Computer Physics Communications 320 (2026) 110003

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Computer Programs in Physics

Dofulator: A tool for calculating degrees of freedom of atoms in molecules with geometry constraints

Stephen Sanderson [a,*], Sobin Alosious [a], Debra J. Searles [a,b,c,*]

[a] Australian Institute for Bioengineering and Nanotechnology, The University of Queensland, 4072, St. Lucia, QLD, Australia
[b] School of Chemistry and Molecular Biosciences, The University of Queensland, 4072, St. Lucia, QLD, Australia
[c] ARC Centre of Excellence for the Green Electrochemical Transformation of Carbon Dioxide, The University of Queensland, 4072, St. Lucia, QLD, Australia

ARTICLE INFO

Editor: Dr. J Ballantyne

Keywords:
Degrees of freedom
Molecular dynamics
Geometry constraints
Temperature
Linked rigid bodies

ABSTRACT

Recently, we proposed a method for calculating per-atom and per-direction degrees of freedom (DoF) in the presence of geometric constraints, enabling fine-grained local kinetic temperature calculations. Here, we discuss relevant implementation details for various constraint geometries, including those which feature kinematic loops (e.g. benzene with rigid bond lengths). Furthermore, by analyzing the effects of deformation of semi-rigid molecules on the DoF of each constituent atom, we gain insight into conditions under which atomic DoF may vary significantly during a simulation. This provides some guidance towards cases where local DoF should be calculated dynamically to obtain reliable local temperature measurements, and cases where using the atomic DoF of the equilibrium geometry as a constant throughout the simulation would be sufficient. We have implemented the presented algorithms in an open-source C library, `dofulator`, which can be used on its own or through a Python interface that includes compatibility with the popular MDAnalysis package.

PROGRAM SUMMARY

Program Title: `dofulator`
CPC Library link to program files: https://doi.org/10.17632/8z6vhd3vw3.1
Developer's repository link: https://github.com/CTCMS-UQ/dofulator
Licensing provisions: MPL-2.0
Programming language: C, Python
Nature of problem: In molecular simulations with geometry constraints, determining the degrees of freedom (DoF) associated with a local kinetic temperature measurement can become non-trivial when constraints include atoms both inside and outside the local subset of interest [1]. The (fractional) DoF of atoms in a rigid body depends on their masses and the molecular geometry. If constraints do not form a rigid body, but instead a semi-rigid fragment, then the partitioning of atomic DoF can vary as the fragment deforms. Furthermore, if directional kinetic temperatures are required, DoF along each direction must be determined, which additionally depend on the orientation of the rigid body or semi-rigid fragment.
Solution method: Atomic DoF can be calculated by the relative contribution of each atom to the inertia of each mode of motion [1]. This software allows rigid bodies and semi-rigid fragments to be defined, from which a plan is constructed for determining said modes and contributions. Once constructed, a plan can be applied repeatedly to calculate atomic DoF on required frames of a molecular dynamics trajectory.
Additional comments including restrictions and unusual features: The core `dofulator` library is provided as a C API, depending only on a BLAS and LAPACK implementation and suitable for direct integration with a molecular dynamics engine (possibly with some modifications). For convenience, a thin Python wrapper is also provided, and this is used to provide plugins to MDAnalysis [2,3] which gives a simple way to read molecular dynamics trajectories and define local selections of atoms for post-processing of local DoF and temperatures.

* Corresponding authors.
E-mail addresses: stephen.sanderson@uq.edu.au (S. Sanderson), salosiou@nd.edu (S. Alosious), d.bernhardt@uq.edu.au (D.J. Searles).

https://doi.org/10.1016/j.cpc.2025.110003
Received 30 January 2025; Received in revised form 14 September 2025; Accepted 12 December 2025
Available online 19 December 2025
0010-4655/© 2025 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

References

[1] S. Sanderson, S. R. Tee, D. J. Searles, Local temperature measurement in molecular dynamics simulations with rigid constraints, Journal of Chemical Theory and Computation 20 (23) (2024) 10615–10624. doi: 10.1021/acs.jctc.4c00957

[2] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, O. Beckstein, MDAnalysis: A toolkit for the analysis of molecular dynamics simulations, Journal of Computational Chemistry 32 (10) (2011) 2319–2327. doi: 10.1002/jcc.21787

[3] R. Gowers, M. Linke, J. Barnoud, T. Reddy, M. Melo, S. Seyler, J. Domański, D. Dotson, S. Buchoux, I. Kenney, O. Beckstein, MDAnalysis: A Python package for the rapid analysis of molecular dynamics simulations, in: Proceedings of the 15th Python in Science Conference, SciPy, SciPy, 2016. doi: 10.25080/majora-629e541a-00e.

Molecular simulations are often carried out in cases where a temperature field is imposed or needs to be determined. An example of a situation where this is important is in a system with a phase boundary (interface) where interfacial thermal resistance can lead to a sharp change in the temperature. However, the presence of geometric constraints, which are commonly used to reduce the calculation time, makes accurate local temperature measurement non-trivial. Recent work has shown the importance of accurately determining the degrees of freedom (DoF) associated with local temperature measurements in cases where geometry constraints include particles both inside and outside the local selection [1–3]. While it is relatively simple to calculate the per-particle DoF of particles which form a rigid body, dynamic updating based on orientation is required for further partitioning into Cartesian directions (of interest, for example, as a check for equilibrium, or when directional driving or anisotropic structure lead to a non-equilibrium state where different directions have different kinetic temperatures). Furthermore, semi-rigid fragments, defined as a set of particles connected by at least one geometric constraint, pose a greater challenge due to the DoF partitioning also depending on the conformation.

Previous work [3] has provided a general theory for determining DoF. This paper presents a practical algorithm for applying this theory, as well as examples demonstrating its application. We have made this DoF calculator algorithm available as open source software, titled `dofulator`, which provides both C and Python library interfaces, along with plugins to the popular molecular dynamics analysis tool MDAnalysis [4,5]. Thus, it is now possible to take trajectories created by a range of different molecular dynamics packages and obtain DoF and temperature information for any subset of atoms (e.g. those in a spatial region, or particular atoms in a molecular species). The calculator can provide both per-atom and per-direction DoF, and supports both rigid bodies and semi-rigid fragments with arbitrary sets of distance constraints, including those which form kinematic loops (e.g. ring structures with all bond lengths fixed).

Theory for imposing other geometric constraints such as bond angle or dihedral constraints is also discussed in detail. Note that while pure bond angle or dihedral constraints which do not constrain bond lengths are possible to support (as described in the following section), in practice, angle constraints are typically imposed along with rigid bond lengths, and dihedral constraints along with both rigid bond lengths and angles. Hence, only distance constraints are necessary to produce identical results for the vast majority of use cases, and support in `dofulator` is therefore limited to such cases in the present version of the software.

Furthermore, it has been suggested that use of average per-particle DoF at equilibrium may provide sufficient accuracy without the additional cost of re-calculating the DoF for each conformation [2,3]. We examine this approximation in comparison to the more accurate treatment enabled by `dofulator`, considering both equilibrium and non-equilibrium systems and a variety of constraint geometries.

1. Theory

The degrees of freedom (DoF) associated with a set of point particles influenced by rigid constraints can be calculated based on their fractional contribution to the inertia of each mode of motion [3]. Consider an arbitrary rigid body consisting of N particles with positions $\boldsymbol{r}_i$, $i \in \{1, \dots, N\}$, which is free to undergo translational and rotational motion (e.g. a rigid molecule). Choosing particle 1 as the origin, the motion of all other particles can be described using a reduced coordinate system which consists of the position of the origin particle, and rotation about the coordinate axes (or in a plane). The velocity of each particle in lab-frame Cartesian coordinates, $\boldsymbol{v} = [\boldsymbol{v}_1; \dots; \boldsymbol{v}_N]$, can hence be obtained in terms of the velocity in the reduced coordinate system, $\dot{\boldsymbol{\theta}} = [\boldsymbol{v}_1; \boldsymbol{\omega}_1]$, as

$$\boldsymbol{v} \equiv \begin{bmatrix} \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_N \end{bmatrix} = \boldsymbol{J} \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{\omega}_1 \end{bmatrix} \equiv \boldsymbol{J}\dot{\boldsymbol{\theta}}. \tag{1}$$

where the $3N \times 6$ Jacobian is given by,

$$\boldsymbol{J} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{1} & -[\boldsymbol{r}_2 - \boldsymbol{r}_1]_\times \\ \vdots & \vdots \\ \mathbf{1} & -[\boldsymbol{r}_N - \boldsymbol{r}_1]_\times \end{bmatrix}. \tag{2}$$

Here, $\mathbf{1}$ is the unit matrix of size 3, and the $\times$ subscript indicates construction of the matrix which applies the cross product,

$$[\boldsymbol{r}]_\times \equiv \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}. \tag{3}$$

That is, $[\boldsymbol{r}]_\times \boldsymbol{x} \equiv \boldsymbol{r} \times \boldsymbol{x}$. Hence, the first block column of the Jacobian matrix corresponds to translational motion, and the second to rotation.

The kinetic energy, K, can be expressed in terms of the reduced coordinate velocities and the associated inertia tensor as

$$2K = \dot{\boldsymbol{\theta}}^T \boldsymbol{\mathcal{I}} \dot{\boldsymbol{\theta}}. \tag{4}$$

It can also be expressed in terms of lab-frame velocities as

$$2K = \boldsymbol{v}^T \boldsymbol{M} \boldsymbol{v}, \tag{5}$$

where $\boldsymbol{M}$ is the diagonal mass matrix given by

$$\boldsymbol{M} = \begin{bmatrix} m_1\mathbf{1} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & m_N\mathbf{1} \end{bmatrix}. \tag{6}$$

Substituting Eq. (1) for $\boldsymbol{v}$ gives

$$2K = \dot{\boldsymbol{\theta}}^T \boldsymbol{J}^T \boldsymbol{M} \boldsymbol{J} \dot{\boldsymbol{\theta}}, \tag{7}$$

and hence the inertia tensor in the reduced coordinate system must be $\boldsymbol{\mathcal{I}} = \boldsymbol{J}^T \boldsymbol{M} \boldsymbol{J}$. This also makes it clear that the contribution to $\boldsymbol{\mathcal{I}}$, and hence K, from a given particle can be obtained by setting the mass of

all other particles to 0 in $\boldsymbol{M}$. Furthermore, noting that the diagonal elements of each $m_i\mathbf{1}$ in $\boldsymbol{M}$ are associated with motion in the three lab-frame directions, the contribution to $\boldsymbol{\mathcal{I}}$ due to motion in a particular direction can be similarly obtained by zeroing the elements in $\boldsymbol{M}$ corresponding to the other directions.

For local temperature measurements where the modes of motion are independent, the equipartition theorem applies and states that, at equilibrium, each quadratic degree of freedom will have an average kinetic energy of $\frac{1}{2}k_BT$. However, in the reduced coordinate system considered here this is not the case in general, since the modes are not independent. The independent modes, or normal modes, which do each have $\frac{1}{2}k_BT$ average kinetic energy [6], can be obtained by diagonalizing $\boldsymbol{\mathcal{I}}$ via eigendecomposition. The eigenvectors, $\boldsymbol{q}_\beta$, form the columns of the matrix $\boldsymbol{Q}$ such that $\boldsymbol{Q\Lambda Q}^T = \boldsymbol{\mathcal{I}}$, where $\boldsymbol{\Lambda}$ is the diagonal matrix of the corresponding eigenvalues, λ_β. β indexes the eigenvalues and eigenvectors, and hence each non-zero λ_β represents a normal mode. Noting that $\boldsymbol{QQ}^T = \mathbf{1}_n$, where $\mathbf{1}_n$ is the unit matrix of size n (the number of reduced coordinates; 6 in the case of a rigid body), this allows a transform to the expression for the kinetic energy,

$$2K = \dot{\boldsymbol{\theta}}^T\boldsymbol{QQ}^T\boldsymbol{\mathcal{I}}\boldsymbol{QQ}^T\dot{\boldsymbol{\theta}}, \tag{8}$$

and hence it is clear that the modal velocities are given by $\boldsymbol{Q}^T\dot{\boldsymbol{\theta}}$, and the modal inertia by $\boldsymbol{Q}^T\boldsymbol{\mathcal{I}}\boldsymbol{Q} = \boldsymbol{\Lambda}$.

It is convenient here, for the sake of isolating particular particles and particular lab-frame directions, to define a new transform, $\boldsymbol{R} \equiv \boldsymbol{M}^{\frac{1}{2}}\boldsymbol{JQ}$, noting that $\boldsymbol{\Lambda} = \boldsymbol{R}^T\boldsymbol{R}$. A particular particle, i, and direction, α, can then be selected with the $3N \times 3N$ switch matrix, $\boldsymbol{S}_{i\alpha}$, which is zero except for a single 1 as the diagonal element corresponding to $i\alpha$, giving its inertia contribution as

$$\boldsymbol{\Lambda}_{i\alpha} \equiv \boldsymbol{R}^T\boldsymbol{S}_{i\alpha}\boldsymbol{R}. \tag{9}$$

Note that by definition,

$$\boldsymbol{\Lambda} = \sum_i^N \sum_{\alpha\in\{x,y,z\}} \boldsymbol{\Lambda}_{i\alpha} \tag{10}$$

$$= \sum_i^N \boldsymbol{R}^T\left[\sum_{\alpha\in\{x,y,z\}} \boldsymbol{S}_{i\alpha}\right]\boldsymbol{R} \tag{11}$$

$$\equiv \sum_i^N \boldsymbol{R}^T\boldsymbol{S}_i\boldsymbol{R}, \tag{12}$$

where $\boldsymbol{S}_i$ is the diagonal switch matrix which gives the total DoF of atom i. Hence, as the DoF of a particular atom in a particular direction, $d_{i\alpha}$, is given by the sum of its fractional contributions to the inertia of each mode it participates in [3], this can be obtained by

$$d_{i\alpha} = \sum_{\beta\in\mathcal{B}} \frac{\hat{\boldsymbol{e}}_\beta^T\boldsymbol{\Lambda}_{i\alpha}\hat{\boldsymbol{e}}_\beta}{\lambda_\beta} \tag{13}$$

where $\hat{\boldsymbol{e}}_\beta$ is the unit vector in the β direction of the $\mathcal{B}$ normal modes (i.e., $\hat{\boldsymbol{e}}_\beta^T\boldsymbol{\Lambda}_{i\alpha}\hat{\boldsymbol{e}}_\beta$ selects the β,β diagonal element of $\boldsymbol{\Lambda}_{i\alpha}$).

The directional DoF can additionally be projected along an arbitrary unit vector, $\hat{\boldsymbol{e}}_\gamma$, through a modified switch matrix,

$$\boldsymbol{S}_{i\gamma} = \boldsymbol{S}_i\begin{bmatrix}\hat{\boldsymbol{e}}_\gamma\\ \vdots \\ \hat{\boldsymbol{e}}_\gamma\end{bmatrix}\begin{bmatrix}\hat{\boldsymbol{e}}_\gamma^T & \cdots & \hat{\boldsymbol{e}}_\gamma^T\end{bmatrix}\boldsymbol{S}_i. \tag{14}$$

This follows simply from taking the component along $\hat{\boldsymbol{e}}_\gamma$ of the atomic velocity in a given mode. Furthermore, the x, y and z DoF of a rigid body rotated by the matrix $\boldsymbol{\Theta}$ from the reference frame in which $\boldsymbol{J}$ was calculated can be obtained from projection in the original reference frame along the new basis vectors $\boldsymbol{\Theta}^{-1}\hat{\boldsymbol{e}}_x$, $\boldsymbol{\Theta}^{-1}\hat{\boldsymbol{e}}_y$, and $\boldsymbol{\Theta}^{-1}\hat{\boldsymbol{e}}_z$.

This method makes calculation of atomic and directional DoF convenient to do computationally by:

1. Constructing the mass-adjusted Jacobian matrix, $\boldsymbol{J}_m \equiv \boldsymbol{M}^{\frac{1}{2}}\boldsymbol{J}$.
2. Left-multiplying it by its transpose to obtain the inertia tensor, $\boldsymbol{\mathcal{I}} = \boldsymbol{J}_m^T\boldsymbol{J}_m$.
3. Finding the eigenvalues and corresponding eigenvectors of $\boldsymbol{\mathcal{I}}$ to obtain $\boldsymbol{\Lambda}$ and $\boldsymbol{Q}$.
4. And calculating $\boldsymbol{R} = \boldsymbol{J}_m\boldsymbol{Q}$.

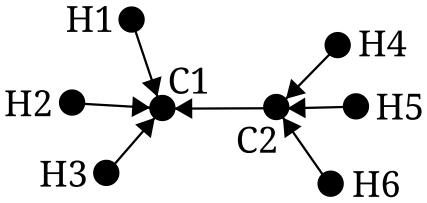


Fig. 1. Example of connectivity matrix construction from tree topology, with C1 chosen as the root atom. Arrows show the unique path from any atom to the root. The connectivity matrix can be obtained by following the arrows from a given atom back to the root and marking a **1** in the block column corresponding to each atom encountered.

At this point, it is trivial to query $\boldsymbol{R}$ and $\boldsymbol{\Lambda}$ to obtain the desired atomic and/or directional DoF as per Eq. (13), noting that

$$\begin{aligned}\hat{\boldsymbol{e}}_\beta^T\boldsymbol{\Lambda}_{i\alpha}\hat{\boldsymbol{e}}_\beta &= \hat{\boldsymbol{e}}_\beta^T\boldsymbol{R}^T\boldsymbol{S}_{i\alpha}\boldsymbol{R}\hat{\boldsymbol{e}}_\beta\\ &= \hat{\boldsymbol{e}}_\beta^T\boldsymbol{R}^T\hat{\boldsymbol{e}}_{i\alpha}\hat{\boldsymbol{e}}_{i\alpha}^T\boldsymbol{R}\hat{\boldsymbol{e}}_\beta\\ &= \left(\hat{\boldsymbol{e}}_{i\alpha}^T\boldsymbol{R}\hat{\boldsymbol{e}}_\beta\right)^2.\end{aligned}$$

That is, for DoF along the basis axes of the current reference frame, the numerator in Eq. (13) is simply the square of the value in row $i\alpha$, column β of $\boldsymbol{R}$, and hence the total DoF of atom i is easily calculated as the sum over the three basis directions.

1.1. Semi-rigid fragments

This same process can be applied to semi-rigid fragments, with the only difference being in the construction of $\boldsymbol{J}$ which we will describe in the following subsections. While rotational and translational invariance holds, deformation of the fragment may result in different DoF partitioning. That is, changes in the position of the fragment will not change the DoF partitioning, and changes in orientation simply rotate the directional DoF as for rigid bodies, but changes to internal coordinates (e.g. lengths of stretchable bonds, or angles between bonds) can have an effect.

1.1.1. Bond constraints

Consider an ethane molecule with rigid bond lengths, C1 connected to H1–H3, and C2 connected to H4–H6 (see Fig. 1). This is an example of an acyclic (open chain) fragment. The constraints have tree topology, meaning there is exactly one path of constraints between any two particles in the fragment, and therefore no kinematic loops. Choosing an arbitrary atom as the root, construction of a Jacobian for such systems can be conveniently simplified to (1) construction of a triangular connectivity matrix which defines, in each block row, the path through the tree from the atom corresponding to that block row back to the root atom, and (2) construction of a block diagonal matrix in which each diagonal element represents the Jacobian for the motion of the corresponding atom relative to its predecessor in the tree [7]. For example, with C1 chosen as the root atom, ethane with rigid bond lengths would have the Jacobian

$$\boldsymbol{J} = \begin{bmatrix}1&0&0&0&0&0&0&0\\1&1&0&0&0&0&0&0\\1&0&1&0&0&0&0&0\\1&0&0&1&0&0&0&0\\1&0&0&0&1&0&0&0\\1&0&0&0&1&1&0&0\\1&0&0&0&1&0&1&0\\1&0&0&0&1&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0\\0&-[\boldsymbol{r}_{\text{C1-H1}}]_\times&0\\0&0&-[\boldsymbol{r}_{\text{C1-H2}}]_\times\\0&0&0\\0&0&0\\0&0&0\\0&0&0\\0&0&0\end{bmatrix} \cdots \tag{15}$$

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \dots\ -[\boldsymbol{r}_{\text{C1-H3}}]_\times & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -[\boldsymbol{r}_{\text{C1-C2}}]_\times & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -[\boldsymbol{r}_{\text{C2-H4}}]_\times & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -[\boldsymbol{r}_{\text{C2-H5}}]_\times & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -[\boldsymbol{r}_{\text{C2-H6}}]_\times \end{bmatrix}$$

where the block rows correspond to C1, H1, H2, H3, C2, H4, H5, and H6, respectively.

More generally, with a constrained bond length, the only motion available to a given particle, s, is rotational motion around its predecessor, p, as described by the matrix $-[\boldsymbol{r}_{p-s}]_\times$. Hence, for general fragments with tree topology, the block row of the Jacobian corresponding to atom s, when applied to $\dot{\boldsymbol{\theta}}$, should give the velocity of p plus the velocity of s relative to p.

1.1.2. Angle constraints

A similar process can be applied for angle constraints. Consider a water molecule with the H–O–H angle fixed, but the O–H bond lengths free to change. One choice of internal coordinate system is to use the O atom as the root, and describe the positions of the H atoms by a rotation of the molecule around O and stretching of the two bond lengths. This gives

$$\boldsymbol{J} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & -[\boldsymbol{r}_{\text{O-H1}}]_\times & \hat{\boldsymbol{r}}_{\text{O-H1}} & \mathbf{0} \\ \mathbf{1} & -[\boldsymbol{r}_{\text{O-H2}}]_\times & \mathbf{0} & \hat{\boldsymbol{r}}_{\text{O-H2}} \end{bmatrix}, \tag{16}$$

where $\hat{\boldsymbol{r}}_{\text{O-H1}}$ is the unit column vector pointing from O to H1, and thus the final two block columns are 1-wide rather than 3-wide (or alternatively could be written as 3-wide with two columns of zeroes, in which case each additional zero column will simply generate a zero eigenvalue).

In cases where an angle constraint applies to the angle between two bonds with constrained lengths, the three participating points become a rigid body, and hence the part of the Jacobian corresponding to them can be constructed as such (by simply omitting the columns corresponding to bond stretching). Alternatively, the angle constraint can be treated as a loop-closing distance constraint (e.g. between the two H atoms in the above example), as detailed below.

1.1.3. Molecules with kinematic loops

Molecules with kinematic loops (including cyclic molecules with rigid bonds forming a ring and acyclic molecules with constraints that form rings) present more of a challenge, but can be managed in a similar manner with some additional preparation. Taking inspiration from algorithms for solving linked rigid body equations of motion [8,9], we first begin by constructing the Jacobian of a spanning tree, $\boldsymbol{J}^{(t)}$. The spanning tree connects all particles in a semi-rigid fragment with a network of constraints such that there is exactly one path between any two particles (see Fig. 2). It is constructed by simply ignoring constraints which would result in kinematic loops, and otherwise following the procedures described in previous sections. For each remaining loop-closing constraint, indexed by k, define $\boldsymbol{V}_k$ as the matrix which, when applied to the vector of atomic velocities (i.e. the result of $\boldsymbol{J}^{(t)}\dot{\boldsymbol{\theta}}$), gives the velocity along loop-closing constraint k. As an example, for a bond length constraint between atoms p and s, $\boldsymbol{V}_k$ would be

$$\boldsymbol{V}_k^{\text{bond}} = \hat{\boldsymbol{r}}_{s-p}^T(\boldsymbol{C}_s - \boldsymbol{C}_p), \tag{17}$$

where the unit vector $\hat{\boldsymbol{r}}_{s-p} \equiv (\boldsymbol{r}_s - \boldsymbol{r}_p)/\left|\boldsymbol{r}_s - \boldsymbol{r}_p\right|$, and the $3 \times 3N$ selector matrices $\boldsymbol{C}_i$ are constructed from 3×3 block columns which are all zero except for the one corresponding to particle i, which contains the 3×3 identity, $\mathbf{1}$.

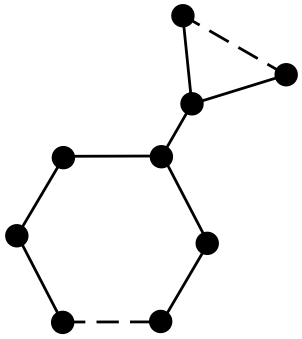


Fig. 2. Illustration of a spanning tree (solid lines) and loop-closing constraints (dashed lines). The loop-closing constraint at the top, forming a triangle, could be considered as an angle constraint. Note, the spanning tree is not unique for constraint topologies with kinematic loops.

Next, define the matrix describing the velocity along each of the N_k loop-closing constraints as

$$\boldsymbol{K} \equiv \begin{bmatrix} \boldsymbol{V}_1 \\ \vdots \\ \boldsymbol{V}_{N_k} \end{bmatrix} \boldsymbol{J}^{(t)}. \tag{18}$$

The loop-closing constraints are only satisfied by $\dot{\boldsymbol{\theta}}$ such that $\boldsymbol{K}\dot{\boldsymbol{\theta}} = \mathbf{0}$. Hence, the null space of $\boldsymbol{K}$ defines the vector subspace in which reduced coordinate velocities of the spanning tree do not violate the loop-closing constraints. With $\boldsymbol{Z}$ an orthonormal basis on $null(\boldsymbol{K})$, the Jacobian which accounts for the loop closures can be written as $\boldsymbol{J} = \boldsymbol{J}^{(t)}\boldsymbol{Z}$. That is, the motion of the spanning tree is constrained to the manifold which preserves the loop-closing constraints, resulting in a Jacobian with matrix rank equal to the total DoF. From this point, the same procedure as before can be followed to determine the normal modes by calculating the eigenvalues, $\boldsymbol{\Lambda}$, and eigenvectors, $\boldsymbol{Q}$, of $\boldsymbol{\mathcal{I}} = \boldsymbol{J}^T\boldsymbol{M}\boldsymbol{J}$, and hence obtain the atomic and directional DoF.

Notably, rather than first defining a spanning tree and then handling loop closures, an alternative approach which works for both cyclic and open-chain structures is to set the initial Jacobian equal to the identity, and then construct $\boldsymbol{K}$ from all constraints in the system. This is useful for verification, and makes rows of $\boldsymbol{K}$ simpler to construct, but it also increases the size of $\boldsymbol{K}$ and thus the expense of computing a basis on $null(\boldsymbol{K})$ (e.g. by singular value decomposition).

A more complicated constraint example is that of a loop-closing angle constraint between atoms i, j and k, which may have unconstrained ij and/or jk distances (see Fig. 3 as an example). With atom j as the reference point, the motion in the constraint space is given by the angular velocity of atom i around j relative to that of k around j, projected onto the ijk plane if such a plane exists. The operator which gives the velocity of i relative to j when applied to $\boldsymbol{J}^{(t)}\dot{\boldsymbol{\theta}}$ is $(\boldsymbol{C}_i - \boldsymbol{C}_j)$. Hence, the operator for the angular velocity of i around j is

$$\boldsymbol{\omega}_{ji} = \frac{[\boldsymbol{r}_{ji}]_\times}{\left|\boldsymbol{r}_{ji}\right|^2}\left(\boldsymbol{C}_i - \boldsymbol{C}_j\right). \tag{19}$$

If the angle is constrained to 0° or 180°, then the constraint imposes $\boldsymbol{\omega}_{ji} = \boldsymbol{\omega}_{jk}$, and hence the velocity along the constraint is given by the difference $\boldsymbol{\omega}_{jk} - \boldsymbol{\omega}_{ji}$. Alternatively, in the more common case that the fixed angle constrains motion only in one plane, the constraint applies the condition

$$\frac{(\boldsymbol{r}_{ji} \times \boldsymbol{r}_{jk})^T}{\left|\boldsymbol{r}_{ji} \times \boldsymbol{r}_{jk}\right|}(\boldsymbol{\omega}_{jk} - \boldsymbol{\omega}_{ji})\boldsymbol{J}^{(t)}\dot{\boldsymbol{\theta}} = 0. \tag{20}$$

Therefore,

$$\boldsymbol{V}_k^{\text{ang}} = \begin{cases} \boldsymbol{\omega}_{jk} - \boldsymbol{\omega}_{ji}, & |\boldsymbol{r}_{ji} \times \boldsymbol{r}_{jk}| = 0, \\ \frac{(\boldsymbol{r}_{ji} \times \boldsymbol{r}_{jk})^T}{|\boldsymbol{r}_{ji} \times \boldsymbol{r}_{jk}|}(\boldsymbol{\omega}_{jk} - \boldsymbol{\omega}_{ji}), & |\boldsymbol{r}_{ji} \times \boldsymbol{r}_{jk}| \neq 0. \end{cases} \tag{21}$$

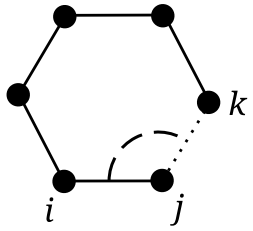


Fig. 3. Example of a loop-closing angle constraint (dashed arc). i, j, and k are already part of the spanning tree (solid lines), so the ijk angle constraint creates a kinematic loop. Note, the angle constraint cannot be represented as an ik distance constraint in this case because the jk bond length (dotted line) is not constrained.

1.1.4. Dihedral constraints

As they apply to four atoms, dihedral constraints are most simply implemented in the same manner as loop-closing angle constraints. For a dihedral constraint on atoms i, j, k and l, it is the angle between the ijk and jkl planes which is held fixed. Hence, the $\boldsymbol{V}_k^{\text{dih}}$ matrix should correspond to the difference in angular velocity between atoms i and l around axis jk, which is given by

$$\begin{aligned}\boldsymbol{V}_k^{\text{dih}} &= \frac{\boldsymbol{r}_{jk}^T}{\left|\boldsymbol{r}_{jk}\right|}\left[\frac{[\boldsymbol{r}_{ji}]_\times}{\left|\boldsymbol{r}_{ji}\right|^2}\left(\boldsymbol{C}_i - \boldsymbol{C}_j^{(t)}\right) - \frac{[\boldsymbol{r}_{kl}]_\times}{\left|\boldsymbol{r}_{kl}\right|^2}\left(\boldsymbol{C}_l - \boldsymbol{C}_k\right)\right], \\ &= \frac{\boldsymbol{r}_{jk}^T}{\left|\boldsymbol{r}_{jk}\right|}(\boldsymbol{\omega}_{ji} - \boldsymbol{\omega}_{lk}). \end{aligned} \tag{22}$$

A similar matrix can be used for improper dihedral constraints, in which case ij is the axis of rotation, and the angle constraint is between the ijk and ijl planes.

This process also illustrates a general method for constructing the $\boldsymbol{V}_k$ matrix for arbitrary geometric constraints.

2. Implementation details

The previous section makes it clear that for a given semi-rigid fragment, knowledge of the topology of the spanning tree constraints and a list of loop-closing constraints (which may be empty) is sufficient to construct the Jacobian, given particle positions and masses. With this in mind, `dofulator` first builds these topology representations from information about constraints in the system, providing a plan for the Jacobian construction. As only distance constraints are currently supported, [1] the spanning tree can be defined simply by storing the index of the predecessor particle, p, of each particle, s, in a given fragment, following similar procedures to those described in Ref. [8]. The list of particles in each fragment is then calculated from predecessor information in an order such that the predecessor of a given particle always comes before that particle. This ensures that when the list is iterated over in order, block row s in $\boldsymbol{J}^{(t)}$ can be constructed simply by copying block row p and then overwriting the zeroes in the columns corresponding to motion of s relative to p. With the plan defined by the topology information stored in a context structure, Jacobians for all fragments, and from there the atomic and directional DoF, can be computed for a given particle configuration by simple iteration over particles in each fragment. The DoF results arestored in the context for querying as described previously, and then overwritten when computed for a new configuration.

In the case of rigid bodies, the fragments are flagged as such during initialization, and the Jacobian is built based on rotation around the center of mass so that only a 3×3 inertia tensor needs to be diagonalized, with translational DoF treated separately. As the total DoF of a given particle in a rigid body does not change, construction of the Jacobian and diagonalization of the inertia tensor only needs to be done once. Choosing up to 3 particles as a reference allows calculation of the orientation of the rigid body relative to its orientation when the Jacobian was calculated, and hence directional DoF can be determined by projection as in Eq. (14).

In the most computationally expensive case we considered, with 560 semi-rigid fragments each containing 23 atoms and 24 constrained bond lengths, giving two loop closures (Fig. 6), it took approximately 2 minutes to process 401 trajectory frames on a laptop with an Intel® Core™ i7-10850H processor. Ignoring time spent loading the trajectory from disk, approximately 10 % of the processing time was spent handling loop closures (primarily in singular value decomposition to find $\boldsymbol{Z}$), and the remaining 90 % was spent mostly on finding the eigenvectors of $\boldsymbol{I}$. This case is significantly more complicated than typical constraint topologies encountered in routine molecular dynamics simulations, which are more likely to have semi-rigid fragments of 4 atoms or less (e.g. CH_2 or CH_3 groups with rigid C–H bond lengths). A 4 atom fragment without loop closures would have a 12×12 inertia tensor to diagonalize, rather than the 67×67 one discussed here, and therefore a significantly lower computational load, with eigenvalue decomposition having a naïve computational complexity of $\mathcal{O}(n^3)$. Even the simple case remains somewhat expensive to evaluate on every step of a simulation with many fragments, though, so if this is required, possible optimizations include batching of matrix operations for GPU acceleration, or perhaps taking advantage of tight integration with the constraint solver, or analytical solutions to simple cases [2]. However, since current run-times are acceptable for the more likely use case of post-processing or evaluation every $N \gg 1$ time steps, we leave such improvements to future work should they become necessary. We note that for long trajectories, parallelization over trajectory frames offers a simple method of acceleration, in which case the current implementation requires only that a separate context is constructed for each parallel worker.

To reduce computational load in the semi-rigid case, it is tempting to separate the Jacobian into smaller matrices corresponding to translation, rotation, and vibrations, and diagonalize those individually. However, in general, the rotational and vibrational modes are inherently coupled (often referred to as Coriolis coupling) and so cannot be separated without approximation. Furthermore, separation of translational modes would require the rovibrational Jacobian to be constructed such that it preserves the center of mass, which, while possible, is not as simple as the procedure described here. Hence, to obtain separate contributions for translational and rovibrational modes of semi-rigid fragments, we separately calculate the translational DoF (which are simply the mass fraction of each atom), and then subtract the result from the total DoF, leaving the rovibrational component. Note, further separation into individual modes is possible in theory by considering particular columns of $\boldsymbol{R}$, but since the normal modes will change as the fragment deforms, there is no guarantee that a given column will correspond to the same mode at different times or between different topologically equivalent fragments.

3. Simulations

With the necessary theory in place, we now simulate systems with various constraint geometries as a means of validation. Unless otherwise specified, results in the following sections used GROMACS (version 2022.3) [10] for molecular dynamics simulation and VMD [11] with the Tachyon renderer [12] for visualization.

[1] This only excludes support for unusual cases where an angle is fixed but the bond lengths can vary, or a dihedral is fixed but the angles and/or bond lengths can vary. While the theory presented in this manuscript is applicable to such cases, they are not normally implemented in MD codes, and hence not currently supported in `dofulator`.

3.1. Open chain molecules: Tree topology

Simulating liquid ethane at equilibrium with constrained bond lengths (as described in Section 1.1) presents a simple test for open chain molecules (having tree topology). In the lowest energy configuration, as given by geometry optimization performed by the Automated Topology Builder (ATB) [13], the carbon atoms have 2.287 DoF, while the hydrogen atoms have 2.071, although both depend on the particular conformation and were adjusted accordingly, per-atom, during analysis.

1362 ethane molecules in a $5 \times 5 \times 5$ nm^3 periodic box were simulated for 1000 ps. The `md-vv-avek` integrator was used with a time step of 0.25 fs, and bond lengths were constrained using the LINCS algorithm [14] with order 4 and 2 iterations. Note, the small time step was required for this system so that the kinetic temperature calculated from the output velocities agreed with the set-point of the thermostat (and with the temperature reported in the GROMACS simulation log). Increasing the LINCS order to 8 and iterations to 4 did not make a noticeable difference. Partial atomic charges, pair-wise interaction parameters, and angle and dihedral restraint parameters were from the ATB [13], based on the GROMOS 54A7 force field [15]. The simulation began from an initial, randomly generated configuration, in which atom velocities were randomly assigned by GROMACS at 185 K. Temperature was controlled at 185 K using a Nosé-Hoover chain thermostat with chain length 3 and a coupling time of 0.1 ps.

Fig. 4 shows the local, instantaneous kinetic temperatures of both types of atom in the system (C and H atoms) as a function of time, averaged over all molecules in the system, with data points every 0.125 ps. For a given type of atom, $\mathcal{A}$, the local instantaneous temperature was calculated as

$$T_{\mathcal{A}}(t) = \frac{\sum_{i \in \mathcal{A}} m_i \boldsymbol{v}_i \cdot \boldsymbol{v}_i}{k_B \sum_{i \in \mathcal{A}} d_i}, \tag{23}$$

where $\boldsymbol{v}_i$ is the velocity of atom i, d_i is its DoF, m_i is its mass, and k_B is Boltzmann's constant. Similarly, the instantaneous temperature of the overall system was calculated as

$$T_{\text{all}}(t) = \frac{\sum_i m_i \boldsymbol{v}_i \cdot \boldsymbol{v}_i}{k_B \sum_i d_i}. \tag{24}$$

Firstly, it is evident that symmetrically equivalent atoms have the same temperature on average, as expected, and the temperatures of both the C and H atoms converge to the total system temperature, indicating that the calculated DoF partitioning is correct. Secondly, comparing against the temperature calculated from all particles in the system, it is clear that relaxation to equilibrium between C and H atoms is relatively slow, as this is not aided by the thermostat which acts only on the overall system temperature. Equilibration time between internal DoF was also observed to be sensitive to the integrator routine, taking significantly longer when the leap-frog algorithm (GROMACS' `md` integrator) was used. This could be influenced by various factors regarding the details of the numerical integrator and constraint algorithm, which we intend to examine in future work. However, a longer equilibration time between internal modes beyond the time for the overall system temperature to relax is an expected result [16,17].

Note, this system has more complex constraint topology than is typical of routine molecular dynamics simulations in which it is often only the C–H bond length that is constrained. In such scenarios, each XH_n group can be treated as an individual semi-rigid fragment. Hence, this system of semi-rigid ethane serves as a test of the algorithm in a plausible, but relatively extreme scenario. The following sections explore even more strenuous circumstances.

3.2. Loop closure

A simple test for handling a constraint topology with a loop closure is the benzene molecule with all bond lengths constrained. Similar to the ethane system, a $5 \times 5 \times 5$ nm^3 periodic unit cell of 845 benzene molecules with constrained bond lengths was simulated for 50 ps using a 0.5 fs time step, and the temperature of each unique atom within a molecule was tracked. The details of the integration scheme, thermostat, and interaction parameters were the same as for the previous simulation of ethane, with the thermostat set to 300 K and initial velocities generated to match that value. Results in Fig. 5 show clearly that accounting for the loop-closing constraints gives a temperature which correctly converges to the set-point after some time for equilibration of internal DoF. Notably, again, equilibration of internal modes took significantly longer than equilibration of the overall system temperature, although shorter than the previous example.

A more complex test with two kinematic loops is presented by 1-methoxybicyclo[2.2.1]heptane with rigid bond lengths. Simulation parameters were as before, with 560 molecules in a $5 \times 5 \times 5$ nm^3 periodic unit cell at 300 K, although again in this case a 0.25 fs time step was used to ensure the output velocities gave a system temperature which

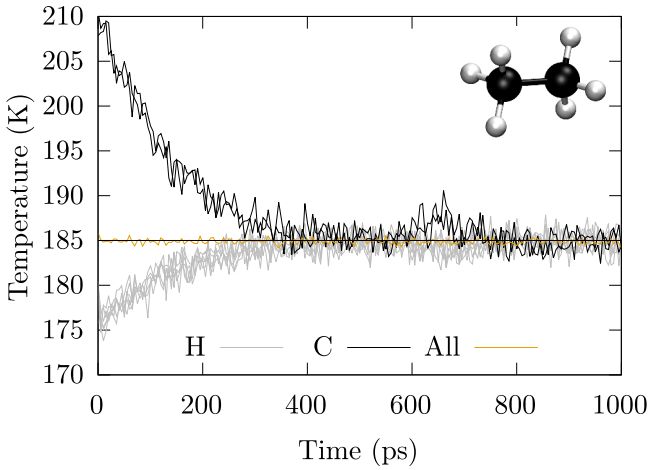


Fig. 4. Temperature of each unique atom in liquid ethane with rigid bond lengths as a function of time after initial random velocity generation, compared against the temperature of the overall system (orange) and the thermostat set-point (dashed line). The temperature was calculated every 0.125 ps, and each data point in the figure represents a block average over 5 ps.

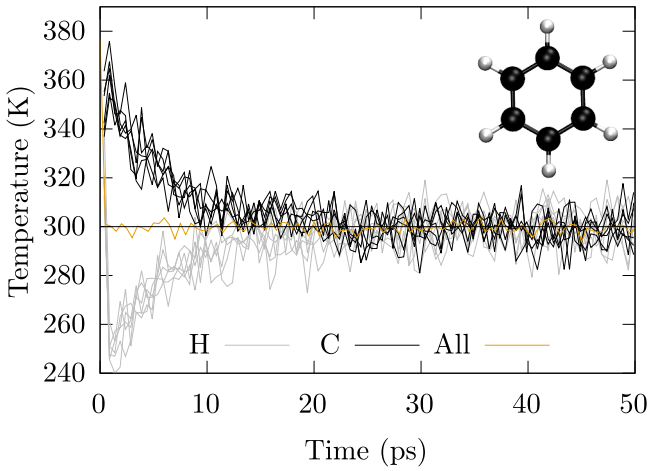


Fig. 5. Temperature of each unique atom in liquid benzene with rigid bond lengths as a function of time after initial random velocity generation, compared against the overall kinetic temperature of all atoms in the system. Atomic temperatures were calculated as per Eq. (23), with lines of the same color representing symmetrically equivalent atoms, and the dashed line indicating the set-point temperature of the Nosé-Hoover thermostat. The temperature was calculated every 0.25 ps and each data point in the figure represents a block average over 0.5 ps.

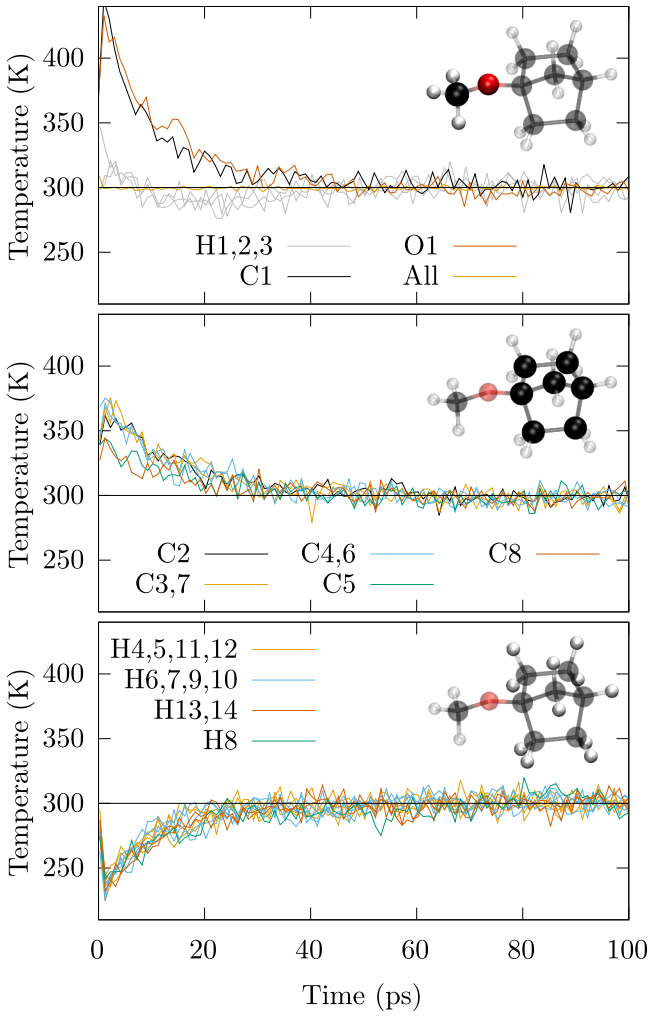


Fig. 6. Temperature of each atom in bulk-phase 1-methoxybicyclo[2.2.1]heptane with all bond lengths held rigid. Temperature was calculated as per Eq. (23), and the dashed line indicates the set-point temperature of the thermostat. Data is split into three panels for clarity, with the top panel showing atoms from the methoxy group and the overall system temperature, the middle panel showing carbon atoms in the ring structure, and the bottom showing hydrogen atoms attached to the ring structure. Insets show the plotted atoms as opaque, and explicit atom labels can be seen in the inset of Fig. 8. The temperature was calculated every 0.25 ps and each data point in the figure represents a block average over 1 ps. Lines of the same color represent symmetrically equivalent atoms.

converged to the set-point, and a LINCS order of 6 was used to ensure all constraints "see" each other. Results in Fig. 6 again show that temperatures of all atoms converge to the thermostat set-point after some equilibration time, confirming that the algorithm for handling loop closures is robust against multiple kinematic loops.

3.3. DoF distribution

Especially for larger semi-rigid fragments, or those with loop-closing constraints, it is tempting for computational performance to avoid dynamically evaluating the DoF of each atom in each conformation by simply assigning a constant value based on the molecule's minimum energy geometry or sampling from a short simulation. Here we examine the extent to which atomic DoF is sensitive to the conformation of a fragment.

Considering carbon atoms in octane with rigid bond lengths, various conformations were generated with H atoms rotated into minimum energy positions for each using the molecular mechanics tool in WebMO [18]. C–C bonds were fixed at 1.53 Å, while C–H bonds were fixed at 1.12 Å. Fig. 7a shows an example of one such conformation, indicating that the atomic DoF can depend strongly on the local constraint environment, as well as the position within the chain. To separate the two components as much as possible, Fig. 7b examines the atomic DoF as a function of the C–C–C bond angles when the bond angles are all equal. It is apparent that the carbon atoms at the ends of the chain are particularly sensitive to the bond angle, changing by 0.36 DoF across the span of angles tested, whereas those in the middle of the chain only changed significantly at large angles. It is also apparent that only considering the conformation of the constraints directly involving a given atom is not enough to predict its DoF; the conformation of the rest of the molecule can also have a significant effect. For example, C7 has 2.082 DoF in the conformation of Fig. 7a, despite only being directly involved in a 180° and 160° C–C–C angle, both of which would suggest a value of approximately 1.95 DoF based on Fig. 7b. This may be explained by the opposite end of the chain being compacted in the Fig. 7a conformation so that the center of mass is further from C7, giving it a greater contribution on average to the inertia of the rotational and bending modes. Interestingly, variations in C–C–C–C dihedral angles (not shown) which maintained fixed bond angles (approx. 107°) and minimum energy H positions had negligible effect on atomic DoF, suggesting that changes in rotational inertia contributions due to extension or compaction of the molecule were compensated by opposing changes in contributions to bending inertia in this case.

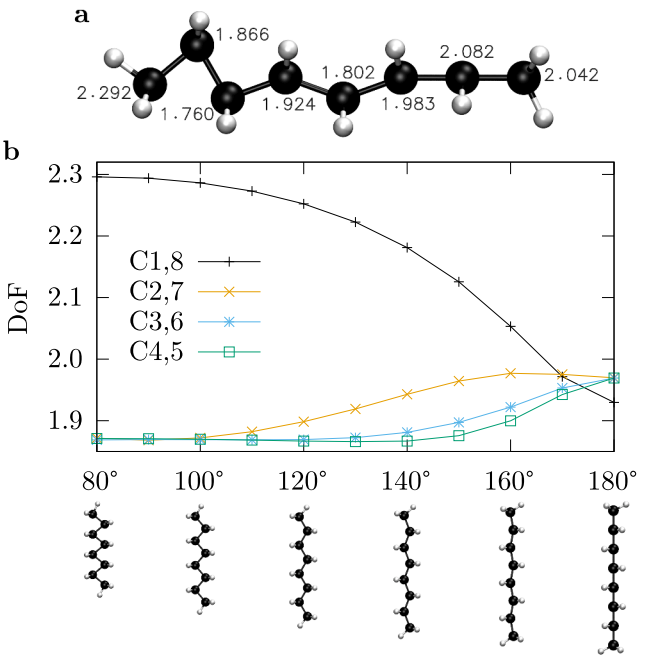


Fig. 7. Sensitivity of carbon atom DoF to bond angles in octane with rigid bond lengths. (a) shows the DoF of each carbon atom in a configuration with C–C bond angles progressing from 80° to 180° in 20° increments. In (b), all C–C–C angles were adjusted simultaneously (x axis). C1 and C8 refer to the carbon atoms at each end of the chain, while C4 and C5 are the central two.

While these results would seem to imply that per-conformation DoF calculations are necessary, there are two factors to be considered. Firstly, for temperatures calculated over a large enough number of equivalent atoms in a similar chemical environment (e.g. the bulk phase), conformations are expected to be within some distribution of the average conformation in that environment, and hence using the average atomic DoF would be an appropriate approximation. For example, calculating the temperature of each atom in ethane with rigid bonds (Section 3.1) using the atomic DoF values of the minimum energy

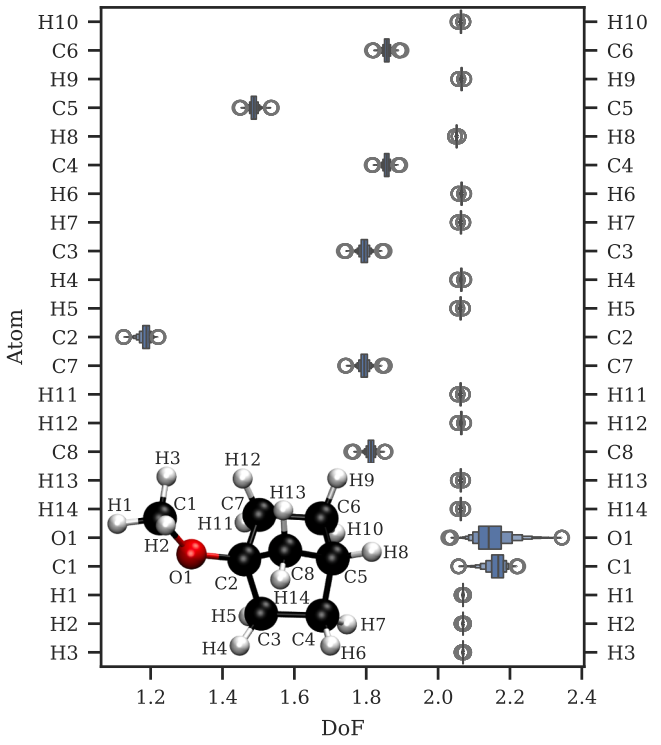


Fig. 8. Letter value plot [19] showing the distribution of DoF of each atom in 1-methoxybicyclo[2.2.1]heptane (bulk phase, 300 K, ~1 g/cm^3). Data from 200 trajectory snapshots, each 0.25 ps apart, with 560 molecules in each snapshot. Box widths indicate the relative sample density, and outliers (representing 0.7 % of the data points) are plotted as circles.

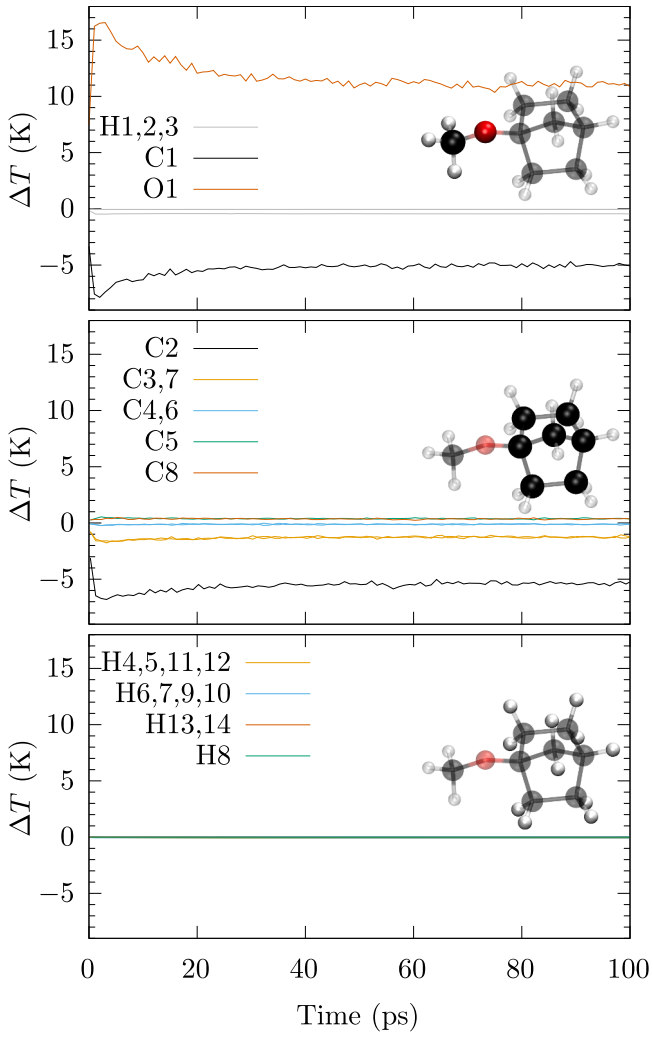


Fig. 9. Difference in temperature, $\Delta T_{\mathcal{A}} = T^0_{\mathcal{A}} - T^{\text{dyn}}_{\mathcal{A}}$, between the temperature calculated using constant per-atom DoF based on the minimum energy geometry, $T^0_{\mathcal{A}}$, and the more accurate temperature calculated using dynamically updated DoF, $T^{\text{dyn}}_{\mathcal{A}}$. The upper, middle, and lower panels show the atoms in the methoxy group, C atoms in the ring structure, and H atoms attached to the ring structure, respectively, as highlighted by the opaque atoms in the insets. The geometry of the insets is that of the minimum energy, in which H1, C1, O1, C2, C8, C5 and H8 are co-planar. Note, all lines in the lower panel are coincident (i.e. $\Delta T_{\mathcal{A}} \approx 0$ for $\mathcal{A} \in \{\text{H4}, \ldots, \text{H14}\}$).

geometry yields near-identical results to those in Fig. 4 which re-calculated atomic DoF for each frame of the trajectory.

Secondly, the range of motion considered here is significantly larger than the range expected to be sampled within a typical simulation, and hence the variance in atomic DoF may be smaller in practice. However, other atomic configurations may have a larger range of motion, or may be more or less sensitive depending on the relative atomic masses. We examine this further by plotting the distribution of atomic DoF encountered in the simulation of bulk phase 1-methoxybicyclo[2.2.1]heptane with rigid bonds as described above, sampling from the final 50 ps. Fig. 8 shows that while the hydrogen atoms are quite insensitive to conformation (as was also the case for octane), the carbon and oxygen DoF can vary significantly. The oxygen atom (O1) and the carbon atom at the end of the methoxy group (C1) were particularly sensitive, while the carbon atoms in the ring structure were less-so.

Given the large variance in DoF of some atoms, particularly O1 and C1, it is interesting here to compare temperatures calculated from dynamically updated DoF, $T^{\text{dyn}}_{\mathcal{A}}$, to those calculated by simply assigning constant atomic DoF equal to that of the minimum energy geometry, $T^0_{\mathcal{A}}$. Fig. 9 plots $\Delta T_{\mathcal{A}} = T^0_{\mathcal{A}} - T^{\text{dyn}}_{\mathcal{A}}$ over the course of the trajectory, and shows clearly that not all atom temperatures converge to a steady value commensurate with the thermostat temperature when a constant DoF value is used. This implies that the average geometry in the equilibrium state does not match the minimum energy geometry determined by the ATB, which is not particularly surprising given that it was calculated for the infinitely dilute case, whereas the simulation is of the liquid phase.

Fig. 9 also shows that one H atom in the methoxy group has a slightly different temperature deviation to the others. This stems from a difference in DoF in the minimum energy geometry between H1 (~2.072 DoF), and H2 and H3 (~2.070 DoF). Although they should be equivalent (and Fig. 8 shows they indeed have the same DoF on average when it is dynamically updated), in the single geometry used for assigning the degrees of freedom in this case, they are not. The difference is in the (instantaneous) environment experienced by the H1 atom, which is co-planar with C1, O1, C2, C8, and H8, compared to the H2 and H3 atoms which are symmetric across that plane. Since the H1–H2–H3 plane is not precisely perpendicular to the C1–O1 bond, H1 has a slightly different inertia contribution to the bending motion of the C1–O1–C2 angle in comparison to H2 or H3. Similarly, the H1–H2–H3 plane is also not perpendicular to the C1–C2 vector, and therefore there is also a difference in contribution to the bending inertia of the O1–C2–C8 angle. Note, these bond bending motions may not necessarily be (instantaneous) normal modes of motion, but illustrate how equivalent atoms may contribute differently to inertia depending on their position. A better estimation of atomic DoF would be obtained by appropriate averaging of equivalent atoms, and the numerical results indicate that this would also give a better approximation of local temperatures.

The other interesting phenomena apparent in Fig. 9 is the magnitude of the fluctuations in $\Delta T_{\mathcal{A}}$. Those atoms with higher variation in DoF (Fig. 8), particularly C1 and O1, and to a lesser extent, C2, show larger

fluctuations, even after the system has equilibrated. Such fluctuations obviously also depend on the number of atoms in the local subset being measured, and hence a dynamic DoF calculation may be required for small subsets of atoms with large DoF variance if accurate time-resolved data is required, whereas a constant, average value would be sufficient for larger subsets and/or those with smaller variance. Time-averaged measurements would also converge to the correct value when a constant atomic DoF is used based on the average value, provided there is no space- or time-dependence. However, molecules around some inhomogeneity/interface may have a different average conformation (and therefore a different DoF distribution) than those in the bulk, or a non-equilibrium process such as a phase change or change in stress may cause the average conformation to change with time.

3.4. Thermal fields

Lastly, we briefly examine a system where there is heat flow across a water-graphene interface due to a thermal gradient. In this nonequilibrium system, accurate local temperature measurement is critical for determination of the temperature profile and examination of the interfacial thermal resistance (Kapitza resistance) between the water and graphene. The Kapitza resistance at the phase boundary between the graphene and water causes a step change in the temperature, and this property is important for thermal management in nanoscale technologies. Using LAMMPS (version 22 Dec 2022) [20], water confined between two walls thermostatted at different temperatures was simulated. Heat flow occurred through the water, which was modelled using two different force fields: the SPC/E (rigid body) potential [21], and a semi-rigid version in which the H–O–H angle was allowed to change with the harmonic potential from the flexible SPC/Fw model [22]. The latter case represented a system of semi-rigid fragments. Bond and angle constraints were applied using the RATTLE [23] algorithm.

The two-layer graphene walls were modelled using a Tersoff potential for intramolecular interactions [24], and a Lennard-Jones potential for interactions between layers [25] and with water [26]. The outer graphene layers were frozen, while the inner layers were free to move but had the total momentum of each layer set to zero each time step to maintain a consistent channel width. The temperature of each graphene wall was controlled with a Langevin thermostat [27,28] with a 50 fs coupling time, while no thermostat was applied to the water molecules.

After initial equilibration of the rigid water system for 2 ns with a 1 fs integration time step and both walls thermostatted to 300 K, the target temperatures of the thermostats on the left and right walls were adjusted to 250 K and 350 K, respectively. The system was then allowed to progress to a nonequilibrium steady state over the course of 3 ns. This configuration was then replicated as a starting point for the semi-rigid water system. For both systems, the time step was adjusted to 0.5 fs and a further 500 ps was simulated to ensure a consistent steady state was achieved. The configuration and velocities were then sampled every 500 fs over 4.5 ns for post-processing.

Fig. 10 shows that for both constraint topologies, the inertia-based DoF partitioning implemented in `dofulator` enables reliable measurement of the temperature profile close to the walls, thereby providing more data from which the temperature gradient, and hence the Kapitza resistance, can be calculated. When the temperature was determined using `dofulator`, the region where the temperature profile is reliable (where it varies linearly with position) has the same extent as that of similar simulations with the fully flexible SPC/Fw model, for which local temperature calculation is trivial. In comparison, naïvely assigning 6/3 (7/3) DoF to each atom in rigid (semi-rigid) water only gives a reliable measurement in the bulk phase, and therefore requires an additional ~5 Å of extrapolation to the walls. The oscillations observed with the naïve DoF assignments are due to the orientational ordering of the water near the walls [3], and therefore the oscillations dampen in the center of the channel where the water is more bulk-like and is randomly oriented. The presence of oscillations would be especially problematic in

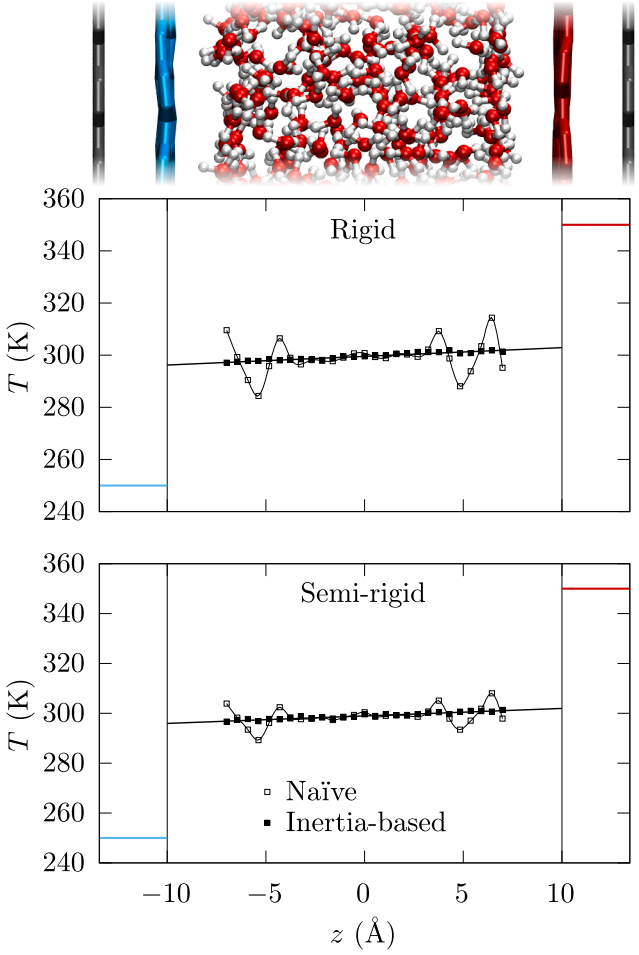


Fig. 10. Temperature profile of water confined between graphene walls, comparing results from inertia-based partitioning (as implemented in `dofulator`) against naïve DoF assignment (each atom within a molecule assigned the same DoF). Vertical dashed lines mark the locations of the inner graphene layers, with the left and right layers thermostatted to 250 K and 350 K, respectively (highlighted blue and red in the pictorial representation). A linear fit to the inertia-based result is shown as a guide to the eye, as is a cubic spline fit to the naïve result. Data points with low average density of water (and therefore poor statistical sampling) were omitted. The pictorial representation shows a slice of the rigid water system aligned with the x axes of the plots. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

narrower channels where there is no bulk region, making it impossible to determine the temperature gradient using the naïve DoF assignment. For the semi-rigid system considered here, using constant DoF values calculated from the equilibrium geometry (2.882 for O, 2.059 for H) gives near-identical results to the dynamic DoF calculation. Hence, in this case, ordering at the interface does not cause significant deviation of the water molecules' average conformation from the minimum energy one, and therefore constant DoF values are sufficient if directional kinetic temperature is not required. Note, though, that this result is likely case-specific, and other nonequilibrium/inhomogeneous systems should be checked before deciding whether using constant atomic DoF is an appropriate approximation.

4. Conclusions

Here we have shown all necessary details to implement calculation of local DoF and temperature in the presence of arbitrary geometric constraints by the method of inertia-based partitioning. The algorithm,

implemented in our open source `dofulator` package, was tested successfully for various constraint topologies, including both open chains and topologies with closed kinematic loops, and for a confined system with a temperature gradient. Simulation results confirmed that, in all cases, on-the-fly determination of the local DoF allowed for accurate local kinetic temperature measurement. Furthermore, it is clear that such measurements present a sensitive test for the equilibration of thermal velocities between internal DoF when such conditions are expected, and this may take significantly longer than equilibration of the overall system temperature. Additionally, we have shown that using the average DoF of a given atom, or even simply its DoF in the minimum energy geometry of the infinitely dilute gas phase, is often sufficient to obtain accurate results, although this may not always be the case.

CRediT authorship contribution statement

Stephen Sanderson: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Conceptualization; **Sobin Alosious:** Writing – review & editing, Investigation; **Debra J. Searles:** Writing – review & editing, Validation, Supervision, Funding acquisition.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors thank the Australian Research Council for its support for this project through the Discovery program (FL190100080). We acknowledge access to computational resources provided by the Pawsey Supercomputing Centre with funding from the Australian Government and the government of Western Australia, and the National Computational Infrastructure (NCI Australia), an NCRIS enabled capability supported by the Australian Government. We also thank Dr. Kim Loi for his assistance in testing the installation process for the software.

References

[1] J.D. Olarte-Plata, F. Bresme, Thermal conductance of the water-gold interface: the impact of the treatment of surface polarization in non-equilibrium molecular simulations, J. Chem. Phys. 156 (20) (2022) 204701. https://doi.org/10.1063/5.0090983
[2] H. Matsubara, D. Surblys, T. Ohara, Degrees of freedom of atoms in a rigid molecule for local temperature calculation in molecular dynamics simulation, Mol. Simul. 49 (13-14) (2023) 1365-1372. https://doi.org/10.1080/08927022.2023.2232466
[3] S. Sanderson, S.R. Tee, D.J. Searles, Local temperature measurement in molecular dynamics simulations with rigid constraints, J. Chem. Theory Comput. 20 (23) (2024) 10615-10624. https://doi.org/10.1021/acs.jctc.4c00957
[4] N. Michaud-Agrawal, E.J. Denning, T.B. Woolf, O. Beckstein, MDAnalysis: a toolkit for the analysis of molecular dynamics simulations, J. Comput. Chem. 32 (10) (2011) 2319-2327. https://doi.org/10.1002/jcc.21787
[5] R. Gowers, M. Linke, J. Barnoud, T. Reddy, M. Melo, S. Seyler, J. Domański, D. Dotson, S. Buchoux, I. Kenney, O. Beckstein, MDAnalysis: a python package for the rapid analysis of molecular dynamics simulations, in: Proceedings of the 15th Python in Science Conference, SciPy, SciPy, 2016. https://doi.org/10.25080/majora-629e541a-00e
[6] A. Jain, I.-H. Park, N. Vaidehi, Equipartition principle for internal coordinate molecular dynamics, J. Chem. Theory Comput. 8 (8) (2012) 2581-2587. https://doi.org/10.1021/ct3002046
[7] G.R. Kneller, K. Hinsen, Generalized Euler equations for linked rigid bodies, Phys. Rev. E 50 (2) (1994) 1559-1564. https://doi.org/10.1103/physreve.50.1559
[8] R. Featherstone, Rigid Body Dynamics Algorithms, Springer US, 2008. https://doi.org/10.1007/978-1-4899-7560-7
[9] A.K. Mazur, Symplectic integration of closed chain rigid body dynamics with internal coordinate equations of motion, J. Chem. Phys. 111 (4) (1999) 1407-1414. https://doi.org/10.1063/1.479399
[10] M.J. Abraham, T. Murtola, R. Schulz, S. Páll, J.C. Smith, B. Hess, E. Lindahl, GROMACS: high performance molecular simulations through multi-level parallelism from laptops to supercomputers, SoftwareX 1-2 (2015) 19-25. https://doi.org/10.1016/j.softx.2015.06.001
[11] W. Humphrey, A. Dalke, K. Schulten, VMD – visual molecular dynamics, J. Mol. Graph 14 (1996) 33–38. .
[12] J. Stone, *An Efficient Library for Parallel Ray Tracing and Animation*, Master's thesis, Computer Science Department, University of Missouri-Rolla, 1998.
[13] M. Stroet, B. Caron, K.M. Visscher, D.P. Geerke, A.K. Malde, A.E. Mark, Automated topology builder version 3.0: prediction of solvation free enthalpies in water and hexane, J. Chem. Theory Comput. 14 (11) (2018) 5834-5845. https://doi.org/10.1021/acs.jctc.8b00768
[14] B. Hess, H. Bekker, H.J.C. Berendsen, J.G.E.M. Fraaije, LINCS: a linear constraint solver for molecular simulations, J. Comput. Chem. 18 (12) (1997) 1463-1472. https://doi.org/10.1002/(sici)1096-987x(199709)18:12\protect$\relax<$1463::aid-jcc4\protect$\relax>$3.0.co;2-h
[15] N. Schmid, A.P. Eichenberger, A. Choutko, S. Riniker, M. Winger, A.E. Mark, W.F. van Gunsteren, Definition and testing of the GROMOS force-field versions 54A7 and 54B7, Eur. Biophys. J. 40 (7) (2011) 843-856. https://doi.org/10.1007/s00249-011-0700-9
[16] B.J. Palmer, B.C. Garrett, Constant pressure-constant temperature simulations of liquid water and carbon dioxide, J. Chem. Phys. 98 (5) (1993) 4047–4058. https://doi.org/10.1063/1.465041
[17] G. Steinebrunner, A.J. Dyson, B. Kirchner, H. Huber, Structural and thermodynamic properties of fluid carbon dioxide from a new ab initio potential energy surface, J. Chem. Phys. 109 (8) (1998) 3153–3160. https://doi.org/10.1063/1.476922
[18] W.F. Polik, J.R. Schmidt, WebMO: web-based computational chemistry calculations in education and research, WIREs Comput. Mol. Sci. 12 (1) (2022) e1554. https://doi.org/10.1002/wcms.1554
[19] H. Hofmann, K. Kafadar, H. Wickham, Letter-Value Plots: Boxplots for Large Data, Technical Report, had.co.nz, 2011.
[20] A.P. Thompson, H.M. Aktulga, R. Berger, D.S. Bolintineanu, W.M. Brown, P.S. Crozier, P.J. i.t. Veld, A. Kohlmeyer, S.G. Moore, T.D. Nguyen, R. Shan, M.J. Stevens, J. Tranchida, C. Trott, S.J. Plimpton, LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales, Comput. Phys. Commun. 271 (2022) 108171. https://doi.org/10.1016/j.cpc.2021.108171
[21] H.J.C. Berendsen, J.R. Grigera, T.P. Straatsma, The missing term in effective pair potentials, J. Phys. Chem. 91 (24) (1987) 6269-6271. https://doi.org/10.1021/j100308a038
[22] Y. Wu, H.L. Tepper, G.A. Voth, Flexible simple point-charge water model with improved liquid-state properties, J. Chem. Phys. 124 (2) (2006) 024503. https://doi.org/10.1063/1.2136877
[23] H.C. Andersen, Rattle: a "velocity" version of the shake algorithm for molecular dynamics calculations, J. Comput. Phys. 52 (1) (1983) 24-34. https://doi.org/10.1016/0021-9991(83)90014-1
[24] L. Lindsay, D.A. Broido, Optimized Tersoff and Brenner empirical potential parameters for lattice dynamics and phonon thermal transport in carbon nanotubes and graphene, Phys. Rev. B 81 (20) (2010) 205441. https://doi.org/10.1103/physrevb.81.205441
[25] L.A. Girifalco, M. Hodak, R.S. Lee, Carbon nanotubes, buckyballs, ropes, and a universal graphitic potential, Phys. Rev. B 62 (19) (2000) 13104-13110. https://doi.org/10.1103/physrevb.62.13104
[26] T. Werder, J.H. Walther, R.L. Jaffe, T. Halicioglu, P. Koumoutsakos, On the water-carbon interaction for use in molecular dynamics simulations of graphite and carbon nanotubes, J. Phys. Chem. B 107 (6) (2003) 1345-1352. https://doi.org/10.1021/jp0268112
[27] T. Schneider, E. Stoll, Molecular-dynamics study of a three-dimensional one-component model for distortive phase transitions, Phys. Rev. B 17 (3) (1978) 1302-1322. https://doi.org/10.1103/physrevb.17.1302
[28] B. Dünweg, W. Paul, Brownian dynamics simulations without Gaussian random numbers, Int. J. Modern Phys. C 02 (03) (1991) 817-827. https://doi.org/10.1142/s0129183191001037