A fast mollified impulse method for biomolecular atomistic simulations

L. Fath a,*, M. Hochbruck a, C.V. Singh b

a Institute for App. and Num. Mathematics, Karlsruhe Institute of Technology, Germany
b Department of Materials Science & Engineering, University of Toronto, Canada

A R T I C L E I N F O
Article history:
Received 18 May 2016
Received in revised form 5 December 2016
Accepted 16 December 2016
Available online 21 December 2016

Keywords:
Mollified impulse method
Highly oscillatory
Resonances
Corotation
Molecular dynamics

A B S T R A C T
Classical integration methods for molecular dynamics are inherently limited due to resonance phenomena occurring at certain time-step sizes. The mollified impulse method can partially avoid this problem by using appropriate filters based on averaging or projection techniques. However, existing filters are computationally expensive and tedious in implementation since they require either analytical Hessians or they need to solve nonlinear systems from constraints. In this work we follow a different approach based on corotation for the construction of a new filter for (flexible) biomolecular simulations. The main advantages of the proposed filter are its excellent stability properties and ease of implementation in standard softwares without Hessians or solving constraint systems. By simulating multiple realistic examples such as peptide, protein, ice equilibrium and ice–ice friction, the new filter is shown to speed up the computations of long-range interactions by approximately 20%. The proposed filtered integrators allow step sizes as large as 10 fs while keeping the energy drift less than 1% on a 50 ps simulation.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

The tremendous rise of available computing power has made simulations based on classical molecular dynamics (MD) a powerful tool for many investigations such as predicting material properties. Those simulations are fruitful in research of mechanisms on a nanoscale and allow for deeper insight on how exactly many phenomena occur. However, the simulations itself are computationally challenging; both time scale and physical scale are so small that investigations usually have to be restricted to very small domains and very short time frames [1–3].

Multiple time-step (MTS) approaches [4–8] can significantly increase the accessible time scale by taking advantage of the specific structure of the underlying problem. Standard MTS methods, such as the impulse method [9,10], increase computational efficiency by evaluating force fields only at their corresponding time scale. Unfortunately, highly oscillatory motions lead to severe resonance instabilities in standard integrators which restrict their step sizes [11–16] to below 5 fs for most problems. Especially the expensive computation of slow, but long-range electrostatic contributions still poses a major bottleneck in many simulations. Therefore, achieving a larger outer time-step in MTS integrators is a pressing objective of research.

For equilibrium simulations, such as sampling conformational equilibria, multiple and very successful attempts to tackle the time scale problem have been proposed; a clever restriction of certain parameters with thermostats, averaging tech-

* Corresponding author.
E-mail addresses: lukas.fath@kit.edu (L. Fath), marlis.hochbruck@kit.edu (M. Hochbruck), chandraveer.singh@utoronto.ca (C.V. Singh).

http://dx.doi.org/10.1016/j.jcp.2016.12.024
0021-9991/© 2016 Elsevier Inc. All rights reserved.
niques, and adding stochastic forces make it possible to avoid resonance problems \[17,19\], and step sizes of 100 fs and larger have been reported. However, these methods do not preserve the dynamics, since a large number of constraints is used to impede energy build-ups in single modes.

Hence, for non-equilibrium simulations, such as dynamic processes like friction, phase transformations, or grain boundaries, the dynamics itself make it questionable to use such methods \[20,21\]. One possibility here is to freeze the fastest degrees of freedom with algorithms such as SHAKE \[22,23\]. However, this rather drastic step removes all contributions which are attributed to the fastest modes. Instead, mild stochastic forces can be used \[24,25,26\], but care has to be taken to choose a weak coupling to not destroy the dynamics. The mollified impulse method \[27\] uses a totally different idea. Resonances are avoided using averaged positions for the evaluation of the slowest potential. In contrast to SHAKE, this method retains its ability to resolve the fastest frequencies. Furthermore, this approach can be carefully combined with methods using mild stochastic forces \[28,29\] reaching even longer time-steps.

A crucial ingredient of the mollified impulse method is its filter. Current filters are either based on averaging \[15\] or projection \[30,31\] techniques. Averaging techniques have been analyzed \[32\] and tested on small systems \[33\]. Their major drawback is expensive forward and backward integration with analytical Hessians of the oscillatory part of the potential. Furthermore, numerical experiments \[30,34\] suggest that these methods are less stable than projection methods where molecules are projected back to their rest positions. However, this requires solving a nonlinear constraint system with SHAKE-like Newton iterations in each time-step.

In this paper we present a new filter technique based on corotation. It is designed for biomolecular simulations and works for arbitrary molecules. After a short discussion of the mollified impulse method in section 2, we show in section 3 how this new corotational filter can be constructed without using analytical Hessians and Newton iterations. Instead, the new filter can be computed explicitly in a very cheap and effective manner. Furthermore, we prove that the new filter applied to common flexible water models is almost the same as a projection method. In section 4 we introduce four typical simulation scenarios and discuss the results in section 5. There we provide numerical evidence that the corotational filter is as stable as projection methods, and is, indeed, very computationally efficient.

### 2. Mollified impulse multiple-timestep method

In classical molecular dynamics, Hamilton’s equations yield atomic trajectories by numerically solving

\[
\begin{align*}
\dot{q}(t) &= \nabla_p H(q, p) \quad \text{with} \quad q(t_0) = q_0 \\
\dot{p}(t) &= -\nabla_q H(q, p) \quad \text{with} \quad p(t_0) = p_0
\end{align*}
\]

for a given (separable) Hamiltonian function

\[H(q, p) = T(p) + U(q) \quad \text{with} \quad T(p) = \frac{1}{2} p^T M^{-1} p,\]

where we denote atomic positions and momenta by vectors \(q(t), p(t) \in \mathbb{R}^{3d}\) at time \(t\), \(M \in \mathbb{R}^{3d \times 3d}\) is a diagonal matrix containing the atomic masses, and \(U(q)\) represents a potential modeling interaction between atoms.

In this paper we assume that the potential \(U\) can be written in the form

\[U(q) = U_{\text{fast}}(q) + U_{\text{medium}}(q) + U_{\text{slow}}(q).\]

Here, the fast intra-molecular part \(U_{\text{fast}}\) consists of the bond, angle and torsion potentials. The short-range pair interactions \(U_{\text{medium}}\) are usually modeled with van der Waals and Coulomb potentials within a certain cut-off radius. Finally, we have the remaining long-range electrostatic contributions \(U_{\text{slow}}\). Most forcefields for biomolecular simulations admit to such a decomposition \[1\] – e.g. the AMBER \[35\], CHARMM \[36\], GROMOS \[37\] and OPLS-AA \[38\] force fields. When numerically computing a trajectory of (1), the impulse method \[9,10\] takes advantage of such a decomposition and allows to evaluate faster potentials more often than slower ones.

However, the step size of the impulse method is inherently limited by severe resonances which are caused by an arbitrary evaluation of the slow force at positions which do not represent the oscillatory nature of a molecule \[12,15\]. Instead, it is advantageous to replace the slow potential \(U_{\text{slow}}\) in (3) by a mollified version \(U_{\text{slow}}(\Psi(q))\) with a filter \(\Psi\). Hence, the slow force \(\nabla(U_{\text{slow}}(\Psi(q))) = \Psi'(q) \nabla U_{\text{slow}}(\Psi(q))\) is evaluated at a filtered position \(\Psi(q)\) and also post-filtered with the transposed Jacobian \(\Psi'(q)\). This is usually referred to as the mollified impulse method \[27\]. Here, resonances are avoided by first time-averaging the position before evaluating the slow force. Then, the Jacobian removes components of the slow force which would excite the fastest mode, and eventually would lead to resonance issues.

In Algorithm 1, we depict this approach when applied to the potential decomposition in (3). For suitably chosen integers \(N\) and \(K\), a step with the mollified impulse method consists of three stages: The inner stage is the application of \(NK\) steps of the velocity-Verlet method with step size \(h/(NK)\) to the kinetic \((T(p))\) and fast part \(U_{\text{fast}}\) of the Hamiltonian. In the middle stage, at begin and end of every \(K\)-th step of the velocity-Verlet method, a half ’kick’ with the potential \(U_{\text{medium}}\) and step size \(h/N\) is added. The outer stage only acts at the begin and end of the entire step, and computes a half ’kick’ with the slow mollified potential \(U_{\text{slow}}(\Psi(q))\) and step size \(h\). Note, that if we use the trivial ’filter’ \(\Psi(q) = q\) in Algorithm 1, we obtain the standard impulse method with three stages.
Algorithm 1: One Step of the Mollified Impulse Method applied to (1) with (2) and (3).

Input: position \( q \), momentum \( p \), step size \( h \), stage factors \( N, K \), filter \( \Psi \)

Output: new position \( q \), new momentum \( p \)

\[ p \leftarrow p - \frac{h}{N} \nabla U_{\text{medium}}(q) \]

for \( i = 1, \ldots, N \)

\[ p \leftarrow p - \frac{h}{K} \psi_1(q) \nabla U_{\text{slow}}(\Psi(q)) \]

for \( j = 1, \ldots, K \)

\[ q \leftarrow q + \frac{h}{K} M^{-1} p \]

\[ p \leftarrow p - \frac{h}{N} \nabla U_{\text{fast}}(q) \]

\[ p \leftarrow p - \frac{h}{N} \nabla U_{\text{medium}}(q) \]

Two types of filters have been proposed in the literature. First, there are filters based on time averaging techniques [15, 32,33]. Positions are integrated forwards and backwards in time using only the fast forces and then averaged by a weight function \( \Psi \):

\[
\Psi(q) = \int_{-\infty}^{\infty} \Phi(\tau) x(\tau) d\tau
\]

with \( \dot{x} = -M^{-1} \nabla U_{\text{fast}}(x) \), \( x(0) = q \), \( \dot{x}(0) = 0 \).

Second, equilibrium type methods [30,31] reset the molecule to its equilibrium position using a constraint function \( g \):

\[
\Psi(q) = q + M^{-1} g(q)^T \lambda,
\]

\[
g(\Psi(q)) = 0.
\]

Note, that for \( m \) constraints, \( g : \mathbb{R}^d \rightarrow \mathbb{R}^m \) is a vector function and \( \lambda \in \mathbb{R}^m \) is a vector of Lagrange multipliers. \( g_q \in \mathbb{R}^{m \times 3d} \) denotes the Jacobian with respect to the position vector \( q \). Due to the nonlinearity of the problem, (5) is usually solved with a Newton-type method. At a first glance resetting internal degrees of freedom with \( g \) might seem too crude of an approximation, but if the outer time-step is large enough this becomes a useful approximation since molecules oscillate around their equilibrium positions.

With these filters, a time-step roughly twice as large compared to a standard multiple time-step method has been achieved. A reliable and stable example is the equilibrium method presented in [30] or a modified version in [31]. Unfortunately, both types of filters need significant computational effort. Averaging filters need to integrate forward and backward (including the derivative!) with the fast forces every outer time-step. Equilibrium methods need to solve nonlinear systems from the constraints.

In the next section, we introduce a new type of filter, which we call corotational filter. They are similar to equilibrium filters in the sense, that corotational filters also reset certain degrees of freedom within molecules to their equilibrium positions. However, filtered molecules are obtained by an approach based on corotation. While this approach yields a similar quality of the filtering process, it is much cheaper in computational cost. Furthermore, the new filter can be implemented completely without computing trajectory averages or solving nonlinear systems. Instead, it is accessible via a simple and explicit algorithm thus reducing algorithmic complexity in the filtering process.

3. Corotational filters

The basic idea of corotation is to decompose motion into rotational and translational parts plus deformations from flexibility in bonds and angles. Corotational filters then discard deformational contributions. However, it would be too much to ask, that for an arbitrary molecule, the structure could be well represented by a single rotation. Therefore, we decompose the (potentially very complex) molecular structure into disjunct clusters of much smaller size, e.g. 2–5 atoms.

3.1. Cluster decomposition of a biomolecular structure

The goal of this decomposition is a division of the molecular structure into many but much smaller substructures. Much like in a ‘divide and conquer’ approach, the substructures are then treated independently, and the connection between substructures is neglected. We call the substructures clusters, and every atom should be in exactly one cluster. The construction of the clusters is such that for a fast bond, both atoms need to be part of the same cluster. Only atoms connected by a slower bond are allowed to be members of different clusters. Finally, we require that each cluster has a distinct central atom, which is usually the heaviest one.

Obviously, the classification of faster and slower bonds allows for some choice. Typically, we try to obtain small enough clusters (e.g. 2–5 atoms) such that the overall rotation of that cluster is still a good approximation of the local rotation for
all its containing atoms, i.e. that there is no torsional degree of freedom inside a cluster. Of course we could also include
trivial clusters containing only a single atom, however, this excludes that atom from the filtering process.

In biomolecular structures, the stretching modes of hydrogen bonds (e.g. H–O, H–N, H–C) are much faster than backbone
connections (C–C, C–N, etc). Thus, most clusters consist of a central atom with fast hydrogen bonds within the cluster, and
slower backbone bonds connecting it to other clusters. Let us take an alkane chain as example (see Fig. 1). It contains slow
backbone bonds (C–C) which connect clusters of methyl (CH3) and methanediyl (CH2) groups: CH3–CH2–...–CH2–CH3. The
cluster decomposition yields two terminal clusters of size 4, and then a chain of clusters of size 3.

While for simple linear molecules, the choice of clusters is straightforward, it becomes a bit more complex in structures
such as aromatic rings. In Fig. 2, such a ring structure is decomposed in two different ways. Numerical experiments did
not indicate a sensitivity towards the specific decomposition, as long as the fastest bonds were contained inside clusters.
Indeed, even very simple decompositions (such as the one used for the peptide in the example section later) yield good
results. There are (semi-)automatic algorithms to identify such cluster decompositions, and they are sometimes used in
constraining algorithms such as the SHAKE algorithm implemented in LAMMPS [39], since such a decomposition allows for
a very efficient implementation.

Similar decomposition approaches are used by many other methods in MD, most notably by coarse-graining methods
(see e.g. [40,41] and references therein) and by internal coordinate molecular dynamics (ICMD) such as the GNEIMo method
[42,43].

3.2. The new corotational filter

After generating a cluster decomposition, the clusters are treated independently. We now illustrate the filter algorithm
for a single cluster containing \( n \) atoms at position \( q = [x_1^T, x_2^T, ..., x_n^T]^T \in \mathbb{R}^{3n} \).

A single cluster is a small structure, and can be well represented by its rotational orientation \( R \in \mathbb{R}^{3 \times 3} \) and center of
mass \( c \in \mathbb{R}^3 \). To get rid of internal deformations, we construct a reference position \( q_0 \in \mathbb{R}^{3n} \), which has the same structure
as this cluster, but with all fast internal degrees of freedom (such as bonds and angles) set back to their corresponding
equilibrium values. This reference configuration is then rotated and shifted to match the cluster’s rotation \( R \) and center of
mass \( c \).

Using Kronecker products \( \otimes \), the new corotational filter can be calculated as

\[
\Psi(q) = [1, \ldots, 1]^T \otimes c + (I_n \otimes RR_0^T)q_0,
\]  \hspace{1cm} (6)
where we assume that $q_0$ has center of mass at $[0,0,0]^T$ and rotational orientation $R_0$. Note, that this can be further simplified, if we choose $q_0$ such that the resulting $R_0$ is the identity matrix $I_3$.

It remains to give an algorithm which quickly approximates the rotational orientation. The given cluster has a central atom with position $x_1$ bonded to $n - 1$ atoms with positions $x_2, \ldots, x_n$. Let us abbreviate $r_i = x_i - x_1$. A fast way of estimating the rotation between $q$ and $q_0$ is described in [44]: We compute an orthonormal frame of vectors $R = [n_1, n_2, n_3] \in \mathbb{R}^{3 \times 3}$ with

$$n_1 = \sum_{i \in S_1} \frac{r_i}{\|r_i\|} \left/ \sum_{i \in S_1} \|r_i\| \right.,$$

$$n_2 = \left( s - s^T n_1 n_1 \right) / \| s - s^T n_1 n_1 \|, \quad s = \sum_{i \in S_2} \frac{r_i}{\|r_i\|},$$

$$n_3 = (n_1 \times n_2) / \| n_1 \times n_2 \|,$$

where we choose $n_1 \in \mathbb{R}^3$ to be an averaged direction of some bonds $i \in S_1$ and for $n_2 \in \mathbb{R}^3$ we choose a different set $S_2$ of bonds. $S_1$ and $S_2$ are suitable chosen, and we illustrate a few common examples in a following section. The vectors $n_1, n_2, n_3$ form an orthonormal basis of $\mathbb{R}^3$. $R_0$ is computed in the same way at reference position $q_0$. Then $RR_0^T$ is an estimate of the rotation from $q_0$ to $q$. In Fig. 3, we sketch the filter algorithm applied to a cluster of three atoms.

### 3.3. Properties of the filter

By construction the filter retains center of mass and rotational orientation of that cluster but by using a reference configuration $q_0$ internal deformations caused by the fastest oscillations are discarded. Also, if a cluster is already in its equilibrium position $q_0$, the filter is the identity map, i.e. $\Psi(q_0) = q_0$. Furthermore, the filter is invariant under rotation and translation in the following sense: If $R$ is a rotation, and $s$ is a shift, it holds that $\Psi(Rq + s) = R\Psi(q) + s$.

In (6), the filter depends only on the center of mass and rotational orientation of that cluster. Since the filter is supposed to smoothen the trajectory and remove highly oscillatory components to suppress resonances, the choice of $S_1$ and $S_2$ has a strong impact on the success of the filter. While the center of mass of a cluster is independent of fast internal motion, we need to ensure that the choice of $S_1$ and $S_2$ leads to a smooth rotational approximation. Constructing $R$ from the direction of the bonds, the fast changes in bond length are already neglected. Furthermore, by a smart combination of neighboring bonds, it is usually possible to remove angle motions. However, not all fast modes can be removed entirely by a linear combination of bond directions, e.g. some asymmetric stretch modes will remain, and we can only reduce the magnitude of those modes. Numerical experiments in section 5 will show though, that this does not pose a problem.

Note, that the filter provides a quite general approach, and there is lots of room for tailoring this filter to one’s needs. First, the reference position is computed using the rigid body of that cluster, that is, all internal degrees of freedom are set to their equilibrium values. Since such a reference position does not change over time, it can be precomputed – and that is why we will use this approach in the numerical experiments. But it also works, if only some faster bonds are reset, and e.g. the current angular position is kept. Second, the extraction of rotational information offers some freedom. The more smooth it behaves (under an oscillatory trajectory), the more effective the filter will be. In the given approach, the rotation is estimated using a linear combination of internal directions around a distinct, central atom. But in general, any other approach which gives a robust estimation can be used.

### 3.4. Examples of sizes 2–4

The filter presented in the compact notation in (6) and (7) is not very instructive. Here, we present examples for cluster sizes $n = 2, 3$ and 4. For $n = 2$ and 3, the filter can be expressed more succinctly.
In the simple case \( n = 2 \), two atoms \( x_1, x_2 \) with masses \( m_1, m_2 \) are connected via a single bond with equilibrium length \( r_0 \). Because of its symmetries, the cluster can be described by a single direction, i.e. \( q_0 \) has no contribution in direction of \( n_2, n_3 \). We choose

\[
\begin{align*}
n_1 &= \frac{x_2 - x_1}{\|x_2 - x_1\|}, \\
n_2, n_3 \text{ s.t. they form an orthonormal basis.}
\end{align*}
\]

Therefore, the filter reads

\[
\Psi(q) = \begin{pmatrix} c + a_1 n_1 \\ c + a_2 n_1 \\ c + a_1 n_2 \\ c + a_2 n_2 \\ c + a_1 n_3 \\ c + a_2 n_3 \end{pmatrix},
\]

with

\[
\begin{align*}
a_1 &= -\frac{m_2}{m_1 + m_2} r_0, \\
a_2 &= \frac{m_1}{m_1 + m_2} r_0.
\end{align*}
\]

(8)

In the formalism of (6) and (7) this corresponds to \( S_1 = \{2\} \) (since it only uses the bond \( r_2 = x_2 - x_1 \)).

For a triatomic cluster \( (n = 3) \), we can reduce by one dimension if we choose the reference cluster \( q_0 \in \mathbb{R}^9 \) in the \((x, y)\)-plane. Since the \( z \)-component is zero, we can omit the computation of \( n_3 \). We consider the following choice:

\[
\begin{align*}
n_1 &= \frac{r_2}{\|r_2\|} + \frac{r_3}{\|r_3\|} = \frac{r_2}{\|r_2\|} + \frac{r_3}{\|r_3\||,} \\
n_2 &= \frac{r_2 - r_1}{\|r_2 - r_1\|} + \frac{r_3}{\|r_3\|}.
\end{align*}
\]

(9)

This particular choice \((S_1 = \{2, 3\}, S_2 = \{2\})\) has the advantage that vibrations due to angle bending and symmetric bond stretching are entirely removed by design; if \( r_2 \) changes as much in an opposite direction as \( r_3 \), then there is no influence on \( n_1, n_2 \) does not change either since \( r_2 \) stays in the same plane. This property holds for any triatomic symmetric molecule, most notably for flexible water models.

The filter then reads \((q_0^i \text{ denotes the } i\text{-th entry of } q_0)\)

\[
\Psi(q) = \begin{pmatrix} c + q_0^1 n_1 + q_0^2 n_2 \\ c + q_0^4 n_1 + q_0^5 n_2 \\ c + q_0^7 n_1 + q_0^8 n_2 \end{pmatrix}.
\]

(10)

However, we have to make one restriction: If \( \theta \) is close or equal to 180° such as in thiocyanate \((\text{SCN})\) or hydrogen cyanide \((\text{HCN})\), \( r_2/\|r_2\| + r_3/\|r_3\| \) threatens to cancel. Thus, for such a cluster, it is much better to use a treatment similar to the case \( n = 2 \) with

\[
\begin{align*}
n_1 &= \frac{r_2}{\|r_2\|} - \frac{r_3}{\|r_3\|} = \frac{r_2}{\|r_2\|} - \frac{r_3}{\|r_3\|}. \\
\end{align*}
\]

(11)

Clusters with four atoms \((n = 4)\) frequently happen with terminal \( \text{CH}_3 \) groups, where the carbon atom has a slow bond connecting it onwards to the next cluster.

Here, the best choice would be \( S_1 = \{2, 3, 4\} \). For \( S_2 \) we choose any two of the ‘satellites’: \( S_2 = \{2, 3\} \) or \( S_2 = \{3, 4\} \) or \( S_2 = \{2, 4\} \). However, in a planar tetra-atomic structure (such as formaldehyde \( \text{H}_2\text{CO} \)) similar to the three atomic linear case we face the danger that \( r_2 + r_3 + r_4 \approx 0 \). Instead, we then choose \( S_1 \) and \( S_2 \) such that \( n_1, n_2 \) reliably characterize the plane in which the cluster lives.

We want to finish by pointing out one detail for the simulation of flexible water. For flexible water, we can directly compare the existing equilibrium filter from [30] to our new filter. The equilibrium filter is given by (5) with

\[
g(q) = \begin{pmatrix} \|x_2 - x_1\|^2 - r_0^2 \\ \|x_3 - x_1\|^2 - r_0^2 \\ \|x_3 - x_2\|^2 - 2r_0^2(1 - \cos \theta_0) \end{pmatrix}
\]

for water molecules at position \( x_1(\text{oxygen}), x_2, x_3 \in \mathbb{R}^3 \) with equilibrium bond length \( r_0 \) and equilibrium angle \( \theta_0 \). Hence for every molecule a small 3-by-3 nonlinear constraint system has to be solved e.g. with a simplified Newton method.
the molecules are close enough to an equilibrium position, this method usually converges within a few steps. Nevertheless, it is more expensive than the corotational filter since the latter only costs roughly as much as a single Newton step.

Now, the interesting point is, that for water molecules (or any symmetric triatomic molecules) equilibrium (5) and corotational filter (6) with (9) are close in the sense that they are exactly equal whenever the two bonds have equal length. This is quite straightforward to show, since a water molecule in its equilibrium position is uniquely identified by its center of mass and rotation matrix \( R \) containing the vectors \( n_i, i = 1, 2, 3 \). Both filters maintain center of mass. For the corotational filter (by design) the vectors \( n_i \) do not change during filtering. So we only need to check the vectors \( n_i \) for the equilibrium filter.

It is, in fact, sufficient to check \( n_1 \) since both filters do not alter the plane in which the molecule lives. Denoting filtered values with tilde, easy calculations reveal that after filtering we have

\[
\frac{\tilde{r}_1}{||\tilde{r}_1||} + \frac{\tilde{r}_2}{||\tilde{r}_2||} = \frac{r_1}{r_0} \left( 1 + \frac{\lambda_1}{m_2} + 2 \frac{\lambda_1}{m_1} \right) + \frac{r_2}{r_0} \left( 1 + \frac{\lambda_2}{m_2} + 2 \frac{\lambda_2}{m_1} \right)
\]

where \( m_i \) is the mass associated with \( x_i \). If we would have \( m_2 = m_3 \) and \( \lambda_1 = \lambda_2 \) then we would have

\[
\frac{\tilde{r}_1}{||\tilde{r}_1||} + \frac{\tilde{r}_2}{||\tilde{r}_2||} = c \left( \frac{r_1}{||r_1||} + \frac{r_2}{||r_2||} \right), \quad c \in \mathbb{R}
\]

and hence \( n_1 \) would not change during filtering for the equilibrium method.

Obviously for water we have \( m_2 = m_3 \). Furthermore, by assumption, both bonds have the same length and there is a solution with \( \lambda_1 = \lambda_2 \). Since \( \frac{\partial \Psi(q)}{\partial x} \) has full rank, this solution is also unique by the implicit function theorem.

3.5. The Jacobian of the corotational filter

Application within the mollified impulse method requires the Jacobian \( \Psi(q) \). In contrast to other filters in the literature, the derivative of the corotational filter is explicitly given and easy to compute. It does not require the solution of any nonlinear systems. Note, that both filter and its derivative are spatially local properties, making a parallel implementation within a domain decomposition strategy fairly easy.

With notation as in (6) and (7) for a cluster of \( n \) atoms we have

\[
\Psi(q) = [1, ..., 1]^T \otimes c + (I_n \otimes RR_0^T)q_0
\]

with

\[
\begin{align*}
n_1 &= \sum_{i \in S_1} \frac{r_i}{||r_i||} \bigg/ \sum_{i \in S_1} \frac{r_i}{||r_i||} = : \frac{u}{||u||}, \\
n_2 &= (s - sT n_1 n_1) \bigg/ ||s - sT n_1 n_1|| = : \frac{v}{||v||}, \\
n_3 &= (n_1 \times n_2) / ||n_1 \times n_2|| = : \frac{w}{||w||}.
\end{align*}
\]

We choose the reference cluster \( q_0 = [q_0^1, q_0^2, ...., q_0^{3n}]^T \in \mathbb{R}^{3n} \) such that \( R_0 \) is the identity \( I_3 \). The derivative then simplifies to:

\[
\Psi_q(q) = [1, ..., 1]^T \otimes \frac{\partial}{\partial q} \left( q_0^{\lambda_1/q} + q_0^{\lambda_2/q} + q_0^{\lambda_3/q} \right) + (I_3 - n_1 n_1^T) \frac{\partial}{\partial q} \left( q_0^{\lambda_1/q} + q_0^{\lambda_2/q} + q_0^{\lambda_3/q} \right)
\]

\[
\begin{align*}
\frac{\partial n_1}{\partial q} &= \frac{1}{||u||} \left( I_3 - n_1 n_1^T \right) \left( \sum_{i \in S_1} \frac{1}{||r_i||} \left( I_3 - r_i r_i^T / ||r_i||^2 \right) \frac{\partial r_i}{\partial q} \right), \\
\frac{\partial n_2}{\partial q} &= \frac{1}{||v||} \left( I_3 - n_2 n_2^T \right) \left[ \left( I_3 - n_1 n_1^T \right) \frac{\partial s}{\partial q} - (s^T n_1 I_3 + n_1 s^T) \frac{\partial n_1}{\partial q} \right], \\
\frac{\partial n_3}{\partial q} &= \frac{1}{||w||} \left( I_3 - n_3 n_3^T \right) \left( \frac{\partial n_1}{\partial q} \times n_2 + n_1 \times \frac{\partial n_2}{\partial q} \right).
\end{align*}
\]

The derivative of the center of mass in (12) is given by
Algorithm 2: Corotational Filter Algorithm.

Input: position \( q \)
Output: filtered position \( \Psi(q) \), Jacobian \( \Psi_q(q) \)
Precompute:
- generate a cluster decomposition
- compute a reference configuration for each cluster
Filter:
for each cluster do
- estimate local rotation via (7)
- compute filtered position via (6)
- compute Jacobian via (12)
-assemble \( \Psi(q), \Psi_q(q) \) from individual clusters

\[
\frac{\partial c}{\partial q} = \frac{1}{m_{\text{all}}} [m_1, m_2, \ldots, m_n] \otimes I_3, \quad \text{with} \quad m_{\text{all}} = \sum_{i=1}^{n} m_i. \tag{14}
\]

Note, that in (13) we have \( \partial \ell / \partial q = (e_1 - e_1) \otimes I_3 \) where \( e_k \) denotes the \( k \)-th canonical unit vector. Furthermore,

\[
\frac{\partial s}{\partial q} = \sum_{e \in S_1} \frac{1}{\| r_i \|} \left( I_3 - \frac{r_i}{\| r_i \|} \otimes \frac{r_i}{\| r_i \|} \right) \frac{\partial r_i}{\partial q} \tag{15}
\]

and a similar expression but for \( S_2 \) already appears in \( \partial n_1 / \partial q \). Thus, if there is an overlap between \( S_1 \) and \( S_2 \), we can reuse part of that calculation. In the last row of (13) we slightly abuse the notation, and the cross products are meant column-wise. The derivative looks quite nasty, but one is reminded, that most quantities are constructed from simple vectors of size 3, which already have been computed with the filter. Also, from an implementational viewpoint, the matrix \( \Psi_q \) does not need to be constructed explicitly, instead only its action on the force vector is required.

For the cases \( n = 2, 3 \) the derivative can be expressed much shorter. In the setting of (8), for a cluster with two atoms the derivative reads

\[
\Psi_q(q) = [1, 1]^T \otimes \frac{\partial c}{\partial q} + \left( \begin{array}{c}
\partial_1 n_1 \\
\partial_2 n_1
\end{array} \right),
\]

\[
\frac{\partial n_1}{\partial q} = \frac{1}{\| x_2 - x_1 \|} (I_3 - n_1 n_1^T) [-I_3, I_3]. \tag{16}
\]

The derivative for a cluster with three atoms as obtained from (10) is

\[
\Psi_q(q) = [1, 1, 1]^T \otimes \frac{\partial c}{\partial q} + \left( \begin{array}{c}
\partial_1 n_1 \\
\partial_2 n_1 \\
\partial_3 n_1
\end{array} \right),
\]

\[
\frac{\partial n_1}{\partial q} = \frac{1}{\| x_2 - x_1 \|} (I_3 - n_1 n_1^T) [-I_3, I_3]. \tag{17}
\]

3.6. The full algorithm

Algorithm 2 summarizes the necessary steps as explained in the previous sections. First, a cluster decomposition of the molecular structure has to be determined and the individual reference configurations for each cluster need to be computed. Since both the cluster decomposition and the reference configurations only need to be computed once for every structure, they can be precomputed.

Now, whenever we need the filtered position and its derivative, for each cluster, we estimate the local rotation. By rotating and shifting the corresponding reference configuration we obtain the filtered position for each cluster. This way internal deformations, which are not represented by overall rotation or center of mass movement of a cluster, are discarded. The filtered position of the entire molecular system then is assembled by simply taking the filtered values from each individual cluster. In a similar manner, we proceed with the Jacobian of the filter.

4. Test problems

We test the corotational filter’s stability and performance on four different problems. First, in problem (A) we choose the most common natural form of ice, that is ice Ih. Ice is very sensitive to changes and its structure is easily destroyed by a misperforming integrator. This makes it an outstanding test problem. As second problem (B) we choose a small peptide solvated in explicit water. It has a huge variety of vibrations from bonds, angles and torsions. It also shows that it is possible to find a suitable choice of clusters. In simulation (C) we investigate ice–ice friction, and in simulation (D) folding phenomena of a protein are investigated. In contrast to the more academic test problems (A) and (B), the dynamics of
problems (C) and (D) are much more complex and interesting. They also indicate that the proposed filtering technique is reliable and stable in application.

These models pose excellent testing problems since expensive long-range forces have to be computed every time-step. However, high oscillations due to bond stretching and angle bending limit the step size of the (unfiltered) impulse method to below 4 fs. They also show that the new filter can easily handle different potential structures. In fact, the filter will work with all flexible molecules regardless of their potential structures. We only have to find a cluster decomposition and identify suitable equilibrium positions for the filter to work.

4.1. Problem (A) for ice Ih

Ice Ih has a hexagonal lattice structure (see [45] for lattice parameters). We set up a cubic simulation box with periodic boundaries containing 2880 molecules (Fig. 4). The simulation size is sufficiently large to get reliable results and obtain a computationally challenging problem (parallel efficiency is a major concern).

There are many atomistic models for water, each with different properties and each is optimized for some special purpose. For a good overview see, for example, [46]. We conduct our investigation by choosing two different flexible models. First we choose the SPC/F model [47], which is based on the rigid three site SPC model and uses harmonic potentials in bonds and angles to model intra-molecular forces. The second model we investigate is TIP4P/2005f [48], a flexible model based on the TIPNP topology. It additionally uses a massless fourth site, \( \Omega \), with a negative charge. Bond interactions are modeled using a Morse potential. The angle potential is again harmonic.

For water, the cluster decomposition is natural: Each molecule is a cluster of size three. The reference configurations then is simply a water molecule with equilibrium bond length and angle as specified in the corresponding water model.

4.2. Problem (B) for a small peptide

We conduct simulations of a small 5-mer peptide solvated in water (Fig. 5). Initial positions and structure can be found in the LAMMPS example section. The peptide has 84 atoms and is solvated in 640 water molecules (total of 2004 atoms). Potentials are modeled according to the CHARMM forcefield [49]. For the cluster decomposition each water molecule forms its own cluster. The decomposition of the peptide is chosen such that all bonds containing a hydrogen and carbonyl groups (C=O) are filtered. Then, the peptide decomposes in clusters mostly of size two and three. However, there are three terminal CH3 groups which form clusters of size four, and four single atoms remain, that we choose not to filter. Of the four unfiltered atoms, three are carbon atoms with slow onwards connections in the aromatic rings, and the last one is a sulfur atom connected with two slow bonds to a CH3 and CH2 group. Note, that those four atoms can be easily included in a neighboring cluster, however, as the numerical results indicate, due to their slow bonds, this is not necessary. In total, the decomposition has 25 clusters of size 2, 646 of size 3 and 3 of size 4.

4.3. Problem (C) for ice–ice friction

The third test problem investigates friction processes between two layers of ice Ih. It has been known for a long time that the creation of a liquid layer in between the ice interfaces plays a significant role. The investigation of the dynamics and mechanisms in this liquid layer is an active area of research from both theoretical/computational and experimental groups [50–54].

Similar to [52], we create two slabs of ice separated by some vacuum in z-direction and bring them slowly into contact. The motion of the two slabs is controlled by harmonically restricting the movement of a layer of water molecules in each slab (indicated by yellow coloring in Fig. 6). This allows us to control friction velocity and contact pressure.

Each slab contains 2880 molecules, in total we have to keep track of 17280 atoms. We choose the TIP4P/2005f model for this investigation. Computationally this is also a very challenging problem, especially if we are interested in slow friction velocities.
Fig. 5. A small peptide, depicted in its ball-stick representation. For simulation (B) it is solvated in 640 water molecules.

Fig. 6. Ice Friction Model for Problem (C) as defined in Section 4.3: two slabs of ice will be brought into contact by applying normal and friction force by harmonically restricting the molecules in yellow indicated areas. Red lines indicate periodic box boundaries. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.4. Problem (D) for bovine pancreatic trypsin inhibitor

Bovine pancreatic trypsin inhibitor (BPTI) is a small protein which can be found in bovine lung tissue (Fig. 7). Its function is suppression of protein digestion and it acts as a competitive inhibitor. For example during heart surgery it can be used to reduce bleeding. Well studied since the 70s/80s, it exhibits complex folding pathways. Structural information was obtained from the RCSB database (ID: 4PTI [55]). BPTI has 892 atoms, and we solvate it in 3165 explicit water molecules (10387 atoms in total). Potentials are modeled according to the CHARMM forcefield [49]. For the filter decomposition, we use the simple heuristic, that bonds containing a hydrogen are fast bonds, and the remaining ones are slow. Thus, we obtain 163 clusters of size 2, 3263 of 3 and 25 of 4.

4.5. Time integration

We investigate three variations of the impulse method with different settings and the Verlet method:

**Impulse Method (IM):** The unfiltered standard impulse method uses three stages with the force splitting described in (3). This corresponds to Algorithm 1 with the trivial filter \( \Psi(q) = q \), so the force kick in the outer stage reads
5.1. Vibration reduction using the corotational filter

In order to avoid resonances in the mollified impulse method, the corotational filter is designed to eliminate fast vibrations in the position of molecules. After equilibration (A: 50 ps, NPT with 150 K and 1 bar pressure, B: 50 ps, NVT with 250 K), we extract trajectories of a selection of molecules for 10 ps (IM with 1 fs outer step size) and use VMDs IR analysis tool to compute vibrational spectra. Then we apply the corotated filter algorithm \( (6) \) directly to each time-step of the extracted trajectories and repeat the analysis. Note that differences in peak heights stem from normalizing.

In Fig. 8 we clearly see how filtering entirely removes frequencies associated with the highest frequencies. There are tiny artifacts remaining though: for example asymmetric bond stretching in the SPC/F model creates a contribution which is barely visible and has drastically reduced amplitude (by more than two orders of magnitude). We do not believe this to be a problem, since the slow force contributions exciting the asymmetric bond stretching modes are filtered out by the derivative. The filter does not seem to influence any other frequencies, the peaks are in exactly the same positions.

Note that in both CIM and EIM, the filtered positions are only used to evaluate the slow forces, hence both integrators (in contrast to SHAKE) still resolve fast bond and angle vibrations.

5.1.2. Energy drift

Geometric integrators \([58]\) have the ability to approximately preserve energy. Hence the energy drift with respect to step size is a property of interest. After equilibration (A: 50 ps, NPT with 150 K and 1 bar pressure, B: 50 ps, NVT with 250 K)
we run 50 ps NVE simulations with IM, CIM and EIM. Since the energy is slightly oscillating we choose to average over the first and last 500 fs to cancel out oscillations and approximate drift. For filtered versions we compute the energy at the filtered slow potential $U_{\text{slow}}(\Psi(q))$.

Fig. 9 indicates that for ice simulations with the SPC/F model, the impulse method can easily handle outer step sizes as big as 4 fs without significant drift. Resonance issues cause, as expected, a dramatic breakdown at step sizes near 4.5 fs. It is a 2:1 resonance failure attributed to the fast bond oscillations. Due to a clear separation of frequency bands, it is a sharp spike. With larger time-steps we can get stable results again. Note that at time-steps between 4.4 and 4.6 fs the simulation actually ‘explodes’ and therefore aborts half way through. Furthermore, we can also detect nonlinear resonances other than the critical 2:1 breakdown. Following [14] we expect 3:1 and 4:1 instabilities, which happen not at half the period, but at a third or a fourth of the associated mode. The peaks in energy drift at around 3.7 fs and 4.2 fs might stem from such higher order resonances, resulting from the angular motion. The situation is found to be similar for the TIP4P/2005f. For this case, the 2:1 instabilities occur near 5 fs (see Fig. 10). The peak at 3.4 fs is clearly a 3:1 instability, and the smaller peaks at 4.1 fs and 4.6 fs could be originated in higher order resonances, resulting from the angular motion. Due to the multitude of different frequencies in the motion of the peptide in (B), it is much harder to assign resonances. In Fig. 11 it is clear, though, that the first dramatic breakdown happens around 5 fs. Other resonances are more smeared and difficult to identify, but clearly visible.
Severe resonance instabilities as observed with the unfiltered IM do not seem to appear in the filtered integrators at all. In fact, step sizes as large as 10 fs (or around 9 fs for the TIP4P model) can be used if we are willing to accept a drift of less than 1% on a 50 ps run with 2880 molecules. This should not be a problem if we are running very long simulations within a NVT/NPT ensemble with a very weak coupling to a target temperature and/or pressure. As for the impulse method, there are artifacts of nonlinear resonances visible for both CIM and EIM. We suspect that they originate from modes associated with angle vibrations at around 20.8 fs (SPC/F) and 20.2 fs (TIP4P/2005f). The peaks around 10 fs are then 2:1 instabilities, those just below 7 fs are 3:1 resonances, and the remaining peaks are probably a mix of 3:1 and 4:1 resonances from different modes. These results clearly show the improved performance of the proposed filters against existing integrators.

5.1.3. Energy distribution
Filtering long-range force contributions successfully removes resonance instabilities which originate from the fastest vibrations. However, filtering interferes with the slow energy exchange between fast and slow modes e.g. as shown theoretically for the Fermi–Pasta–Ulam problem in [58], chap. XIII and references therein. We compare differences in the potential energy of $U_{\text{fast}}$, $U_{\text{medium}}$ and $U_{\text{slow}}$. After 250 ps equilibration (A: NPT at 180 K and 1 bar, B: NPT at 250 K and 1 bar) with each method we collect these values for 10 ps. Fig. 12 plots the relative difference $\Delta U = |U_{\text{CIM}} - U_{\text{IM}}|/|U_{\text{IM}}|$, where we use the CIM with an 8 fs outer time-step and the IM as a reference solution with outer time-step of 1 fs.

These plots indicate that there is a slight difference, primarily in bonded interactions. This is somewhat expected since we mollify the long-range force contributions exactly in the direction of these interactions. We further investigate the impact of filtering and analyze the radial distribution functions.

5.1.4. Radial distribution function
A radial distribution function (RDF) is a structural fingerprint. Here we investigate the RDF between oxygen atoms obtained after 250 ps equilibration in NPT at 180 K and 1 bar with each method.

In Fig. 13 we compare RDFs obtained with a large step CIM at 8 fs to a reference solution using the IM with a small outer step size of 1 fs. Bond and pair forces are computed every 0.25 fs. The data is collected and averaged over 10 ps. The results are almost indistinguishable, and we have peaks at exactly the same positions. There are small differences in height.
for the two models, with a slight emphasis on the low energy positions for the proposed filter. However, these discrepancies are so little that they can be safely neglected in most applications. Investigating both the energy distribution and the RDFs, we have no indication that filtering overly manipulates the energy flow.

5.1.5. Computational cost

Computational and implementational efficiency are the main motivations behind design and construction of our new filter. It should be noted that timing information is not only extremely hardware sensitive, but also strongly depends on problem size and implementation. We give results for problem (A), and the other problems are expected to behave in a very similar way.

The cost of the long-range electrostatic forces plays a major role; the more expensive they are the more we can gain by using new filters developed here. As our previous results indicate, the CIM can easily use an outer step size which is twice as large. This means it only evaluates the slow force half as often compared to a regular IM with a 4 fs step size. In case the slow force is computationally cheap, we can shift the cutoff radius so that a bigger portion of the force is computed in the outer stage. Also, using a GPU can significantly decrease the pair time [59].

Timing results in Fig. 14 are obtained from short 50 ps simulations after equilibration. The results are averaged over ten runs each. They are obtained on the bwUniCluster (see Acknowledgments) using 128 cores (8 nodes with 16 cores each). Categories are as follows: 'Bond' contains the computation of intra-molecular forces, 'Pair' the time spent on evaluating all forces between molecules within a certain cutoff. 'Long-Range' is the time used calculating the long-range electrostatic contributions. 'Comm' represents all time due to communication between processors. 'Filter' signifies all computations assigned to the pre- and post-filtering. Remaining time is summarized in 'Other'.

The improvement of CIM over a regular Verlet method is about eight times less time spent on long-range computations. Compared to a standard IM, the CIM saves half the computational cost of long-range interactions. The additional computation of the filter itself is very cheap and costs less than 0.1% overall CPU time. Therefore, in Fig. 14, it is represented by a very narrow black line as the CIM and 'Other'. Obviously, we also timed the EIM, but do not give separate results. Qualitatively, it is the same as the CIM, and only differs in the filtering time, where it tends to be 50% more expensive. However, the overall variation between the ten runs each is of a similar magnitude.

In general at least 15–20% overall speed up can be expected by using the filtered CIM compared to a standard IM; and >50% gains over the traditional Verlet algorithms can be accomplished.

5.2. Simulation of ice-friction in problem (C)

The following simulations are computed with the CIM, using an outer step size of 8 fs. The gain in computational efficiency makes it possible to investigate slower friction velocities and compute longer trajectories. First, we investigate ice friction between two slabs of ice Ih with the TIP4P/2005f model.

5.2.1. Simulation protocol

For equilibration we perform 800 ps in NPT targeting 1.0 bar and slowly heating up to 180 K. After equilibration, the simulation runs in NVE except for the constrained layers, where temperature is controlled by a Nose–Hoover thermostat. Throughout the process, friction causes heat to be generated at the interface. Hence with temperature control of constrained layers, we model heat dissipation out of the domain (Fig. 15).
Then each slab is set to an opposing velocity and we start applying a normal force to the constrained layers of molecules. We run for another 8.4 ns where we slowly increase the target temperature of the constrained layers to 210 K. Afterwards, we keep the temperature of the constrained layers fixed at 210 K and run for another 8 ns.

The friction velocity is chosen ranging from 0.01 m/s to 10 m/s. For the imposed normal force we select 10 and 100 kcal/mol-Å, half of each is applied to the top layer, the other half is applied in the opposite direction to the bottom layer.

5.2.2. Friction force
Previous experimental studies [53,54] found that friction forces are rather sensitive to a variety of parameters such as velocity, normal force and temperature. Unfortunately, it takes a long simulation to get from static friction to kinetic friction. Fig. 16 shows one simulation run, plotting the average friction force over time (at 100 kcal/mol-Å normal force and 5 m/s velocity). We can see that as soon as the slabs come into contact, due to static friction, there is a high peak in friction. The strong connections subsequently begin to break down, and we observe sliding between the layers. The force of friction drops quickly in the beginning, but much slower afterwards. It takes a long time to converge.

Results obtained by varying velocity and normal force are plotted in Fig. 17. Force values are averaged over the last 4 ns of the 17.2 ns runs. They seem to indicate two relations: first, the higher the velocity, the larger the friction, and second, higher normal force, as compared to lower normal force, is correlated with significantly lower friction.

5.2.3. Temperature distribution and liquid layer
In the simulation, friction generates heat and temperature increases around the interface. A liquid layer is formed at the interface which has an increased density. In [60], an order parameter as indicator for the structure of ice Ih is introduced. It would be 1 in a perfect tetrahedral network, or 0 in an ideal gas. As noted in [52], in practice we can expect it to be around 0.95 in ice, and around 0.5 to 0.85 in liquid water.
Fig. 17. Average final friction force for 10 and 100 kcal/mol-Å normal force and different friction velocities, friction force [kcal/mol-Å] vs. friction velocity [m/s] for Problem (C).

Fig. 18. The thickness of melted layer at 10 kcal/mol-Å for different friction velocities in Problem (C), order/temperature [K] vs. spatial [Å].

Fig. 19. BPTI in Problem (D): RMSD evolution over time [ns] for residues 5 to 54, backbone atoms only.

In Fig. 18, we plot order parameter and temperature along the z-axis for three different friction velocities. As expected, it indicates that a liquid layer forms in the middle, and also that the temperature is significantly higher at the friction interface than in the exterior of the domain.

5.3. Simulation of BPTI in problem (D)

We equilibrate BPTI to 300 K at 1 bar (NPT) for 210 ps using a regular IM with an outer step size of 4 fs. Afterwards, we switch to the CIM with an outer step size of 8 fs and run at 300 K (NVT) for roughly 10 million steps, achieving a simulation time of 80 ns. This is still a very short trajectory, however, and major transitions happen on larger time scales [61,62]. And yet, we could still observe small changes using the root-mean-square deviation (RMSD) as a reaction coordinate.

5.3.1. RMSD indicates two configurations

The RMSD is plotted in Fig. 19 for backbone atoms of residues 5 to 54 versus time. After an initial equilibration, it seems to settle down just above 1 Å until around 45 ns. It then quickly moves to just below 2 Å to find equilibrium again. In Fig. 20, we plot the initial structure and snapshots after 39 ns and 80 ns. They indicate the origin of this behavior: the lower left part of the backbone partially rotates. This is also supported by Fig. 21, which shows the RMSD for each residue...
Fig. 20. BPTI snapshots from Problem (D) after approx. 0 ns, 39 ns and 80 ns simulation time.

Fig. 21. Backbone RMSD per residue averaged over 800 ps at 39 ns and 80 ns simulation time for BPTI in Problem (D).

during those time frames. After 39 ns, the structure is still relatively unchanged, as compared to the original. It is no surprise that the end tail clearly moves. There is an additional small peak at id 38 which indicates changes near the middle of the protein chain. Towards 80 ns, the start tail also unfolds. More interestingly, residues 10 to 15 flip and move away from the center.

6. Conclusions

For biomolecular simulations with flexible bonds and angles, our new filter based on corotation successfully avoids resonance instabilities which inherently limit the impulse method. With our method, it is possible to use step sizes which are twice as large as those which can be used in the standard impulse method. It is stable, reliable and computationally cheap. The filter is most effective in simulations with expensive long-range calculations where we can obtain approximately 20% speed-up at no visible loss of accuracy. Its major advantage, compared to other filters, is its ease in implementation and its versatility.

Acknowledgements

We thank the reviewer for the thorough review and highly appreciate the comments and suggestions, which significantly contributed to improving the quality of this paper. The authors are grateful to Volker Grimm for valuable comments and suggestions, and to Brianna Robinson for proofreading. This research was partially supported by a scholarship from the DAAD (German Academic Exchange Service). Most simulations were performed on the computational resource bwUniCluster funded by the Ministry of Science, Research and the Arts Baden-Württemberg and the Universities of the State of Baden-Württemberg, Germany, within the framework program bwHPC. Some computations were also performed on the gpc supercomputer at the SciNet HPC Consortium. SciNet [63] is funded by: the Canada Foundation for Innovation under the auspices of Compute Canada; the Government of Ontario; Ontario Research Fund – Research Excellence; and the University of Toronto.

References

Some parameter scales for the simulations of biomolecules.


http://dx.doi.org/10.1137/S1064827596313851.


