POLITECNICO DI TORINO
Repository ISTITUZIONALE

From GROMACS to LAMMPS: GRO2LAM : A converter for molecular dynamics software

Original
From GROMACS to LAMMPS: GRO2LAM : A converter for molecular dynamics software / CHAVEZ THIELEMANN, H.N., Cardellini, A., Fasano, M., Bergamasco, L., Alberghini, M., Ciorra, G., Chiavazzo, E., Asinari, P.. - In: JOURNAL OF MOLECULAR MODELING. - ISSN 1610-2940. - ELETTRONICO. - 25:6(2019), p. 147. [10.1007/s00894-019-4011-x]

Availability:
This version is available at: 11583/2732958 since: 2019-05-10T19:06:48Z

Publisher:
Springer Nature Publishing AG

Published
DOI:10.1007/s00894-019-4011-x

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

23 October 2026

Noname manuscript No.
(will be inserted by the editor)

From GROMACS to LAMMPS: GRO2LAM

A converter for molecular dynamics software

Hernan Chavez Thielemann · Annalisa Cardellini · Matteo Fasano · Luca Bergamasco · Matteo Alberghini · Gianmarco Ciorra · Eliodoro Chiavazzo · Pietro Asinari

Received: date / Accepted: date

Abstract Atomistic simulations have progressively attracted attention in the study of physical-chemical properties of innovative nanomaterials. GROMACS and LAMMPS are currently the most widespread open-source software for molecular dynamics simulations thanks to their good flexibility, numerous functionalities and responsive community support. Nevertheless, the very different formats adopted for input and output files are limiting the possibility to transfer GROMACS simulations to LAMMPS. In this article we present GRO2LAM, a modular and open-source Python 2 code for rapidly translating input files and parameters from GROMACS to LAMMPS format. The robustness of the tool has been assessed by comparing the simulation results obtained by GROMACS and LAMMPS, after the format conversion by GRO2LAM. Specifically, three nanoscale configurations of interest in both engineering and biomedical fields are studied, namely a carbon nanotube, an iron oxide nanoparticle, and a protein immersed in water. In perspective, GRO2LAM may be the first step to achieve a full interoperability between molecular dynamics software. This would allow to easily exploit their complementary potentialities and post-processing functionalities. Moreover, GRO2LAM could facilitate the cross-check of simulation results, guaranteeing the reproducibility of molecular dynamics models and testing their robustness.

Keywords Reproducibility · Molecular Dynamics · GROMACS · LAMMPS · Conversion

1 Introduction

Nanostructured materials are nowadays at the frontier of innovation in a broad variety of industries, spanning from engineering [1–6] to biomedical [7–10] fields. However, the design of nanostructured materials often encounters challenges linked

Department of Energy, Politecnico di Torino - Corso Duca degli Abruzzi 24, 10129 Torino - Italy
E-mail: pietro.asinari@polito.it (Pietro Asinari)

to the prediction of their macroscopic properties on the bases of geometrical, physical and chemical material's parameters. The progressive improvement of the computational capabilities together with the speed-up of the most recent high performance computing have enhanced the role of modelling in the rational design of nanostructured materials.

Several computational methods are typically used to investigate and guide the design of innovative materials, for example: the numerical solution of Navier-Stokes equations and Fourier's law (macroscale), the numerical solution of Boltzmann transport equation (mesoscale), the integration of Newton's law at the atomistic scale (nanoscale), and combinations thereof [11–17]. In particular, Molecular Dynamics (MD) simulations offer a suitable platform both to describe nanoscale phenomena, for instance heat and mass transport mechanisms at solid-liquid interface [18–20], and to estimate macroscopic properties, such as thermal conductivity and viscosity [21–23]. The first MD studies were developed by Alder and Wainwright [24] in the fifties and by Rahman [25] in the sixties. Thanks to the Pantagruelian progress in the software and hardware industry, MD has then become a powerful tool to provide new physical and chemical insights at molecular level, therefore assisting the rational design of nanostructured materials.

Among the many numerical codes developed to perform atomistic simulations [26–28], GROMACS [29] and LAMMPS [30] are open-source software that are attracting particular interest in the scientific community, because of their good flexibility, numerous functionalities and responsive community support. Both software have been developed to numerically integrate a set of coupled differential equations (Newton's law) describing the trajectory of each atom. Although these software all share the same methodology, they have been implemented and optimized to reach different goals, leading to the birth of a large variety of architectures and structures. For example, while GROMACS is particularly suitable for modelling soft matter and biological systems, like proteins and lipid bilayers, LAMMPS has several advantages in simulating hard matter, such as metals and semiconductors. As a result, the two MD codes manage different tools and force fields according to their own needs. For instance, Tersoff [31] and COMPASS [32] force fields are implemented in LAMMPS, while they are not available in GROMACS. In addition, the growing request of reproducibility of the simulation data encourages the scientific community to validate models and results by using different MD platforms. With this background in mind, it seems very much convenient to easily convert input files and parameters from one format to another, according to the specific requirements of users. Some preliminary attempts for the translation of MD files between specific formats have been recently developed, for example the TopoGromacs [33], MDWiZ [34], DL_FIELD [35, 36], ParmEd, and InterMol [37] tools. However, we are still far from achieving a general transferability among different MD engines by means of a fully-flexible, user-friendly and open-source code.

In this article, we present a Python 2 program, GRO2LAM, whose aim is to convert MD input files from one format to another. The algorithm, freely available on the open-source platform *GitHub*[1] under the MIT license, is designed to deal with a potential multitude of MD engines, software versions, force fields and simulation parameters. In the first release, GRO2LAM automates the conversion

[1] https://github.com/hernanchavezthielemann/GRO2LAM

of data and input files from GROMACS to LAMMPS, two of the most employed MD software. The code has been developed in a modular way, in order to facilitate the addition of functionalities tailored to the user's needs. Moreover, an intuitive Graphical User Interface (GUI) guides the users through the whole conversion process until the creation and run of the LAMMPS data and input files. Finally, the converter has been validated using study cases of particular interest in both engineering and biomedical fields.

2 Methods

2.1 Overall structure

The algorithm presented in this article is implemented as an open-source Python 2 code compatible with Linux operating systems, and it converts an input MD setup from the GROMACS to the LAMMPS format. The ultimate objective is to realize an extendible open code, which could be upgraded to include a broader variety of MD systems and software.

The GRO2LAM algorithm is supported by a GUI organized into three main sections: (i) Data file creation, (ii) Input file creation, and (iii) Running LAMMPS simulations (see the overall flow chart in Fig. 1). In the current release, the algorithm first creates the LAMMPS data file by importing from GROMACS the information concerning the atom types, including masses and partial charges, the force field parameters (bonded and non-bonded interactions), and the coordinates of atoms. Second, a collection of input parameters, both extracted from GROMACS files and set by the user, is translated into the LAMMPS input file. Finally, advanced input settings can be selected to tailor the desired molecular dynamics setup and to run the simulation in LAMMPS.

In the following sub-sections, a detailed description of GRO2LAM is presented and the main steps of the algorithm explained.

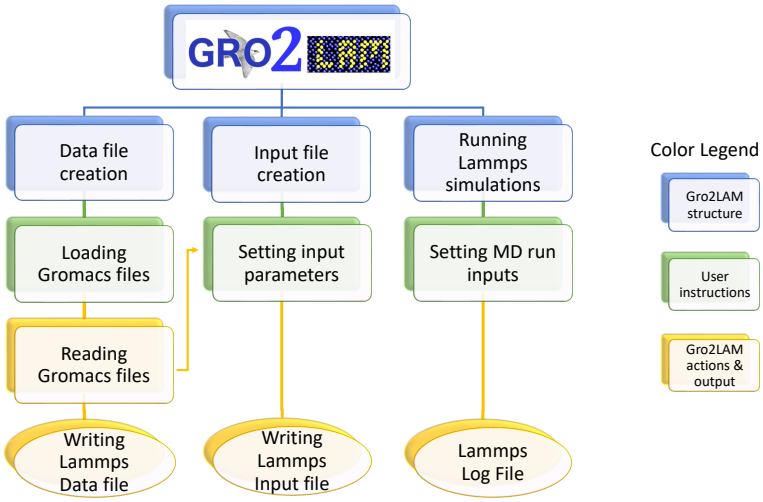

Fig. 1 Flow chart of the GRO2LAM code.

2.2 Functionalities

2.2.1 Data file creation

An intuitive Graphical User Interface (GUI), organized in three windows, guides the user towards a complete conversion of the input files from GROMACS to LAMMPS format. The *Data File Creation* window is employed to load the GROMACS input files and convert the geometry and force field information into the LAMMPS data file (see Fig. 2). Specifically, the atom coordinates and the simulation box size are imported from the `*.gro` file, while atom types, masses and partial charges are read from the `*.top` file. The whole force field folder with the relative files can be automatically loaded by the *autoload* button; alternatively, the force field files can be also provided by the user one by one.

The bond, angle and dihedral potential parameters are taken from the `*.itp` file including bonded coefficients. In particular, the current version of GRO2LAM can handle the following potentials:

- Bond potential: harmonic and Morse.
- Angle potential: harmonic and Urey-Bradley.
- Proper dihedral potential: harmonic, Fourier and Ryckaert-Bellemans.
- Improper dihedral potential: harmonic.

It is worth noticing that the harmonic constants in LAMMPS format (k_{LAMMPS}) already include the usual 1/2 factor; thus, both bond and angle coefficients from GROMACS ($k_{GROMACS}$) are converted as:

$$k_{LAMMPS} = \frac{k_{GROMACS}}{2}. \tag{1}$$

To model van der Waals interactions, GRO2LAM reads from the standard GROMACS force field file (`forcefield.itp`) whether a 12-6 Lennard-Jones or a

Buckingham potential [38] should be utilized. The 12-6 Lennard-Jones interaction (V_{LJ}) can be imported in the typical form:

$$V_{LJ}(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right], \tag{2}$$

where r_{ij} is the distance between i-th and j-th atoms, σ and ϵ are the Lennard-Jones parameters imported from the specific `*.itp` file including non-bonded coefficients. The code is also capable of handling the 12-6 Lennard-Jones potential in the A, B form, namely

$$V_{LJ}(r_{ij}) = \frac{A}{r_{ij}^{12}} - \frac{B}{r_{ij}^{6}}, \tag{3}$$

where $\sigma = (A/B)^{\frac{1}{6}}$ and $\epsilon = B/4\sigma^6$. The Buckingham potential (V_{BU}) is instead implemented as:

$$V_{BU}(r_{ij}) = Ce^{-r_{ij}/\rho} - \frac{D}{r_{ij}^{6}}, \tag{4}$$

where the energy term C, the distance term ρ and the D coefficient are parameters specified in the non-bonded `*.itp` file.

In the *Data File Creation* window, the user can also choose the desired *atom style* to be implemented in LAMMPS (see Fig. 2). Such information regards the force field structure used during the molecular dynamics simulations. For example, the `atomic style` may be selected to implement only van der Waals interactions; the `angle style` should be used to consider both van der Waals interactions and bond and angle potentials; while the `full style` should be set to include van der Waals and electrostatic interactions in combination with bond, angle and dihedrals potentials. If `full style` is chosen as input option, the intermolecular electrostatic term (V_{Coul}) is implemented as:

$$V_{Coul}(r_{ij}) = \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{\epsilon_r r_{ij}}, \tag{5}$$

where q_i and q_j are the charges of the i-th and j-th atoms, respectively, ϵ_0 is the vacuum permittivity and ϵ_r the relative permittivity. Further options of *atom style* are also available in GRO2LAM (see the LAMMPS manual for additional details). Finally, we specify that – in the current version of GRO2LAM – only three-points water models, e.g. SPC, SPC/E [39] or TIP3P, can be treated.

After importing all information about geometry and force field, GRO2LAM writes the LAMMPS data file.

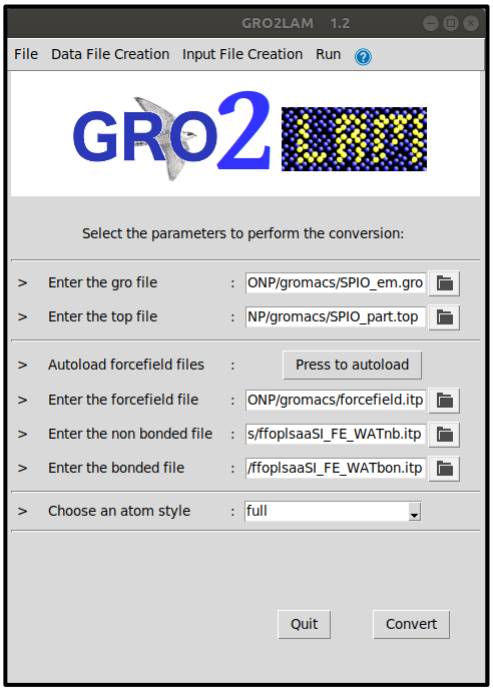

Fig. 2 Screenshot of the Graphical User Interface (GUI) in GRO2LAM. The *Data File Creation* window is reported.

2.2.2 Input file creation and run

Once the creation of the data file is completed, GRO2LAM automatically opens the *Input File Creation* window, which contains the main settings for running the MD simulation in LAMMPS (see Fig. 3).

First, all information previously collected in the data file are loaded, while the desired time step of integration can be defined through the GUI. Moreover, a default periodic boundary condition along x, y, and z directions is implemented and translated in the input file. In the *Input File Creation* window, the user manages the relevant parameters describing the statistical ensembles to be adopted during the simulation, namely:

- *NVE* ensemble, where the number of time steps is required;
- *NVT* ensemble, where the number of time steps, the system temperature ramp, and the damping factor should be set;
- *NPT* ensemble, where both the temperature and pressure with the respective damping factors are asked [22].

The *Advanced setting* icon, found in the *Input File Creation* window (see Fig. 3), can be manually selected by the user to complete the writing process of the input file. Here, GRO2LAM allows to define the order of MD simulations to be performed in the previously defined statistical ensembles. It is worth noting that, in the NPT and NVT ensembles, Nose-Hoover thermostat [40, 41] and Parrinello-Rahman barostat [42] are chosen by default; whereas, Verlet velocity algorithm

is considered for integrating the equations of motion. Although such settings are implemented by default, simple modifications of the Python 2 code can be applied in order to customize the converter to the user's needs. For instance, the user can implement additional thermostatting or barostatting options by changing the `write_lammps_input` function in the `lammps.py` file. Moreover, the GUI allows to fix simulation parameters such as the cut-off radius for the pairwise interactions, the long-range interaction solver with the relative error, and the update of neighbouring searching algorithm. The desired Lennard-Jones mixing rules (arithmetic, geometric, or sixth-power), and the coefficient for the 1-4 pair interactions can be specified within the *Advanced settings* window. Note that, the default options for the latter parameters are imported from the standard GROMACS `forcefield.itp` file loaded in the previous step, and the user is informed by a warning message that any manual modification of them would alter the simulation conditions with respect to the original files.

Finally, the creation of input file can be completed by adding position restraints and constraints. Particularly, the user can set both harmonic restraints to specific groups of atoms in the selected simulation step, as well as bond and angle constraints according to the SHAKE algorithm [43].

The third window in the GUI of GRO2LAM is dedicated to automatically run the input file created in the previous steps (see Fig. 4). The option to select the number of computational cores for running the simulation is also provided in this window. Note that, before launching the MD simulation, the user should take care of the installation of the LAMMPS software on the local machine.

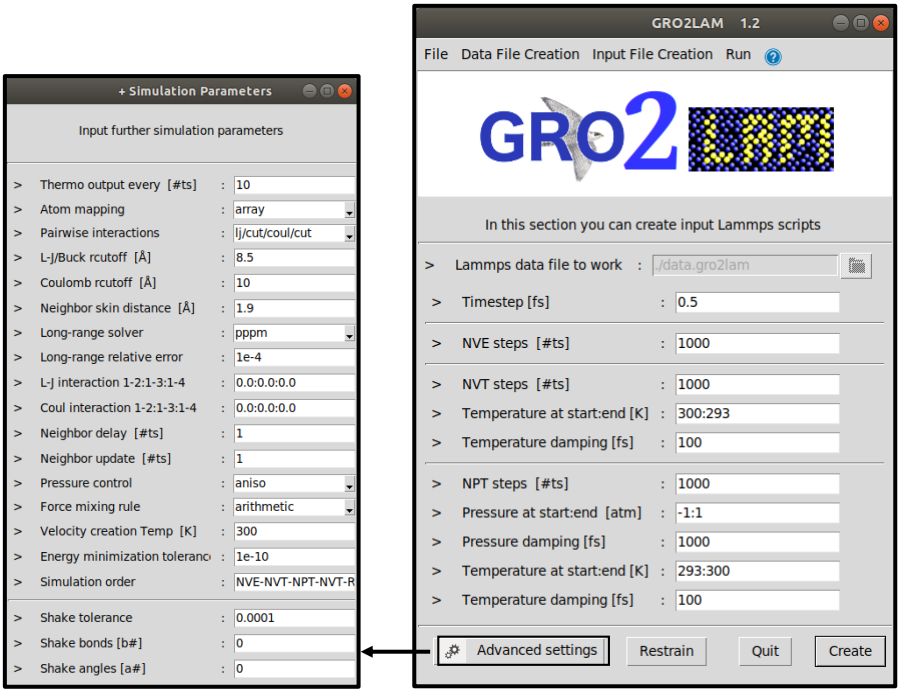

Fig. 3 Screenshot of the Graphical User Interface (GUI) in GRO2LAM. The *Input File Creation* and the related *Advanced settings* windows are reported.

Fig. 4 Screenshot of the Graphical User Interface (GUI) in GRO2LAM. The *Run* window is reported.

3 Results

In this section we present three MD study cases used to validate GRO2LAM: a Carbon Nanotube (CNT), an Iron Oxide Nanoparticle (IONP), and a protein solvated in a water box. First, molecular dynamics simulations of these setups are carried out using GROMACS; second, GRO2LAM is employed to convert the GROMACS input files into LAMMPS input files (processing time: less than one second); third, MD simulations of the converted setups are performed in LAMMPS. Finally, the results obtained from the two MD engines are compared to assess the robustness of the format conversion process. To be consistent with the most known conversion software in the literature [34,37], the validation of GRO2LAM has been based on the comparison of the mean energies computed in the LAMMPS and GROMACS runs. For the sake of completeness, a quantitative text comparison of the input files with the converted ones is also carried out for the analysed cases.

3.1 Carbon nanotube

CNTs find many applications in biomedical and engineering fields [44–47]. The system studied here consists of a single-walled CNT made up of 480 carbon atoms, which is solvated in 1561 molecules of SPC/E water [39]. The initial configuration is built in GROMACS, and it is represented in Fig. 5.

Since the objective of the analysis is mainly related to assess the robustness of the conversion algorithm, here we model carbon bonds by a simplified harmonic potential, which has a uniform C-C equilibrium distance of 0.142 nm and a single energy constant of 478900 kJ/mol/nm^2 [48]. Moreover, we consider an equilibrium angle of 120° with harmonic constant equal to 562.2 kJ/mol/rad^2 for all C-C-C angles and neglect all dihedrals. Finally, Lennard-Jones parameters of carbon atoms are set to $\sigma = 0.355$ nm and $\epsilon = 0.292$ kJ/mol, while partial charges are set to zero [48]. The bond length and angle of water molecules are constrained with the SHAKE algorithm [43]. Water-water electrostatic interactions are modelled by Coulomb potential, with the long-range term computed by either Particle Mesh Ewald (PME, GROMACS implementation [49]) or Particle-Particle-Particle-Mesh (PPPM, LAMMPS implementation [50]) methods. Although the electrostatic long-range interactions are computed by different approaches, we expect to observe no relevant differences among the energy values computed by the two MD software, since PME method is simply an extension of PPPM one. A quantitative indication of the relative error in the electrostatic energy computed with the two methods has been detailed elsewhere [37]. The cut-off radius of non-bonded interactions is set to 1.3 nm, while the skin distance of neighbours to 0.1 nm. Periodic boundary conditions are applied along all box directions.

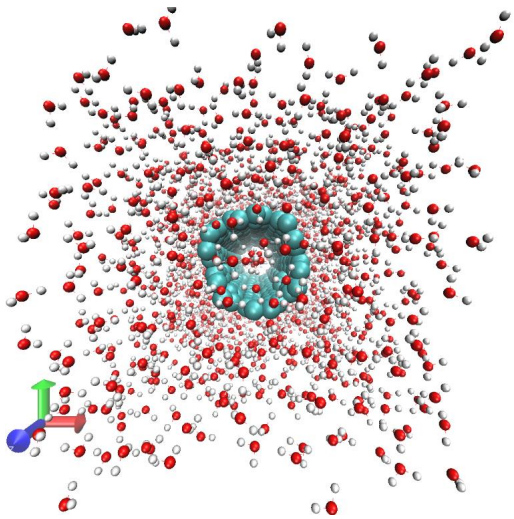

Fig. 5 Frontal view in the xy plane of the simulated single-walled carbon nanotube (CNT) in water. The image has been created by VMD software [26] with the following color code: cyan, carbon atoms; red, oxygen atoms; white, hydrogen atoms.

First, the above mentioned input details are summarized in GROMACS topology and force field files, then GRO2LAM is used to generate the data and input files compatible with LAMMPS format, and finally the same MD simulation protocol is performed with both software separately. Specifically, the solvated system is first energy minimized using a steepest descent algorithm. Atomic velocities are then initialized according to Maxwell-Boltzmann distribution at 300 K. Positions of carbon atoms are restrained by a harmonic potential with constant $k = 1000\,\mathrm{kJ/mol/nm^2}$, and the system is equilibrated in two different stages: i) 100 ps of NVT thermalization at 300 K using the Berendsen thermostat with 0.1 ps time constant; ii) 50 ps of isothermal-isobaric (NPT) equilibration, using the Nose-Hoover thermostat ($T = 300$ K and 0.2 ps time constant) and Parrinello-Rahman barostat ($p = 1$ bar and 2 ps time constant). After these equilibration steps, restraints are removed from carbon atoms and the MD simulation is continued up to 150 ps in the NPT ensemble (1 fs time step).

The last 100 ps of MD trajectories obtained by both software are post-processed in Matlab® to extract energies. Figure 6 shows the energy ratios ($Ratio_E$) as a function of simulation time, which are obtained as

$$Ratio_E = \frac{E_{LAMMPS}}{E_{GROMACS}}, \tag{6}$$

where E_{LAMMPS} and $E_{GROMACS}$ are either bonded or non-bonded potential energies computed in LAMMPS and GROMACS, respectively. In particular, the top panels in Fig. 6 show angle (E_{ang}) and bond (E_{bon}) interaction energies; whereas, bottom plots refer to van der Waals (E_{vdW}) and Coulomb (E_{Coul}) interaction energies. Note that the total energy is the summation of the following terms:

$$E_{tot} = E_{kin} + E_{ang} + E_{bon} + E_{vdW} + E_{Coul}, \tag{7}$$

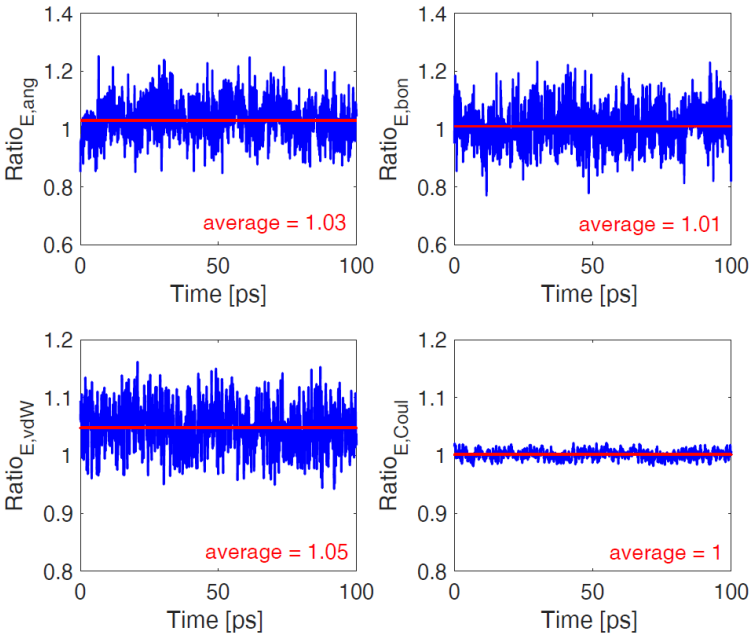

Fig. 6 Ratio between the energies of the solvated carbon nanotube obtained by LAMMPS and GROMACS simulations. Top panels (from left to right): energy given by angle (E_{ang}) and bond (E_{bon}) interaction potentials. Bottom panel (from left to right): energy given by van der Waals (E_{vdW}) and Coulomb (E_{Coul}) interaction potentials.

being E_{kin} the kinetic energy and

$$E_{Coul} = E_{Coul_{SR}} + E_{Coul_{LR}}. \tag{8}$$

In the latter equation, E_{Coul} is the overall energy due to electrostatic interactions, which can be in turn split into short-range ($E_{Coul_{SR}}$) and long-range ($E_{Coul_{LR}}$) Coulomb interactions.

The reported results show a good agreement between the energies computed by GROMACS and LAMMPS, with oscillations around the unit ratio due to the stochastic nature of velocity initialisation in MD simulations. In addition, we found that the average of total energy ratio ($Ratio_{E,tot}$, where E_{tot} is given by Eq. (7)) is equal to 1.01. Considering the different algorithms used to evaluate long-range interactions, the obtained results confirm the robustness of the GRO2LAM converter. Furthermore, the consistency of conversion is also verified by quantitatively comparing the main characteristics of input files (*i.e.* GROMACS original vs. LAMMPS conversion), as detailed in Tab. 1 in Appendix 2.

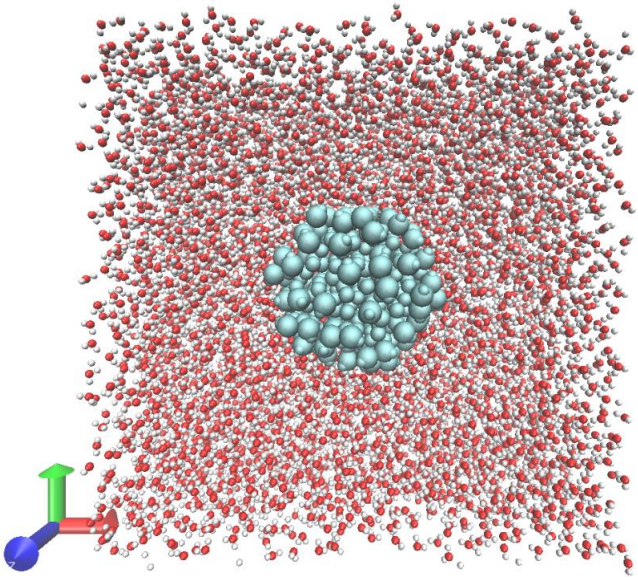

Fig. 7 Snapshot of the iron oxide nanoparticle solvated in water. The image has been created using the VMD software [26]. For the sake of clarity, water molecules at $z > z_{max}/2$ ($z_{max} = 7\,\text{nm}$) have been hidden in the current visualization. Color code: cyan, magnetite atoms; red, oxygen atoms; white, hydrogen atoms.

3.2 Iron oxide nanoparticle

The second test case deals with the equilibrium MD simulation of an iron oxide nanoparticle (IONP) in water. This setup is of particular interest because IONP are currently studied for both diagnostic and therapeutic biomedical applications [9]. For example, these nanoparticles can assist hyperthermia treatments, which – when used together with radiotherapy and chemotherapy – have shown exceptional results in killing breast cancer, melanoma, head tumours, cervix cancer and glioblastom [51]. In fact, thanks to their magnetic properties, IONPs can be accumulated in a specific tissue and then heated up by magnetic field, therefore inducing a localized heat transfer only in the target region of hyperthermia [52].

The IONP investigated here is made of magnetite (Fe_3O_4, 413 atoms), and it is solvated in a box of 11200 SPC/E water molecules. The dimensions of the simulation box are 7 nm $\times$ 7 nm $\times$ 7 nm, and the surface of nanoparticle presents hydroxyl groups [48]. The whole system includes 10 atom types, 8 bond types and 17 angle types (see Tab. 2 in Appendix 3). The energy minimization step has been realized similarly to the previous CNT simulation. Figure 7 shows a snapshot of the system after energy minimization.

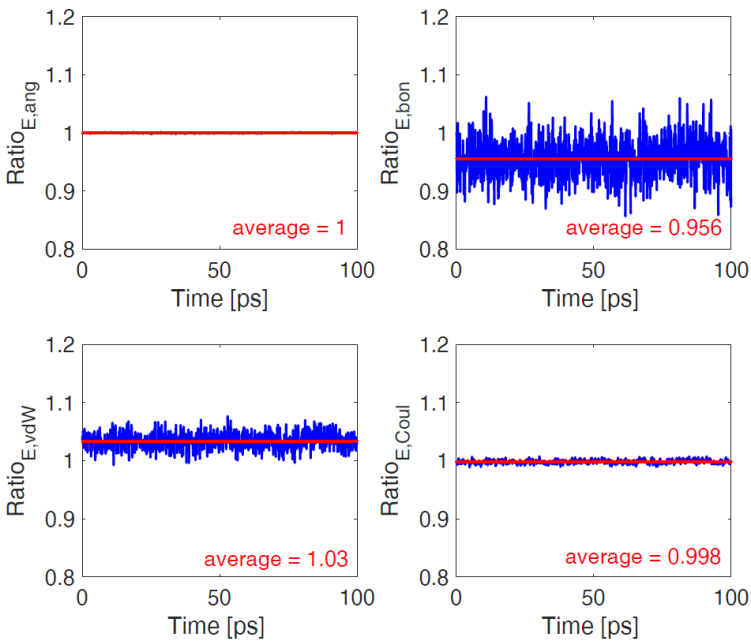

Fig. 8 Ratios between the energies of the solvated iron oxide nanoparticle obtained from GROMACS and LAMMPS simulations. Top panels (from left to right): energy given by angle E_{ang} and bond E_{bon} interaction potentials. Bottom panel (from left to right): energy given by van der Waals E_{vdW} and Coulomb E_{Coul} interaction potentials.

To decrease the complexity of the test case, we decided to set all the bond and angle harmonic constants to 400000 kJ/mol/nm^2 and 400 kJ/mol/rad^2, respectively. In Tab. 3 and Tab. 4 in Appendix 3, the considered equilibrium bond lengths and angles are shown, according to the crystal structure of the magnetite [48]. Non-bonded interactions are modelled by Lennard-Jones and Coulomb potentials, whose parameters are reported in Tab. 2 in Appendix 3 [48]. The protocol adopted for the MD simulation of the solvated IONP is the same one described in the CNT case study: both GROMACS and LAMMPS simulation results are compared to assess the effectiveness of GRO2LAM in converting data and input files.

The results obtained from the two MD engines are separately post-processed, and the ratios between the energies measured in GROMACS and LAMMPS simulations are shown in Fig. 8. For the iron oxide nanoparticle, we found that the average of the total energy ratio ($Ratio_{E,tot}$) is equal to 1.06. The accurate matching between the computed energies demonstrates once again the robustness of the GRO2LAM conversion, as also proved by the quantitative comparison of input files characteristics reported in Tab. 1 in Appendix 2.

3.3 Lysozyme protein

The third study case includes the conversion of a solvated protein, namely Lysozyme in water. The protein geometry has been downloaded from the RCSB of the Protein Data Bank (https://www.wwpdb.org/) [53] (1AKI), and the topology has been completed by including the OPLS-AA force field [54]. The Lysozyme has been solvated in a box of water, where solvent molecules are described by SPC/E model [39]. To achieve the electrostatic neutrality of the system, ions are added before minimizing the energy. The resulting configuration of the solvated protein and the related force field files have been provided to GRO2LAM for the conversion. A snapshot of the Lysozyme in water is reported in Fig. 9.

Once the data and input files are obtained, two parallel MD simulations have been carried out in GROMACS and LAMMPS, respectively. Specifically, a single simulation step of 100 ps has been performed to equilibrate the system in the canonical ensemble (v-rescale thermostat on both water and protein at 300 K). During the equilibration, the protein atoms have been restrained with a harmonic potential. The production run has been carried out for 40 ps in NVT ensemble by applying a Nose-Hoover thermostat [40, 41] at 300 K. Electrostatic and van der Waals interactions are modelled with Coulomb and LJ potential according to the parameters defined in the OPLS-AA force field. During the production run, the restraints are removed from protein's atoms.

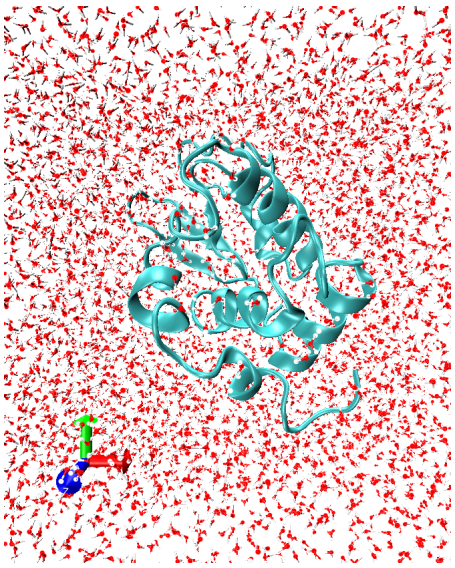

Fig. 9 Snapshot of the Lysozyme protein (PDB ID: 1AKI) solvated in water. The image has been created using the VMD software [26]. For the sake of clarity, water molecules at $z > z_{max}/2$ ($z_{max} = 7\,\text{nm}$) have been hidden in the current visualization. Color code: cyan, protein; red, oxygen atoms; white, hydrogen atoms.

The results obtained from the two MD engines are separately post-processed, and the ratios between the energies measured in GROMACS and LAMMPS simulations are shown in Fig. 10. The accurate matching between the computed en-

ergies demonstrates once again the robustness of the GRO2LAM conversion. The consistency of conversion is also verified by quantitatively comparing the main characteristics of input files, as reported in Tab. 1 in Appendix 2.

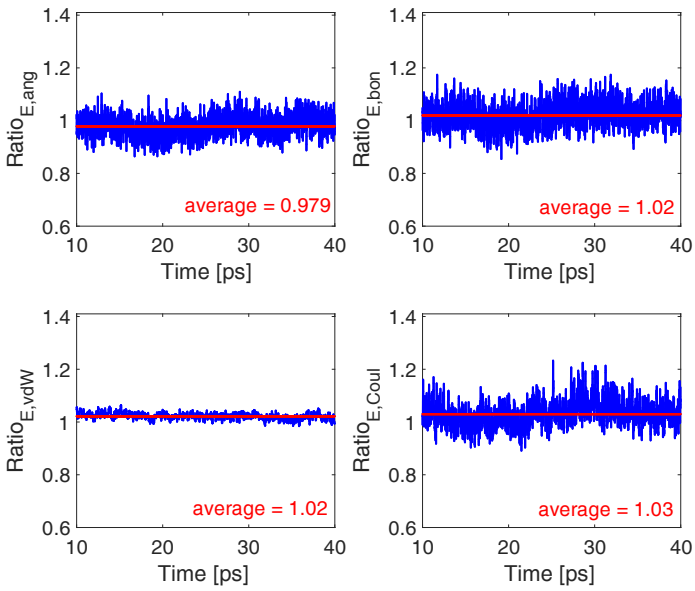

Fig. 10 Ratios between the energies of the solvated Lysozyme obtained from GROMACS and LAMMPS simulations. Top panels (from left to right): energy given by angle E_{ang} and bond E_{bon} interaction potentials. Bottom panel (from left to right): energy given by van der Waals E_{vdW} and Coulomb E_{Coul} interaction potentials.

4 Discussion

GROMACS and LAMMPS are emerging as the most complete, flexible and widely used open-source software to carry out atomistic simulations. However, the incompatibility between the different formats of their input files and parameters is still limiting a fast and reliable transferability of the simulations from one software to the other. Moreover, the complexity and the management of different input files strongly demotivate the integrated use of the two software. The possibility to easily transfer geometry, force field and simulation parameters from GROMACS to LAMMPS format is in fact desirable to fully exploit the complementary simulation protocols and post-processing functionalities implemented in the two software. The perspective improvement of the interoperability between MD platforms will help data reproducibility and independence of the simulation results. Additionally, a strong communication between GROMACS and LAMMPS would benefit the connection between scientists from different communities, such as bioinformatics, engineers and material scientists. This added value would differentiate

GRO2LAM from previous translation software, which were mainly focused on biomaterials [33, 37].

GRO2LAM is a modular Python 2 code, supported by a clear and intuitive GUI, able to translate input files and parameters from GROMACS to LAMMPS format. The aim of the code is to build the core of an extendible open-source software, which can be successively upgraded to include a broader variety of molecular dynamics setups and force fields. While its modular structure allows future upgrades and extensions, the current release of GRO2LAM comes with the following features:

- conversion from GROMACS (version 5) to LAMMPS;
- three-points water models;
- ions;
- Coulomb, 12-6 Lennard-Jones and Buckingham non-bonded interactions;
- harmonic and Morse bond potentials;
- harmonic and Urey-Bradley angle potentials;
- periodic, Fourier and Ryckaert-Bellemans proper dihedral potentials;
- harmonic improper dihedral potentials;
- integration with standard GROMACS force fields, e.g. OPLS-AA [54].

5 Conclusions

In this article we have presented GRO2LAM, a Python 2 software with an intuitive graphical user interface that automates the porting of input and data files between two of the most commonly used molecular dynamics software: GROMACS and LAMMPS. After importing the data from GROMACS input files, the code generates the molecular dynamics data and input files in the LAMMPS format. The robustness of GRO2LAM has been verified by three molecular dynamics test cases of interest in both biomedical and engineering fields: a carbon nanotube, an iron oxide nanoparticle, and a protein in water. First, MD simulations have been performed using GROMACS; then, input files have been converted by GRO2LAM; finally, MD simulations have been carried out in LAMMPS. Results showed a good agreement between the energies obtained by the two different molecular dynamics software, proving the robustness of the GRO2LAM conversion. A quantitative text comparison between the original (GROMACS) and converted (LAMMPS) input files further proved the correctness of the conversion.

The GRO2LAM package is extremely user friendly, modular, open-source and easy to be modified and extended to further MD setups, such as proteins and biosystems. In perspective, GRO2LAM would allow to easily exploit the complementary simulation protocols and post-processing functionalities of different molecular dynamics software. The interoperability across different MD platforms will facilitate the reproducibility of simulation data obtained with different codes and algorithms.

Acknowledgements

The authors acknowledge the high-performance computing initiative at Politecnico di Torino (HPC@Polito) and the CINECA Iscra C projects MISURPAC

(HP10CJOR5E) and NANOCLUS (HP10CYC6UC) for the availability of high-performance computing resources and support. The authors would also like to acknowledge Dr. Rajat Srivastava for his useful suggestions. The authors declare no competing financial interests. This is a post-peer-review, pre-copyedit version of an article published in Journal of Molecular Modeling. The final authenticated version is available online at: https://doi.org/10.1007/s00894-019-4011-x

Appendix 1: Installation guide

The following instructions pertain to the download and installation of the latest version of GRO2LAM:

1. Download the latest version of GRO2LAM from: https://github.com/hernanchavezthielemann/GRO2LAM
2. Extract the downloaded version in the GRO2LAM folder;
3. Execute the setup file;
4. Execute the run script.

Equivalently, the following sequence of bash commands can be used:

- ```
  $ wget https://github.com/hernanchavezthielemann/
      GRO2LAM/archive/27ene19.zip
  ```

- ```
  $ unzip 27ene19.zip
  ```

- ```
  $ cd GRO2LAM-27ene19
  ```

- ```
  $ python setup
  ```

- ```
  $ ./run
  ```

Notice that the name of the installation zip file may change with the current version of GRO2LAM, which should be always checked at the related GitHub page before starting the installation.

Alternatively, the user could also copy and paste the following command in the bash console and execute it, and this will download and execute the latest version of GRO2LAM all at once:

- ```
  $ wget https://raw.githubusercontent.com/
      hernanchavezthielemann/utils/master/grotolam/
      G2L_installer && bash G2L_installer
  ```

Appendix 2: Comparison of input files characteristics

	CNT in water		IONP in water		PROTEIN in water	
	GROMACS original	*LAMMPS converted*	*GROMACS original*	*LAMMPS converted*	*GROMACS original*	*LAMMPS converted*
Atoms	5163	5163	34013	34013	33876	33876
Non-water atoms	480	480	413	413	1968	1968
Oxygen atoms	1561	1561	11200	11200	10821	10821
Water molecules	1561	1561	11200	11200	10636	10636
Bonds	3830	3830	23038	23038	1984	1984
Angles	2953	2953	13111	13111	3547	3547
Dihedrals	708	708	554	554	5187	5187
VdW pot. types	5	5	13	13	813	813
Stretch pot. types	6	6	14	14	300	300
Angle pot. types	13	13	28	28	929	929
Dihedral pot. types	23	23	78	78	1202	1202
Overall charge	0	0	0	0	0	0

Table 1 Consistency check between input files from GROMACS and files converted into LAMMPS language by GRO2LAM, for the CNT, IONP and PROTEIN study cases. The number of atoms, non-water atoms, etc. are reported, as well as the overall charge of the simulation box. For the case of protein, the vdW, stretch, angle and dihedral potential types refer to the whole OPLS-AA force field, which is entirely converted by GRO2LAM. The overall charge for the solvated protein is considered after ion neutralization.

Appendix 3: Force field for iron oxide nanoparticle

Atom type	σ[nm]	ϵ[kJ/mol]	q [e]
H	0.00	0.00	0.40
Fe2	0.43	24.94	0.00
Fe3	0.43	24.94	0.00
FeS2	0.43	24.94	1.21
FeS3	0.43	24.94	-1.13
OFe	0.38	5.34	0.00
OHF2	0.38	5.34	-1.61
OHF3	0.38	5.34	-1.13

Table 2 Non-bonded parameters of iron oxide nanoparticles [48]: σ and ϵ are used to describe the 12-6 Lennard-Jones potential, while q (expressed in terms of elementary charge) to compute the Coulomb interactions. The atom types are defined as: H, hydrogen atom in the surface hydroxyl groups; OHF2 and OHF3, oxygen atoms in the surface hydroxyl groups; OFe, oxygen atom in the magnetite core; Fe2 and Fe3, divalent and trivalent iron atoms in the magnetite core, respectively; FeS2 and FeS3, divalent and trivalent iron atoms on the magnetite surface, respectively.

Bond type	L[nm]
Fe2 - OFe	0.188
Fe3 - OFe	0.206
FeS2 - OFe	0.188
FeS3 - OFe	0.206
FeS2 - OHF2	0.189
FeS3 - OHF3	0.206
H - OHF2	0.096
H - OHF3	0.096

Table 3 Equilibrium distances (L) of the bond interaction potentials adopted in the iron oxide nanoparticle simulations [48].

Angle type	θ [deg]
OFe - Fe2 - OFe	110.0
OFe - Fe3 - OFe	87.4
OFe - FeS2 - OFe	109.0
OFe - FeS3 - OFe	87.6
OHF2 - FeS2 - OFe	110.0
OHF3 - FeS3 - OFe	92.3
OHF2 - FeS2 - OHF2	109.0
OHF3 - FeS3 - OHF3	87.6
Fe2 - OFe - Fe3	124.0
Fe2 - OFe - FeS3	124.0
FeS2 - OFe - Fe3	124.0
FeS2 - OFe - FeS3	124.0
Fe3 - OFe - Fe3	92.4
FeS3 - OFe - Fe3	92.1
FeS3 - OFe - FeS3	92.0
FeS2 - OHF2 - H	129.0
FeS3 - OHF3 - H	129.0

Table 4 Equilibrium angles (θ) of the angle interaction potentials adopted in the iron oxide nanoparticle simulations [48].

References

1. S. Lin, C.J. Shih, V. Sresht, A.G. Rajan, M.S. Strano, D. Blankschtein, Advances in colloid and interface science **244**, 36 (2017)
2. A.G. Rajan, K.S. Silmore, J. Swett, A.W. Robertson, J.H. Warner, D. Blankschtein, M.S. Strano, Nature materials **18**, 129 (2019)
3. M.B. Bigdeli, M. Fasano, International Journal of Thermal Sciences **117**, 98 (2017)
4. M. Fasano, D. Borri, E. Chiavazzo, P. Asinari, Applied Thermal Engineering **101**, 762 (2016)
5. M.B. Bigdeli, M. Fasano, A. Cardellini, E. Chiavazzo, P. Asinari, Renewable and Sustainable Energy Reviews **60**, 1615 (2016)
6. M. Fasano, T. Humplik, A. Bevilacqua, M. Tsapatsis, E. Chiavazzo, E.N. Wang, P. Asinari, Nature communications **7**, 12762 (2016)

7. M. Ahmadi, A. Nowroozi, M. Shahlaei, Journal of Molecular Graphics and Modelling **61**, 243 (2015)
8. A. Cardellini, M. Fasano, E. Chiavazzo, P. Asinari, Physics Letters A **380**(20), 1735 (2016)
9. A. Gizzatov, J. Key, S. Aryal, J. Ananta, A. Cervadoro, A.L. Palange, M. Fasano, C. Stigliano, M. Zhong, D. Di Mascolo, et al., Advanced functional materials **24**(29), 4584 (2014)
10. S.K. Choubey, J. Jeyaraman, Journal of Molecular Graphics and Modelling **70**, 54 (2016)
11. R. Kamali, A. Kharazmi, Computer Physics Communications **184**(10), 2316 (2013)
12. I.A. Cosden, J.R. Lukes, Computer Physics Communications **184**(8), 1958 (2013)
13. E. Somogyi, A.A. Mansour, P.J. Ortoleva, Computer Physics Communications **202**, 337 (2016)
14. P. Neumann, H. Flohr, R. Arora, P. Jarmatz, N. Tchipev, H.J. Bungartz, Computer Physics Communications **200**, 324 (2016)
15. F. Mackay, S.T. Ollila, C. Denniston, Computer Physics Communications **184**(8), 2021 (2013)
16. L. Bergamasco, S. Izquierdo, A. Ammar, Journal of Non-Newtonian Fluid Mechanics **201**, 29 (2013)
17. L. Bergamasco, S. Izquierdo, I. Pagonabarraga, N. Fueyo, Chemical Engineering Science **126**, 471 (2015)
18. A. Cardellini, M. Fasano, M.B. Bigdeli, E. Chiavazzo, P. Asinari, Journal of Physics: Condensed Matter **28**(48), 483003 (2016)
19. A.S. Tascini, J. Armstrong, E. Chiavazzo, M. Fasano, P. Asinari, F. Bresme, Physical Chemistry Chemical Physics **19**(4), 3244 (2017)
20. M. Morciano, M. Fasano, A. Nold, C. Braga, P. Yatsyshin, D.N. Sibley, B.D. Goddard, E. Chiavazzo, P. Asinari, S. Kalliadasis, The Journal of Chemical Physics **146**(24), 244507 (2017)
21. P. Allen, D. Tildesley, *Computer Simulation of Liquids*. Oxford Science Publ (Clarendon Press, 1989)
22. D. Frenkel, B. Smit, *Understanding molecular simulation: from algorithms to applications*, vol. 1 (Academic press, 2001)
23. M. Fasano, M.B. Bigdeli, M.R.V. Sereshk, E. Chiavazzo, P. Asinari, Renewable and Sustainable Energy Reviews **41**, 1028 (2015)
24. B. Alder, T. Wainwright, The Journal of Chemical Physics **27**(5), 1208 (1957)
25. A. Rahman, Physical Review **136**(2A), A405 (1964)
26. W. Humphrey, A. Dalke, K. Schulten, Journal of molecular graphics **14**(1), 33 (1996)
27. J.C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R.D. Skeel, L. Kale, K. Schulten, Journal of computational chemistry **26**(16), 1781 (2005)
28. D.A. Pearlman, D.A. Case, J.W. Caldwell, W.S. Ross, T.E. Cheatham, S. DeBolt, D. Ferguson, G. Seibel, P. Kollman, Computer Physics Communications **91**(1-3), 1 (1995)
29. B. Hess, C. Kutzner, D. Van Der Spoel, E. Lindahl, Journal of chemical theory and computation **4**(3), 435 (2008)
30. S. Plimpton, Journal of computational physics **117**(1), 1 (1995)
31. J. Tersoff, Physical Review B **39**(8), 5566 (1989)
32. H. Sun, The Journal of Physical Chemistry B **102**(38), 7338 (1998)
33. J.V. Vermaas, D.J. Hardy, J.E. Stone, E. Tajkhorshid, A. Kohlmeyer, Journal of Chemical Information and Modeling **56**(6), 1112 (2016)
34. V.H. Rusu, V.A. Horta, B.A. Horta, R.D. Lins, R. Baron, Journal of Molecular Graphics and Modelling **48**, 80 (2014)
35. W. Smith, C. Yong, P. Rodger, Molecular Simulation **28**(5), 385 (2002)
36. C.W. Yong. Descriptions and implementations of dl_f notation: a natural chemical expression system of atom types for molecular simulations (2016)
37. M.R. Shirts, C. Klein, J.M. Swails, J. Yin, M.K. Gilson, D.L. Mobley, D.A. Case, E.D. Zhong, Journal of computer-aided molecular design **31**(1), 147 (2017)
38. R.A. Buckingham, in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 168 (The Royal Society, 1938), vol. 168, pp. 264–283
39. H. Berendsen, J. Grigera, T. Straatsma, Journal of Physical Chemistry **91**(24), 6269 (1987)
40. S. Nosé, The Journal of chemical physics **81**(1), 511 (1984)
41. W.G. Hoover, Physical Review A **31**(3), 1695 (1985)
42. M. Parrinello, A. Rahman, Journal of Applied physics **52**(12), 7182 (1981)
43. J.P. Ryckaert, G. Ciccotti, H.J. Berendsen, Journal of Computational Physics **23**(3), 327 (1977)

44. M. Fasano, E. Chiavazzo, P. Asinari, Nanoscale research letters **9**(1), 1 (2014)
45. S.S. Razavi, S.M. Hashemianzadeh, H. Karimi, Journal of molecular modeling **17**(5), 1163 (2011)
46. K. Salazar-Salinas, C. Kubli-Garfias, J.M. Seminario, Journal of molecular modeling **19**(7), 2797 (2013)
47. M. Fasano, A. Crisafulli, A. Cardellini, L. Bergamasco, E. Chiavazzo, P. Asinari, Molecular Simulation **45**(4-5), 417 (2018)
48. E. Chiavazzo, M. Fasano, P. Asinari, P. Decuzzi, Nature communications **5**, 3565 (2014)
49. T. Darden, D. York, L. Pedersen, The Journal of chemical physics **98**(12), 10089 (1993)
50. R. Hockney, J. Eastwood, *Computer Simulation Using Particles* (CRC Press, 1988)
51. M. Falk, R. Issels, International Journal of Hyperthermia **17**(1), 1 (2001)
52. L. Bergamasco, M. Alberghini, M. Fasano, A. Cardellini, E. Chiavazzo, P. Asinari, Entropy **20**(2), 126 (2018)
53. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, Nucleic acids research **28**(1), 235 (2000)
54. G.A. Kaminski, R.A. Friesner, J. Tirado-Rives, W.L. Jorgensen, The Journal of Physical Chemistry B **105**(28), 6474 (2001)