



Discrete gradients in short-range molecular dynamics simulations

Volker Grimm¹ · Tobias Kliesch¹ · G. R. W. Quispel²

Received: 21 August 2023 / Accepted: 28 November 2023
© The Author(s) 2023

Abstract

Discrete gradients (DG) or more exactly discrete gradient methods are time integration schemes that are custom-built to preserve first integrals or Lyapunov functions of a given ordinary differential equation (ODE). In conservative molecular dynamics (MD) simulations, the energy of the system is constant and therefore a first integral of motion. Hence, discrete gradient methods seem to be a natural choice as an integration scheme in conservative molecular dynamics simulations.

Keywords Discrete gradient · Geometric numerical integration · Molecular dynamics · Linked cell method · Parallel computing

Mathematics Subject Classification (2010) 65L05 · 65Y05 · 65P10 · 82M37

1 Introduction

In molecular dynamics (MD) simulations, or, more generally, in particle methods, one is usually not interested in the trajectory of a single particle but in derived and/or averaged quantities that are often related to geometric properties of the underlying equations of motion. For a meaningful simulation, the preservation of these geometric

Volker Grimm, Tobias Kliesch and G. R. W. Quispel contributed equally to this work

✉ Volker Grimm
volker.grimm@kit.edu
Tobias Kliesch
tobias.kliesch@kit.edu
G. R. W. Quispel
r.quispel@latrobe.edu.au

- ¹ Department of Mathematics, Karlsruhe Institute of Technology (KIT), D-76131 Karlsruhe, Germany
- ² Department of Mathematical and Physical Sciences, La Trobe University, Bundoora, Melbourne, Victoria 3086, Australia

properties is important (e.g. [21, 30, 34]). In a microcanonical or (NVE) ensemble, important macroscopic variables are the total number of particles (N), the system's volume (V) and the total energy (E). Microcanonical MD simulations should therefore preserve the total energy, which corresponds to a first integral of the equations of motion (e.g. [51]). In the Hamiltonian formulation, the total energy or Hamiltonian $H : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, the positions $\mathbf{q} \in \mathbb{R}^d$ and momenta $\mathbf{p} \in \mathbb{R}^d$ determine the equations of motion as $\mathbf{q}' = \nabla_{\mathbf{p}} H(\mathbf{q}, \mathbf{p})$, $\mathbf{p}' = -\nabla_{\mathbf{q}} H(\mathbf{q}, \mathbf{p})$. In addition to the preserved energy, Hamiltonian systems possess another important structural property, the symplecticity of the Hamiltonian flow (cf. [29, 52]). Unfortunately, the symplectic structure and the total energy can generally not be preserved exactly at the same time. A class of geometric integrators that can preserve first integrals and Lyapunov functions exactly are discrete gradient methods. They have first been considered energy-conserving schemes (e.g. [12, 13, 24]), and they have then been generalised to arbitrary first integrals of Hamiltonian and non-Hamiltonian systems (cf. [46]) and Lyapunov functions (cf. [36, 56]). Other methods that are able to preserve integrals and Lyapunov functions are projection methods (e.g. [18]), which are related to discrete gradient methods (cf. [43]). In addition to many applications, discrete gradient methods can be used to preserve the energy of suitably discretised variational partial differential equations (e.g. [8, 9, 40–42, 44, 60]) as well as to ensure the dissipation of gradient systems in image processing (e.g. [16, 17, 19, 49]). They have been generalised to manifolds (cf. [6, 7]) and inspire new ideas in smooth optimization (e.g. [11, 48]) and deep learning (e.g. [5]).

Following this introduction, discrete gradients are discussed in Section 2. Their use for Hamiltonian systems in MD simulations is studied in Section 3. Special discrete gradients for molecular dynamics are presented in Section 4. The parallelisation of DG methods is discussed in Section 5. Finally, a brief conclusion is given in Section 6.

2 Discrete gradients in geometric integration

In order to remind the reader of discrete gradients, we follow [35].

Definition 1 (Gonzalez 1996) *Let $V : \mathbb{R}^n \rightarrow \mathbb{R}$ be continuously differentiable. The function $\bar{\nabla}V : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a discrete gradient of V iff it is continuous and*

$$\left\{ \begin{array}{l} \langle \bar{\nabla}V(\mathbf{u}, \mathbf{u}'), (\mathbf{u}' - \mathbf{u}) \rangle = V(\mathbf{u}') - V(\mathbf{u}), \\ \bar{\nabla}V(\mathbf{u}, \mathbf{u}) = \nabla V(\mathbf{u}), \end{array} \right. \quad \text{for all } \mathbf{u}, \mathbf{u}' \in \mathbb{R}^n.$$

The discrete gradient is symmetric, if and only if

$$\bar{\nabla}V(\mathbf{u}, \mathbf{u}') = \bar{\nabla}V(\mathbf{u}', \mathbf{u}) \quad \text{for all } \mathbf{u}, \mathbf{u}' \in \mathbb{R}^n.$$

For continuously differentiable V , the second defining property follows from the first and might be omitted (cf. [10]). An interpretation of the following proposition, that can be found as Proposition 3.2 in [35], is that the component of any discrete gradient in the direction $(\mathbf{u}' - \mathbf{u})/\|\mathbf{u}' - \mathbf{u}\|$ is $(V(\mathbf{u}') - V(\mathbf{u}))/\|\mathbf{u}' - \mathbf{u}\|$.

Proposition 1 $\bar{\nabla}V(\mathbf{u}, \mathbf{u}')$ is a discrete gradient if and only if it is continuous and

$$\bar{\nabla}V(\mathbf{u}, \mathbf{u}') = \frac{V(\mathbf{u}') - V(\mathbf{u})}{\|\mathbf{u}' - \mathbf{u}\|^2}(\mathbf{u}' - \mathbf{u}) + w(\mathbf{u}, \mathbf{u}'), \quad (\mathbf{u} \neq \mathbf{u}'),$$

where $w(\mathbf{u}, \mathbf{u}')$ is a vector-valued function such that

$$\begin{cases} \langle w(\mathbf{u}, \mathbf{u}'), (\mathbf{u}' - \mathbf{u}) \rangle = 0, & (\mathbf{u} \neq \mathbf{u}'), \\ \lim_{\mathbf{u}' \rightarrow \mathbf{u}} w(\mathbf{u}, \mathbf{u}') - P_{(\mathbf{u}' - \mathbf{u})^\perp} \nabla V(\mathbf{u}) = 0, \end{cases}$$

where $P_{(\mathbf{u}' - \mathbf{u})^\perp}$ is the projection on the space perpendicular to $(\mathbf{u}' - \mathbf{u})$.

For the Euclidean inner product, $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{v}^T \mathbf{u}$, for $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, the projection can be written as the matrix

$$P_{(\mathbf{u}' - \mathbf{u})^\perp} = I_n - \frac{(\mathbf{u}' - \mathbf{u})(\mathbf{u}' - \mathbf{u})^T}{\|\mathbf{u}' - \mathbf{u}\|^2}$$

where I_n is the $n \times n$ identity matrix and \cdot^T means transpose. The following simple fact is quite useful.

Lemma 1 Let $\bar{\nabla}V^i$ be discrete gradients for $V^i, i = 1, \dots, N$. Then,

$$\bar{\nabla}V = \sum_{i=1}^N \bar{\nabla}V^i \quad \text{is a discrete gradient for} \quad V = \sum_{i=1}^N V^i.$$

We omit the obvious proof of Lemma 1. Examples of well-known discrete gradients are the *midpoint discrete gradient* or *Gonzalez discrete gradient* (cf. [12])

$$\bar{V}_{MP}(\mathbf{u}, \mathbf{u}') = \nabla V\left(\frac{\mathbf{u} + \mathbf{u}'}{2}\right) + \frac{V(\mathbf{u}') - V(\mathbf{u}) - \left\langle \nabla V\left(\frac{\mathbf{u} + \mathbf{u}'}{2}\right), \mathbf{u}' - \mathbf{u} \right\rangle}{\|\mathbf{u} - \mathbf{u}'\|^2}(\mathbf{u}' - \mathbf{u}), \tag{1}$$

where $\mathbf{u} \neq \mathbf{u}'$, the *mean value discrete gradient* (cf. [22])

$$\bar{V}_{MV}(\mathbf{u}, \mathbf{u}') = \int_0^1 \nabla V((1 - \xi)\mathbf{u} + \xi\mathbf{u}') d\xi, \quad (\mathbf{u} \neq \mathbf{u}'), \tag{2}$$

and the *coordinate increment discrete gradient* (cf. [24])

$$\bar{V}_{CI}(\mathbf{u}, \mathbf{u}') = \begin{pmatrix} \frac{V(u'_1, u_2, \dots, u_n) - V(u_1, u_2, \dots, u_n)}{u'_1 - u_1} \\ \frac{V(u'_1, u'_2, \dots, u_n) - V(u'_1, u_2, \dots, u_n)}{u'_2 - u_2} \\ \vdots \\ \frac{V(u'_1, \dots, u'_{n-1}, u_n) - V(u'_1, \dots, u_{n-1}, u_n)}{u'_{n-1} - u_{n-1}} \\ \frac{V(u'_1, \dots, u'_{n-1}, u'_n) - V(u'_1, \dots, u'_{n-1}, u_n)}{u'_n - u_n} \end{pmatrix}, \tag{3}$$

where $0/0$ is understood to be $\partial V/\partial u_i$. The midpoint discrete gradient is symmetric, i.e. $\bar{V}_{MP}(\mathbf{u}, \mathbf{u}') = \bar{V}_{MP}(\mathbf{u}', \mathbf{u})$ for all $\mathbf{u} \neq \mathbf{u}'$, as is the mean value discrete gradient. The coordinate increment discrete gradient is not symmetric, but it can be symmetrised (cf. [37])

$$\bar{V}_{SI}(\mathbf{u}, \mathbf{u}') = \frac{1}{2} (\bar{V}_{CI}(\mathbf{u}, \mathbf{u}') + \bar{V}_{CI}(\mathbf{u}', \mathbf{u})) .$$

A discrete gradient can be used as the discretised version of the gradient of a first integral or the gradient of a Lyapunov function of an ordinary differential equation (ODE). If the ODE is smooth enough and possesses a first integral or Lyapunov function V , then the ODE can be written as

$$\mathbf{y}' = f(\mathbf{y}) = A(\mathbf{y})\nabla V(\mathbf{y}), \tag{4}$$

where $A(\mathbf{y})$ is an equally smooth antisymmetric matrix, when V is a first integral, and a negative semidefinite matrix, when V is a Lyapunov function (cf. [35]). A *discrete gradient method* then reads

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \tau \tilde{A}(\mathbf{y}_n, \mathbf{y}_{n+1}, \tau) \bar{\nabla} V(\mathbf{y}_n, \mathbf{y}_{n+1}), \quad n = 0, 1, 2, \dots , \tag{5}$$

where $\tilde{A}(\mathbf{y}, \mathbf{y}, 0) = A(\mathbf{y})$, for consistency, and $\bar{\nabla} V$ is a discrete gradient for V . If the discrete gradient is symmetric and $\tilde{A}(\mathbf{y}_{n+1}, \mathbf{y}_n, -\tau) = \tilde{A}(\mathbf{y}_n, \mathbf{y}_{n+1}, \tau)$ holds for all possible values, then the discrete gradient method (5) is called time-symmetric or self-adjoint (cf. Definition 3.2 in [36]).

3 DG methods in MD simulations

The equations of motion in molecular dynamics can be conveniently stated in *Hamiltonian* form. Given an arbitrary (smooth) Hamiltonian function $H : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ on the phase space $\mathbb{R}^d \times \mathbb{R}^d$, $d \geq 1$, the corresponding Hamiltonian equations of motion are

$$\begin{aligned} \mathbf{q}' &= \nabla_{\mathbf{p}} H(\mathbf{q}, \mathbf{p}), \\ \mathbf{p}' &= -\nabla_{\mathbf{q}} H(\mathbf{q}, \mathbf{p}). \end{aligned} \tag{6}$$

The Hamiltonian corresponds to the total energy of the system that is preserved in a conservative simulation. In order to recognise the fact that (6) is a special case of (4), we set $\mathbf{y} = (\mathbf{q}, \mathbf{p})^T$ and define the matrix

$$J = \begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix} \in \mathbb{R}^{2d \times 2d},$$

where I_d is the identity matrix of dimension d . With these definitions, system (6) reads

$$\mathbf{y}' = J \nabla_{\mathbf{y}} H(\mathbf{y}) . \tag{7}$$

Due to

$$\frac{d}{dt} H(\mathbf{y}(t)) = \nabla_{\mathbf{y}} H(\mathbf{y}(t))^T \mathbf{y}'(t) = \nabla_{\mathbf{y}} H(\mathbf{y}(t))^T J \nabla_{\mathbf{y}} H(\mathbf{y}(t)) = 0 ,$$

the Hamiltonian H (energy) is preserved along solutions of the system. With an arbitrary discrete gradient satisfying Definition 1, the corresponding discrete gradient method reads

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \tau J \bar{\nabla} H(\mathbf{y}^n, \mathbf{y}^{n+1}), \tag{8}$$

with step size $\tau > 0$. The discrete gradient method also preserves the Hamiltonian, since Definition 1 gives

$$\begin{aligned} H(\mathbf{y}^{n+1}) - H(\mathbf{y}^n) &= \bar{\nabla} H(\mathbf{y}^n, \mathbf{y}^{n+1})^T (\mathbf{y}^{n+1} - \mathbf{y}^n) \\ &\stackrel{(8)}{=} \tau \bar{\nabla} H(\mathbf{y}^n, \mathbf{y}^{n+1})^T J \bar{\nabla} H(\mathbf{y}^n, \mathbf{y}^{n+1}) = 0. \end{aligned}$$

The method (8) is time-symmetric, whenever the chosen discrete gradient is symmetric.

3.1 Separable Hamiltonian systems

If the Hamiltonian is *separable*, i.e.

$$H(\mathbf{q}, \mathbf{p}) = T(\mathbf{p}) + V(\mathbf{q}), \tag{9}$$

one can apply a different discrete gradient to T or V , respectively, in order to obtain a discrete gradient for $H(\mathbf{q}, \mathbf{p})$. System (7) now reads

$$\begin{bmatrix} \mathbf{q} \\ \mathbf{p} \end{bmatrix}' = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \bar{\nabla}_{\mathbf{q}} V \\ \bar{\nabla}_{\mathbf{p}} T \end{bmatrix}. \tag{10}$$

Given two discrete gradients $\bar{\nabla}_{\mathbf{q}} V$ for V and $\bar{\nabla}_{\mathbf{p}} T$ for T , respectively, the discrete gradient method is

$$\begin{aligned} \mathbf{q}^{n+1} &= \mathbf{q}^n + \tau \bar{\nabla}_{\mathbf{p}} T(\mathbf{p}^n, \mathbf{p}^{n+1}), \\ \mathbf{p}^{n+1} &= \mathbf{p}^n - \tau \bar{\nabla}_{\mathbf{q}} V(\mathbf{q}^n, \mathbf{q}^{n+1}). \end{aligned} \tag{11}$$

Method (11) exactly preserves the energy, which is noted in the following lemma.

Lemma 2 *For the separable Hamiltonian system (10) and two discrete gradients $\bar{\nabla}_{\mathbf{q}} V$ and $\bar{\nabla}_{\mathbf{p}} T$, method (11) preserves the energy exactly, i.e.*

$$H(\mathbf{q}^{n+1}, \mathbf{p}^{n+1}) = H(\mathbf{q}^n, \mathbf{p}^n), \quad n = 0, 1, 2, \dots$$

Proof From Definition 1 followed by (11), we find

$$\begin{aligned} H(\mathbf{q}^{n+1}, \mathbf{p}^{n+1}) - H(\mathbf{q}^n, \mathbf{p}^n) &= T(\mathbf{p}^{n+1}) - T(\mathbf{p}^n) + V(\mathbf{q}^{n+1}) - V(\mathbf{q}^n) \\ &= \bar{\nabla}_{\mathbf{p}} T(\mathbf{p}^n, \mathbf{p}^{n+1})^T (\mathbf{p}^{n+1} - \mathbf{p}^n) + \bar{\nabla}_{\mathbf{q}} V(\mathbf{q}^n, \mathbf{q}^{n+1})^T (\mathbf{q}^{n+1} - \mathbf{q}^n) \\ &= -\tau \bar{\nabla}_{\mathbf{p}} T(\mathbf{p}^n, \mathbf{p}^{n+1})^T \bar{\nabla}_{\mathbf{q}} V(\mathbf{q}^n, \mathbf{q}^{n+1}) + \tau \bar{\nabla}_{\mathbf{q}} V(\mathbf{q}^n, \mathbf{q}^{n+1})^T \bar{\nabla}_{\mathbf{p}} T(\mathbf{p}^n, \mathbf{p}^{n+1}) \\ &= 0. \end{aligned}$$

□

The proof of Lemma 2 also shows that

$$\begin{bmatrix} \bar{\nabla}_{\mathbf{p}} T \\ \bar{\nabla}_{\mathbf{q}} V \end{bmatrix} = \bar{\nabla} H$$

is a discrete gradient for H for any choice of a discrete gradient $\bar{\nabla}_{\mathbf{q}} V$ and $\bar{\nabla}_{\mathbf{p}} T$ whenever the Hamiltonian H is separable, cf. (9). If both discrete gradients in (11) are symmetric then the method is time-symmetric.

In molecular dynamics, the Hamiltonian often is of the even simpler form

$$H(\mathbf{q}, \mathbf{p}) = \frac{1}{2} \mathbf{p}^T M^{-1} \mathbf{p} + V(\mathbf{q}), \quad \text{i.e. } T(\mathbf{p}) = \frac{1}{2} \mathbf{p}^T M^{-1} \mathbf{p} \quad (12)$$

is a quadratic function that corresponds to the kinetic energy. M^{-1} is a diagonal matrix with the inverses of the masses of the corresponding particles. For quadratic functions, any discrete gradient reduces to the midpoint rule. Choosing the midpoint discrete gradient for $\bar{\nabla}_{\mathbf{p}} T$, one thus obtains

$$\begin{aligned} \mathbf{q}^{n+1} &= \mathbf{q}^n + \tau M^{-1} \frac{\mathbf{p}^{n+1} + \mathbf{p}^n}{2}, \\ \mathbf{p}^{n+1} &= \mathbf{p}^n - \tau \bar{\nabla}_{\mathbf{q}} V(\mathbf{q}^n, \mathbf{q}^{n+1}). \end{aligned} \quad (13)$$

Inserting the second equation in the first leads to the system

$$\begin{aligned} \mathbf{q}^{n+1} &= \mathbf{q}^n + \tau M^{-1} \mathbf{p}^n - \frac{\tau^2}{2} M^{-1} \bar{\nabla}_{\mathbf{q}} V(\mathbf{q}^n, \mathbf{q}^{n+1}), \\ \mathbf{p}^{n+1} &= \mathbf{p}^n - \tau \bar{\nabla}_{\mathbf{q}} V(\mathbf{q}^n, \mathbf{q}^{n+1}), \end{aligned} \quad (14)$$

which will be used for the computation. The first equation is implicit in \mathbf{q}^{n+1} and takes some effort to solve numerically. The momenta \mathbf{p}^{n+1} are easily computed, once the first equation has been solved. If the first appearance of the discrete gradient was replaced by the true gradient at \mathbf{q}^n and the second discrete gradient by the average of the true gradients at \mathbf{q}^{n+1} and \mathbf{q}^n , respectively, we would recover the well-known Velocity-Störmer-Verlet method. Method (14) might therefore be called Velocity-DG method.

Proposition 2 *The method (13) or method (14), respectively, exactly preserves the Hamiltonian (12). If the discrete gradient $\bar{\nabla}_{\mathbf{q}} V$ is symmetric, then the scheme is time-symmetric (reversible). If the discrete gradient $\bar{\nabla}_{\mathbf{q}} V$ is symmetric and sufficiently smooth, then the method is of second order for sufficiently smooth V .*

Proof The scheme (14) exactly preserves the Hamiltonian (12), since it is just a reformulation of scheme (11) for this Hamiltonian and Lemma 2 applies. Exchanging $\tau \leftrightarrow -\tau$, $\mathbf{q}^{n+1} \leftrightarrow \mathbf{q}^n$ and $\mathbf{p}^{n+1} \leftrightarrow \mathbf{p}^n$ shows the time-symmetry for symmetric discrete gradients. Finally, Definition 1 and the symmetry of the scheme show second-order accuracy (cf. Theorem 8.10 in [20]). □

A more elaborate proof of second order for the midpoint discrete gradient applied to the full system (7) can be found as the proof of Theorem 8.5.4 in [56]. For separable systems, this corresponds to a special case of our Proposition 2.

Analogous to the elimination of the velocities (momenta) in the Verlet algorithm, one may derive a two-step formulation by adding the first line of (14) to the one obtained with negative time step $-\tau$, i.e.

$$\mathbf{q}^{n+1} - 2\mathbf{q}^n + \mathbf{q}^{n-1} = -\frac{\tau^2}{2}M^{-1} \left(\bar{\nabla}_{\mathbf{q}}V(\mathbf{q}^n, \mathbf{q}^{n+1}) + \bar{\nabla}_{\mathbf{q}}V(\mathbf{q}^n, \mathbf{q}^{n-1}) \right).$$

For the solution of the first equation in (14), the equation is transformed to $F(\mathbf{u}) = \mathbf{0}$, where

$$F(\mathbf{u}) = \mathbf{u} - \mathbf{q}^n - \tau M^{-1}\mathbf{p}^n + \frac{\tau^2}{2}M^{-1}\bar{\nabla}_{\mathbf{q}}V(\mathbf{q}^n, \mathbf{u})$$

and the Newton method is applied in the vicinity of \mathbf{q}^n . This leads to the iteration

$$\frac{\partial F}{\partial \mathbf{u}}(\mathbf{u}^m)\Delta\mathbf{u}^m = -F(\mathbf{u}^m), \quad \Delta\mathbf{u}^m = \mathbf{u}^{m+1} - \mathbf{u}^m, \quad m = 0, 1, 2, \dots \quad (15)$$

where $\mathbf{u}^0 \approx \mathbf{q}^n$ and

$$\frac{\partial F}{\partial \mathbf{u}}(\mathbf{u}^m) = I + \frac{\tau^2}{2}M^{-1}\frac{\partial \bar{\nabla}_{\mathbf{q}}V}{\partial \mathbf{u}}(\mathbf{q}^n, \mathbf{u}^m). \quad (16)$$

In order to reduce the computational work, one can also use the quasi-Newton iteration

$$J_F(\mathbf{u}^m)\Delta\mathbf{u}^m = -F(\mathbf{u}^m), \quad \Delta\mathbf{u}^m = \mathbf{u}^{m+1} - \mathbf{u}^m, \quad m = 0, 1, 2, \dots \quad (17)$$

where $J_F(\mathbf{u}^m)$ is an approximation to the full Jacobian. For example, for the midpoint discrete gradient (1),

$$J_F(\mathbf{u}^m) = I + \frac{\tau^2}{4}M^{-1}\frac{\partial \nabla V}{\partial \mathbf{u}}\left(\frac{\mathbf{u}^m + \mathbf{q}^n}{2}\right) \quad (18)$$

might be used. This is just the Jacobian that would occur in the implicit midpoint rule. This way, a loop over all particles to compute the potential $V(\mathbf{u}^m)$, the gradient $\nabla V(\mathbf{u}^m)$ and the norm of the difference of the positions is avoided. The matrices $\frac{\partial F}{\partial \mathbf{u}}$ and $J_F(\mathbf{u}^m)$ are symmetric, and the linear systems in (15) and (17), respectively, can be solved efficiently by the conjugate gradients (CG) method (cf. [23]). Newton iterations also appear in the discretisation schemes SHAKE and RATTLE that are custom-built to pose constraints during molecular dynamics simulations (cf. [1, 50]). The method is often also referred to as SHAKE and RATTLE, since they have been found to be equivalent in [31].

4 Discrete gradients for molecular dynamics

In this section, we discuss previously known as well as new discrete gradients that are designed for use in particle and molecular dynamics. A basic idea for discrete gradients custom-built for molecular dynamics is to mimic the standard forces by

discrete gradients. That is, for a pairwise force, the discrete gradient will be built upon two particles. For angle forces, three particles are involved, and for torsion forces, four particles are involved. Therefore, we consider N particles with masses m_1, \dots, m_N , positions $\mathbf{q}_i, i = 1, \dots, N$, and momenta $\mathbf{p}_i, i = 1, \dots, N$. Here, \mathbf{p}_i or \mathbf{q}_i are three-dimensional vectors. The momenta are given as $\mathbf{p}_i = m_i \mathbf{v}_i$, where \mathbf{v}_i are the velocities of the particles. We designate by $\mathbf{r}_{ij} = \mathbf{q}_j - \mathbf{q}_i$ the vector, that points from particle i to particle j . Its length is noted as $r_{ij} = \|\mathbf{q}_j - \mathbf{q}_i\|$, where $\|\cdot\|$ designates the Euclidean norm. The connection with the vectors used in the previous sections is given as

$$\mathbf{q} = \begin{pmatrix} \mathbf{q}_1 \\ \vdots \\ \mathbf{q}_N \end{pmatrix}, \quad \mathbf{p} = \begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \end{pmatrix}.$$

The vector \mathbf{q} collects the positions of the particles, and the vector \mathbf{p} the momenta, respectively.

4.1 Discrete gradients for pairwise forces

A discrete gradient for pairwise forces has been known for quite some time. The first appearance is due to LaBudde & Greenspan [26–28]. Since then, this discrete gradient has been studied by several scientists (e.g. [14, 47, 54, 55]). If $V(r_{ij})$ is a pairwise potential for the particles i and j , then we obtain, with the finite difference

$$\Delta_{r_{ij}} V(\mathbf{q}, \mathbf{q}') := \frac{V(r'_{ij}) - V(r_{ij})}{r'_{ij} - r_{ij}}, \tag{19}$$

the discrete gradient

$$\bar{\nabla} V(\mathbf{q}, \mathbf{q}') = \begin{pmatrix} \bar{\nabla}_{\mathbf{q}_1} V(\mathbf{q}, \mathbf{q}') \\ \vdots \\ \bar{\nabla}_{\mathbf{q}_N} V(\mathbf{q}, \mathbf{q}') \end{pmatrix}$$

with the non-zero components

$$\bar{\nabla}_{\mathbf{q}_i} V(\mathbf{q}, \mathbf{q}') = -\Delta_{r_{ij}} V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{ij} + \mathbf{r}_{ij}}{r'_{ij} + r_{ij}}, \quad \bar{\nabla}_{\mathbf{q}_j} V(\mathbf{q}, \mathbf{q}') = \Delta_{r_{ij}} V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{ij} + \mathbf{r}_{ij}}{r'_{ij} + r_{ij}}, \tag{20}$$

and $\bar{\nabla}_{\mathbf{q}_k} V(\mathbf{q}, \mathbf{q}') = 0$ for $k \neq i, j$. The formulas (20) form a discrete gradient for pairwise potentials $V(r_{ij})$. This can be seen by checking the conditions in Definition 1. The following theorem is due to LaBudde and Greenspan (cf. [26]). The theorem and its proof might also be found as Theorem 5.1 in [21].

Theorem 1 *The method (13) with (20) for a system of N particles with Hamiltonian*

$$H(\mathbf{q}, \mathbf{p}) = \frac{1}{2} \mathbf{p}^T M^{-1} \mathbf{p} + V(\mathbf{q}), \quad V = \sum_{k=1}^M V^k,$$

where V^k are pairwise potentials, is a second-order symmetric implicit method that conserves the energy, the total linear momentum $\mathbf{P} = \sum_{i=1}^N \mathbf{p}_i$ and the total angular momentum $\mathbf{L} = \sum_{i=1}^N \mathbf{q}_i \times \mathbf{p}_i$.

Note that the important part of the theorem is the conservation of the energy. Several schemes, such as the explicit Verlet scheme, preserve the total linear momentum and the total angular momentum. The statement about the total linear momentum and the total angular momentum in the theorem is given in order to highlight that the discrete gradient methods with the proposed discrete gradients also preserve these quantities.

4.2 Experiment with two Lennard–Jones particles

The Lennard–Jones potential is one of the most used models for the interaction of neutral particles (cf. [32]). We will use the standard 12–6 potential

$$V(r_{ij}) = 4\epsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right) = 4\epsilon \left(\frac{\sigma}{r_{ij}} \right)^6 \cdot \left(\left(\frac{\sigma}{r_{ij}} \right)^6 - 1 \right). \quad (21)$$

In our experiment, we use $\sigma = 1$ and $\epsilon = 5$. The initial conditions are shown in Fig. 1. In the positions section, the atoms are numbered together with their initial location. The velocities are given in the next section, while the particle is identified by its number. The particles are attracted to each other and then repelled again. This gives a periodic movement. Moreover, due to the initial velocities, they are turning around each other.

With this setup, we first compute the trajectories of the particles up to time 10 with different step sizes. The error is shown in Fig. 2 on the left-hand side. The error is shown versus the chosen step sizes. All three methods show a second-order error behaviour. The black line indicates the second-order slope in the logarithmic plot. On the right-hand side, the calculated energies are shown for all time steps with step size

Code fragment 1 Data

Positions

1	6.010216	5.000000	5.000000
2	5.000000	5.000000	5.000000

Velocities

1	0.000000	1.000000	0.000000
2	0.000000	-1.000000	0.000000

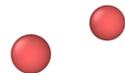


Fig. 1 Initial conditions for the experiment with two Lennard–Jones particles: the positions and velocities are given on the left-hand side. The first column refers to the particle number, the next three columns to the three coordinates. On the right-hand side, the initial positions are given as a plot made by the open visualization tool (OVITO) (cf. [57])

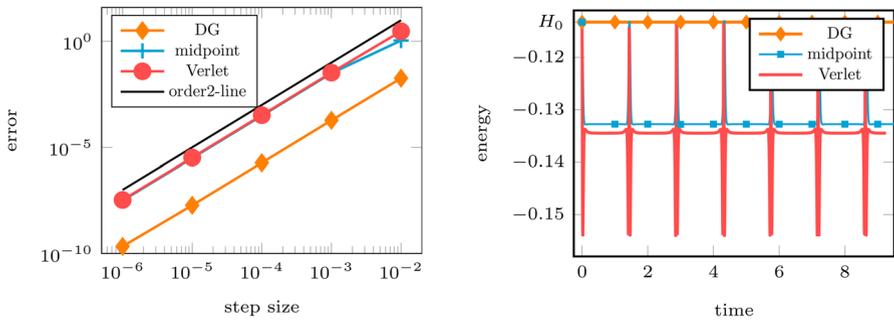


Fig. 2 Results of the experiment with two Lennard–Jones particles: the error versus the time step is shown on the left-hand side for the discrete gradient (DG) method, the midpoint rule, and the Verlet scheme. On the right-hand side, the energy is shown over the time span $[0, 10]$ for step size $\tau = 0.005$ for all three methods. H_0 designates the exact energy

$\tau = 0.005$. The discrete gradient method preserves the energy correctly, while the Verlet scheme and the implicit midpoint rule show a periodic change of the energy. The behaviour of the Verlet scheme and the implicit midpoint rule is expected, since symplectic integrators preserve a modified Hamiltonian. The total linear momentum and the total angular momentum are preserved by all three methods. The implicit equation in (14) is solved with the Newton method, (15), with the full Jacobian (16). The Jacobian is a symmetric matrix, and the CG method is used to solve the linear systems. The Jacobian is not explicitly formed. Instead, the action of the Jacobian on a vector is directly computed. This way, the computational effort is comparable to computing the forces. Sometimes, this is called a matrix-free implementation.

4.3 Discrete gradients for bond angles

We discuss some discrete gradients for bond angles. Besides general discrete gradients, e.g. (1), (2) and (3), restricted to the bond angles, a symmetric discrete gradient for the bond angles has recently been proposed in [53]. We discuss a slight generalisation of this discrete gradient. The standard angle potential, $V(\theta)$, is assumed to depend smoothly on the angle θ (cf. Fig. 3). The angle $\theta = \theta_{ijk}$ can be expressed in terms of the distances r_{ji} , r_{jk} and r_{ij} between the three atoms i , j and k as given on the right-hand side of Fig. 3. This is due to the following lemma.

Lemma 3 *We have the following representation of the scalar product*

$$\langle \mathbf{r}_{ji}, \mathbf{r}_{jk} \rangle = \frac{1}{2} \left(r_{ji}^2 + r_{jk}^2 - r_{ik}^2 \right).$$

Proof The proof is a simple calculation.

$$\begin{aligned} r_{ji}^2 + r_{jk}^2 - r_{ik}^2 &= \langle \mathbf{q}_i - \mathbf{q}_j, \mathbf{q}_i - \mathbf{q}_j \rangle + \langle \mathbf{q}_k - \mathbf{q}_j, \mathbf{q}_k - \mathbf{q}_j \rangle - \langle \mathbf{q}_k - \mathbf{q}_i, \mathbf{q}_k - \mathbf{q}_i \rangle \\ &= 2 \left(\langle \mathbf{q}_j, \mathbf{q}_j \rangle - \langle \mathbf{q}_i, \mathbf{q}_j \rangle - \langle \mathbf{q}_k, \mathbf{q}_j \rangle + \langle \mathbf{q}_i, \mathbf{q}_k \rangle \right) = 2 \cdot \langle \mathbf{q}_i - \mathbf{q}_j, \mathbf{q}_k - \mathbf{q}_j \rangle \end{aligned}$$

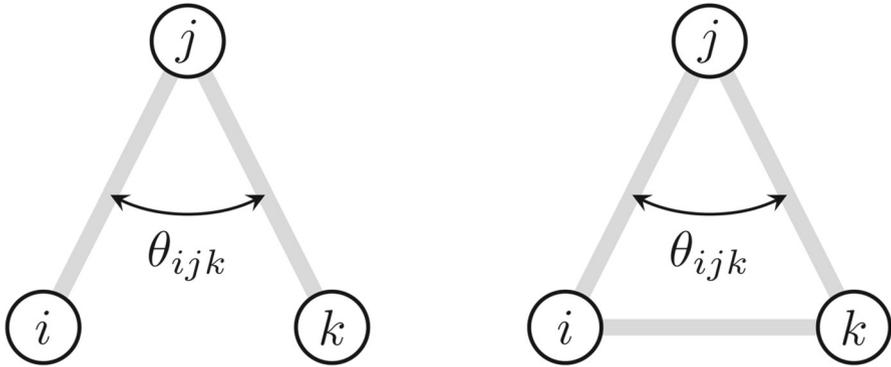


Fig. 3 Sketch of bond angle: From left to right, first, the bonds and the bond angle between the atoms i , j , and k are shown. The second sketch shows the distances between the atoms i , j , and k used in the discrete gradient

$$= 2 \cdot \langle \mathbf{r}_{ji}, \mathbf{r}_{jk} \rangle .$$

□

With Lemma 3, one readily obtains the representation of the angle in terms of distances

$$\theta_{ijk} = \arccos \left(\frac{\langle \mathbf{r}_{ji}, \mathbf{r}_{jk} \rangle}{\|\mathbf{r}_{ji}\| \cdot \|\mathbf{r}_{jk}\|} \right) = \arccos \left(\frac{r_{ji}^2 + r_{jk}^2 - r_{ik}^2}{2r_{ji} \cdot r_{jk}} \right) .$$

So the angle potential only depends on three distances $V(r_{ji}, r_{jk}, r_{ik})$. We first give a non-symmetric discrete gradient that is similar to the Itoh–Abe gradient. In order to write the discrete gradient down in a concise way, we use the following finite differences with respect to the distances of the particles

$$\begin{aligned} \Delta_{r_{ji}}^\ell V(\mathbf{q}, \mathbf{q}') &:= \frac{V(r'_{ji}, r_{jk}, r_{ik}) - V(r_{ji}, r_{jk}, r_{ik})}{r'_{ji} - r_{ji}} , \\ \Delta_{r_{jk}}^\ell V(\mathbf{q}, \mathbf{q}') &:= \frac{V(r'_{ji}, r'_{jk}, r_{ik}) - V(r'_{ji}, r_{jk}, r_{ik})}{r'_{jk} - r_{jk}} , \\ \Delta_{r_{ik}}^\ell V(\mathbf{q}, \mathbf{q}') &:= \frac{V(r'_{ji}, r'_{jk}, r'_{ik}) - V(r'_{ji}, r'_{jk}, r_{ik})}{r'_{ik} - r_{ik}} . \end{aligned}$$

The finite differences are labelled with ℓ , because the variables with primes are filled in from the left. This discrete gradient might be seen as the Itoh–Abe (coordinate increment) discrete gradient for $V(r_{ji}, r_{jk}, r_{ik})$. With these finite differences, our first discrete gradient $\bar{\nabla}^\ell V$ has the non-zero components

$$\bar{\nabla}_{\mathbf{q}_i}^\ell V(\mathbf{q}, \mathbf{q}') = \Delta_{r_{ji}}^\ell V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{ji} + \mathbf{r}_{ji}}{r'_{ji} + r_{ji}} - \Delta_{r_{ik}}^\ell V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{ik} + \mathbf{r}_{ik}}{r'_{ik} + r_{ik}} ,$$

$$\begin{aligned} \bar{\nabla}_{\mathbf{q}_i}^\ell V(\mathbf{q}, \mathbf{q}') &= -\Delta_{r_{ji}}^\ell V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{ji} + \mathbf{r}_{ji}}{r'_{ji} + r_{ji}} - \Delta_{r_{jk}}^\ell V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{jk} + \mathbf{r}_{jk}}{r'_{jk} + r_{jk}}, \\ \bar{\nabla}_{\mathbf{q}_k}^\ell V(\mathbf{q}, \mathbf{q}') &= \Delta_{r_{jk}}^\ell V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{jk} + \mathbf{r}_{jk}}{r'_{jk} + r_{jk}} + \Delta_{r_{ik}}^\ell V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{ik} + \mathbf{r}_{ik}}{r'_{ik} + r_{ik}}, \end{aligned} \tag{22}$$

where all others are zero. This discrete gradient is not symmetric, since the finite differences are not symmetric, e.g.

$$\Delta_{r_{ji}}^\ell V(\mathbf{q}, \mathbf{q}') \neq \Delta_{r_{ji}}^\ell V(\mathbf{q}', \mathbf{q}).$$

Hence, we can define another discrete gradient $\bar{\nabla}^r V$ by using coordinate decrease. The finite differences are now

$$\begin{aligned} \Delta_{r_{ik}}^r V(\mathbf{q}, \mathbf{q}') &:= \Delta_{r_{ik}}^\ell V(\mathbf{q}', \mathbf{q}) = \frac{V(r_{ji}, r_{jk}, r'_{ik}) - V(r_{ji}, r_{jk}, r_{ik})}{r'_{ik} - r_{ik}}, \\ \Delta_{r_{jk}}^r V(\mathbf{q}, \mathbf{q}') &:= \Delta_{r_{jk}}^\ell V(\mathbf{q}', \mathbf{q}) = \frac{V(r_{ji}, r'_{jk}, r'_{ik}) - V(r_{ji}, r_{jk}, r'_{ik})}{r'_{jk} - r_{jk}}, \\ \Delta_{r_{ji}}^r V(\mathbf{q}, \mathbf{q}') &:= \Delta_{r_{ji}}^\ell V(\mathbf{q}', \mathbf{q}) = \frac{V(r'_{ji}, r'_{jk}, r'_{ik}) - V(r_{ji}, r'_{jk}, r'_{ik})}{r'_{ji} - r_{ji}}. \end{aligned}$$

With these finite differences, we obtain another discrete gradient that is also not symmetric. The finite differences are labelled r , because the variables with primes are filled in from the right. Since symmetric discrete gradients lead to symmetric time-integration methods, which are of second order, one might prefer symmetric discrete gradients. In the same way as the Itoh–Abe discrete gradient, we can symmetrise our gradients here. This leads to the discrete gradient with the (symmetric) finite differences

$$\Delta_v^s V(\mathbf{q}, \mathbf{q}') := \frac{1}{2} \left(\Delta_v^\ell V(\mathbf{q}, \mathbf{q}') + \Delta_v^r V(\mathbf{q}, \mathbf{q}') \right), \quad v \in \{r_{ji}, r_{jk}, r_{ik}\}.$$

And the non-zero coefficients of the symmetric discrete gradient $\bar{\nabla}^s V$ read

$$\begin{aligned} \bar{\nabla}_{\mathbf{q}_i}^s V(\mathbf{q}, \mathbf{q}') &= \Delta_{r_{ji}}^s V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{ji} + \mathbf{r}_{ji}}{r'_{ji} + r_{ji}} - \Delta_{r_{ik}}^s V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{ik} + \mathbf{r}_{ik}}{r'_{ik} + r_{ik}}, \\ \bar{\nabla}_{\mathbf{q}_j}^s V(\mathbf{q}, \mathbf{q}') &= -\Delta_{r_{ji}}^s V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{ji} + \mathbf{r}_{ji}}{r'_{ji} + r_{ji}} - \Delta_{r_{jk}}^s V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{jk} + \mathbf{r}_{jk}}{r'_{jk} + r_{jk}}, \\ \bar{\nabla}_{\mathbf{q}_k}^s V(\mathbf{q}, \mathbf{q}') &= \Delta_{r_{jk}}^s V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{jk} + \mathbf{r}_{jk}}{r'_{jk} + r_{jk}} + \Delta_{r_{ik}}^s V(\mathbf{q}, \mathbf{q}') \cdot \frac{\mathbf{r}'_{ik} + \mathbf{r}_{ik}}{r'_{ik} + r_{ik}}. \end{aligned} \tag{23}$$

These three expressions are indeed discrete gradients, which can be seen by checking Definition 1.

There are more discrete gradients. One might prescribe any pattern of primes to the three distances in the finite differences. For example, the pattern prime, no prime, prime and then changing from prime to no prime and vice versa from left to right

$$\begin{aligned}\Delta_{r_{ji}}^f V(\mathbf{q}, \mathbf{q}') &:= \frac{V(r'_{ji}, r_{jk}, r'_{ik}) - V(r_{ji}, r_{jk}, r'_{ik})}{r'_{ji} - r_{ji}}, \\ \Delta_{r_{jk}}^f V(\mathbf{q}, \mathbf{q}') &:= \frac{V(r'_{ji}, r'_{jk}, r'_{ik}) - V(r'_{ji}, r_{jk}, r'_{ik})}{r'_{jk} - r_{jk}}, \\ \Delta_{r_{ik}}^f V(\mathbf{q}, \mathbf{q}') &:= \frac{V(r'_{ji}, r_{jk}, r'_{ik}) - V(r'_{ji}, r_{jk}, r_{ik})}{r'_{ik} - r_{ik}}.\end{aligned}$$

All discrete gradients of this type can be symmetrised by symmetrising the finite differences.

4.4 Discrete gradients for dihedral angles

The dihedral angle

$$\phi_{ijkl} = \text{sign}(\langle \mathbf{r}_{ij}, \mathbf{r}_{jk} \times \mathbf{r}_{kl} \rangle) \arccos \varphi_{ijkl}, \quad \varphi_{ijkl} = \frac{\langle \mathbf{r}_{ij} \times \mathbf{r}_{jk}, \mathbf{r}_{jk} \times \mathbf{r}_{kl} \rangle}{\|\mathbf{r}_{ij} \times \mathbf{r}_{jk}\| \|\mathbf{r}_{jk} \times \mathbf{r}_{kl}\|}, \quad (24)$$

denotes the angle between the planes spanned by the atoms i, j and k and j, k and ℓ , respectively. The sign of the dihedral angle ϕ_{ijkl} , i.e. $\text{sign}(\langle \mathbf{r}_{ij}, \mathbf{r}_{jk} \times \mathbf{r}_{kl} \rangle)$, designates on which side of the plane through j, k and ℓ the particle i lies. We consider potentials $V(\phi)$ that depend smoothly on the dihedral angle ϕ . The computation of the forces based on the dihedral angle is based on the expression as given above in (24) and leads to the most used formulas as described in [3]. To our best knowledge, the discrete gradient that we propose in the following is new. It is based on the same idea as the discrete gradients before, namely, the idea to express the angle with distances. The dihedral angle can also be expressed in terms of the distances between all atoms as indicated on the right-hand side of Fig. 4. The additional lines on the right-hand side of Fig. 4 are the additional distances that will be used. The representation of the angle in distances is given in the following lemma.



Fig. 4 Sketch of united-atom butane: On the left-hand side, the molecule with the particles i, j, k and ℓ is shown including the bonds between atoms. On the right-hand side, the distances between the particles i, j, k and ℓ used by the discrete gradient are given

Lemma 4 We have the following representation in terms of distances

$$\varphi_{ijkl} = \frac{\langle \mathbf{m}, \mathbf{n} \rangle}{\|\mathbf{m}\| \|\mathbf{n}\|} = \frac{(r_{kl}^2 + r_{jk}^2 - r_{jl}^2)(r_{ij}^2 + r_{jk}^2 - r_{ik}^2) - 2r_{jk}^2(r_{jk}^2 + r_{il}^2 - r_{jl}^2 - r_{ik}^2)}{\sqrt{4r_{jk}^2 r_{ij}^2 - (r_{ij}^2 + r_{jk}^2 - r_{ik}^2)^2} \sqrt{4r_{jk}^2 r_{kl}^2 - (r_{jk}^2 + r_{kl}^2 - r_{jl}^2)^2}}$$

where $\mathbf{m} = \mathbf{r}_{ij} \times \mathbf{r}_{jk}$ and $\mathbf{n} = \mathbf{r}_{jk} \times \mathbf{r}_{kl}$.

Proof The proof is a tedious calculation along the following steps:

$$\begin{aligned} \frac{\langle \mathbf{r}_{ij} \times \mathbf{r}_{jk}, \mathbf{r}_{jk} \times \mathbf{r}_{kl} \rangle}{\|\mathbf{r}_{ij} \times \mathbf{r}_{jk}\| \|\mathbf{r}_{jk} \times \mathbf{r}_{kl}\|} &= (-1) \cdot \frac{\langle r_{jk}^2 \mathbf{r}_{ij} - \langle \mathbf{r}_{ij}, \mathbf{r}_{jk} \rangle \mathbf{r}_{jk}, r_{jk}^2 \mathbf{r}_{kl} - \langle \mathbf{r}_{kl}, \mathbf{r}_{jk} \rangle \mathbf{r}_{jk} \rangle}{\|r_{jk}^2 \mathbf{r}_{ij} - \langle \mathbf{r}_{ij}, \mathbf{r}_{jk} \rangle \mathbf{r}_{jk}\| \|r_{jk}^2 \mathbf{r}_{kl} - \langle \mathbf{r}_{kl}, \mathbf{r}_{jk} \rangle \mathbf{r}_{jk}\|} \\ &= \frac{\langle \mathbf{r}_{kl}, \mathbf{r}_{jk} \rangle \langle \mathbf{r}_{ij}, \mathbf{r}_{jk} \rangle - r_{jk}^2 \langle \mathbf{r}_{ij}, \mathbf{r}_{kl} \rangle}{\sqrt{r_{jk}^2 r_{ij}^2 - \langle \mathbf{r}_{ij}, \mathbf{r}_{jk} \rangle^2} \sqrt{r_{jk}^2 r_{kl}^2 - \langle \mathbf{r}_{jk}, \mathbf{r}_{kl} \rangle^2}} \\ &= \frac{(r_{kl}^2 + r_{jk}^2 - r_{jl}^2)(r_{ij}^2 + r_{jk}^2 - r_{ik}^2) - 2r_{jk}^2(r_{jk}^2 + r_{il}^2 - r_{jl}^2 - r_{ik}^2)}{\sqrt{4r_{jk}^2 r_{ij}^2 - (r_{ij}^2 + r_{jk}^2 - r_{ik}^2)^2} \sqrt{4r_{jk}^2 r_{kl}^2 - (r_{jk}^2 + r_{kl}^2 - r_{jl}^2)^2}} \end{aligned}$$

□

So the dihedral potential depends on six distances $V(r_{ij}, r_{jk}, r_{kl}, r_{ik}, r_{j\ell}, r_{i\ell})$. The representation in Lemma 4 suggests an alternative way to represent any potential dependent on the dihedral angle in terms of distances. But we will consider potentials that are dependent on the cosine of the dihedral angle, because we will use such potentials in our experiments later. Let

$$U_t(\phi) = k_\phi \sum_{n=0}^m a_n \cos^n \phi = \tilde{U}_t(\cos \phi), \quad \tilde{U}_t(\varphi) = k_\phi \sum_{n=0}^m a_n \varphi^n,$$

be the potential for the torsion angles. Then, we have

$$\begin{aligned} \frac{\partial}{\partial \mathbf{q}_u} U_t(\phi_{ijkl}) &= \tilde{U}'_t(\cos \phi_{ijkl}) \cdot (-1) \cdot \sin(\phi_{ijkl}) \text{sign}(\langle \mathbf{r}_{ij}, \mathbf{n} \rangle) \frac{-1}{\sin \phi_{ijkl}} \cdot \frac{\partial}{\partial \mathbf{q}_u} \phi_{ijkl} \\ &= \tilde{U}'_t(\varphi_{ijkl}) \cdot \frac{\partial}{\partial \mathbf{q}_u} \varphi_{ijkl}, \quad u = i, j, k, \ell. \end{aligned}$$

The potential reduces to $\tilde{U}_t(\varphi_{ijkl})$. Hence, for the discrete gradient, we use the representation

$$V(r_{ij}, r_{jk}, r_{kl}, r_{ik}, r_{j\ell}, r_{i\ell}) = \tilde{U}_t(\varphi_{ijkl}),$$

with φ_{ijkl} expressed in the distances as in Lemma 4.

We first discuss non-symmetric discrete gradients. In order to write the discrete gradient down in a concise way, we again use finite differences with respect to the distances of the particles.

$$\Delta_{r_{ij}}^\ell V(\mathbf{q}, \mathbf{q}') := \frac{V(r'_{ij}, r_{jk}, r_{kl}, r_{ik}, r_{j\ell}, r_{i\ell}) - V(r_{ij}, r_{jk}, r_{kl}, r_{ik}, r_{j\ell}, r_{i\ell})}{r'_{ij} - r_{ij}}$$

$$\begin{aligned} & \vdots \\ \Delta_{r_{i\ell}}^\ell V(\mathbf{q}, \mathbf{q}') & := \frac{V(r'_{ij}, r'_{jk}, r'_{kl}, r'_{ik}, r'_{j\ell}, r'_{i\ell}) - V(r'_{ij}, r'_{jk}, r'_{kl}, r'_{ik}, r'_{j\ell}, r_{i\ell})}{r'_{i\ell} - r_{i\ell}} \end{aligned}$$

This can also be seen as the Itoh–Abe (coordinate increment) discrete gradient for V . With these finite differences, our first discrete gradient reads as follows:

$$\begin{aligned} \bar{\nabla}_{\mathbf{q}_i}^\ell V(\mathbf{q}, \mathbf{q}') &= -\Delta_{r_{ij}}^\ell V \cdot \frac{\mathbf{r}'_{ij} + \mathbf{r}_{ij}}{r'_{ij} + r_{ij}} - \Delta_{r_{ik}}^\ell V \cdot \frac{\mathbf{r}'_{ik} + \mathbf{r}_{ik}}{r'_{ik} + r_{ik}} - \Delta_{r_{i\ell}}^\ell V \cdot \frac{\mathbf{r}'_{i\ell} + \mathbf{r}_{i\ell}}{r'_{i\ell} + r_{i\ell}}, \\ \bar{\nabla}_{\mathbf{q}_j}^\ell V(\mathbf{q}, \mathbf{q}') &= \Delta_{r_{ij}}^\ell V \cdot \frac{\mathbf{r}'_{ij} + \mathbf{r}_{ij}}{r'_{ij} + r_{ij}} - \Delta_{r_{jk}}^\ell V \cdot \frac{\mathbf{r}'_{jk} + \mathbf{r}_{jk}}{r'_{jk} + r_{jk}} - \Delta_{r_{j\ell}}^\ell V \cdot \frac{\mathbf{r}'_{j\ell} + \mathbf{r}_{j\ell}}{r'_{j\ell} + r_{j\ell}}, \\ \bar{\nabla}_{\mathbf{q}_k}^\ell V(\mathbf{q}, \mathbf{q}') &= \Delta_{r_{ik}}^\ell V \cdot \frac{\mathbf{r}'_{ik} + \mathbf{r}_{ik}}{r'_{ik} + r_{ik}} + \Delta_{r_{jk}}^\ell V \cdot \frac{\mathbf{r}'_{jk} + \mathbf{r}_{jk}}{r'_{jk} + r_{jk}} - \Delta_{r_{kl}}^\ell V \cdot \frac{\mathbf{r}'_{kl} + \mathbf{r}_{kl}}{r'_{kl} + r_{kl}}, \\ \bar{\nabla}_{\mathbf{q}_\ell}^\ell V(\mathbf{q}, \mathbf{q}') &= \Delta_{r_{i\ell}}^\ell V \cdot \frac{\mathbf{r}'_{i\ell} + \mathbf{r}_{i\ell}}{r'_{i\ell} + r_{i\ell}} + \Delta_{r_{j\ell}}^\ell V \cdot \frac{\mathbf{r}'_{j\ell} + \mathbf{r}_{j\ell}}{r'_{j\ell} + r_{j\ell}} + \Delta_{r_{k\ell}}^\ell V \cdot \frac{\mathbf{r}'_{k\ell} + \mathbf{r}_{k\ell}}{r'_{k\ell} + r_{k\ell}}. \end{aligned} \tag{25}$$

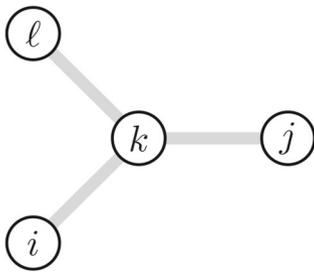
This discrete gradient is not symmetric. The coordinate decrement version

$$\begin{aligned} \Delta_{r_{ij}}^r V(\mathbf{q}, \mathbf{q}') &:= \frac{V(r_{ij}, r'_{jk}, r'_{kl}, r'_{ik}, r'_{j\ell}, r'_{i\ell}) - V(r'_{ij}, r'_{jk}, r'_{kl}, r'_{ik}, r'_{j\ell}, r'_{i\ell})}{r_{ij} - r'_{ij}} \\ & \vdots \\ \Delta_{r_{i\ell}}^r V(\mathbf{q}, \mathbf{q}') &:= \frac{V(r_{ij}, r_{jk}, r_{kl}, r_{ik}, r_{j\ell}, r_{i\ell}) - V(r_{ij}, r_{jk}, r_{kl}, r_{ik}, r_{j\ell}, r'_{i\ell})}{r_{i\ell} - r'_{i\ell}} \end{aligned}$$

leads to another discrete gradient that is also not symmetric. Similar to what we saw in the previous case, one might prefer symmetric discrete gradients. The symmetrised discrete gradient makes use of the symmetric finite differences

$$\Delta_r^s V(\mathbf{q}, \mathbf{q}') := \frac{1}{2} \left(\Delta_r^\ell V(\mathbf{q}, \mathbf{q}') + \Delta_r^r V(\mathbf{q}, \mathbf{q}') \right), \quad r \in \{r_{ij}, r_{jk}, r_{kl}, r_{ik}, r_{j\ell}, r_{i\ell}\}$$

These three expressions are indeed discrete gradients, which can be seen by checking Definition 1. Actually, there are many more discrete gradients. One might prescribe any pattern of primes to the six distances. Changing the primed distances to not-primed distances and vice versa from left to right when the finite difference ‘passed’ this distance, we obtain a new unsymmetric discrete gradient. These unsymmetric discrete gradients can all be symmetrised. Besides the conventional torsion potential, where the atoms i, j, k and ℓ are consecutively connected, there is also the possibility that three atoms are connected to one as in Fig. 5 on the left-hand side. Here, the so-called improper dihedral angle is defined as the angle between the planes spanned by the atoms i, j and k and j, k and ℓ , respectively. As shown on the right-hand side of Fig. 5, this angle can be expressed in exactly the same way as before for the standard dihedral angle (24). Therefore, the distance definition of the dihedral angle in



$$\omega_{ijkl} = \arccos(\varphi_{ijkl}),$$

$$\varphi_{ijkl} = \frac{\langle \mathbf{r}_{ij} \times \mathbf{r}_{jk}, \mathbf{r}_{jk} \times \mathbf{r}_{kl} \rangle}{\|\mathbf{r}_{ij} \times \mathbf{r}_{jk}\| \|\mathbf{r}_{jk} \times \mathbf{r}_{kl}\|}$$

Fig. 5 Improper dihedral angle: On the left-hand side, the typical situation is shown. On the right-hand side, the formula for the improper dihedral angle ω_{ijkl} is given. φ_{ijkl} is exactly the same term as for the standard dihedral angle (cf. (24))

Lemma 4 also works for the improper dihedral angle, and the corresponding discrete gradients are constructed analogously.

$$\omega_{ijkl} = \arccos(\varphi_{ijkl}),$$

$$\varphi_{ijkl} = \frac{\langle \mathbf{r}_{ij} \times \mathbf{r}_{jk}, \mathbf{r}_{jk} \times \mathbf{r}_{kl} \rangle}{\|\mathbf{r}_{ij} \times \mathbf{r}_{jk}\| \|\mathbf{r}_{jk} \times \mathbf{r}_{kl}\|}$$

With discrete gradients for all standard short-range forces at hand, we are able to formulate our main theorem for the considered discrete gradients.

Theorem 2 *The method (13) for a particle system of N particles with Hamiltonian*

$$H(\mathbf{q}, \mathbf{p}) = \frac{1}{2} \mathbf{p}^T M^{-1} \mathbf{p} + V(\mathbf{q}), \quad V = \sum_{k=1}^M V^k,$$

where V^k are pairwise, angle and dihedral potentials, is a first-order non-symmetric implicit method if at least one non-symmetric discrete gradient is used, and a second-order symmetric implicit method if all discrete gradients are symmetric. All methods preserve the energy, the total linear momentum $\mathbf{P} = \sum_{i=1}^N \mathbf{p}_i$ and the total angular momentum $\mathbf{L} = \sum_{i=1}^N \mathbf{q}_i \times \mathbf{p}_i$.

Proof The order of the method follows from Proposition 2. The preservation of the energy directly follows from the use of discrete gradients, Lemma 1 and Lemma 2. From (20), (23) (and its variants) or (25) (and its variants), respectively, we find by a tedious but simple calculation that for arbitrary \mathbf{q}' and \mathbf{q} , we have that

$$\sum_{i=1}^N \bar{\nabla}_{\mathbf{q}_i} V_{\mathbf{q}_i}^k(\mathbf{q}, \mathbf{q}') = 0, \quad \text{for } k = 1, \dots, M.$$

Hence,

$$\begin{aligned} \mathbf{P}^{n+1} &= \sum_{i=1}^N \mathbf{p}_i^{n+1} = \sum_{i=1}^N \left(\mathbf{p}_i + \tau \sum_{k=1}^M \bar{\nabla} V_{\mathbf{q}_i}^k(\mathbf{q}^n, \mathbf{q}^{n+1}) \right) \\ &= \sum_{i=1}^N \mathbf{p}_i^n + \tau \sum_{k=1}^M \sum_{i=1}^N \bar{\nabla} V_{\mathbf{q}_i}^k(\mathbf{q}^n, \mathbf{q}^{n+1}) = \sum_{i=1}^N \mathbf{p}_i^n = \mathbf{P}^n. \end{aligned} \tag{26}$$

For all potentials and all proposed discrete gradients and for arbitrary \mathbf{q}' and \mathbf{q} , we have that

$$\sum_{i=1}^N (\mathbf{q}'_i + \mathbf{q}_i) \times \bar{\nabla}_{\mathbf{q}_i} V^k(\mathbf{q}, \mathbf{q}') = \mathbf{0}.$$

This is again a tedious but simple calculation. From this, \mathbf{q}' and \mathbf{q} arbitrarily chosen, we find

$$\begin{aligned} \sum_{i=1}^N (\mathbf{q}'_i + \mathbf{q}_i) \times \bar{\nabla}_{\mathbf{q}_i} V(\mathbf{q}, \mathbf{q}') &= \sum_{i=1}^N (\mathbf{q}'_i + \mathbf{q}_i) \times \left(\sum_{k=1}^M \bar{\nabla}_{\mathbf{q}_i} V^k(\mathbf{q}, \mathbf{q}') \right) \\ &= \sum_{k=1}^M \left(\sum_{i=1}^N (\mathbf{q}'_i + \mathbf{q}_i) \times \bar{\nabla}_{\mathbf{q}_i} V^k(\mathbf{q}, \mathbf{q}') \right) = \mathbf{0}. \end{aligned}$$

Now, we turn to method (13). Then, we have by the equation above that

$$\sum_{i=1}^N (\mathbf{q}_i^{n+1} + \mathbf{q}_i^n) \times (\mathbf{p}_i^{n+1} - \mathbf{p}_i^n) = -\tau \sum_{i=1}^N (\mathbf{q}_i^{n+1} + \mathbf{q}_i^n) \times \bar{\nabla}_{\mathbf{q}_i} V(\mathbf{q}^n, \mathbf{q}^{n+1}) = \mathbf{0}. \tag{27}$$

Due to

$$(\mathbf{q}_i^{n+1} - \mathbf{q}_i^n) \times (\mathbf{p}_i^{n+1} + \mathbf{p}_i^n) = \frac{\tau}{2m_i} (\mathbf{p}_i^{n+1} + \mathbf{p}_i^n) \times (\mathbf{p}_i^{n+1} + \mathbf{p}_i^n) = \mathbf{0},$$

we also have

$$\sum_{i=1}^N (\mathbf{q}_i^{n+1} - \mathbf{q}_i^n) \times (\mathbf{p}_i^{n+1} + \mathbf{p}_i^n) = \mathbf{0}. \tag{28}$$

Adding (27) and (28) shows the preservation of the angular momentum. □

Theorem 2 shows that all standard short-range forces in a classical molecular dynamics simulation can be modelled with discrete gradients in such a way that the statement of Theorem 1 by LaBudde and Greenspan for pairwise forces is carried over to all standard short-range forces. Since even large protein simulations are often based on these standard short-range potentials and since the standard potentials do not change whether more atoms are connected to atoms defining one of the standard interactions or not, the new Theorem 2 is quite general with respect to its applicability.

Additional insight in the computation of the gradient of a dihedral angle potential might be gained from the representation of the dihedral angle with six distances. The computation of this gradient is not straightforward. It is usually based on formula (24), also called the cross-product definition of the dihedral angle. The scalar product of the cross-products can be represented as a scalar product without cross-products, the so-called scalar product definition for the torsion angle. The cross-product definition leads to the most used formulas for the computation of the forces induced by the dihedral potential as given in [3]. The representation of the dihedral angle in terms of distances leads to the interesting alternative formulas

$$\begin{aligned} \nabla_{\mathbf{q}_i} V(\mathbf{q}) &= -V_{r_{ij}} \frac{\mathbf{r}_{ij}}{r_{ij}} - V_{r_{ik}} \frac{\mathbf{r}_{ik}}{r_{ik}} - V_{r_{il}} \frac{\mathbf{r}_{il}}{r_{il}}, & \nabla_{\mathbf{q}_j} V(\mathbf{q}) &= V_{r_{ij}} \frac{\mathbf{r}_{ij}}{r_{ij}} - V_{r_{jk}} \frac{\mathbf{r}_{jk}}{r_{jk}} - V_{r_{jl}} \frac{\mathbf{r}_{jl}}{r_{jl}}, \\ \nabla_{\mathbf{q}_k} V(\mathbf{q}) &= V_{r_{ik}} \frac{\mathbf{r}_{ik}}{r_{ik}} + V_{r_{jk}} \frac{\mathbf{r}_{jk}}{r_{jk}} - V_{r_{kl}} \frac{\mathbf{r}_{kl}}{r_{kl}}, & \nabla_{\mathbf{q}_l} V(\mathbf{q}) &= V_{r_{il}} \frac{\mathbf{r}_{il}}{r_{il}} + V_{r_{jl}} \frac{\mathbf{r}_{jl}}{r_{jl}} + V_{r_{kl}} \frac{\mathbf{r}_{kl}}{r_{kl}}, \end{aligned}$$

where

$$V_{r_{mn}} = V_{r_{mn}}(r_{ij}, r_{jk}, r_{kl}, r_{ik}, r_{jl}, r_{il}), \quad r_{m,n} \in \{r_{ij}, r_{jk}, r_{kl}, r_{ik}, r_{jl}, r_{il}\},$$

designate the derivative with respect to the corresponding distance. The representation of the dihedral angle in terms of the distances in Lemma 4 might be called the ‘distance’ definition of the dihedral angle, as a supplement to the cross-product definition and the scalar product definition of the dihedral angle.

Since we have ‘distance’ definitions of the bond angles as well as the dihedral angles, and since the other potentials are dependent on distances in a natural way, one could evaluate all forces with respect to these potentials in a unified way. We use these representations here in order to use the same technique to construct a discrete gradient. But this unified way to compute the forces, i.e. the gradients of the potentials, might be interesting for the evaluation of the short-range forces in the standard Verlet scheme, too.

4.5 Experiment with butane

We use a united-atom model of butane. The Lennard–Jones potential is chosen as before with the parameters given in Table 1. It is only applied for every two atoms that belong to different molecules. The bond potential V_b and the angle potential V_a are chosen as

$$V_b(r) = k_b(r - r_0)^2 \quad \text{and} \quad V_a(\theta) = k_\theta(\cos(\theta) - \cos(\theta_0))^2. \quad (29)$$

Table 1 Parameters for the simulation of butane

$k_b = 8.75 \frac{\text{MJ}}{\text{mol}\cdot\text{nm}^2}$	$r_0 = 1.53 \text{ \AA}$	Bond potential
$k_\theta = 65 \frac{\text{kJ}}{\text{mol}}$	$\theta_0 = 109.47 \text{ degree}$	Angle potential
$k_\phi = 8.31451 \frac{\text{kJ}}{\text{mol}}$		Torsion potential
$\sigma = 3.923 \text{ \AA}$	$\epsilon = 0.5986 \frac{\text{kJ}}{\text{mol}}$	Lennard–Jones potential

The bond potential and the angle potential are available in LAMMPS (cf. [58]) as `bond_style harmonic` and `angle_style cosine/squared`, respectively. And the potential for the torsion potential in terms of the dihedral angle ϕ in the IUPAC convention (cf. [25]) reads

$$U_t(\phi) = k_\phi(1.116 - 1.462 \cos \phi - 1.578 \cos^2 \phi + 0.368 \cos^3 \phi + 3.156 \cos^4 \phi + 3.788 \cos^5 \phi). \quad (30)$$

The parameter k_ϕ is also given in Table 1 (cf. [59]). The dihedral potential is also available in LAMMPS as `dihedral_style nharmonic`. Since we are using realistic data, we first remove the units by scaling. We use $\tilde{\sigma} = 10^{-9} \text{ m} = 1 \text{ nm}$, $\tilde{\epsilon} = 1 \text{ kJ/mol}$, $\tilde{m} = 1 \text{ u}$ and $\tilde{\alpha} = \tilde{\sigma} \sqrt{\tilde{m}/\tilde{\epsilon}} \approx 10^{-12} \text{ s} = 1 \text{ ps}$. The dimensionless quantities then read

$$\begin{aligned} m' &= m/\tilde{m}, & \mathbf{x}'_i &= \mathbf{x}_i/\tilde{\sigma}, & \mathbf{r}'_{ij} &= \mathbf{r}_{ij}/\tilde{\sigma}, & E' &= E/\tilde{\epsilon}, & V' &= V/\tilde{\epsilon}, \\ \sigma' &= \sigma/\tilde{\sigma}, & \epsilon' &= \epsilon/\tilde{\epsilon}, & t' &= t/\tilde{\alpha}. \end{aligned}$$

The initial configuration for the atoms in the scaled quantities is shown in Fig. 6 on the right-hand side. The molecules are plotted with OVITO (cf. [57]). The first column in Code fragment 2 on the left-hand side refers to the atom number, the second column to the molecule the atom belongs to. Two consecutive atoms within the same molecule are connected by a bond. Three consecutive atoms within the same molecule are ruled by the angle potential and all four atoms of the molecule by the given torsion potential. That is, this experiment uses all potentials discussed so far.

In Fig. 7 on the left-hand side, one can see that the methods perform as expected with respect to the order. The integration time for the error plot was up to $T = 2.0$ with the step sizes indicated on the abscissa. The error measured in the standard Euclidean norm is shown on the ordinate. If only one of the discrete gradients used in the DG method is unsymmetric then the DG method is of first order. This is shown by the blue circle-marked line. If all discrete gradients are symmetric, the method is

Code fragment 2 Data butane

Positions

1	1	0.949003	1.144251	1.100000
2	1	1.000000	1.000000	1.000000
3	1	1.153000	1.000000	1.000000
4	1	1.203997	1.144251	0.900000
5	2	0.375077	0.588336	1.000000
6	2	0.500000	0.500000	1.000000
7	2	0.624923	0.588336	1.000000
8	2	0.749846	0.500000	1.000000



Fig. 6 Initial positions for the experiment with two united-atom butane molecules: the positions are given on the left-hand side. The first column numbers the particles. The second column assigns molecule numbers. If this number is the same, a bond is added between the particles. The third to fifth columns are the coordinates of the positions. The velocities are set to zero and not given in the code fragment. On the right-hand side, the initial configuration is given as a plot made by the open visualization tool (OVITO) (cf. [57])

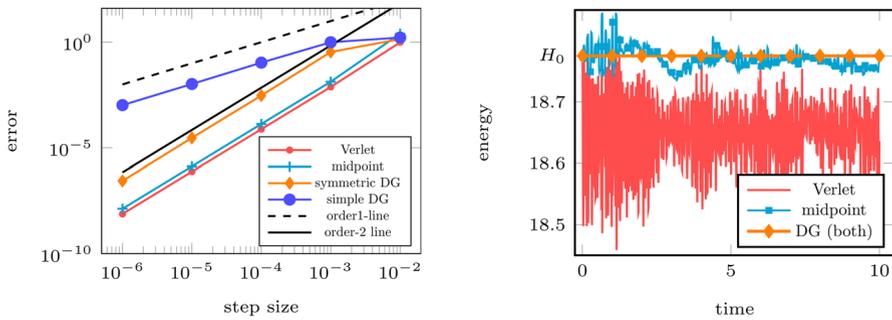


Fig. 7 Results of the experiment with two united-atom butane molecules: the error versus the time step is shown on the left-hand side for the Verlet scheme, the symmetric discrete gradient (DG) scheme, the midpoint rule and the unsymmetric simple discrete gradient (DG) method. On the right-hand side, the energy is shown over the time span $[0, 10]$ for step size $\tau = 0.005$ for the Verlet scheme, the midpoint rule and the two discrete gradient (DG) schemes. H_0 designates the exact energy

of second order as well as the implicit midpoint rule and the Verlet scheme. For the energy plot on the right-hand side, we computed the solutions up to time $T = 10$ with step size $\tau = 0.005$. All DG methods preserve the energy up to round-off error. The implicit midpoint rule and the Verlet scheme deviate from the constant energy that should be preserved. Since the Verlet scheme deviates significantly more than the implicit midpoint rule, we also calculated the energy with LAMMPS. The energy behaviour turned out to be exactly the same. Setting the NVE ensemble in LAMMPS, the red, solid energy curve is the outcome, which coincides with our own implementation of the Verlet scheme. The implicit equation in (14) is again solved with the quasi-Newton method, (17), where the approximate Jacobian J_F , cf. (18), includes the full Hessian with respect to the Lennard–Jones potentials and the bond potentials but omits the part with respect to the bond angle and dihedral angle potentials.

5 Parallelisation of DG methods

In order to show the usefulness of DG methods in molecular dynamics, it is indispensable to take care of parallelisation. Many codes for molecular dynamics simulations are based on the basic parallel treatment of short-range forces (e.g. [4, 33, 45, 58]).

5.1 Parallelisation of the evaluation of discrete gradients for Lennard–Jones forces with cut-off

In this section, we discuss the parallelisation of Lennard–Jones forces, induced by the standard potential (cf. (21)). Since we are only interested in the short-range part of the potential and since we will also make use of the Hessian, the Lennard–Jones potential

with cut-off function should be twice continuously differentiable. For this reason, we will use the switching function proposed in [38] as

$$s(r) = \begin{cases} 1, & 0 \leq r < r_m, \\ (1-x)^3(1+3x+6x^2), & r_m \leq r \leq r_{\text{cut}}, \\ 0, & r_{\text{cut}} < r, \end{cases} \quad x = \frac{r-r_m}{r_{\text{cut}}-r_m},$$

with $r_m = \frac{r_{\text{cut}}}{2}$. The function and the first two derivatives restricted to $[r_m, r_{\text{cut}}]$ read

$$\begin{aligned} s(r) &= (1-x)^3(1+3x+6x^2) \\ s'(r) &= \frac{1}{r_{\text{cut}}} - r_m \cdot (-30) \cdot x^2(x-1)^2 \\ s''(r) &= \frac{1}{(r_{\text{cut}} - r_m)^2} \cdot (-60) \cdot x(x-1)(2x-1) \end{aligned}, \quad x = \frac{r-r_m}{r_{\text{cut}}-r_m}.$$

The Lennard–Jones potential with switching function, i.e. $V(r) = U(r) \cdot s(r)$, is a twice continuously differentiable short-range potential. This switching function is available in LAMMPS as `pair_style lj/mdf`.

The Lennard–Jones interactions are implemented with the linked cell method as described, e.g. in Chapter 3 of [15]. Since DG methods are implicit methods, the particle structure is extended by the possible future positions of the particles during the iterative solution of (14). Due to this, Lennard–Jones forces pose a special challenge with DG methods or implicit methods, in general. While a border neighbourhood of one cell is enough for the Verlet scheme, we need a border neighbourhood of two cells for DG methods. The reason is that the particle structure not only carries the position at the given time, but also the position of the next step. Two particles could become close in the next step that are not close in the given time step. This is illustrated in Fig. 8. Since possible future positions are needed for the evaluation of the discrete gradient (20) in the difference (19), particles with a distance of $2r_{\text{cut}}$ need to be known, when we assume that the step size is chosen so small that the particles cannot travel for more than $\frac{2}{3}r_{\text{cut}}$ in a time step. To sweep all particles in a border neighbourhood of two cells of a given cell is enough to catch these events. As an alternative, one might use cells with dimensions larger than $2r_{\text{cut}}$. Then, a border neighbourhood with a width of one cell would be enough for the larger cells. We will stick with the smaller cells. There are some computations, for example the potential with the cut-off function, where we only need a border neighbourhood of one cell. Hence, using the smaller cells saves a bit of computing time.

For the parallelisation, domain decomposition is used. If we decompose the two-dimensional domain in Fig. 9 in six larger parts based on the cells given by the linked cell method, every processor only knows the particles in its domain. Since each processor might run on its own node with its own memory in a cluster computer, the access to the data of adjacent processors is not immediate (cf. [2]). The standard technique is to extend the domain that the processor is handling by further cells, the border neighbourhood, and to retrieve the necessary information from the adjacent processors in these cells. The current processor also has to send the information needed by the adjacent processors from his domain to the neighbours. For the exchange of the data between processors, message passing is used. That is, the processors send messages

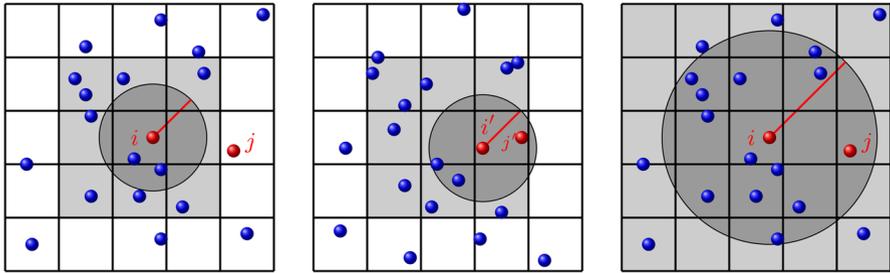


Fig. 8 Linked cell method: Simulation domain is decomposed into square cells of size $r_{\text{cut}} \times r_{\text{cut}}$. The dark-shaded circle is the cut-off radius r_{cut} about the red particle i . On the left-hand side, the current time step is shown. For the standard force computation, only the particles in the 3×3 grid of light-grey cells need to be taken into account. In the middle, the situation at the next time step is shown. The future positions i' of particle i and j' of particle j , respectively, are now within the cut-off range. While particle i and j of the current time step do not interact, they interact after they moved to the positions i' and j' in the next time step. In the discrete gradient method, the positions in the next time step are needed to compute the discrete gradient. Therefore, a 5×5 grid of cells around the cell with the red particle i needs to be taken into account for DG methods, if the particles are assumed to travel not more than $2/3$ of the cut-off in one time step. The cut-off radius of the DG method around the red particle i and the cells that need to be taken into account are shown on the right-hand side

to each other. This is standardised in the message passing interface (MPI) (cf. [39]). Due to the discussion above, we need to extend the processor's domain by a border neighbourhood of the size of two cells as shown in Fig. 10 on the left-hand side. We will conduct three-dimensional experiments. The corresponding domain and its neighbourhood are shown in Fig. 10 on the right-hand side.

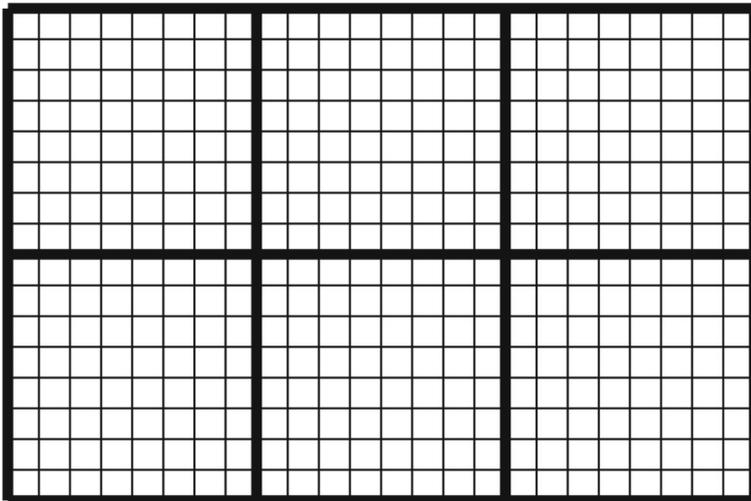


Fig. 9 Decomposition of the simulation domain: the domain Ω is divided in cells indicated by thin lines. The domain is subdivided further into six subdomains as indicated by the thick lines. Each of the six subdomains is assigned to its own processor that handles the computations in this subdomain according to the linked cell method

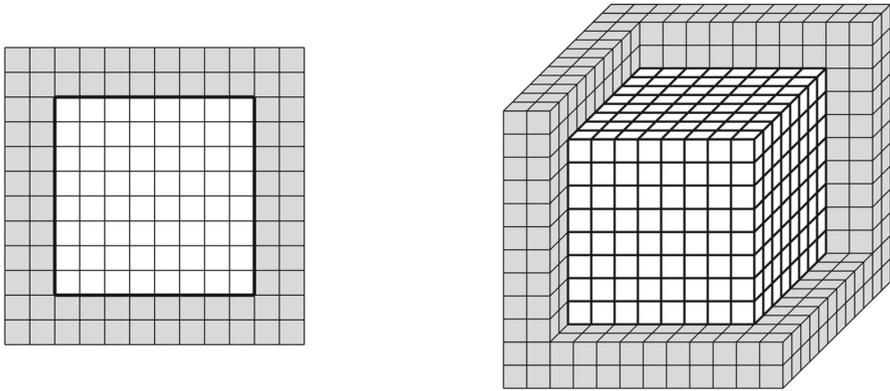


Fig. 10 Subdomain and border neighbourhood: On the left-hand side, a typical subdomain is shown as the square with the white cells. The processor responsible for this domain needs the data from the adjacent domains. A border neighbourhood of cells, given in grey, is added to the subdomain. The border neighbourhood is filled with copies of the particles in the adjacent domains. The adjacent processors have to send the data to the current processor by messages, according to the message passing paradigm standardised in the message passing interface (MPI) (cf. [39]). For implicit methods and DG methods, a border neighbourhood of two cells is necessary in order to compute the forces by the linked cell method. On the right-hand side, the situation for a three-dimensional subdomain is illustrated

5.2 Collision of two bodies

As a test problem for the parallelisation, we use the collision of two bodies as described in Section 4.5.1 of [15]. In Fig. 11 on the right-hand side, the initial configuration of the experiment is shown. The two bodies consist of $10 \times 10 \times 10$ and $10 \times 30 \times 30$ cubic cells, with 4000 and 36000 particles, respectively. Each cell contains four particles in a face-centred cubic grid, as shown on the left-hand side of Fig. 11. The four particles, which are counted for this cell, are the lower left corner as well as the particles in the

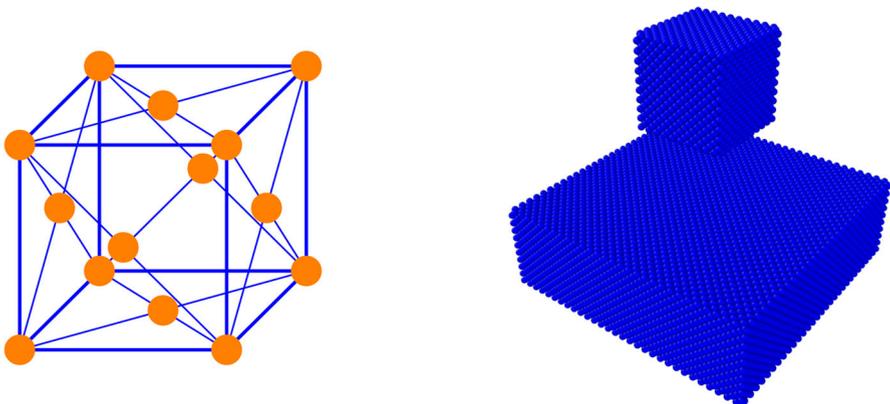


Fig. 11 Sketch of a face-centred cubic lattice on the left-hand side. On the right-hand side, the initial positions of the particles that form the smaller and the large body are shown in a plot made by the open visualization tool (OVITO) (cf. [57])

Table 2 Parameters for the simulation of the collision

L_1	=	150σ	L_2	=	150σ	L_3	=	150σ
ε	=	120	σ	=	3.4			
m	=	39.95	\mathbf{v}	=	(0, 0, -20.4)			
dist. part	=	$2^{1/6} \sigma$	N_1	=	4000	N_2	=	36,000
r_{cut}	=	2.5σ	τ	=	0.001			

centre of the front, bottom and left face. The others belong to regular grids spanned by these four particles. The shortest distance in the grid is $2^{1/6}\sigma$, according to the equilibrium length of the Lennard–Jones potential. At the beginning of the simulation, the smaller body moves with a high velocity \mathbf{v} towards the resting larger body. The simulation cell of size $[0, 150\sigma]^3$ is equipped with periodic boundary conditions. All data are given in Table 2. Several snapshots of the simulation are shown in Fig. 12. The simulation has been conducted with four processors. For our proof-of-concept implementation, we used four standard personal computers (PCs) as processors that had been connected by a one-gigabit local area network (1 GB LAN). The particles in Fig. 12 are colour-coded with respect to the processor that handles the particles.

The simulation is run up to (scaled) time $T = 30$. All results are given in scaled quantities. In Fig. 13, one can see that the DG method preserves the energy very well. The implicit midpoint rule overestimates the energy when the small body penetrates the larger body. The Verlet scheme underestimates the energy during this phase. If the larger body is destroyed, the energy computed by the midpoint rule decreases and the energy computed by the Verlet scheme increases. In order to check our computation, we also conduct the same experiment with LAMMPS and only one single processor

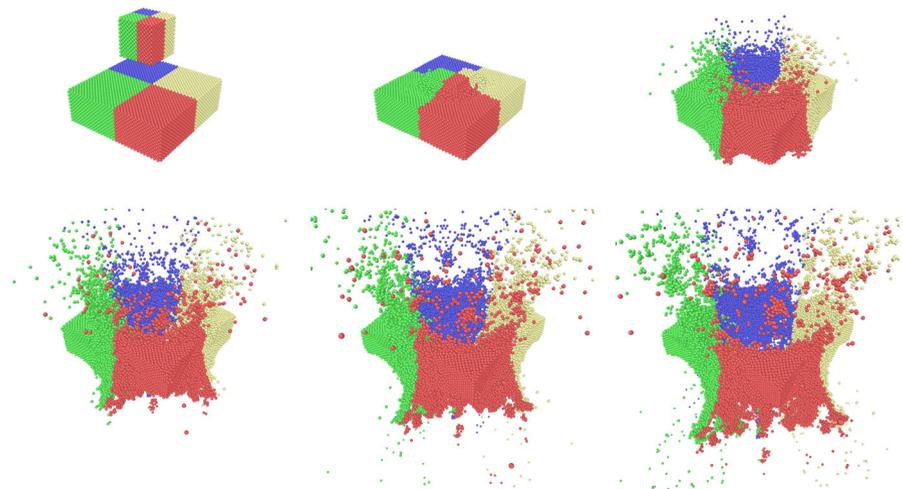


Fig. 12 Simulation of the collision of two bodies at times $t = 0, 4, 8, 12, 16, 20$ from top left to bottom right. The simulation is run with four processors identified by colour. The same colour means that the same processor is handling the particles

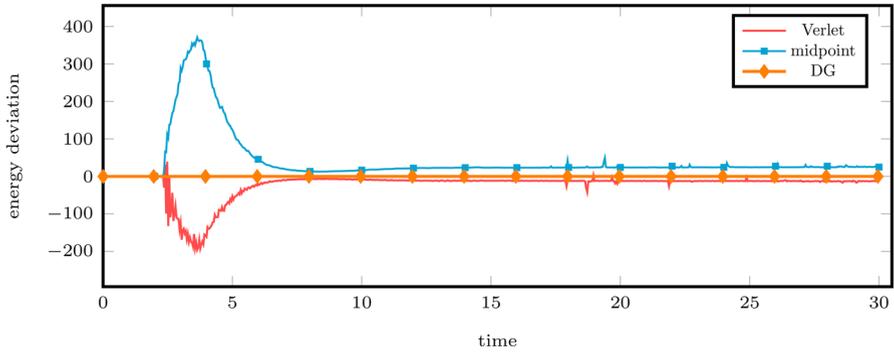


Fig. 13 Energy preservation for the collision of two bodies: the deviation from the exact energy is shown over the time span $[0, 30]$ for step size $\tau = 0.001$ for the Verlet scheme, the midpoint rule and the discrete gradient (DG) scheme. All three methods have been computed in parallel with four processors on a cluster computer (cf. [2])

(no parallelisation). The observed energy behaviour of the independent Verlet implementation in LAMMPS shows exactly the same energy curve, including the small peaks later on. That is, only the discrete gradient method is able to simulate a true microcanonical ensemble. The implicit equation in (14) is solved with the Newton method, (15), with the full Jacobian (16). The action of the Jacobian on a vector is directly computed. This way, one can transfer the linked cell method to the computation of the action of the Jacobian on a vector. The total linear momentum is preserved by all three methods. The total angular momentum can not be preserved, by any of the methods, due to the periodic boundary conditions. Only for free space or repelling boundary conditions, methods can preserve the total angular momentum in a MD simulation.

5.3 Rate of acceleration

A modification of the above experiment is used in order to quantify the rate of acceleration by the parallelisation. The modification is necessary to obtain a reasonable load balance for the processors in a parallel computation. It is inspired by the experiments in Section 4.4 of [15].

In a face-centred cubic lattice, 39,754 particles are placed in a box with dimensions $L \times L \times L$, where $L = (N/\rho)^{1/3}$. The density ρ is given in Table 3. Periodic boundary conditions are imposed on the faces of the box. The particles interact by short-range Lennard–Jones potentials whose parameters are also given in Table 3. The initial

Table 3 Parameters for the experiment to quantify the acceleration

ϵ	=	1	σ	=	1
m	=	1			
ρ	=	0.8442	T	=	1.44
r_{cut}	=	2.5σ	τ	=	0.00462

velocities of the particles are superimposed by a small thermal motion according to a Maxwell–Boltzmann distribution with reduced temperature T given in Table 3.

A fundamental measure to evaluate the increase in performance by parallelisation is the speedup

$$S(p) = \frac{T}{T(p)},$$

where p denotes the number of processors used in the computation, $T(p)$ denotes the time needed by the parallel computation with p processors, and T denotes the time needed by the sequential programme. We use $T(1)$, that is the time needed by the parallel programme with one processor, for the sequential execution time T . Table 4 shows the speedup for one time step of the corresponding integrators with up to 8 processors. The speedup of the Verlet scheme is as expected. The DG method and the midpoint rule show the same, or even slightly better, speedup with the parallelisation strategy discussed in Sect. 5.1. That is, the acceleration by parallelisation for the DG schemes is as successful as the parallelisation of the Verlet scheme.

5.4 Parallelisation of the evaluation of discrete gradients for bonded forces

The parallelisation of the bonded forces is simpler in the sense that it works in the standard way, which is illustrated in Fig. 14. On the left-hand side of Fig. 14, the domain with its border neighbourhood is shown. Then, the adjacent processors send the necessary particles, while this processor sends the particles needed by neighbouring processors. This is the state in the middle of Fig. 14.

Then, the particles in the border neighbourhood are linked based on the molecule numbers and atom numbers stored with the particles. After that, the current processor sees all data needed to compute the discrete gradients with respect to bonded forces for the particles in its domain. Recall, that the new positions are also stored with the corresponding particles. This state is illustrated on the right-hand side of Fig. 14. After their computation, the discrete gradients are also stored with the particles. Then, the particles in the border neighbourhood are separated again, which brings us back to the state in the middle of the figure. Finally, the particles in the border neighbourhood are deleted, because they are no longer needed. This procedure also works for the matrix-free computation of the Hessian of the bonded potentials times a vector.

5.5 Experiment with butane

We run an experiment with 64 united-atom butane molecules. The initial condition and a snapshot of the simulation at time $T = 4$ can be seen in Fig. 15. The data for the

Table 4 Speedup

Processors	1	2	4	8
Verlet	1	1.88	3.57	5.78
DG	1	1.86	3.62	6.52
Midpoint	1	1.89	3.55	6.31

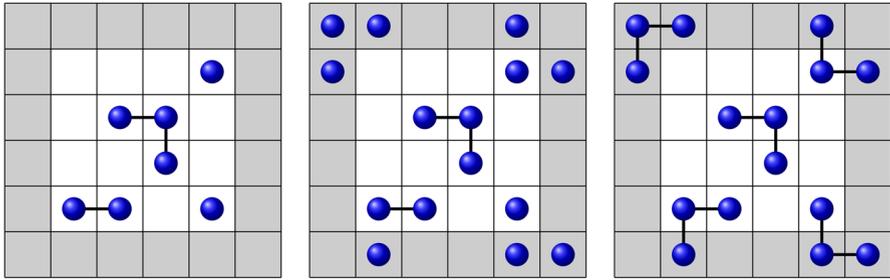


Fig. 14 Reconnection and separation of atoms in the border neighbourhood: If the subdomain of the processor receives atoms that are bonded within a molecule, the processor needs to reattach the atoms at the correct sites. This is illustrated here from left to right in a two-dimensional example. First, the subdomain with the knowledge of the processor is shown. In the middle, the situation is shown that results after the processor has received the particles necessary for the computation of the bonded forces from the adjacent processors. With the help of the molecule and particle numbers, the processor reattaches the molecules in the correct way. The result is illustrated on the right-hand side. After the computation of the bonded forces, the process is reversed from right to left

potentials have been chosen as in Table 1. Also, the potentials are chosen as before, i.e. the angle potential is given in (29) and the torsion potential is given in (30). The simulation is set in a periodic box of size $4r_{\text{cut}}$, $r_{\text{cut}} = 2.5\sigma$, and σ as in Table 1 in scaled variables. The whole simulation is scaled as before in the butane simulation.

The deviation from the exact energy over the simulation time with step size $\tau = 0.0001$ is shown in Fig. 16. The simulation has been run with four processors. While the DG methods preserve the energy up to round-off error, the implicit midpoint scheme and the Verlet scheme deviate from the constant energy. The peaks in the energy of the Verlet scheme are real. We also computed the energy for the given initial value with LAMMPS with one processor. This simulation reproduced exactly the same peaks as our code. This means that the particles do not evolve with respect to a genuine NVE ensemble at these peaks. The solid red line is the closest a standard molecular dynamics

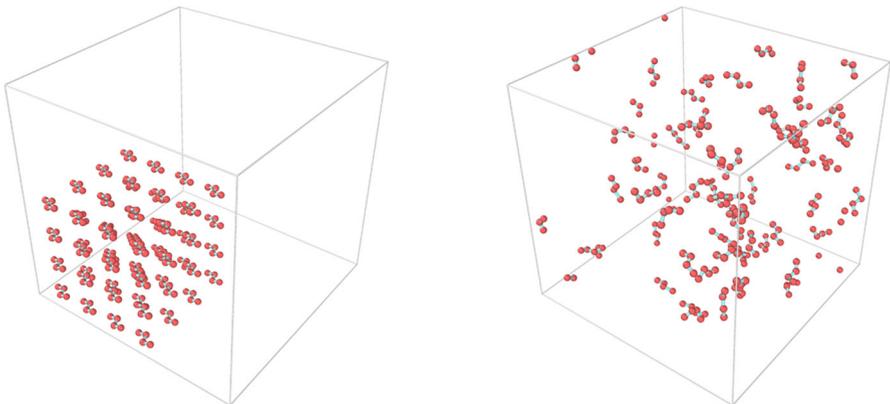


Fig. 15 Simulation of butane in a periodic box at times $t = 0, 4$. The plot is made by the open visualization tool (OVITO) (cf. [57])

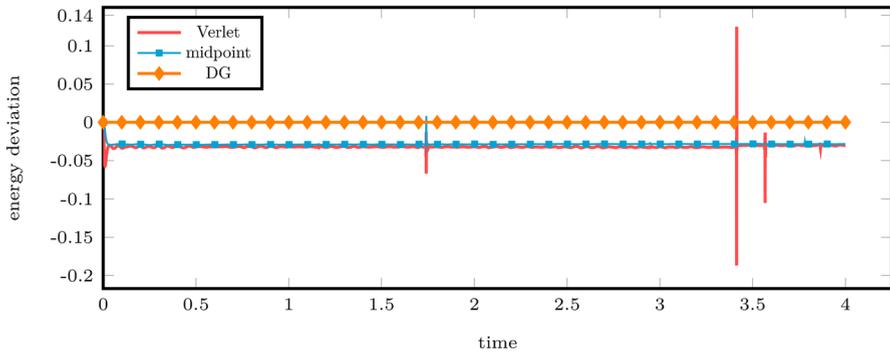


Fig. 16 Energy preservation during the simulation of 64 united-atom butane molecules: the deviation from the exact energy is shown over the time span $[0, 4]$ with step size $\tau = 0.0001$ for the Verlet scheme, the midpoint rule and the discrete gradient (DG) method

package with the Verlet scheme and setting the ensemble to the NVE ensemble can get. If one wishes more accuracy with respect to energy preservation, discrete gradient methods are an interesting alternative.

6 Conclusion

This work shows that all standard short-range interactions in a classical conservative molecular dynamics simulation can be computed by discrete gradient methods. These methods reliably preserve the total energy in the system, along with the total linear momentum and the total angular momentum in free space simulations. The simple and unified idea to construct the discrete gradients is to express all standard short-range interactions in terms of distances between atoms. The new discrete gradients for the dihedral angle potentials also suggest an interesting way to compute the gradient of dihedral angle potentials based on distances for the use in standard time-integration schemes. Furthermore, the discrete gradient methods can be parallelised. We proposed the necessary changes to the linked cell method for the parallel evaluation of the discrete gradients with respect to truncated Lennard–Jones potentials as well as the necessary changes for bonded forces. As a result, the proposed DG methods can be computed in parallel.

Acknowledgements The authors would like to dedicate this paper to John C. Butcher on the occasion of his 90th birthday. Of course, John is very well known for his ground-breaking contributions to Runge–Kutta methods. What may be less well known is John’s unwavering support of younger mathematicians over many years. Indeed, one of the current authors (GRWQ) is grateful to John Butcher for the invitation to participate and speak at ANODE, the Auckland Numerical ODE workshop, in July 1998 (and on several other occasions). A record of some of the talks presented then is given in [34].

Author contribution The authors contributed equally to this work.

Funding Open Access funding enabled and organized by Projekt DEAL. We thank the Isaac Newton Institute for Mathematical Sciences, Cambridge, for support during the programme ‘Geometry, compatibility

and structure preservation in computational differential equations' and the German Science Foundation (DFG) for support under project GRK 2450.

Data availability The datasets generated during the current study are available from the corresponding author on reasonable request.

Declarations

Ethical approval Not applicable

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Anderson, H.C.: Rattle: a "velocity" version of the shake algorithm for molecular dynamics calculations. *J. Comput. Phys.* **52**, 24–34 (1983). [https://doi.org/10.1016/0021-9991\(83\)90014-1](https://doi.org/10.1016/0021-9991(83)90014-1)
2. Bauke, H., Mertens, S.: Cluster computing. Springer (2006). <https://doi.org/10.1007/3-540-29928-9>
3. Bekker, H.: Molecular dynamics simulation methods revised. Proefschrift (Thesis), Rijksuniversiteit Groningen, (1996). <https://research.rug.nl/en/publications/molecular-dynamics-simulation-methods-revised>
4. Berendsen, H., Spoel, D., Vandrunen, R.: GROMACS—a message-passing parallel molecular-dynamics implementation. *Comput. Phys. Commun.* **91**, 43–56 (1995). [https://doi.org/10.1016/0010-4655\(95\)00042-E](https://doi.org/10.1016/0010-4655(95)00042-E)
5. Celledoni, E., Ehrhardt, M., Etmann, C., McLachlan, R.I., Owren, B., Schönlieb, C.-B., Sherry, F.: Structure-preserving deep learning. *Euro. J. Appl. Math.* **32**(5), 888–936 (2021). <https://doi.org/10.1017/S0956792521000139>
6. Celledoni, E., Eidnes, S., Owren, B., Ringholm, T.: Energy-preserving methods on Riemannian manifolds. *Math. Comp.* **89**(322), 699–716 (2020). <https://doi.org/10.1090/mcom/3470>
7. Celledoni, E., Eidnes, S., Owren, B., Ringholm, T.: Dissipative numerical schemes on Riemannian manifolds with applications to gradient flows. *SIAM J. Sci. Comput.* **40**, A3789–A3806 (2018). <https://doi.org/10.1137/18M1190628>
8. Celledoni, E., Grimm, V., McLachlan, R.I., McLaren, D.I., O'Neale, D., Owren, B., Quispel, G.R.W.: Preserving energy resp dissipation in numerical PDEs using the "average vector field" method. *J. Comput. Phys.* **231**, 6770–6789 (2012). <https://doi.org/10.1016/j.jcp.2012.06.022>
9. Dahlby, M., Owren, B.: A general framework for deriving integral preserving numerical methods for PDEs. *SIAM J. Sci. Comput.* **33**(5), 2318–2340 (2011). <https://doi.org/10.1137/100810174>
10. Eidnes, S.: Order theory for discrete gradient methods. *BIT Numer. Math.* **62**(4), 1207–1255 (2022). <https://doi.org/10.1007/s10543-022-00909-z>
11. Ehrhardt, M. J., Erlend, S. R., Ringholm, T., Schönlieb, C.-B.: A geometric integration approach to smooth optimisation: foundations of the discrete gradient method, preprint. [arXiv:1805.06444](https://arxiv.org/abs/1805.06444). <https://doi.org/10.48550/arXiv.1805.06444>
12. Gonzalez, O.: Time integration and discrete Hamiltonian systems. *J. Nonlinear Sci.* **6**, 449–467 (1996). <https://doi.org/10.1007/BF02440162>

13. Gonzalez, O.: Design and analysis of conserving integrators for nonlinear Hamiltonian systems with symmetry. Ph.D. thesis, Stanford University, Stanford, CA (1996). <https://searchworks.stanford.edu/view/3219326>
14. Gonzalez, O., Simo, J.C.: On the stability of symplectic and energy-momentum algorithms for nonlinear Hamiltonian systems with symmetry. *Comput. Methods Appl. Mech. Engrg.* **134**, 197–222 (1996). [https://doi.org/10.1016/0045-7825\(96\)01009-2](https://doi.org/10.1016/0045-7825(96)01009-2)
15. Griebel, M., Knapek, S., Zumbusch, G.: Numerical simulation in molecular dynamics. Springer-Verlag, Berlin, (2007). <https://doi.org/10.1007/978-3-540-68095-6>
16. Grimm, V., McLachlan, R. I., McLaren, D. I., Quispel, G. R. W., Schönlieb, C.-B.: Discrete gradient methods for solving variational image regularisation models. *J. Phys. A* **50**, 295201, 21 (2017). <https://doi.org/10.1088/1751-8121/aa747c>
17. Grimm, V.: Implementation of discrete gradient methods for dissipative PDEs in image processing on GPUs. in *Geometric Numerical Integration and its Applications, Math-for-Industry (MI) Lecture Note Series, 74*. Institute of Mathematics for Industry, Kyushu University, Fukuoka, 69–71, (2017). https://www.imi.kyushu-u.ac.jp/wp-content/uploads/2022/07/mil_74.pdf
18. Grimm, V., Quispel, G.R.W.: Geometric integration methods that preserve Lyapunov functions. *BIT* **45**, 709–723 (2005). <https://doi.org/10.1007/s10543-005-0034-z>
19. Grimm, V., Henn, S., Witsch, K.: A higher-order PDE-based image registration approach. *Numer. Linear Algebra Appl.* **13**(5), 399–417 (2006). <https://doi.org/10.1002/nla.467>
20. Hairer, E., Nørsett, S. P., Wanner, G.: Solving ordinary differential equations. I, second edition, Springer-Verlag, Berlin, (1993). <https://doi.org/10.1007/978-3-540-78862-1>
21. Hairer, E., Lubich, C., Wanner, G.: Geometric numerical integration, second edition. Springer-Verlag, Berlin, (2006). <https://doi.org/10.1007/3-540-30666-8>
22. Harten, A., Lax, P. D., van Leer, B.: On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Rev.* **25**, 35–61 (1983). <https://www.jstor.org/stable/2030019>
23. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.* **49**, 409–436 (1952). <https://doi.org/10.6028/jres.049.044>
24. Itoh, T., Abe, K.: Hamiltonian-conserving discrete canonical equations based on variational difference quotients. *J. Comput. Phys.* **76**(1), 85–102 (1988). [https://doi.org/10.1016/0021-9991\(88\)90132-5](https://doi.org/10.1016/0021-9991(88)90132-5)
25. IUPAC-IUB Commission on Biochemical Nomenclature, Abbreviations and symbols for the description of the conformation of polypeptide chains. Tentative rules (1969). *Biochem. J.* **9**, 3471–3479 (1970). [https://doi.org/10.1016/S0021-9258\(18\)62561-X](https://doi.org/10.1016/S0021-9258(18)62561-X)
26. LaBudde, R.A., Greenspan, D.: Discrete mechanics - a general treatment. *J. Comput. Phys.* **15**, 134–167 (1974). [https://doi.org/10.1016/0021-9991\(74\)90081-3](https://doi.org/10.1016/0021-9991(74)90081-3)
27. LaBudde, R. A., Greenspan, D.: Energy and momentum conserving methods of arbitrary order of the numerical integration of equations of motion. I. Motion of a single particle. *Numer. Math.* **25**(4), 323–346 (1975/76). <https://doi.org/10.1007/BF01396331>
28. LaBudde, R. A., Greenspan, D.: Energy and momentum conserving methods of arbitrary order of the numerical integration of equations of motion. II. Motion of a system of particles. *Numer. Math.* **26**(1), 1–16 (1976). <https://doi.org/10.1007/BF01396562>
29. Leimkuhler, B., Reich, S.: Simulating Hamiltonian dynamics. Cambridge University Press, Cambridge, (2004). <https://doi.org/10.1017/CBO9780511614118>
30. Leimkuhler, B., Matthews, C.: *Mol. Dynamics*, Springer, Cambridge, (2015). <https://doi.org/10.1007/978-3-319-16375-8>
31. Leimkuhler, B., Skeel, R.D.: Symplectic numerical integrators in constrained Hamiltonian systems. *J. Comput. Phys.* **112**, 117–125 (1994). <https://doi.org/10.1006/jcph.1994.1085>
32. Lennard-Jones, J.E.: On the determination of molecular fields. *Proc. R. Soc. Lond. A* **106**(738), 463–477 (1924). <https://doi.org/10.1098/rspa.1924.0081>
33. Lindahl, E., Hess, B., van der Spoel, D.: GROMACS 3.0: a package for molecular simulation and trajectory analysis. *J. Mol. Model.* **7**, 306–317 (2001). <https://doi.org/10.1007/s008940100045>
34. McLachlan, R.I., Quispel, G.R.W.: Six lectures on the geometric integration of ODEs. *London Math. Soc. Lecture Note Ser.* **284**, 155–210 (2001). <https://doi.org/10.1017/CBO9781107360198.008>
35. McLachlan, R. I., Quispel, G. R. W., Robidoux, N.: Geometric integration using discrete gradients. *R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci.* **357**, 1021–1045 (1999). <https://doi.org/10.1098/rsta.1999.0363>

36. McLachlan, R.I., Quispel, G.R.W., Turner, G.S.: Numerical integrators that preserve symmetries and reversing symmetries. *SIAM J. Numer. Anal.* **35**, 586–599 (1998). <https://doi.org/10.1137/S0036142995295807>
37. McLaren, D. I., Quispel, G. R. W.: Bootstrapping discrete-gradient integral-preserving integrators to fourth order. In: Daniel, M., Rajasekar, S. (eds.) *Nonlinear Dynamics*, 157–172. Narosa Publishing House, (2008), http://www.narosa.com/books_display.asp?catgcode=978-81-7319-941-7
38. Mei, J., Davenport, J.W., Fernando, G.W.: Analytic embedded-atom potentials for fcc metals: application to liquid and solid copper. *Phys. Rev. B* **43**(6), 4653–4658 (1990). <https://doi.org/10.1103/PhysRevB.43.4653>
39. Message Passing Interface Forum, MPI: a message-passing interface standard, (2023). <https://www.mpi-forum.org/>
40. Miyatake, Y.: Structure-preserving model reduction for dynamical systems with a first integral. *Jpn. J. Ind. Appl. Math.* **36**(3), 1021–1037 (2019). <https://doi.org/10.1007/s13160-019-00378-y>
41. Miyatake, Y., Cohen, D., Furihata, D., Matsuo, T.: Geometric numerical integrators for Hunter-Saxton-like equations. *Jpn. J. Ind. Appl. Math.* **34**(2), 441–472 (2017). <https://doi.org/10.1007/s13160-017-0252-1>
42. Miyatake, Y., Matsuo, T.: Conservative finite difference schemes for the Degasperis-Procesi equation. *J. Comput. Appl. Math.* **236**(15), 3728–3740 (2012). <https://doi.org/10.1016/j.cam.2011.09.004>
43. Norton, R.A., McLaren, D.I., Quispel, G.R.W., Stern, A., Zanna, A.: Projection methods and discrete gradient methods for preserving first integrals of ODEs. *Discrete Contin. Dyn. Syst.* **35**, 2079–2098 (2015). <https://doi.org/10.3934/dcds.2015.35.2079>
44. Perse, B., Kormann, K., Sonnendrücker, E.: Geometric particle-in-cell simulations of the Vlasov-Maxwell system in curvilinear coordinates. *SIAM J. Sci. Comput.* **43**(1), B194–B218 (2021). <https://doi.org/10.1137/20M1311934>
45. Plimpton, S.: Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.* **117**, 1–19 (1995). <https://doi.org/10.1006/jcph.1995.1039>
46. Quispel, G.R.W., Turner, G.S.: Discrete gradient methods for solving ODEs numerically while preserving a first integral. *J. Phys. A* **29**, L341–L349 (1996). <https://doi.org/10.1088/0305-4470/29/13/006>
47. Reich, S.: Enhancing energy conserving methods. *BIT. Numer. Math.* **36**(1), 122–134 (1996). <https://doi.org/10.1007/BF01740549>
48. Riis, E.S., Ehrhardt, M.J., Quispel, G.R.W., Schönlieb, C.-B.: A geometric integration approach to nonsmooth, nonconvex optimisation. *Found. Comput. Math.* **22**, 1351–1394 (2022). <https://doi.org/10.1007/s10208-020-09489-2>
49. Ringholm, T., Lazić, J., Schönlieb, C.-B.: Variational image regularization with Euler’s elastica using a discrete gradient scheme. *SIAM J. Imaging Sci.* **11**(4), 2665–2691 (2018). <https://doi.org/10.1137/17M1162354>
50. Ryckaert, J.P., Ciccotti, G., Berendsen, H.J.C.: Numerical integration of the Cartesian equations of motion of a system with constraints: molecular dynamics of *n*-alkanes. *J. Comput. Phys.* **23**, 327–341 (1977). [https://doi.org/10.1016/0021-9991\(77\)90098-5](https://doi.org/10.1016/0021-9991(77)90098-5)
51. Saluena, C., Avalos, J. B.: Molecular dynamics algorithm enforcing energy conservation for microcanonical simulations. *Phys. Rev. E* **89**, 053314, 10 (2014). <https://doi.org/10.1103/PhysRevE.89.053314>
52. Sanz-Serna, J.M., Calvo, M.P.: *Numerical Hamiltonian problems*. Chapman & Hall, London (1994)
53. Schiebl, M., Romero, I.: Energy-momentum conserving integration schemes for molecular dynamics. *Comput. Mech.* **97**, 915–935 (2021). <https://doi.org/10.1007/s00466-020-01971-6>
54. Simo, J.C., Gonzalez, O.: Assessment of energy-momentum and symplectic schemes for stiff dynamical systems. *Merican Society of Mechanical Engineers, ASME Winter Annual Meeting, New Orleans, LA* (1993)
55. Simo, J.C., Tarnow, N., Wong, K.K.: Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics. *Comput. Methods Appl Mech. Eng.* **100**, 63–116 (1992). [https://doi.org/10.1016/0045-7825\(92\)90115-Z](https://doi.org/10.1016/0045-7825(92)90115-Z)
56. Stuart, A.M., Humphries, A.R.: *Dynamical systems and numerical analysis*. Cambridge University Press, Cambridge (1996)
57. Stukowski, A.: Visualization and analysis of atomistic simulation data with OVITO - the Open Visualization Tool. *Model. Simul. Mater. Sci. Eng.* **18**, 015012 (2010). <https://doi.org/10.1088/0965-0393/18/1/015012>

58. Thompson, A. P., Aktulga, H. M., Berger, R., Bolintineanu, D. S., Brown, W. M., Crozier, P. S., in 't Veld, P. J., Kohlmeyer, A., Moore, S. G., Nguyen, T. D., Shan, R., Stevens, M. J., Tranchida, J., Trott, C., Plimpton S. J.: LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comput. Phys. Commun.* **271**, 108171 (2022). <https://doi.org/10.1016/j.cpc.2021.108171>
59. Toxvaerd, S.: Comment on constrained molecular dynamics of macromolecules. *J. Chem. Phys.* **87**, 6140–6143 (1987). <https://doi.org/10.1063/1.453488>
60. Yaguchi, T., Matsuo, T., Sugihara, M.: The discrete variational derivative method based on discrete differential forms. *J. Comput. Phys.* **231**(10), 3963–3986 (2012). <https://doi.org/10.1016/j.jcp.2012.01.035>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.