from section 4.4 we consider the linearization with the Hessian matrix of just the bond potential as well as the combination with the Hessian of the bond and angle potential, and bond, angle, and torsional potential. We use the same notation as in (4.25), this means that we consider the three matrices

$$A_1 = -H_{V_B}(q^e), \quad A_2 = -H_{V_B}(q^e) - H_{V_A}(q^e), \quad A_3 = -H_{V_B}(q^e) - H_{V_A}(q^e) - H_{V_T}(q^e).$$

We plot the error, with respect to the reference solution, made by the Verlet integrator and the Gautschi-type integrator with different linearization matrices as a function of the stepsize on the top of figure 6.6. On the top left, there is no inner filter, i.e., $\Phi \equiv 1$. For small stepsizes the unfiltered Gautschi-type integrators are more accurate than the Verlet

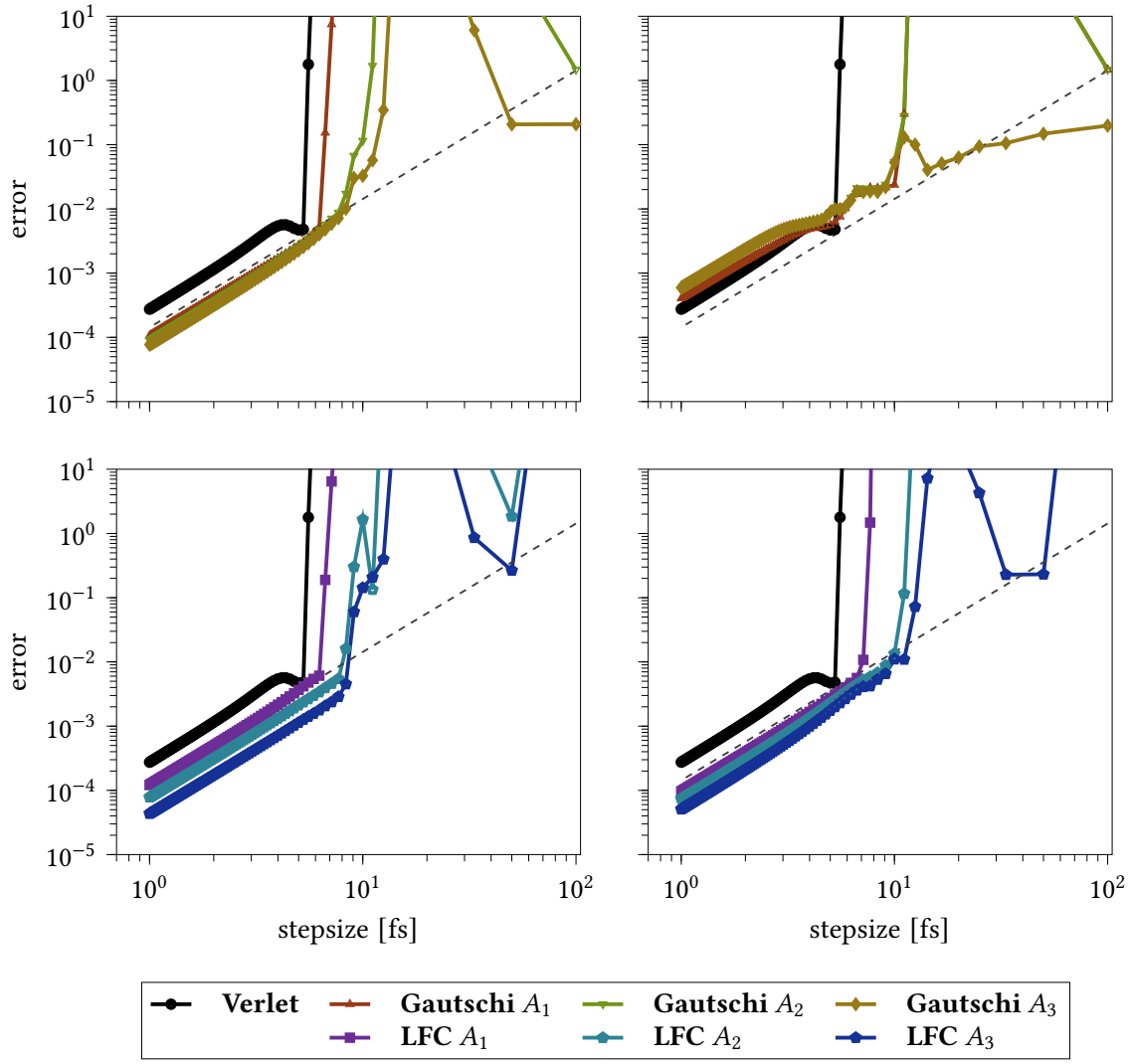


Figure 6.6. Global error at $t = 100$ fs for the Verlet, unfiltered (top, left) and filtered (top, right) Gautschi-type integrator and Leapfrog-Chebyshev integrator, with $\eta = 0$ (bottom, left) and $\eta = 1$ (bottom, right), when applied to the graphene sheet as a function of the stepsize τ[fs] and a gray dashed reference line with slope 2.

method for the same stepsize. However, due to the numerical resonances the unfiltered Gautschi-type integrator does not allow for significantly larger stepsizes. As discussed, the stepsize can be increased by employing an inner filter function, $\phi(\xi) = \text{sinc}^2(\xi)$, and thus averaging the position at which the nonlinear part is evaluated. We can see on the top right of figure 6.6 that this idea is successful even for larger systems such as this graphene sheet. We will revisit this example with a focus on the computational cost involved in performing a step of the Gautschi-type integrators as well as their behavior over large number of timesteps in section 7.3.

On the bottom of figure 6.6 one can see on the left that the error of the unstabilized Leapfrog-Chebyshev method shows a similar behavior as the error of the unfiltered Gautschi-type integrator. In comparison to the filtered Gautschi-type integrator, the stabilized Leapfrog-

	H_{V_B}	$H_{V_B} + H_{V_A}$	$H_{V_B} + H_{V_A} + H_{V_T}$
Unfiltered Gautschi $m_{\max}$	5	5	5
Filtered Gautschi $m_{\max}$	10	10	10
Leapfrog-Chebyshev p	6	8	10

Table 6.1.: Default values of the filtered and unfiltered Gautschi-type and the Leapfrog-Chebyshev integrators depending on the linearization matrix.

Chebyshev method suffers from numerical resonances for larger stepsizes as shown on the right of figure 6.6.

6.3.3 Default Values for the Gautschi-type and Leapfrog-Chebyshev Integrators

In section 6.3.1 we motivated the default values for the dimension of the Krylov subspace $m_{\max} = 5$ of the unfiltered Gautschi-type integrator and polynomial degree $p = 6$ for the Leapfrog-Chebyshev integrator for problems that include the linearization of the bond potential. We extend these default values for the filtered Gautschi-type integrator as well as for problems that additionally include linearizations of the angle and torsional potential and collect the values in table 6.1. The default value for the dimension of the Krylov subspace for the unfiltered Gautschi-type integrator is independent of the specific combination of the three linearization matrices. We did not see a significant improvement when choosing larger maximal dimensions. However, the filtered Gautschi-type integrator needs a higher accuracy for the Krylov subspace approximation. We therefore set the default value of the filtered Gautschi-type integrator to $m_{\max} = 10$ for all configurations of the linearization matrix. The stability region of the Leapfrog-Chebyshev integrator depends directly on the degree of the polynomials, as we can observe for example in figure 6.3. In figure 4.4 we show that the size of the largest possible stepsize for the Gautschi-type integrator increases with each added linearization matrix. To obtain this behavior with the Leapfrog-Chebyshev integrator we increase the default value for the polynomial degree p with each additional linearization.

If not further specified, all following integrations are performed with these default values.

6.4 Efficient Parallelization

Parallelization is a significant part in accelerating the computations of molecular dynamics simulations. In order to apply the integrator to relevant examples from the applied sciences it was imperative to develop an efficient parallel implementation. We again use as many of the given structures in LAMMPS as possible. We first present important, elementary concepts of parallel implementation with MPI before turning to the techniques used in the implementation of the semi-analytical integrators.

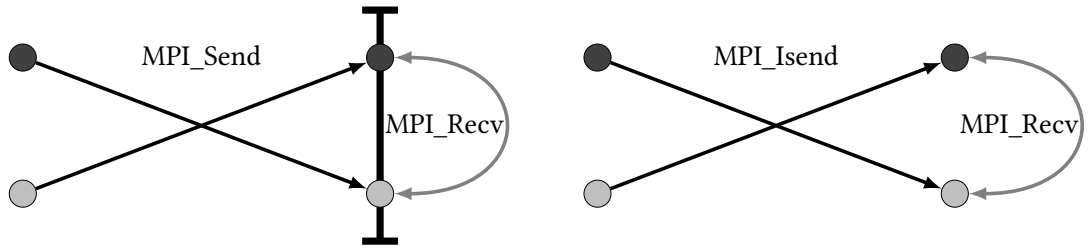


Figure 6.7. Thread 1 (black) and thread 2 (gray) are sending to and receiving information from each other. The Communication on the left is blocked, since both threads are waiting for the other to confirm that the message is received. On the right, the threads do not wait for the receive confirmation and the communication is successful.

6.4.1 Parallelization with MPI

In parallel computing a large task gets decomposed into several smaller parts. Each part gets assigned to a thread. The key element of working with multiple threads on the same problem is the communication between the threads. *MPI*, the *Message Passing Interface*, sets a widely used standard for message-passing since June 1994. It is governed and expanded at the *MPI Forum* (mpi-forum.org). The focus of this section are not the technical details but the general workings of MPI. For more details we refer to the official documentation [23].

Point-to-Point Communication

Each thread in the parallel architecture can *send* a message to and *receive* a message from every other thread. There is differentiation between *blocking* and *non-blocking* communication. There are benefits and drawbacks to both approaches.

Blocking Communication

With `MPI_Send` a blocking send is executed. The call only returns when the operation has been completed, that is the message has been received. Every call made after this blocking send can be sure that it is safe to use and alter the buffer with the sent data. The procedure also results in the sending thread being dependent on the receiver. The sender has to wait until the receiver successfully handled the message. Analogously, `MPI_Recv` only continues after the asked for message has been received. Combining a blocking send with a blocking receive command leads to a deadlock, since the threads are waiting for each other to receive confirmation. We illustrate the concept on the left of figure 6.7.

Non-Blocking Communication

The non-blocking send, executed by `MPI_Isend`, returns immediately after the call has been made. The sender can continue independently of the receiver. There is however, no knowledge whether the operation has been completed. As a consequence, working on the sent data might cause conflicts. Again, the `MPI_Irecv` call works analogously. In practice one often combines blocking and non-blocking communication as visualized in the right-hand side of figure 6.7.

Datatype Matching

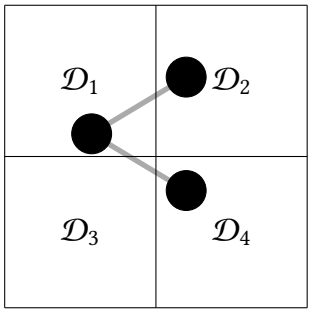
For a successful transmission of data there needs to be an agreement over the size and type of the passed message between the involved threads. With each `MPI_(I)Send` the number and datatype of sent elements has to be specified. This information has to match with the corresponding `MPI_(I)Recv` command. As a result, additional communication might be necessary in order to specify these details ahead of the main communication.

6.4.2 Collective Communication

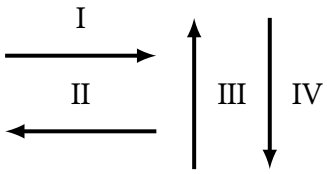
In addition to Point-to-Point communication MPI allows for collective communication. Collective communication involves a group of multiple threads. Apart from that it is very similar to the standard Point-to-Point communication. There are blocking and non-blocking variations. If blocking communication is used, collective communication additionally synchronizes all participating processes. The datatype must match between sending and receiving processes as described above. For some collective routines there is a single thread sending (receiving) to (from) all other threads. This thread is called the *root*. An example, used in this work, is broadcasting, where the root sends information to all other threads.

6.5 Parallel Semi-Analytical Integration

For the matrix function approximation in the Krylov subspace we use the symmetric Lanczos algorithm to compute the basis of the Krylov space (cf. section 3.2). There are three distinct operations that require communication between threads. The sparse matrix-vector product, the full inner product and the computation of the norm. The parallelization of the norm-computation of a vector, as well as the inner product of two vectors that are equally distributed over multiple threads is a standard procedure. Every thread computes the local operation on their part of the vector and the terms are summed up with a collective communication. The computation of an efficient, parallel, sparse matrix-vector product is a more challenging task. We present our approach that makes use of the specific structure of molecular dynamics simulations. The considerations for the matrix-vector product are



(a) Simulation domain decomposed in four subdomains



(b) Order of communication in 2 dimensions

Figure 6.8. Visualization of communication details in a 2-dimensional system.

of course also applicable for the matrix vector products in the Leapfrog-Chebyshev scheme.

Consider the system displayed in figure 6.8a, where the domain $\mathcal{D}$ is split up in four subdomains, $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ and $\mathcal{D}_4$ and contains three bonded particles. The particles are each located in a different subdomain, $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_4$. Per default, LAMMPS assigns a bond uniquely to one subdomain (cf. section 5.2.1). Let both bonds be assigned to the subdomain $\mathcal{D}_1$. As a result only this subdomain has information about the existence and properties of the bonds. In order to maintain the approach presented in algorithm 7 we have to establish communication between the three subdomains. We make use of two separate communications. First the thread that has ownership of the bond, i.e., subdomain $\mathcal{D}_1$, informs the other subdomains, $\mathcal{D}_2$ and $\mathcal{D}_4$, about the existence and type of the shared bond. With this information and the particles in the ghost region all subdomains can compute the Hessian entry corresponding to that bond. Before every matrix vector product, the entries in the vector belonging to the particles in the shared bond are communicated. Only then may both subdomains compute the entry of the matrix-vector product that corresponds to their particle.

Note that the first communication procedure, i.e., sharing the information about the bonds, only needs to be repeated when the neighborhood list is updated. In contrast, the second procedure is repeated for every matrix-vector product. For that reason, we spend more time on carefully setting up the communication path in the first step in order to allow for a fast communication in the second step.

Since the efficient implementation of the aforementioned procedure is integral for a fast and parallel integration scheme, we present the details of the implementation here. We limit the explanation to the two-dimensional case, but give a short note on the three-dimensional case in the end.

The data transfer between two threads requires a communication channel between the threads. Establishing and terminating the communication channel between threads takes a significant amount of time. Consequently, we want to use the existing channels as efficient as possible and open as few communication channels as possible. A common communication strategy, and the one that LAMMPS as well as our approach follows, admits a fixed communication order, as sketched in figure 6.8b. Every thread first sends

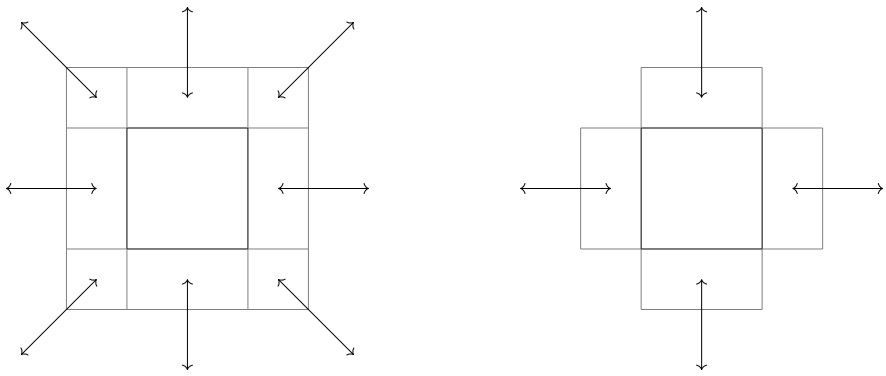


Figure 6.9. The directions of communications of a subdomain. The necessary communication channels are sketched as an extension to the domain. The sketch on the left includes the diagonal communication channels. The subdomain on the right only uses communications in x and y directions.

data to its right neighbor, therefore receiving data from its left neighbor. This procedure is then repeated with the left, top, and bottom neighbor and ensures that no blocking occurs. Note that there is no diagonal communication in this procedure, e.g., from bottom left to top right. While it is technically possible to establish a diagonal communication channel, it is not efficient. As shown in figure 6.9, including the diagonal communications would double the total number of communication channels. If we want to transfer information between the subdomains $\mathcal{D}_1$ and $\mathcal{D}_4$ in figure 6.8a, we use the subdomains $\mathcal{D}_2$ and $\mathcal{D}_3$ as *bridge* domains. Instead of the direct path

$$\mathcal{D}_1 \rightleftarrows \mathcal{D}_4$$

we use the indirect paths

$$\mathcal{D}_1 \to \mathcal{D}_2 \to \mathcal{D}_4 \qquad \mathcal{D}_4 \to \mathcal{D}_3 \to \mathcal{D}_1,$$

as shown in figure 6.10. While the amount of data that gets sent and received is twice as large as in the direct approach, we only use communication channels that are already needed for the data transfer between the other subdomains. However, before the data can be transferred in this way, the subdomains $\mathcal{D}_2$ and $\mathcal{D}_3$ need to be informed about the incoming data and where it is supposed to be sent to. We present the procedure of setting up the communication in algorithm 8.

Algorithm 8 Communication Setup

```
for each bond ∈ 𝓑 do
    if bond is heterogeneous then
        Find subdomain that owns foreign particle
        Create direction vector commDir∈ ℤ^d
    end if
end for
Communicate the existence of the diagonal bonds
Sort new diagonal bonds
Communicate the existence and type of all heterogeneous bonds
Post process                ▷ Assign needed memory, sort bonds, calculate Hessian
```

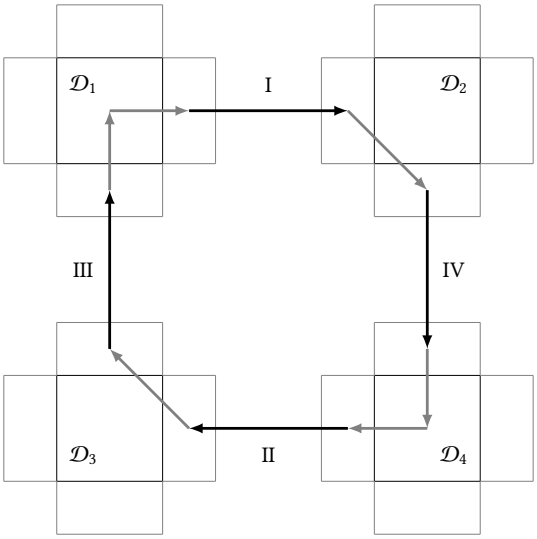


Figure 6.10. Visualization of the communication path for the bonded particles with origins in $\mathcal{D}_1$ and $\mathcal{D}_4$ (compare figure 6.8a) where the necessary communication channels are sketched as an extension to the domain

Every thread checks for each of its owned bonds if it is a *heterogeneous* bond, i.e., a bond between particles in different subdomains. After localizing the subdomain of the foreign particle in the bond, the relative direction to this subdomain is determined. We store the relative direction from subdomain $\mathcal{D}_i$ to the subdomain $\mathcal{D}_j$ as a d dimensional vector $commDir_{ij}$ with one entry with value $-1, 0$ or 1 for each dimension of the subdomain grid. The first entry corresponds to the x direction and is 1, when the subdomain with the foreign particle is on the right, 0 when it is on the same x level, and -1 when it is on the left. The consecutive entries correspond to the y and z direction. We follow the order of communication specified in figure 6.8b. In the system in figure 6.8a, this means that the subdomain $\mathcal{D}_2$ can be reached from $\mathcal{D}_1$ with $commDir_{12} = [1, 0]$ and subdomain $\mathcal{D}_4$ with $commDir_{14} = [1, -1]$.

We inform the threads of the subdomains with foreign particles of the existence of the bond in two steps. First we communicate the existence of the diagonal bonds, i.e., the bonds that use neighboring subdomains as bridges to the destination. This step informs the thread governing the destination subdomain and the bridge threads about the existence of the bond. In the second step we repeat this procedure and include the heterogeneous bonds between particles in neighboring subdomains. Note that the direction from $\mathcal{D}_j$ to $\mathcal{D}_i$ is given by $commDir_{ji} = -commDir_{ij}$ (compare figure 6.10). This means that for non-direct communications, the path back to the origin subdomain follows a different route. After the second step, every thread knows which data it receives and which it sends to its neighboring threads.

Every thread takes this information and sorts it into its existing lists that govern the bonds and calculate the Hessian entry corresponding to this bond. Additional memory that is needed to store the incoming data may also be allocated at this point.

This elaborate scheme allows us to efficiently communicate the entries in the vector belonging to the particles in the shared bonds. Since every thread knows exactly which data it will receive from which thread and which data it sends to its neighboring threads, the communication of the full vector is straight forward. We present the pseudocode in algorithm 9, where N_{local} is the number of particles that are in the subdomain.

Algorithm 9 Vector Communication

```
for n = 1, 2, 3 do                          ▷ dimensions of the domain grid
    for d = 1, 2 do                         ▷ left, right; up, down; front, back
        Send data to neighbor
        Receive data from neighbor
        Sort incoming data according to destination
    end for
end for
```

This procedure sends only the required amount of data with the least amount of communication steps possible.

In a three-dimensional grid of subdomains we have to consider an additional communication direction. While the general procedure as described in algorithms 8 and 9 does not change, particles that share a bond may now lie in two subdomains $\mathcal{D}_i, \mathcal{D}_j$ that are only reachable through two other subdomains, e.g., $commDir_{ij} = [1, 1, 1]$.

One measurement of parallel efficiency is the total speedup in relation to the number of threads that are involved in the computation. If T_s is the serial computation time and T_p the computation time with parallelism, the total speedup due to parallelism is given by

$$speedup = \frac{T_s}{T_p}.$$

In general, one cannot expect a one to one relation with the number of parallel threads. This is on one hand due to the necessary and costly communications between threads, and on the other due to the parts of the algorithm that cannot be parallelized at all.

6.5.1 Parallel Semi-Analytical Integration of the Model Problem

To showcase the parallel efficiency of our approach, we consider again the 12 800-particle system introduced in section 5.4. The particles in this system are evenly spread around in the subdomains which makes the system very well suited for parallelization. We apply the Verlet, the stabilized Leapfrog-Chebyshev, and the unfiltered Gautschi-type integrator with their default values and updated Hessian to the system with a stepsize of 1 fs. After executing the integration for 10 000 steps, i.e., 10 ps, with 1 to 10 threads we compute the speedup for each of the three integrators in comparison to its serial execution. In figure 6.11, we plot the result of the relative parallel speedup as a function of the number of threads.

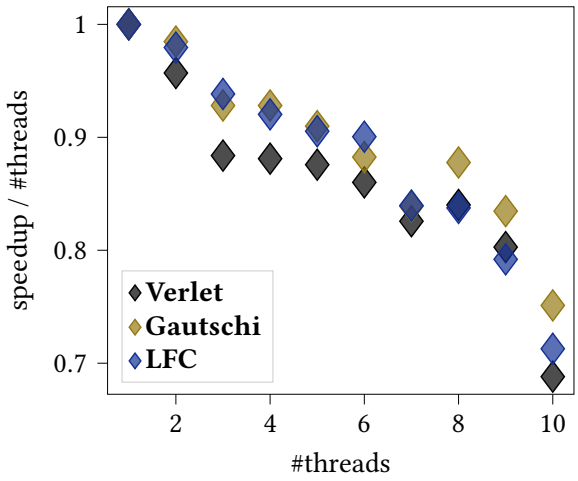


Figure 6.11. The relative parallel speedup of the Verlet, unfiltered Gautschi-type and stabilized Leapfrog-Chebyshev integrators as a function of the number of threads.

As seen in figure 6.11, the relative speedup decreases for every added thread for all three integrators, since the overhead cost due to the communication increases as discussed before. However, the rate of the decrease varies a bit. The Gautschi-type integrator has the overall best parallel efficiency, closely followed by the Leapfrog-Chebyshev integrator. The main reason is that the total computational effort for the semi-analytical integrators is larger. This computational effort is, for the most part, concentrated on the matrix-vector product, which is a mostly local operation (cf. section 6.2). Thanks to the efficient parallel implementation, the additional effort due to communication is low compared to the effort of evaluating and communication the particle interactions. These have to be performed independent of the integrator and are responsible for the main costs of parallelization.

In general, the parallel efficiency depends on the structure of the system. However, this example shows that our parallel approach is very well suited for the application in molecular dynamics and allows the semi-analytical approach to have a competitive parallel performance. In all applications that we run during the development of the integrators, the parallel efficiency of the Verlet integrator did not outperform our parallel implementation of the semi-linear approach. In the next chapter we will take a closer look at the distribution of the computational effort in the context of long-time integrations.

7 | Long-Time Behavior and Validation

In the previous chapters we presented the theory, properties, and implementation details of the semi-analytical approach and now focus on the application to long-time simulations. In contrast to the simulations over relatively short time periods, we do not compare the solution to a reference solution. Instead, we consider known physical properties of the studied systems which we compare with the approximations made with the semi-analytical approach and the Verlet method. At first, we take a further look at the large scale system constructed in section 5.4. This academic system has been constructed to act as a model problem which we use to showcase the differences between the Leapfrog-Chebyshev and Gautschi-type integrator. The application to a graphene sheet has already been discussed in section 6.3.2. We extend this discussion to longer time intervals and consider it in the context of a molecular system studied by computational physicists. The final results presented in this chapter are the fruits of a long-lasting collaboration with applied scientists inside the RTG 2450. The results of the simulations of pentane under different temperatures and pressures show the robustness of the Gautschi-type integrator over multiple millions steps on a high performance cluster. With these three problems we cover a large field of different problems that might be considered with molecular dynamics simulations.

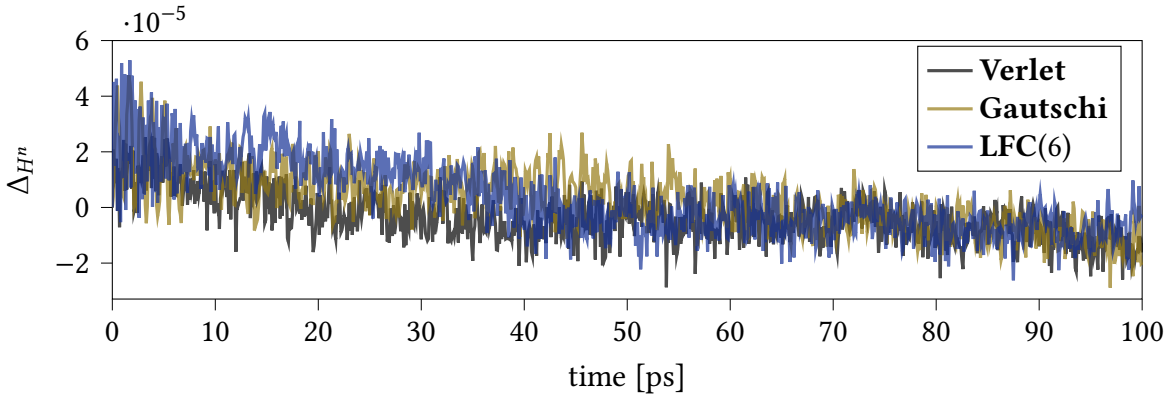


Figure 7.1. Relative deviation Δ_{H^n} from the initial energy for the Verlet, unfiltered Gautschi-type and stabilized Leapfrog-Chebyshev integrators when applied to the model system defined in section 6.3.1 with stepsize 1 fs as a function of the elapsed time [ps].

7.1 Application to the Large Scale Model System

We have already seen the computational advantages of the semi-analytical approach when applied to the large scale system introduced in section 5.4 for simulations over short time periods in section 6.3.1. Motivated by these results we now investigate and compare the long-time behavior of the unfiltered Gautschi-type integrator and the stabilized Leapfrog-Chebyshev method. We use the default parameters for the Gautschi-type integrator as motivated in section 6.3.1. This means that the matrix function vector product of the Gautschi-type integrator is approximated in a Krylov subspace of dimension $m_{\max} = 5$. We further apply the stabilized Leapfrog-Chebyshev method with different polynomial degrees p. We denote by *LFC(p)* the stabilized Leapfrog-Chebyshev integrator with Chebyshev polynomials of degree p. An important measure for the quality of the numerical time-integration is the energy conservation. We have shown in section 2.1 that the energy, i.e., the Hamiltonian, is preserved along the exact solution of the Hamiltonian system.

To see how well our implementation of the semi-analytical approach conserve the energy over long time periods we first equilibrate the system with 12 800 particles shown in figure 5.7 over 25 ps with the Verlet method and stepsize $\tau = 0.5$ fs. We then apply the Verlet, unfiltered Gautschi-type and the stabilized Leapfrog-Chebyshev integrator with a stepsize of 1 fs to the equilibrated system. We compute the Hamiltonian $H^n = H(q^n, p^n)$ for every 100 steps over a total of 100 000 fs = 100 ps and consider the relative deviation from the starting value $H^0 = H(q^0, p^0)$ at timestep t_n given by

$$\Delta_{H^n} = \frac{H^n - H^0}{|H^0|}. \tag{7.1}$$

In figure 7.1 we plot the relative deviation of the three integrators with stepsize $\tau = 1$ fs as a function of the elapsed time. For this stepsize, all three integrators exhibit similar energy conservation with a small relative error $|\Delta_{H^n}| < 6 \cdot 10^{-5}$ for all timesteps t_n. Even though

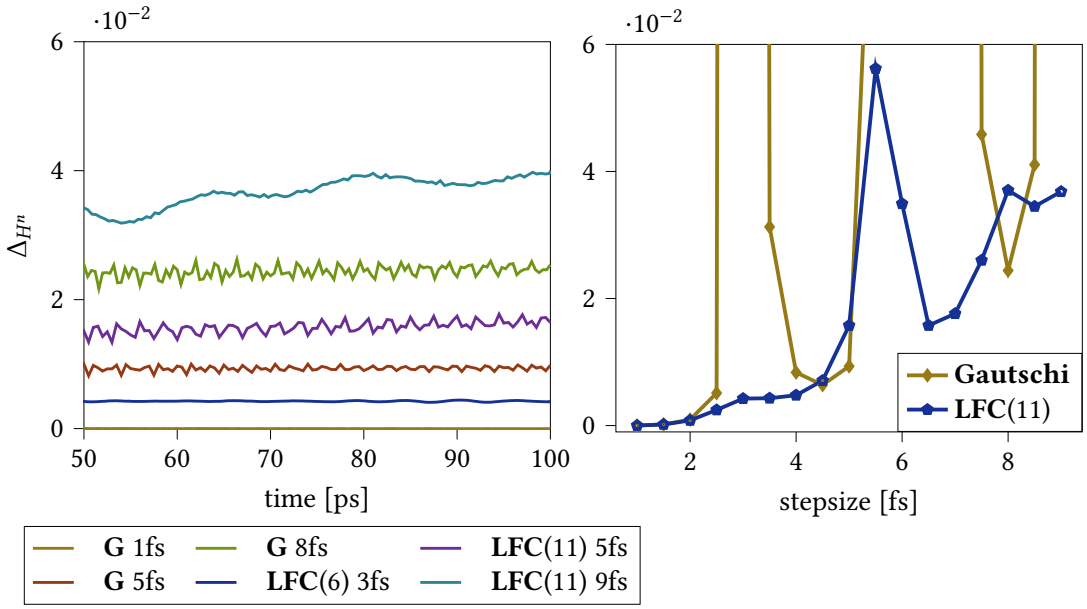


Figure 7.2. Relative deviation from the initial Hamiltonian for the unfiltered Gautschi-type (G) and stabilized Leapfrog-Chebyshev (LFC) integrators with different stepsizes when applied to the model system defined in section 6.3.1 with different stepsizes as a function of the elapsed time [ps] (left) and absolute, average deviation over the last 50 ps as a function of the stepsize τ[fs] (right).

the system has been equilibrated prior to the integration, all integrators show a peak in energy at the first integration step. This results in strong oscillations of the energy. These oscillations smoothen out over time and are, for all three integrators with $\tau = 1$ fs, very close to the original energy H^0. For that reason, we view this as a second equilibration phase and consider the relative energy deviation for larger stepsizes only after 50 ps, as our interest lies in the behavior over long time periods.

The energy conservation is deemed to be sufficiently good, when there are no drifts after this second equilibration phase. In section 6.3.1 we have already seen that the Verlet scheme is only stable for stepsizes up to 1 fs, whereas the Gautschi-type and the Leapfrog-Chebyshev integrators are stable for larger stepsizes (see figure 6.4). To observe the long-time energy conservation for these stepsizes, we integrate the system with larger stepsizes using the semi-analytical approaches and plot the results for selected stepsizes on the left in figure 7.2. For additional context, we calculate the absolute value of the average deviation over the last 50 ps on the right of figure 7.2.

On the left of figure 7.2, we see that after the second equilibration phase both integrators are able to produce solutions with bounded, albeit slightly larger energies with larger stepsizes. The unfiltered Gautschi-type integrator fails, in contrast to the stabilized Leapfrog-Chebyshev integrator, to produce solutions with stepsize 3 fs. However, the energy conservation with stepsize 5 fs is better for the Gautschi-type integrator than the Leapfrog-Chebyshev integrator. One can even observe a slight energy drift for the Leapfrog-Chebyshev integrator. As depicted in the legend, we manually adapted the degree of the Chebyshev polynomials to allow for stepsize up to 9 fs. While the scheme with

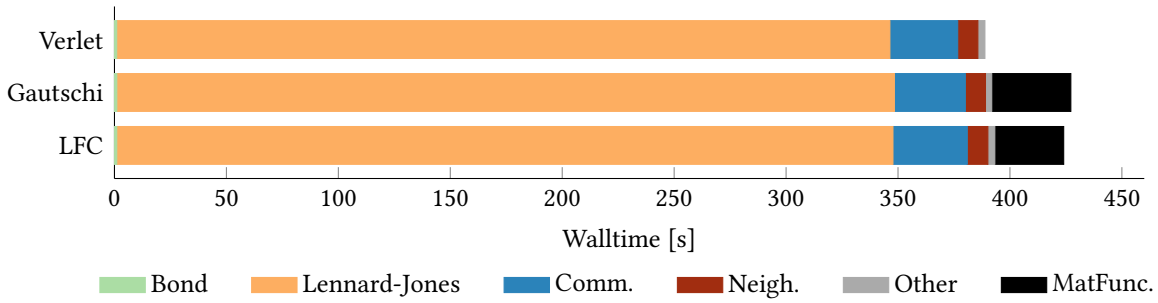


Figure 7.3. Total walltime for the Verlet, unfiltered Gautschi-type and stabilized Leapfrog-Chebyshev integrators when applied to the model problem from section 5.4 with $\tau = 1\,\text{fs}$ over 100 ps, separated into its different components.

smaller polynomial degrees was not stable for this stepsize, larger polynomial degrees did not improve the energy conservation. In comparison, the Gautschi-type integrator showed similar performance for different Krylov space dimensions with $m_{\max} \geq 5$. The differences between the two integration schemes for larger stepsizes can be seen on the right of figure 7.2 in more detail. Here, the resonance effects of the unfiltered Gautschi-type integrator for stepsizes between 2.5 fs and 4 fs coincide with those seen in figure 6.2. Note that due to the lack of an equilibrium position q^e, we can not use our averaging approach to overcome these resonances (compare section 6.3.1).

We take a closer look at the time the computations need to complete. In addition to the computations, there are also overhead costs such as initialization, data transfer and data output. We therefore consider the walltime. The walltime refers to the actual time that passes between the start and end of a program's execution. The computations have been performed in parallel with 10 threads on an *Intel® Xeon® Silver Processor 4210R*. To get an understanding what factors play which role in the total walltime, we separate the walltime into its individual components. We consider the time needed for inter- and intramolecular force computations, the additional effort from the semi-analytical integrators and the parallelization cost. The cost of parallelization is in itself split into two parts. LAMMPS has to keep track of the neighbor list in the ghost region for each processor (`Neigh.`), as described in section 5.2.1, before the actual communication (`Comm.`) can be performed in each step. We collect all remaining time-consuming actions, such as output generation in `Other`.

Figure 7.3 shows, that the cost for computing the bond potential is negligible compared to the long-range Lennard-Jones calculations. Despite the fact that we compute multiple matrix-vector products with a matrix of dimension $38\,400 \times 38\,400$ in each step, the extra effort needed for the semi-analytical approaches is comparatively small. This is due to the implicit matrix-vector product introduced in section 6.2 (see algorithm 7) as well as our efficient implementation in LAMMPS. A significant amount of the walltime is spent on communication between the threads. The additional communication, that is described in section 6.5, is already included in the '`MatFunc.`' section. Thanks to our efficient parallel implementation, this extra effort is small compared to the communications '`Comm.`' needed for the force computations. A significant portion of this work has been committed to create

	$100\,000 \times 1$ fs	$33\,334 \times 3$ fs	$20\,000 \times 5$ fs
Walltime [s]			
Verlet	465.4	/	/
Gautschi	494.84	/	113.4
Leapfrog-Chebyshev	507.8	168.82	113.13
Timesteps/seconds $[\frac{1}{s}]$			
Verlet	214.87	/	/
Gautschi	202.1	/	176.3
Leapfrog-Chebyshev	196.93	197.45	176.8

Table 7.1.: Comparison of the total walltime of the Verlet, unfiltered Gautschi-type and stabilized Leapfrog-Chebyshev integrators and number of timesteps per second when applied to the model system with different stepsizes. The polynomial degree for the Leapfrog-Chebyshev integrator is 6 for $\tau \leq 3$ fs and 11 else.

this efficient and parallel implementation of the semi-analytical approach. Without this effort, the integrators would not be able to compete against the highly optimized Verlet scheme.

We collect the walltimes for the simulations with 1 fs, 3 fs and 5 fs in table 7.1 to get a comparison of the walltimes with different stepsizes. From table 7.1 we see that we do not get a one to one relation between the number of steps and the total walltime. The reason for that is not an increased computational effort for the matrix function approximation but rather the fact that LAMMPS needs more neighborhood updates and a larger neighborhood domain for larger stepsizes. This is due to the fact that particles are moving larger distances during one timestep. We can in fact see, that the additional five matrix-vector multiplications needed for the Leapfrog-Chebyshev scheme with $p = 11$ have no significant influence on the computational cost. The main effort comes from updating the neighbor list and the Hessian of the bond potential.

7.2 Application to a Graphene Sheet

To analyze the long-time behavior of the filtered Gautschi-type integrator with the full Hessian matrix approximating bond, angle, and torsional potential we take a second look at the graphene sheet introduced in section 6.3.2. We consider a graphene sheet in vacuum and set the initial velocities of the particles such that the system has a temperature of 300 Kelvin. The simulation is then carried out over 50 picoseconds. As before, we consider the energy conservation of the system as a first measurement for the quality of the simulation. We follow the same steps as for the system in the previous section and plot on the left in figure 7.4 the relative energy deviation Δ_{H^n} (see (7.1)) as a function of the elapsed time. On the right-hand side of figure 7.4 we further plot the absolute value of the average over the last 25 ps.

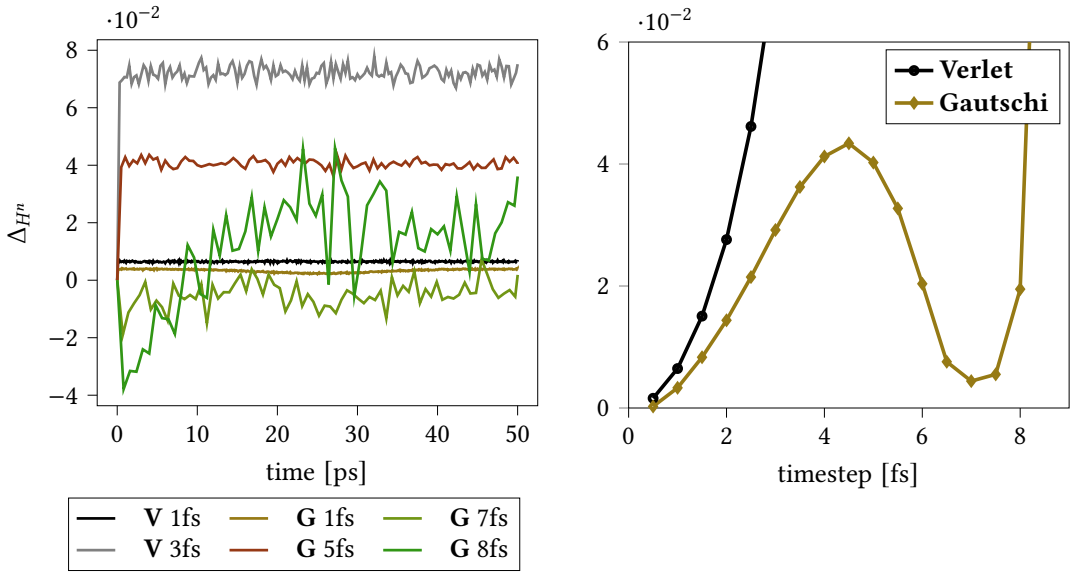


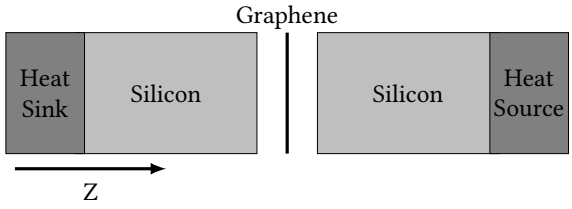
Figure 7.4. Relative deviation from the initial Hamiltonian for the Verlet (V) and filtered Gautschi-type (G) integrators with different stepsizes when applied to a graphene sheet with 680 carbon particles as a function of elapsed time (left) and absolute, average deviation over the last 25 ps as a function of the stepsize τ[fs] (right).

From figure 7.4 we deduct that the inner filter is able to filter out resonances not only over short time periods, as seen in figure 6.6, but also over longer time periods. Due to the larger mass of the carbon atoms, the Verlet scheme is stable for up to 3 fs. However, for the same stepsize the Gautschi-type integrator always performs with a better accuracy. Up to a stepsize of $\tau = 8$ fs, the energy produced by the Gautschi-type integrator is in a regime of 10^{-2} around the starting Hamiltonian H^0. While the amplitude of the oscillations around the energy increases for larger stepsizes, as seen on the left of figure 7.4, the absolute value of the averaged deviation stays bounded.

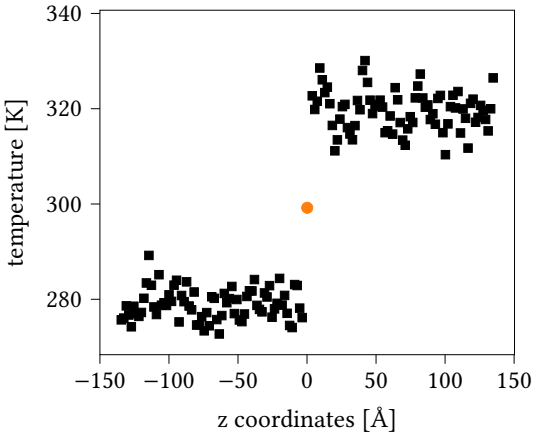
In contrast to the system studied in section 7.1, there are only intramolecular potentials in this system. This means the split of the total potential $V(q)$ in fast, cheap potential $W(q)$ and slow, expensive $U(q)$ is not as distinct, and we can therefore not expect the Gautschi-type integrator to give a large computational advantage. In fact, the Verlet integrator is roughly three times faster for the same stepsize. With 95%, the majority of the computation for the Gautschi-type integrator is used to approximate the matrix function. In order to make use of the higher accuracy and larger possible stepsize we extend this system to a more complex example.

7.3 Heat Transport Through a Silicon-Graphene Medium

Graphene is not only a good example to showcase the full functionality of the Gautschi-type integrator, but also a material of high interest in material sciences. One specific



(a) Structure of the system, simulating a heat transport through two silicon blocks divided by a graphene sheet.



(b) Heat distribution in the system as simulated with the Verlet integrator with $\tau = 1\,\text{fs}$ after 50 ps. The simulation domain is decomposed in 150 slices in z direction and the temperature of each slice is plotted against its center coordinate. The Graphene sheet (orange circle) acts as an insulator between the silicon blocks (black boxes).

Figure 7.5. Structure of the system (left) and simulation result (right) of the Silicon Graphene heat transport.

application is the use in solar cells in combination with crystalline silicon [56]. We here consider a system loosely based on a paper by M. Shen, P. K. Schelling, and P. Keblinski [73] that focuses on the heat transfer mechanism across few-layer graphene. The authors consider graphene between two crystalline, face center cubic, silicon blocks with (111) surfaces (see appendix B.2 for more information on crystalline structures) and a length of 11.2 nanometers in $z-$direction, see figure 7.5a.

7.3.1 Initialization of the System

The system for this section is based on the one in the paper, with one main difference. The authors use a *Tersoff* potential [79, 80] to describe the intramolecular interactions of the graphene sheet. For more details on the Tersoff potentials we refer to appendix B.1.2. We want to make use of the intramolecular potentials with the Hessian described in chapter 4. To still simulate a continuous graphene sheet in x and y direction, we overlap the graphene sheet at the periodic boundary and fix the overlapping particles in place. The system is composed of two silicon blocks with each $7 \times 17 \times 24$ unit cells and a lattice constant of $a = 5.431$ for a total of 46 648 Si atoms. The graphene sheet consisting of 1260 carbon atoms lies in between the two silicon blocks. The intermolecular interaction between the graphene sheet and the silicon blocks are governed by the Lennard-Jones potential with the parameters

$$\epsilon_{\text{C-Si}} = 0.005\,955\,\text{eV}, \qquad \sigma_{\text{C-Si}} = 3.555\,\text{Å}.$$

The interactions between Si atoms are described by the Stillinger-Weber potentials [75] (cf. appendix B.1.1) combined with the Lennard-Jones potential with

$$\epsilon_{\text{Si-Si}} = 0.011\,82\,\text{eV}, \qquad \sigma_{Si-Si} = 3.7\,\text{Å}$$

	Verlet 1 fs	Gautschi 7 fs	Gautschi 8 fs
Thermal conductivity	−39.9	−39	−40.6
Walltime	2h24min	50min	46min

Table 7.2.: Comparison of the total computational time and thermal conductivity of Verlet and filtered Gautschi-type applied to the silicon-graphene material with different stepsizes.

for long range interactions.

7.3.2 Simulation Run

The simulation is carried out in three stages. First, we set the temperature of the system to 300 Kelvin by changing the velocities of the system. Under the NVT ensemble, i.e., with a Norse-Hoover thermostat, the simulation is equilibrated over 10 picoseconds. We perform this first stage with the Verlet integrator using a 1 fs stepsize. The second stage now introduces the heat source and heat sink, as shown in figure 7.5a. The temperature in the heat source and heat sink is controlled with a *Langevin* thermostat. The cold region gets cooled down to 280 Kelvin and the hot region heated up to 320 Kelvin. We simulate this setup over 50 ps so that the temperature flow in the system has reached a steady state. After this second equilibration step, we compute the thermal conductivity over 100 ps. The final state for the Verlet integration with $\tau = 1$ fs is shown in figure 7.5b. After decomposing the simulation domain in 150 slices in z direction we measure the temperature of each slice and plot it as a function of the center coordinate of the slice.

We simulate the system with the Verlet integrator and a stepsize of $\tau = 1$ fs to get a reference value for the thermal conductivity and compare it with the value obtained by the filtered Gautschi-type integrator with stepsizes 7 fs and 8 fs. Table 7.2 shows the three values as well as the walltimes after the system has been equilibrated.

The data in table 7.2 shows that we can use significantly larger stepsizes with the filtered Gautschi-type integrator. The result of the simulation is in close accordance with the result of the Verlet integrator. Even though the necessary computations for the semi-linear approach are large for the isolated graphene sheet, as we have seen in section 7.2, we get a significant speedup compared to the Verlet algorithm. This is due to the fact that the majority of computation time is needed for the silicon-silicon interactions.

7.4 Application to Pentane

In order to further validate the quality of the semi-analytical approach for long-time simulations there is a close collaboration with researchers using LAMMPS for their studies inside the RTG 2450 'Tailored Scale-Bridging Approaches to Computational Nanoscience'. The work presented in this section is the result of a particularly close collaboration with

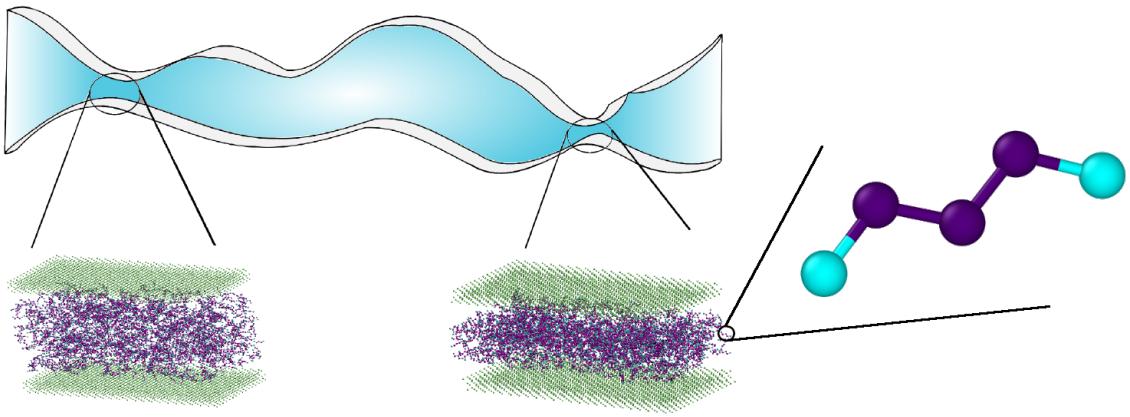


Figure 7.6. The macroscopic problem on top is divided into many microscopic problems. Properties of the macroscopic problem influence the microscopic ensemble and vice versa.

M. T. Elewa. In order to describe the adhesion and friction between solid and soft matter the behavior of liquid pentane between two gold layers is observed. Varying thickness of the spacing between the gold layers under different temperature and pressure conditions give an insight to the flow on a molecular scale. This information can then be used to simulate the behavior on a larger scale with fluid dynamics simulation [47]. Figure 7.6 gives a schematic representation of the system. The main goal of this section is to validate the results generated by the semi-analytical approach as well as showcasing the functionality of the parallelization on a high performance computer. For this reason, we limit the study to the Gautschi-type integrator.

The system is constrained in $z-$direction by two face-centered cubic (FCC) gold (111) surfaces (see appendix B.2). The pentane molecule is described by a united atom model, meaning that the carbon hydrogen groups are united into a new pseudo-atom. As a result the fastest harmonic oscillations, stemming from the hydrogen carbon bond, is already eliminated from the system. The slower, harmonic oscillations of the bonds between the pseudo-atom are however still present. This is a common approach in molecular dynamics simulation to allow for faster computations. The fluid is controlled by the *Transferable Potential for Phase Equilibria* (TraPPE) force field [58] with the parameters

$$\begin{aligned} \bar{\rho}_B &= 1.54\,\text{Å}, & k_B &= 1000\frac{\text{kcal}}{\text{mol}}, \\ \bar{\theta} &= 114^\circ, & k_A &= 124.2\frac{\text{kcal}}{\text{mol}}, \end{aligned}$$

for the harmonic bond and angle potential (cf. (2.14) and (2.15)). The torsional potential is given by

$$\begin{aligned} V_T(\phi_{ijkl}) = \quad & \frac{1}{2}K_1[1+\cos(\,\phi_{ijkl}\,)] + \frac{1}{2}K_2[1-\cos(2\phi_{ijkl})] \\ & + \frac{1}{2}K_3[1+\cos(3\phi_{ijkl})] + \frac{1}{2}K_4[1-\cos(4\phi_{ijkl})] \end{aligned}$$

with parameters

$$K_1 = 1.411, \quad K_2 = -0.271, \quad K_3 = 3.145, \quad K_4 = 0.$$

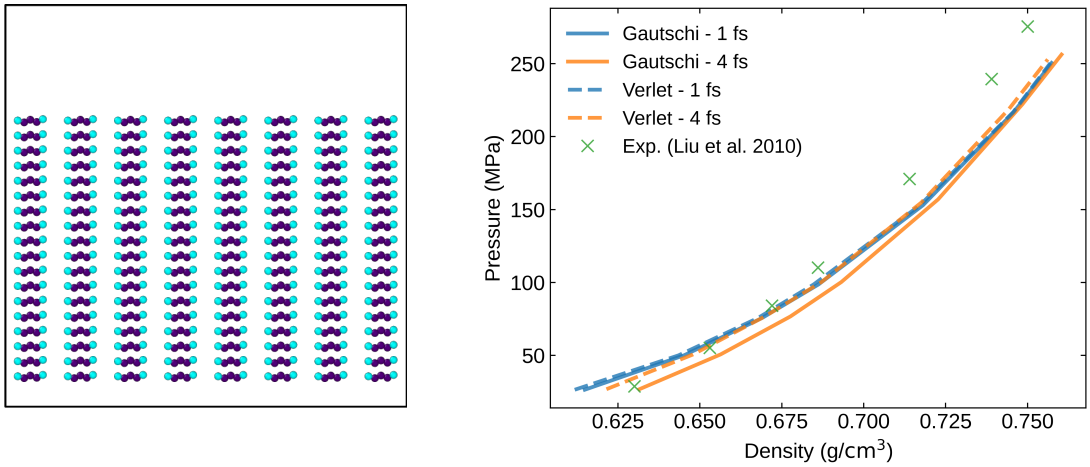


Figure 7.7. Comparison of the pressure-density relation at 326 K, computed with the Verlet and unfiltered Gautschi-type integrator on a 14 400 n-pentane system with an initial $8 \times 20 \times 18$ grid (left with front view), with the experimental data by [57] (right).

The intermolecular interactions are described by a Lennard-Jones potential with parameters

$$\sigma_{CH_3} = 0.1950 \qquad \epsilon_{CH_3} = 3.75$$
$$\sigma_{CH_2} = 0.0914 \qquad \epsilon_{CH_2} = 3.95,$$

and a cutoff radius of $\rho_{\text{cut}} = 10$ Å.

7.4.1 Equation of State

In order to validate the accuracy of the parametrization of a liquid system it is common to compute an equation of state and compare it with experimental data. There are different types of equations of state. In general, they give a relation of thermodynamic properties, such as the pressure, volume, density, and temperature of a gas, liquid, or solid. We consider the relation of pressure and density of the liquid system at 326 K. For a comparison with experimental data, we use the findings of K. Liu et al. [57].

The simulation is performed on a system with 14 400 united atom n-pentane molecules that are initially placed in a $8 \times 20 \times 18$ grid inside a simulation domain with periodic boundaries. The initial configuration and can be seen on the left in figure 7.7. We consider the system at 285, 546, 830, 1088, 1688, 2364 and 2719 atmospheric pressure [atm] (1 MPa $\approx$ 9.869 atm). For each pressure setting we follow the same workflow. Under an isothermal-isobaric (NPT) ensemble imposed by a Nose-Hoover style non-Hamiltonian equation of motion (cf. section 5.3) the system is equilibrated for 10 picoseconds with the Verlet integrator and the stepsize $\tau = 1$ fs. Both the Verlet and the unfiltered Gautschi-type integrator with the Hessian of the bond potential are then applied with $\tau = 1$ fs and $\tau = 4$ fs to the equilibrated

	Verlet	Gautschi
$\tau = 1\,\text{fs}$	174.02	102.85
$\tau = 4\,\text{fs}$	167.64	99.63

Table 7.3.: Average number of timesteps per second of the Verlet and unfiltered Gautschi-type integrator when applied to 14 400 particles on 80 threads.

system for 1 000 000 steps. Note that the pentane molecules are translating and rotating freely in the domain. We thus update the Hessian in every step and cannot use an inner filter function. After these simulations, the pressure in the system has converged, and we get one data point of the relation shown on the right in figure 7.7.

The Verlet and unfiltered Gautschi-type integrators give for both stepsizes, $\tau = 1\,\text{fs}$ and $\tau = 4\,\text{fs}$, similar results. The deviation from the experimental data is explained with the parametrization of the force fields and not the numerical integration. Due to the united atom model the fastest oscillations limit the stepsize of the Verlet integrator to four instead of the one femtosecond that is usually imposed by the covalent hydrogen bonds in biomechanical systems. For one femtosecond, the results of two integrators almost align. Only for the larger 4 fs one can see a separation, especially for lower pressures. However, the qualitative behavior is correct for both methods at this timestep. The computations where performed on the *bwUniCluster* with 80 threads. We collect the average computation times for the Verlet and Gautschi-type integrator over all simulations in table 7.3.

As in the model problem, the simulation become more expensive for larger stepsizes (compare table 7.1). Larger stepsizes result in larger skin distances and thus in larger neighbor lists. For both stepsizes, the Gautschi-type integrator is almost 70% slower. A lack of inner filter functions and the resulting stepsize restriction means that we are not able to outperform the Verlet integrator for this system.

7.4.2 Liquid-Gas Membrane

An important property of the pentane system, and interfacial systems in general, that can be studied with molecular dynamics simulation is the surface tension. An excellent introduction to the computational techniques used in the calculation of the surface tension is given in [28]. A. Mulero, I. Cachadiña, and M. I. Parra [62] provide experimental data which we use to validate our simulation results. To compute the surface tension of the vapour-liquid interface of pentane, we construct a system consisting of a liquid phase and a gas phase that are in contact with each other. The system is contained within a periodic box. We simulate a total of 3108 united atom pentane molecules. For this simulation the long range interactions between the liquid and gas phase are of special interest. For that reason we increase the cutoff radius of the Lennard-Jones interactions to $\rho_{\text{cut}} = 14\,\text{Å}$. A snapshot of the simulation at 300 Kelvin can be seen in figure 7.8. The droplet in the middle of the simulation domain disappears for higher temperatures. This is one of the reasons,

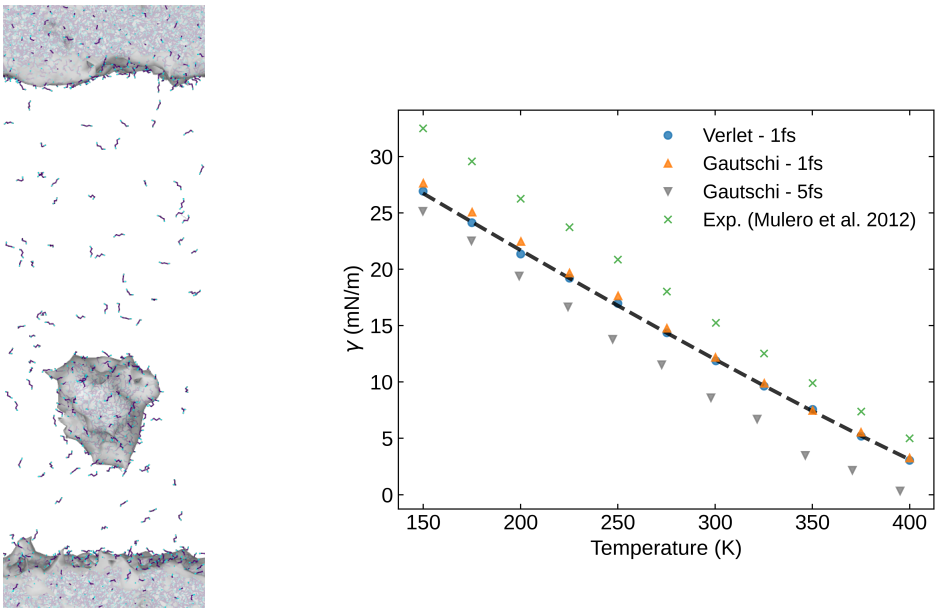


Figure 7.8. Comparison of the approximate surface tension of n-pentane with the experimental data by [62].

why M. T. Elewa computed the surface tension on the planar interface at the top and bottom of the simulation cell.

As for the previous computations, we follow the same workflow for multiple data points. We consider the system from 150 K to 400 K with 25 K increments for a total of 11 data points. Under the canonical (NVT) ensemble imposed by a Nose-Hoover style non-Hamiltonian equation of motion (cf. section 5.3) the system is equilibrated for 10 picoseconds with the Verlet integrator and the stepsize $\tau = 1$ fs. The Verlet integrator with 1 fs or the Gautschi-type integrator with $\tau = 1$ fs and $\tau = 5$ fs is then applied to the equilibrated system for 3 000 000 steps. In a post-processing step, of which we omit the details, the surface tension is calculated. We plot the resulting relation in figure 7.8. It is important to note that the Verlet integrator is not stable for the stepsize $\tau = 5$ fs.

The results of the two integrators coincide again for $\tau = 1$ fs. The plot generated with the Gautschi-type integrator with $\tau = 5$ fs shows a deviation from those simulated with the smaller stepsize. However, the qualitative behavior is still correct compared with the experimental data. The simulations where again performed on the *bwUniCluster* with 80 threads. We collect the average number of timesteps per second for the Verlet and Gautschi-type integrator over all simulation steps in table 7.4.

Significantly more time is spent on the Lennard-Jones interactions, due to the larger cutoff radius. As a result, the Verlet integration takes almost 65% longer than in the previous simulation (compare table 7.3). This is despite the fact that we consider almost 80% fewer particles. Therefore, the extra costs needed for the Gautschi-type integrator is small in relation to the cost of the intermolecular interactions. For the same stepsize, the Gautschi-type integrator is only 24% slower.

	Verlet	Gautschi
$\tau = 1\,\mathrm{fs}$	62.22	47.26
$\tau = 5\,\mathrm{fs}$		48.52

Table 7.4.: Average number of timesteps per second of the Verlet and unfiltered Gautschi-type integrator when applied to 3108 particles on 80 threads.

To conclude this section, we note that this system is not ideal for the semi-analytical approach. On one hand the fastest harmonic oscillations were already taken out of the system by using the united atom approach. On the other hand we can not employ an inner filter function and are forced to continually reevaluate the Hessian matrix. Nevertheless, we were able to validate the implementation of the unfiltered Gautschi-type integrator on a large system over multiple millions of timesteps. We have shown that the qualitative result are correct and that the computational effort is competitive with that of the Verlet integrator.

8 Conclusion

In the first half of this thesis we present the theoretical and technical details of semi-analytical integration. It becomes clear that the use of integrators of this type requires a deeper understanding of the physical background compared to the standard Verlet integration. We discuss the different scales in molecular dynamics and introduce a harmonic approximation to the potentials that are responsible for the fastest motions in the system.

In the work we present Hessian matrices for the three most commonly used intramolecular potentials. We did not find the expressions nor an implementation of their evaluation in the literature. We were able to show that the approximation of the nonlinear intramolecular potentials with these Hessians allows for accurate semi-analytical integration with large stepsize. There are more potentials for which the accuracy of the linear approximation is not so clear. Computing, implementing and analyzing the effect of the Hessian of further potentials in the context of semi-analytical integration is a field that might be explored further.

The efficient implementation of the semi-analytical integrators revolves around a suitable approximation of matrix-valued functions multiplied with certain vectors. We present and follow two different approaches, the Gautschi-type integrators with Krylov subspace approximations and the Leapfrog-Chebyshev scheme. Their efficient implementation into molecular dynamics software such as LAMMPS is not trivial. A significant effort has been put into the development of these schemes in LAMMPS which allows the integrators to be competitive with the well established Verlet scheme.

Throughout this work we repeatedly compare the damping approach of the Leapfrog-Chebyshev integrator with the inner filter function of the Gautschi-type integrator. Both ideas are able to overcome, in specific settings, numeric resonances in the system. With the stabilization parameter of the Leapfrog-Chebyshev integrator, no additional information of the particle structure is needed. As a result, the stabilization approach can easily be applied to many problems. However, for problems with fixed particle structure we have shown that the inner filter function used in the Gautschi-type integrator is able to filter out the resonances more reliably.

As a result of this work, we have a running implementation of two different semi-analytical integration schemes that can, in certain circumstances, beat the standard Verlet integrator in both accuracy and computation time. The implementation of these integrators consists

of six different fixes and one additional atom style and is applicable to a wide range of problems inside LAMMPS.

We apply the integration schemes to three different molecular dynamics problems. With these problems, we show that the efficiency of the semi-analytical approach strongly depends on the system that it is applied to. Through our collaboration with scientists that use molecular dynamics simulation for their research we can mutually improve the understanding for these computations. As a result, the integrator was thoroughly tested and optimized on realistic examples that cover a wide range of problems. We also sparked the interest of applied scientists in faster computations, not through reduced models, but improved numerical integration. This interdisciplinary effort is needed if one wants to improve the quality of computational scientific research.

While this work alone can not bridge the scales of molecular dynamics simulations from nano- to microsecond resolution, it is a further step in closing the gap between theoretical knowledge and practical application.

Bibliography

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. doi:10.1137/1.9780898719604.

[2] P. Bauer, B. Hess, and E. Lindahl. Gromacs 2022.4 manual, 2022. doi:10.5281/zenodo.7323409.

[3] H. Bekker, H. J. C. Berendsen, E. J. Dijkstra, S. Achterop, R. Vondrumen, D. Vanderspoel, A. Sijbers, H. Keegstra, and M. K. R. Renardus. Gromacs: A parallel computer for molecular-dynamics simulations. In *4th International Conference on Computational Physics (Physics Computing '92)*, pages 252–256. World Scientific Publishing, 1993. URL https://hdl.handle.net/11370/84ef8765-003b-421d-b570-b9c5a3f78a9b.

[4] H. Bekker, H. J. C. Berendsen, and W. F. Van Gunsteren. Force and virial of torsional-angle-dependent potentials. *J. Comput. Chem.*, 16(5):527–533, 1995. doi:10.1002/jcc.540160502.

[5] G. Benettin and A. Giorgilli. On the Hamiltonian interpolation of near-to-the-identity symplectic mappings with application to symplectic integration algorithms. *J. Statist. Phys.*, 74(5-6):1117–1143, 1994. doi:10.1007/BF02188219.

[6] H. J. C. Berendsen. *Simulating the Physical World: Hierarchical Modeling from Quantum Mechanics to Fluid Dynamics*. Cambridge University Press, Cambridge, 2007. doi:10.1017/CBO9780511815348.

[7] H. J. C. Berendsen, J. P. M. Postma, W. F. Van Gunsteren, A. Dinola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.*, 81(8):3684–3690, 1984. doi:10.1063/1.448118.

[8] M. A. Botchev, L. Knizhnerman, and M. Schweitzer. Krylov subspace residual and restarting for certain second order differential equations. 2022. URL https://doi.org/10.1137/22M1503300.

[9] H. Bowdler, R. S. Martin, C. Reinsch, and J. H. Wilkinson. *The QR and QL Algorithms for Symmetric Matrices*, pages 227–240. Springer Berlin Heidelberg, Berlin, Heidelberg, 1971. doi:10.1007/978-3-662-39778-7_14.

[10] D. W. Brenner. Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films. *Phys. Rev. B*, 42:9458–9471, 1990. doi:10.1103/PhysRevB.42.9458.

[11] B. R. Brooks, C. L. Brooks III, A. D. M. Jr., L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, and Others. Charmm: the biomolecular simulation program. *J. Comput. Chem.*, 30(10):1545–1614, 2009. doi:10.1002/jcc.21287.

[12] S. Buchholz, L. Gauckler, V. Grimm, M. Hochbruck, and T. Jahnke. Closing the gap between trigonometric integrators and splitting methods for highly oscillatory differential equations. *IMA J. Numer. Anal.*, 38(1):57–74, 2018. doi:10.1093/imanum/drx007.

[13] C. Carle. *On leapfrog-Chebyshev schemes for second-order differential equations*. PhD thesis, Karlsruher Institut für Technologie (KIT), 2022. doi:10.5445/ir/1000147725.

[14] C. Carle, M. Hochbruck, and A. Sturm. On leapfrog-Chebyshev schemes. *SIAM J. Numer. Anal.*, 58(4):2404–2433, 2020. doi:10.1137/18M1209453.

[15] D. A. Case, H. M. Aktulga, K. Belfon, I. Y. Ben-Shalom, J. T. Berryman, S. R. Brozell, D. S. Cerutti, T. E. Cheatham III, G. A. Cisneros, V. W. D. Cruzeiro, T. A. Darden, R. E. Duke, G. Giambasu, M. K. Gilson, H. Gohlke, A. W. Goetz, R. Harris, S. Izadi, S. A. Izmailov, K. Kasavajhala, M. C. Kaymak, E. King, A. Kovalenko, T. Kurtzman, T. S. Lee, S. Legrand, P. Li, C. Lin, J. Liu, T. Luchko, R. Luo, M. Machado, V. Man, M. Manathunga, K. M. Merz, Y. Miao, O. Mikhailovskii, G. Monard, H. Nguyen, K. A. O'Hearn, A. Onufriev, F. Pan, S. Pantano, R. Qi, A. Rahnamoun, D. R. Roe, A. Roitberg, C. Sagui, S. Schott-verdugo, A. Shajan, J. Shen, C. L. Simmerling, N. R. Skrynnikov, J. Smith, J. Swails, R. C. Walker, J. Wang, J. Wang, H. Wei, R. M. Wolf, X. Wu, Y. Xiong, Y. Xue, D. M. York, S. Zhao, and P. A. Kollman. Amber 22 reference manual. 2022. URL https://www.ambermd.org.

[16] C. A. De Coulomb. Premier mémoire sur l'electricité et le magnétisme. *Mem.Acad.r.Sci.Paris*, 569:54, 1785.

[17] P. Deuflhard. A study of extrapolation methods based on multistep schemes without parasitic solutions. *Z. Angew. Math. Phys.*, 30(2):177–189, 1979. doi:10.1007/BF01601932.

[18] R. O. Dror, R. M. Dirks, J. P. Grossman, H. Xu, and D. E. Shaw. Biomolecular simulation: A computational microscope for molecular biology. *Annu. Rev. Biophys.*, 41:429–452, 2012. doi:10.1146/annurev-biophys-042910-155245.

[19] A. Dubrulle. A short note on the implicit QL algorithm for symmetric tridiagonal matrices. *Numer. Math.*, 15(5):450–450, 1970. doi:https://doi.org/10.1007/BF02165514.

[20] P. P. Ewald. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann. Phys.*, 369(3): 253–287, 1921. doi:10.1002/andp.19213690304.

[21] L. Fath, M. Hochbruck, and C. V. Singh. A fast mollified impulse method for biomolecular atomistic simulations. *J. Comput. Phys.*, 333:180–198, 2017. doi:10.1016/j.jcp.2016.12.024.

[22] E. Fermi, P. Pasta, S. Ulam, and M. Tsingou. Studies of the nonlinear problems. Technical report, Los Alamos National Lab., 1955. doi:10.2172/4376203.

[23] M. P. I. Forum. *MPI: A Message-Passing Interface Standard Version 4.0*, 2021. URL https://www.mpi-forum.org/docs/mpi-4.0/mpi40-report.pdf.

[24] L. Galgani, A. Giorgilli, A. Martinoli, and S. Vanzini. On the problem of energy equipartition for large systems of the Fermi-Pasta-Ulam type: analytical and numerical estimates. *Phys. D*, 59(4):334–348, 1992. doi:10.1016/0167-2789(92)90074-W.

[25] E. Gallopoulos and Y. Saad. Efficient solution of parabolic equations by krylov approximation methods. *SIAM J. Sci. Comput.*, 13:1236–1264, 1992. doi:10.1137/0913071.

[26] B. García-Archilla, J. M. Sanz-Serna, and R. D. Skeel. Long-time-step methods for oscillatory differential equations. *SIAM J. Sci. Comput.*, 20(3):930–963, 1999. doi:10.1137/S1064827596313851.

[27] W. Gautschi. Numerical integration of ordinary differential equations based on trigonometric polynomials. *Numer. Math.*, 3:381–397, 1961. doi:10.1007/BF01386037.

[28] A. Ghoufi, P. Malfreyt, and D. J. Tildesley. Computer modelling of the surface tension of the gas–liquid and liquid–liquid interface. *Chem. Soc. Rev.*, 45(5):1387–1409, 2016. doi:10.1039/C5CS00736D.

[29] B. Giovanni, C. Giovanni, and G. Pietro. *Computer Meets Theoretical Physics: The New Frontier of Molecular Simulation.* Springer, Berlin, 2020. doi:10.1086/719725.

[30] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987. doi:10.1016/0021-9991(87)90140-9.

[31] M. Griebel, G. Zumbusch, and K. Stephan. *Numerical Simulation in Molecular Dynamics - Numerics, Algorithms, Parallelization, Applications.* Springer, Berlin Heidelberg, 1 edition, 2007. doi:10.1007/978-3-540-68095-6.

[32] V. Grimm and M. Hochbruck. Error analysis of exponential integrators for oscillatory second-order differential equations. *J. Phys. A*, 39(19):5495–5507, 2006. doi:10.1088/0305-4470/39/19/S10.

[33] H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten. Generalized Verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Mol. Simul.*, 6(1-3):121–142, 1991. doi:10.1080/08927029108022142.

[34] E. Hairer. *Long-time Energy Conservation*, pages 162–180. London Mathematical Society Lecture Note Series. Cambridge University Press, 2006. doi:10.1017/CBO9780511721571.005.

[35] E. Hairer and C. Lubich. Long-time energy conservation of numerical methods for oscillatory differential equations. *SIAM J. Numer. Anal.*, 38(2):414–441, 2000. doi:10.1137/S0036142999353594.

[36] E. Hairer, C. Lubich, and G. Wanner. Geometric numerical integration illustrated by the Störmer-Verlet method. *Acta Numer.*, 12:399–450, 2003. doi:10.1017/S0962492902000144.

[37] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I.* Springer, Heidelberg, 1993. doi:10.1007/978-3-540-78862-1.

[38] E. Hairer, G. Wanner, and C. Lubich. *Geometric Numerical Integration - Structure-Preserving Algorithms for Ordinary Differential Equations.* Springer, Berlin Heidelberg, 2 edition, 2006. doi:10.1007/3-540-30666-8.

[39] W. R. Hamilton. Second essay on a general method in dynamics. *Philos. Trans. R. Soc.*, 125:95–144, 1835. URL http://www.jstor.org/stable/108119.

[40] J. Hersch. Contribution à la méthode des équations aux différences. *Z. Angew. Math. Phys.*, 9a:129–180, 1958. doi:10.1007/BF01600630.

[41] N. J. Higham. *Functions of Matrices.* Society for Industrial and Applied Mathematics, 2008. doi:10.1137/1.9780898717778.

[42] M. Hochbruck and C. Lubich. On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 34(5):1911–1925, 1997. doi:10.1137/S0036142995280572.

[43] M. Hochbruck and C. Lubich. Exponential integrators for quantum-classical molecular dynamics. *BIT*, 39(4):620–645, 1999. doi:10.1023/A:1022335122807.

[44] M. Hochbruck and C. Lubich. A Gautschi-type method for oscillatory second-order differential equations. *Numer. Math.*, 83(3):403–426, 1999. doi:10.1007/s002110050456.

[45] M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numer.*, 19:209–286, 2010. doi:10.1017/S0962492910000048.

[46] O. Hoffman-Ostenhof, W. E. Cohn, A. E. Braunstein, P. Karlson, B. Keil, W. Klyne, C. Liebecq, E. Slater, E. C. Webb, and W. Whelan. IUPAC-IUB commission on biochemical nomenclature. abbreviations and symbols for the description of the conformation of polypeptide chains. *Biochem.*, 9(18):3471–3479, 1970. doi:10.1021/bi00820a001.

[47] H. Holey, A. Codrignani, P. Gumbsch, and L. Pastewka. Height-averaged Navier–Stokes solver for hydrodynamic lubrication. *Tribol. Lett.*, 70(2):1–15, 2022. doi:10.1007/s11249-022-01576-5.

[48] W. G. Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A*, 31:1695–1697, 1985. doi:10.1103/PhysRevA.31.1695.

[49] A. Hospital, J. R. Goñi, M. Orozco, and J. L. Gelpí. Molecular dynamics simulations: Advances and applications. *Adv. Appl. Bioinform. Chem*, pages 37–47, 2015. doi:10.2147/AABC.S70333.

[50] W. Humphrey, A. Dalke, and K. Schulten. VMD – Visual Molecular Dynamics. *J. Mol. Graph.*, 14:33–38, 1996. doi:10.1016/0263-7855(96)00018-5.

[51] A. I. Jewett, D. Stelter, J. Lambert, S. M. Saladi, O. M. Roscioni, M. Ricci, L. Autin, M. Maritan, S. M. Bashusqeh, T. Keyes, R. T. Dame, J. E. Shea, G. J. Jensen, and D. S. Goodsell. Moltemplate: A tool for coarse-grained modeling of complex biological matter and soft condensed matter physics. *J. Mol. Biol.*, 433(11):166841, 2021. doi:10.1016/j.jmb.2021.166841.

[52] J. E. Jones and S. Chapman. On the determination of molecular fields.—I. From the variation of the viscosity of a gas with temperature. *Proc. Math. Phys. Eng.*, 106(738):441–462, 1924. doi:10.1098/rspa.1924.0081.

[53] J. E. Jones and S. Chapman. On the determination of molecular fields. —II. From the equation of state of a gas. *Proc. Math. Phys. Eng.*, 106(738):463–477, 1924. doi:10.1098/rspa.1924.0082.

[54] B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2005. doi:10.1017/CBO9780511614118.

[55] B. J. Leimkuhler, S. Reich, and R. D. Skeel. *Integration methods for molecular dynamics*, volume 82 of *IMA Vol. Math. Appl.*, pages 161–185. Springer, New York, 1996. doi:10.1007/978-1-4612-4066-2_10.

[56] X. Li, H. Zhu, K. Wang, A. Cab, J. Wei, C. Li, Y. Jia, Z. Li, X. Li, and D. Wu. Graphene-on-silicon Schottky junction solar cells. *Adv Mater*, 22(25):2743–2748, 2010. doi:10.1002/adma.200904383.

[57] K. Liu, Y. Wu, M. A. Mchugh, H. Baled, R. M. Enick, and B. D. Morreale. Equation of state modeling of high-pressure, high-temperature hydrocarbon density data. *J. Supercrit. Fluids*, 55(2):701–711, 2010. doi:10.1016/j.supflu.2010.10.004.

[58] M. G. Martin and J. I. Siepmann. Transferable potentials for phase equilibria. 1. United-atom description of n-alkanes. *J. Phys. Chem. B*, 102(14):2569–2577, 1998. doi:10.1021/jp972543.

[59] D. L. Michels and M. Desbrun. A semi-analytical approach to molecular dynamics. *J. Comput. Phys.*, 303:336–354, 2015. doi:10.1016/j.jcp.2015.10.009.

[60] W. H. Miller. *A treatise on crystallography*. J. & JJ Deighton, 1839.

[61] T. Morishita. Fluctuation formulas in molecular-dynamics simulations with the weak coupling heat bath. *J. Chem. Phys.*, 113(8):2976–2982, 2000. doi:10.1063/1.1287333.

[62] A. Mulero, I. Cachadiña, and M. I. Parra. Recommended correlations for the surface tension of common fluids. *J. Phys. Chem. Ref. Data*, 41(4):043–105, 2012. doi:10.1063/1.4768782.

[63] S. Nosé. A molecular dynamics method for simulations in the canonical ensemble. *Mol. Phys.*, 52(2): 255–268, 1984. doi:10.1080/00268978400101201.

[64] J. C. Phillips, D. J. Hardy, J. D. C. Maia, J. E. Stone, J. V. Ribeiro, R. C. Bernardi, R. Buch, G. Fiorin, J. Hénin, W. Jiang, R. Mcgreevy, M. C. R. Melo, B. K. Radak, R. D. Skeel, A. Singharoy, Y. Wang, B. Roux, A. Aksimentiev, Z. Luthey-schulten, L. V. Kalé, K. Schulten, C. Chipot, and E. Tajkhorshid. Scalable molecular dynamics on cpu and gpu architectures with NAMD. *J. Chem. Phys.*, 153(4):044130, 2020. doi:10.1063/5.0014475.

[65] H. Poincaré. *Théorie du potentiel Newtonien: Leçons professées à la Sorbonne pendant le premier semestre 1894-1895*. Carré et Naud, 1899.

[66] D. C. Rapaport. *The art of molecular dynamics simulation*. Cambridge university press, Cambridge, 2 edition, 2004. doi:10.1017/CBO9780511816581.

[67] S. Reich. Backward error analysis for numerical integrators. *SIAM J. Numer. Anal.*, 36(5):1549–1570, 1999. doi:10.1137/S0036142997329797.

[68] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 29(1):209–228, 1992. doi:10.1137/0729014.

[69] Y. Saad. *Numerical methods for large eigenvalue problems: revised edition*. SIAM, 2011. doi:10.1137/1.9781611970739.

[70] T. Schlick. *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Springer, New York, 2 edition, 2010. doi:10.1007/978-1-4419-6351-2.

[71] T. Schlick, M. Mandziuk, R. D. Skeel, and K. Srinivas. Nonlinear resonance artifacts in molecular dynamics simulations. *J. Comput. Phys.*, 140(1):1–29, 1998. doi:10.1006/jcph.1998.5879.

[72] T. Schlick, R. D. Skeel, A. T. Brunger, L. V. Kalé, J. A. Board, Jr., J. Hermans, and K. Schulten. Algorithmic challenges in computational molecular biophysics. *J. Comput. Phys.*, 151(1):9–48, 1999. doi:10.1006/jcph.1998.6182. Computational molecular biophysics.

[73] M. Shen, P. K. Schelling, and P. Keblinski. Heat transfer mechanism across few-layer graphene by molecular dynamics. *Phys. Rev. B*, 88(4):045444, 2013. doi:10.1103/PhysRevB.88.045444.

[74] D. Sølvason, J. Kolafa, H. G. Petersen, and J. W. Perram. A rigorous comparison of the Ewald method and the fast multipole method in two dimensions. *Comput. Phys. Commun.*, 87(3):307–318, 1995. doi:10.1016/0010-4655(95)00027-D.

[75] F. Stillinger and T. Weber. Computer simulation of local order in condensed phases of silicon. *Physical Review B*, 31:5262–5271, 1985. doi:10.1103/PhysRevB.31.5262.

[76] S. D. Stoddard and J. Ford. Numerical experiments on the stochastic behavior of a Lennard-Jones gas system. *Phys. Rev. A*, 8:1504–1512, 1973. doi:10.1103/PhysRevA.8.1504.

[77] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506–517, 1968. doi:10.1137/0705041.

[78] A. Stukowski. Visualization and analysis of atomistic simulation data with ovito-the open visualization tool. *Model. Simul. Mater. Sci. Eng.*, 18(1), 2010. doi:10.1088/0965-0393/18/1/015012.

[79] J. Tersoff. New empirical approach for the structure and energy of covalent systems. *Phys. Rev. B*, 37: 6991–7000, 1988. doi:10.1103/PhysRevB.37.6991.

[80] J. Tersoff. Modeling solid-state chemistry: Interatomic potentials for multicomponent systems. *Phys. Rev. B*, 39:5566–5568, 1989. doi:10.1103/PhysRevB.39.5566.

[81] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comput. Phys. Commun.*, 271:108–171, 2022. doi:10.1016/j.cpc.2021.108171.

[82] H. F. Trotter. On the product of semi-groups of operators. *Proc. Amer. Math. Soc.*, 10:545–551, 1959. doi:10.2307/2033649.

[83] M. Tuckerman, B. J. Berne, and G. J. Martyna. Reversible multiple time scale molecular dynamics. *J. Chem. Phys.*, 97(3):1990–2001, 1992. doi:10.1063/1.463137.

[84] L. Verlet. Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Phys. Rev.*, 159:98–103, 1967. doi:10.1103/PhysRev.159.98.

[85] D. S. Watkins. *The Matrix Eigenvalue Problem.* Society for Industrial and Applied Mathematics, 2007. doi:10.1137/1.9780898717808.

[86] H. Weyl. *The classical groups: Their invariants and representations.* 1. Princeton university press, 1946. doi:https://doi.org/10.2307/j.ctv3hh48t.

A Numerical Linear Algebra

Here we collect some auxiliary result of numerical linear algebra that are used at some point during the thesis.

A.1 Compressed Sparse Row Format

A storage efficient way to store sparse matrix $A \in \mathbb{R}^{n\times n}$ with nnz nonzero entries in an unknown sparsity structure is the compressed row (or column) storage. It consists of three vectors. The first vector, `data`, lists all nonzero entries of the matrix in row-wise order. The second vector, `indices`, contains the column indices of each nonzero entry in `data`. The column index of $\texttt{data}(k)$ is $\texttt{indices}(k)$. The `ptr` stores the number of entries in each row implicitly and thus the row index of every nonzero entry. It holds that $\texttt{ptr}(1) = 1$ and that $\texttt{ptr}(k+1)-\texttt{ptr}(k)$ is equal to the number of nonzero elements in row k. Consequently, the last entry is defined by $\texttt{ptr}(n+1) = \text{nnz} + 1$. With this approach only $2\text{nnz}+n+1$ instead of n^2 elements need to be stored. For a small example we consider the matrix

$$A = \begin{bmatrix} 0 & a_{12} & 0 & a_{14} \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{44} \end{bmatrix}.$$

For this matrix, the compressed sparse row format reads

$$\begin{aligned} \texttt{data} &= [a_{12} \quad a_{14} \quad a_{21} \quad a_{22} \quad a_{23} \quad a_{44}] \\ \texttt{indices} &= [\ 2 \quad\ 4 \quad\ 1 \quad\ 2 \quad\ 3 \quad\ 4\ ] \\ \texttt{ptr} &= [1 \quad 3 \quad 6 \quad 6 \quad 7] \end{aligned}$$

Algorithm 10 shows how a sparse matrix vector product may be implemented. The complexity of the sparse matrix vector product is in $O(n \cdot \text{nonzero})$.

Algorithm 10 Sparse Matrix Vector Product

Input $A \in \mathbb{C}^{n\times n}$ in CSR format, $b \in \mathbb{C}^n$
  for $i = 1, \ldots, n$ **do**
    for $j = \texttt{ptr}[i], \ldots, \texttt{ptr}[i+1] - 1$ **do**
      $\texttt{out}[i] \leftarrow \texttt{out}[i] + \texttt{data}[j] \cdot b[\texttt{indices[j]}]$
    end for
  end for
Output out $\in \mathbb{C}^n$

A.2 The Minimal Polynomial

We define the minimal polynomial of a matrix $A \in \mathbb{C}^{n\times n}$ and give important properties with respect to matrix function evaluations.

Definition 9. *The monic polynomial p of least degree such that*

$$p(A) = 0, \quad \textit{with} \quad A \in \mathbb{C}^{n\times n}$$

is called minimal polynomial of A. *We denote it by p_A.*

We further define the minimal polynomial of $A \in \mathbb{C}^{n\times n}$ with respect to a vector $b \in \mathbb{C}^n$.

Definition 10. *The monic polynomial p of least degree such that*

$$p(A)b = 0, \quad \textit{with} \quad A \in \mathbb{C}^{n\times n},\ b \in \mathbb{C}^n$$

is called minimal polynomial of A with respect to b. *We denote it by $p_{A,b}$.*

The degree of the minimal polynomial of A with respect to b is of special interest for matrix function evaluation.

Theorem 4 (Higham 41, Thm 13.2)**.** *Let f be defined on the spectrum of $A \in \mathbb{C}^{n\times n}$ and let $p_{A,b}$ be the minimal polynomial of A with respect to b. Then $f(A)b = q(A)b$, where q is the unique Hermite interpolating polynomial of degree less than $\sum_{i=1}^{s} l_i = deg(p_{A,b})$ that satisfies the interpolation conditions*

$$q^{(i)}(\lambda_i) = f^{(j)}(\lambda_i), \qquad j = 0, \ldots, l_i - 1, \quad i = 1, \ldots, s.$$

A.3 Matrix Functions

We present the definition of a matrix function via the *Jordan* canonical form and give some basic results. For a more detailed look we refer to [41]. The Jordan canonical form of a matrix $A \in \mathbb{C}^{n\times n}$ is given by

$$J = \operatorname{diag}(J_1, J_2, \ldots, J_p) = S^{-1}AS, \quad J_k = J_k(\lambda_k) = \begin{bmatrix} \lambda_k & 1 & & \\ & \lambda_k & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_k \end{bmatrix} \in \mathbb{C}^{m_k\times m_k}, \tag{A.1}$$

with nonsingular transformation matrix S and $\sum_{k=1}^{p} m_k = n$. By $\lambda_1, \ldots, \lambda_k$ we denote the eigenvalues of A. A matrix function can be defined by

Definition 11 (Higham 41, Def. 1.1)**.** *The function f is said to be defined on the spectrum of $A \in \mathbb{C}^{n\times n}$ if the values*

$$f^{(j)}(\lambda_k), \qquad j = 0, \ldots, m_k - 1 \quad , k = 1, \ldots, p$$

exist. These are called the values of the function f on the spectrum of A.

Definition 12 (Higham 41, Def. 1.2)**.** *Let f be defined on the spectrum of $A \in \mathbb{C}^{n\times n}$ and let A have the Jordan canonical form (A.1). Then*

$$f(A) := Sf(J)S^{-1} = Sdiag(f(J_k))S^{-1},$$

where

$$f(J_k) := \begin{bmatrix} f(\lambda_k) & f'(\lambda_k) & \ldots & \frac{f^{(m_k-1)}(\lambda_k)}{(m_k-1)!} \\ & f(\lambda_k) & \ddots & \vdots \\ & & \ddots & f'(\lambda_k) \\ & & & f(\lambda_k) \end{bmatrix}.$$

If the matrix A is diagonalizable all Jordan blocks J_k are one-dimensional and J is a diagonal matrix. Consequently, $f(J)$ is a diagonal matrix.

B | Molecular Structures and Force Fields

Here we collect some additional information for structures and force fields in molecular dynamics.

B.1 Further Potentials

We present some more elaborate potentials. These potentials are constructed with a specific purpose. While they are more complicated and computational expensive than conventional potentials, they have been shown to describe certain systems with higher accuracy.

B.1.1 The Stillinger-Weber Potential

The *Stillinger-Weber* potential [75] is a classical potential that is used for the description of crystalline structures. The main purpose is the simulation of silicon crystalline structures. The 3−body Stillinger-Weber potential is implemented in LAMMPS in the following way.

$$
\begin{aligned}
E &= \sum_i \sum_{j>i} \phi_2(r_{ij}) + \sum_i \sum_{j\neq i} \sum_{k>j} \phi_3(r_{ij}, r_{ik}, \theta_{ijk}) \\
\phi_2(r_{ij}) &= A_{ij}\varepsilon_{ij} \left[B_{ij} \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{p_{ij}} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{q_{ij}} \right] \exp\left(\frac{\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}}\right) \\
\phi_3(r_{ij}, r_{ik}, \theta_{ijk}) &= \lambda_{ijk}\varepsilon_{ijk} \left[\cos(\theta_{ijk}) - \cos(\bar{\theta})\right]^2 \exp\left(\frac{\gamma_{ij}\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}}\right) \exp\left(\frac{\gamma_{ik}\sigma_{ik}}{r_{ik} - a_{ik}\sigma_{ik}}\right)
\end{aligned}
$$

There are two types of interactions modeled by the potential. The two-body interactions are described by ϕ_2 and the three-body interactions by ϕ_3. The parameters A, B, p, q are used for the two-body interactions, λ and $\bar{\theta}$ are parameters that are used for the three body interactions, and ε, σ and a are used for the two- and three-body interactions. All these parameters are system dependent and have to be defined in advanced.

B.1.2 The Tersoff Potential

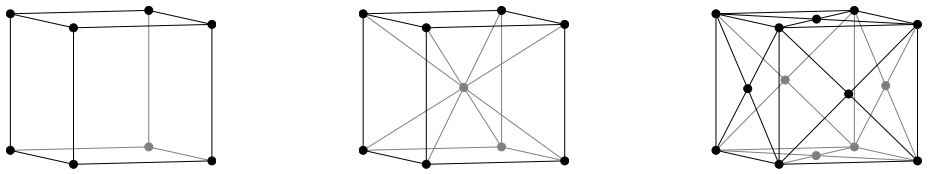
The Tersoff potential [79] belongs to the class of bond order potentials. Bond order potentials are a class of analytical potentials [10, 79, 80]. In contrast to the conventional potentials, these potentials can describe different bonding states of a particle. In particular chemical reactions that involve the breaking and forming of bonds can be modeled with this class of potentials. There is a broad range of application for analytical potentials from crystalline structures to description of liquids. The Tersoff potential is the most used analytical, inter-atomic potential to describe covalently bonded carbon particles in graphene. We give the definition of the 3−body Tersoff potential as implemented in LAMMPS.

$$
\begin{aligned}
E &= \frac{1}{2}\sum_i \sum_{j\neq i} V_{ij} \\
V_{ij} &= f_C(r_{ij}+\delta)\left[f_R(r_{ij}+\delta) + b_{ij} f_A(r_{ij}+\delta)\right] \\
f_C(r) &= \begin{cases} 1 & r < R-D \\ \frac{1}{2} - \frac{1}{2}\sin(\frac{\pi}{2}\frac{r-R}{D}) & R-D < r < R+D \\ 0 & r > R+D \end{cases} \\
f_R(r) &= A\exp(-\lambda_1 r) \\
f_A(r) &= -B\exp(-\lambda_2 r) \\
b_{ij} &= (1+\beta^n \zeta_{ij}^n)^{-\frac{1}{2n}} \\
\zeta_{ij} &= \sum_{k\neq i,j} f_C(r_{ik}+\delta) g[\theta_{ijk}(r_{ij}, r_{ik})] \exp[\lambda_3^m (r_{ij}-r_{ik})^m] \\
g(\theta) &= \gamma_{ijk}\left(1 + \frac{c^2}{d^2} - \frac{c^2}{d^2 + (\cos(\theta) - cos(\bar{\theta}))^2}\right)
\end{aligned}
$$

Here, δ is an optional negative shift of the equilibrium bond length. There are two types of interactions modeled by the potential. The two-body interactions are described by f_R and the three-body interactions by f_A. The parameters $n, \beta, \lambda_1, \lambda_2, A, B$ are used for the two-body interactions, $m, \gamma, \lambda_3, c, d$ and $\bar{\theta}$ are used for the three body interactions, and R and D are distance parameters that are used for the two- and three-body interactions. All these parameters are system dependent and have to be defined in advanced.

B.2 Crystal Structure

A crystalline structure is a repeating arrangement of atoms, molecules, or ions in a three-dimensional pattern. In a crystalline structure, the atoms are arranged in a highly ordered and repeating pattern called a crystal lattice, which determines the material's physical and chemical properties. A crystal lattice is made up of a repeating unit called a unit cell, which is the smallest portion of the crystal lattice that contains all the information necessary to reproduce the entire lattice structure. The shape and the number of the unit cells determine the crystal's overall shape and dimensions, as well as its physical and chemical properties. In figure B.1 we present three commonly used unit cell types, the simple, body-centered, and face-centered cubic.



(a) Simple cubic **(b)** Body-centered cubic **(c)** Face-centered cubic

Figure B.1. Three different unit cell types from which a large crystalline structure may be created.

When joining the crystal lattice points by straight lines, one gets the crystal planes. The orientation of these crystal planes can be described by the *Miller* indices [60]. Depending on this orientation the structure has different properties, e.g., conductivity. The Miller indices, denoted by (h, k, l), describe the intercepts of the plane along the crystal planes. We present a selection of miller indices in the context of the unit cube in figure B.2.

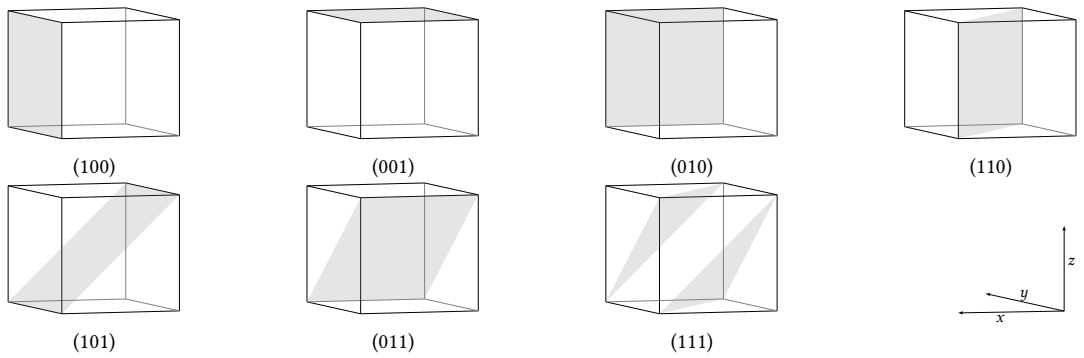


Figure B.2. A selection of planes with different Miller indices determining the orientation of the crystalline strucure.

By combining the structure and number of unit cells in each dimension with the orientation given by the Miller index one can uniquely define a complete large scale crystalline structure.