

MPolSys Modeler, a tool for computational modeling of linear polymer systems

MPolSys Modeler, uma ferramenta para modelagem computacional de sistemas poliméricos lineares

MpolSys Modeler, una herramienta para el modelado computacional de sistemas de polímeros lineales

Received: 11/28/2021 | Reviewed: 12/04/2021 | Accept: 12/05/2021 | Published: 12/14/2021

Filipe Morais Frade de Faria

ORCID: <https://orcid.org/0000-0002-2905-6441>

Federal Institute of Minas Gerais, Brazil

E-mail: filipe.frade@hotmail.com

Reginaldo Gonçalves Leão Junior

ORCID: <https://orcid.org/0000-0002-3596-2256>

Federal Institute of Minas Gerais, Brazil

E-mail: reginaldo.junior@ifmg.edu.br

Abstract

The computational study of intermolecular relationships of a given material can be used as a route for predicting quantities impossible or difficult to be determined experimentally. Furthermore properties of new materials can also be predicted by techniques of this type, when they are still in the modeling phase. This technique reproduces the classical dynamic relationships between the constituent elements of the material, atoms or unicorpuscular approximations of molecules, from interaction potential models called force fields. This work aims to develop a tool that performs the composition of linear polymeric chain systems through a self-avoided walk. For this, the concept of self-experimentation of long walks (SAWLC) was used, together with the Python language to develop MpolSys Modeler. This tool is a non-overlapping polymer chain generator, which in turn generates outputs that can be used as input to Moltemplate. To validate the tool's results, experiments were carried out in which the numbers and polymerization chains of the simulated polymer were varied, observing the overlap or not of the molecules that make up the simulation. At the end of the simulations, there were positive results that indicate a promising usage of the tool for the creation of polymers with a high number of chains and degrees of polymerization.

Keywords: MPolSys; Moltemplate; SAWLC; Polymer.

Resumo

O estudo computacional das relações intermoleculares de um determinado material pode ser usado como uma rota para prever quantidades que são impossíveis ou difíceis de determinar experimentalmente. Propriedades de novos materiais também podem ser previstas por técnicas desse tipo, quando ainda estão em fase de modelagem. Esta técnica reproduz as relações dinâmicas clássicas entre os elementos constituintes do material, átomos ou aproximações de interações unicorpusculares de moléculas, a partir de modelos potenciais denominados campos de força. Este trabalho tem como objetivo desenvolver uma ferramenta que realiza a composição de sistemas de cadeias poliméricas lineares por meio de uma caminhada auto evitada. Para isso, foi utilizado o conceito de SAWLC (caminhadas auto-evitantes de longas distâncias), em conjunto com a linguagem Python 3 para desenvolver MPolSys Modeler. Esta ferramenta é um gerador de cadeia de polímero não sobreposto, que por sua vez gera saídas que podem ser usadas como entrada para Moltemplate. Para validar os resultados da ferramenta, foram realizados experimentos nos quais foram variados os números e as cadeias de polimerização do polímero simulado, observando-se a sobreposição ou não das moléculas que compõem a simulação. Ao final das simulações, houve resultados positivos que indicam uma possibilidade promissora de utilização da ferramenta para a criação de polímeros com elevado número de cadeias e grau de polimerização.

Palavras-chave: MPolSys; Moltemplate; SAWLC; Polímero.

Resumen

El estudio computacional de las relaciones intermoleculares de un material dado puede usarse como una ruta para predecir cantidades que son imposibles o difíciles de determinar experimentalmente. Las propiedades de los nuevos materiales también pueden predecirse mediante técnicas de este tipo, cuando aún se encuentran en la fase de modelado. Esta técnica reproduce las relaciones dinámicas clásicas entre los elementos constituyentes del material, átomos o aproximaciones de interacciones unicorpusculares de moléculas, a partir de modelos potenciales denominados campos de fuerza. Este trabajo tiene como objetivo desarrollar una herramienta que realice la composición de sistemas de cadenas poliméricas lineales a

través de un paseo autoevitado. Para ello, se utilizó el concepto de SAWLC (caminata autoevitante de largas distancias), junto con el lenguaje Python 3 para desarrollar MpolSys Modeler. Esta herramienta es un generador de cadenas de polímeros que no se superponen, que a su vez genera salidas que se pueden utilizar como entrada para Moltemplate. Para validar los resultados de la herramienta, se realizaron experimentos en los que se variaron los números y las cadenas de polimerización del polímero simulado, observando el solapamiento o no de las moléculas que componen la simulación. Al final de las simulaciones, se obtuvieron resultados positivos que indican una posibilidad prometedora de utilizar la herramienta para la creación de polímeros con alto número de cadenas y grados de polimerización.

Palabras clave: MpolSys; Moltemplate; SAWLC; Polímero.

1. Introduction

Developing computational molecular dynamics (CMD) techniques is an arduous task that is associated with many possibilities of error during the modeling process. This is due to the need to determine the spatial arrangement of the model elements in the initial condition, correct description of the connections between particles and accurate selection of the reference simulation potentials (force fields). Because of this, the input composition of multipurpose simulation systems such as Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS), a classical molecular dynamics code with a focus on materials modeling developed by Plimpton et al. (2020), is a task that is almost impossible to accomplish manually.

Moreover, as in all other fields of science, but especially in molecular dynamics, the reproducibility of experiments is the only way to guarantee the reliability of the method, data and the assertions resulting from successful research. Humbert et al. (2019) emphasizes the importance of data reproducibility by discussing the role of post-processing tools for CMD, an argument that applies even to pre-processing tools.

Gartner III and Jayaraman (2019), reviewed modeling and simulating polymers by CMD studies. They pointed out results from a wrong model, or a poorly parameterized simulation, do not appear to be obvious errors and can, even in these cases be understood as acceptable.

In this sense, and in the same line of thought as Humbert et al. (2019), the tools for CMD preprocessing, act as an interface between the developer of the molecular model and the solver used for simulation, besides facilitate the input development processes, they reduce the probability of error in the modeling stage and maximize their reproducibility. Among those available for LAMMPS, the VMD TopoTools, Avogadro, Packmol and Moltemplate, the latter adopted in this research. Kohlmeyer et al. (2020); Hanwell et al. (2012); Martínez et al. (2009); Jewett et al. (2021). Such tools, however, do not exempt the researchers from the setup of their models, because this process is highly dependent on the system to be studied and what is to be studied.

In this work we also present MPolSys Modeler acronym for Multiple Polymer Chain Systems, a complementary tool to Moltemplate for the composition of linear polymeric chain systems whose intermediate and extreme monomeric components (tail and head) have been previously created using data from some force field. The system aims to facilitate access to the self-avoided walk techniques to determine the positions of the elementary components, as well as to quantify the effectiveness of the modeling technique.

This work aims to present the main features of the current version of the software and the results of functionality tests performed.

2. Materials and Methods

2.1 Determining the shape of the chains

In our system, the shapes of the modeled chains are determined by movements of arbitrary direction governed by a kernel that determines them via self-avoiding three-dimensional wormlike chains (SAWLCs) (Kratky & Porod's, 1949). When a particle system is spatially represented, the space occupied by each component of the system constitutes a prohibitive condition for the occurrence of a

second particle in the volume occupied by the first, requiring that individual positions be unique. However, the arrangement of particles in a real system, a polymer in an arbitrary solvent for example, does not obey any deterministic condition and is completely random (Gujrati & Leonove, 2010). Because of this, the initialization of the system to be modeled should, in principle, obey both conditions so that the potential energy of the system is not excessively high to the point of causing an error in simulation time, nor so low that demanded excessive computing time or made convergence difficult (Birta & Arbez, 2013).

By definition, SAWLCs are theoretically capable of meeting both requirements, as they perform random walks (in R^3 in our case) avoiding positions that have already been covered before, and simultaneously maintaining some semi-flexibility, as is common in polymer chains. The kernel for producing the positions in the created system was developed by Stefko and Douglass (2017) in the *Laboratory of Experimental Biophysics of École polytechnique fédérale de Lausanne, Switzerland*, the system uses Eigen/C++ library for matrices, vectors, and numeric computational routines (Guennebaud & Jacob, 2010).

The Stefko and Douglas numeric method is a discretized version of Kratky and Porod's (1949) SAWLCs continuous model, it considers the chain as a series of segments of the same length. The changes in direction of each of the successive segments are determined by the zenith angles θ and azimuth φ . While the azimuth varies randomly between 0 and 2π , the zenith varies according to a probability distribution function that simulates the stiffness of the polymer. If the persistence length is taken as l_p , then the zenith is determined by Equation (1).

$$\theta = \sqrt{\frac{-2}{l_p} \ln(1 - X)} \quad (1)$$

Where X is a random variable normally distributed between 0 and 1. In our experiments the persistence length was considered equal to 5 Å.

2.2 Chemical bond size correction

Due to the maximum bond diameter generated by PolymerCpp is unity, MPolSys Modeler provides an intermonomeric bond distance correction tool, **positon_crr** function. When active, the user must provide an ideal separation distance for each of the monomers belonging to the modeled chains. It is desirable that this distance is generally consistent with those force field standards used in the study, this reduces the total energy of the system, reducing the pre-simulation energy minimization time, or even totally eliminating the need for this step.

The correction method operates on the Cartesian distance L between the monomers considering a pair of coordinates as a constant k determined by Equations (2) and (3).

$$L^2 = \Delta c_1^2 + \Delta c_2^2 + c_3^2 \quad (2)$$

$$k = \Delta c_2^2 + \Delta c_3^2 \quad (3)$$

Getting L expressed by Equation (4).

$$L^2 = \Delta c_i^2 + k \quad (4)$$

If the corrected bond length is expressed by L' , then the length correction factor in one of the orthogonal directions of the system is given by Equation (5).

$$L'^2/L^2 = (C'^2 + k)/(C^2 + k) \quad (5)$$

Where C' is the corrected length which is given by Equation (6).

$$C'^2 = (L'/L)^2 \cdot (C^2 + k) - k \quad (6)$$

The algorithmic method performs the correction of Equation (6) first in the X direction and checks if the new inside coordinate over one before it, if yes, performs the same test for Y, if the coincidence persists, performs the test for Z. If there is a coincidence for the three directions, the attribution of the correction in the Z direction is kept and the coincident values added to a list that is informed to

the user to take appropriate action.

2.3 Determining the complexity of the system

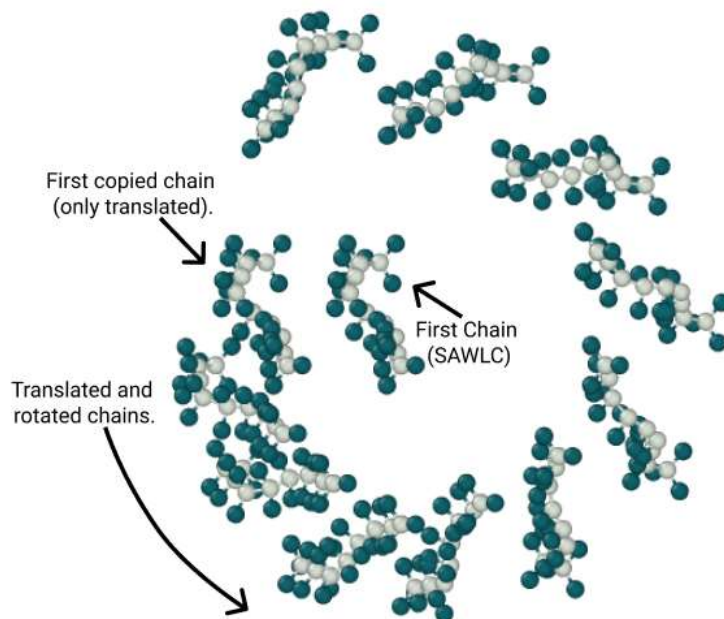
MPolSys Modeler aims not only to model chains of arbitrary lengths, but to compose entire systems for the simulation of multiple chains by CMD. However, PolymerCpp does not perform operations between different chains, leaving the risk that components of distinct chains will overlap.

This risk was corrected by MPolSys Modeler, replicating a chain with the polymerization degree (PD) desired by the user until the number of chains (NC) desired for the simulation is reached.

To avoid overlapping units, the head of the first chain is positioned at a reference position (0,0,0), the second chain translated from Δr_{ax} to a position r in a direction perpendicular to the Cartesian axis on which the chain has the largest component, the third translated $\Delta r_{ax} + \delta$ and rotated around the axis normal to the plane of translation of a Ψ angle, forming a spiral of distribution of chains around the original. The values of Δr_a , r , δ and the axes of rotation and translation are dynamically obtained through automatic measurements taken from the dimensions of the original chain.

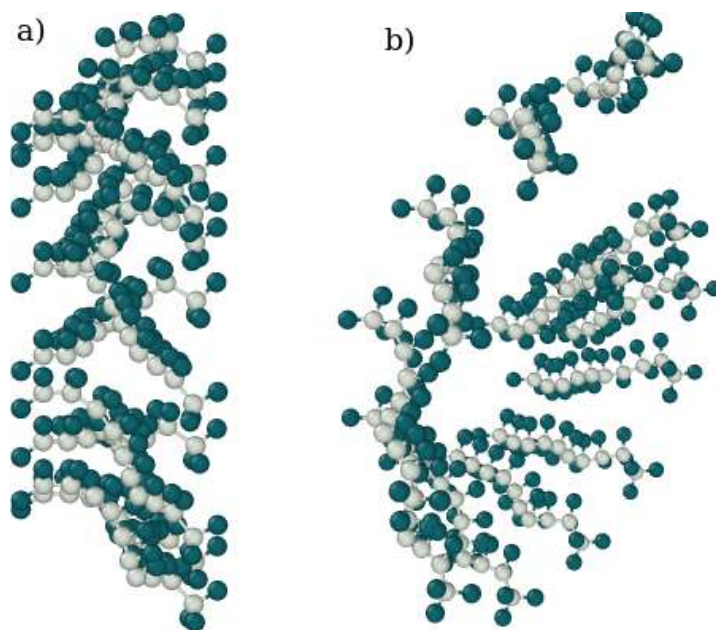
In Figure 1 a front view (from Z) of the modeled system can be seen, as well as the indications of each of chains used in the system's composition. In Figure 2 can be seen a side view (a) and a perspective view (b) of the same system represented in Figure (1).

Figure 1. Representation of the chain organization scheme in MPolSys Modeler in an arbitrary system with NC = 12 and PD = 10..



Source: Authors, generated with OVITO (Stukowski, 2009).

Figure 2. Side representation a) and perspective b) of the chain organization scheme in MPolSys Modeler with NC = 12 and PD = 10.



Source: Authors, generated with OVITO (Stukowski, 2009).

The chains are distributed spirally and not circularly concentric to avoid the risk of unwanted bias during the simulation, caused by a regular distribution of units of the same chemical nature (Ovchinnikov & Karplus, 2012), (Calio et al., 2020), (Tribello et al., 2012).

The sequencing of repeat units is carried out through the **sequence** method used by MPolSys Modeler, which receives the chemical forms of the tail, head and intermediate monomers, besides the polymerization degree (PD) and the number of desired chains (NC). This process is sufficient to determine the positions of all monomers in the system to be simulated.

Unfortunately, this approach does not prioritize simulation volume reduction, which is an important computational aspect, but a method of chain translation and simulation volume optimization is currently under development and will replace this in later versions.

The transfer of the model to Moltemplate, is done through the **GenPoly** class belonging to Moltemplate itself. MPolSys Modeler already provides the matrix adjustment of chain coordinate format in a suitable shape compatible with that system. The software also performs automatic scaling of the simulation volume as the PolymerCpp kernel increases system complexity. The system dimensions are automatically incorporated into the appropriate field of the Moltemplate input. The algebraic and statistical methods for performing this analysis were derived from the NumPy, SciPy and StatsModel libraries (Van Der Walt, Colbert & Varquaux, 2011; Harris et al, 2020; Virtanen et al., 2019; Seabold & Perktold, 2010).

2.4 Modeling test

The system was tested under 6 different modeling conditions, prioritizing the combination of low PD with a large NC, and a small number of chains with a high degree of polymerization. Table 1 shows the combinations of GP and NC used in the automated tests that were performed.

Table 1. Modeling experiments performed with MPolSys Modeler.

Polymerization Degree	Number of Chains
10	10, 100, 500
100	10, 100
1000	10

Source: Authors.

For each configuration, 500 computational experiments were performed to measure the measured tail to head distance R , and the radius of gyration R_g , the **verifyWLC** method was used in the process (Stefko & Douglass, 2017). The values of the mean radius and mean turning radius were used to compose a histogram of the 500 experiments performed. In addition, the system's adherence to the wormlike chains theoretical model (WLC) was also tested in a scalability assessment. A bond distance of 1.0 \AA was used as a standard for the experiments.

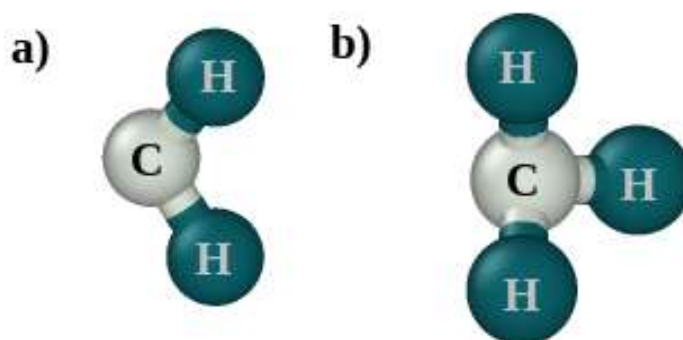
Polyethylene chains with CH_2 and CH_3 monomers modeled according to the standard configurations of COMPASS force field were chosen. The characteristics of the functional groups modeled and extracted from Compass can be seen in Table 2, a graphical representation of the groups can also be seen in Figure 3 (Sun, 1998; Kondratyuk & Pisarev, 2019).

Table 2. Positioning coordinates of functional groups centered on zero.

Functional Groups	Element	Elementary Coordinates x, y, z (\AA)
CH_2	C	0.0, 0.0, 0.0
	H	0.0, 0.631044, 0.892431 0.0, 0.631044, -0.892431
CH_3	C	0.0, 0.0, 0.0
	H	0, 0.631044, 0.892431 0, 0.631044, -0.892431 -0.892431, 0.631044, 0.0

Source: Authors.

Figure 3. Graphic representation of the CH_2 (a) and CH_3 (b) groups used in the test experiments.

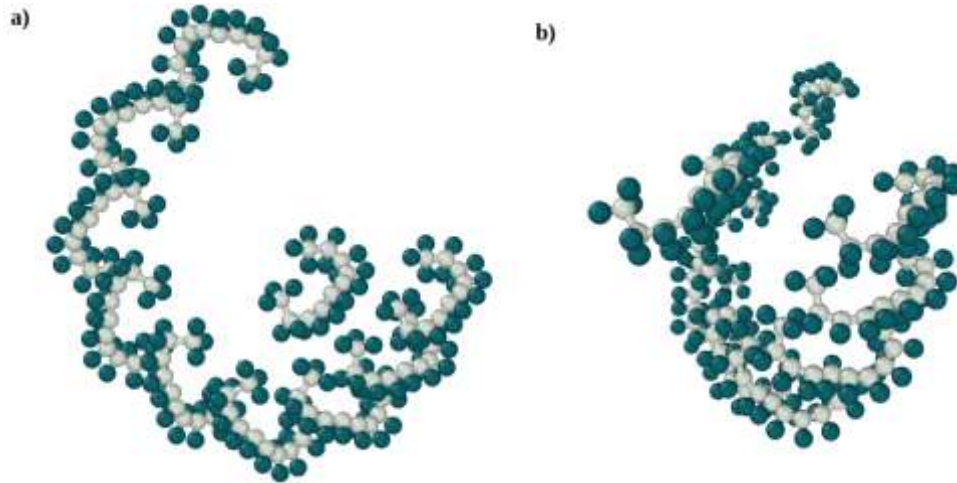


Source: Authors, generated with OVITO (Stukowski, 2009).

3. Results and Discussion

In Figures 4 and 5 show a two-view representation of the PD-10 polyethylene for a system with 10 and 100, chains. In a), the front views of each system and in b) the perspective views.

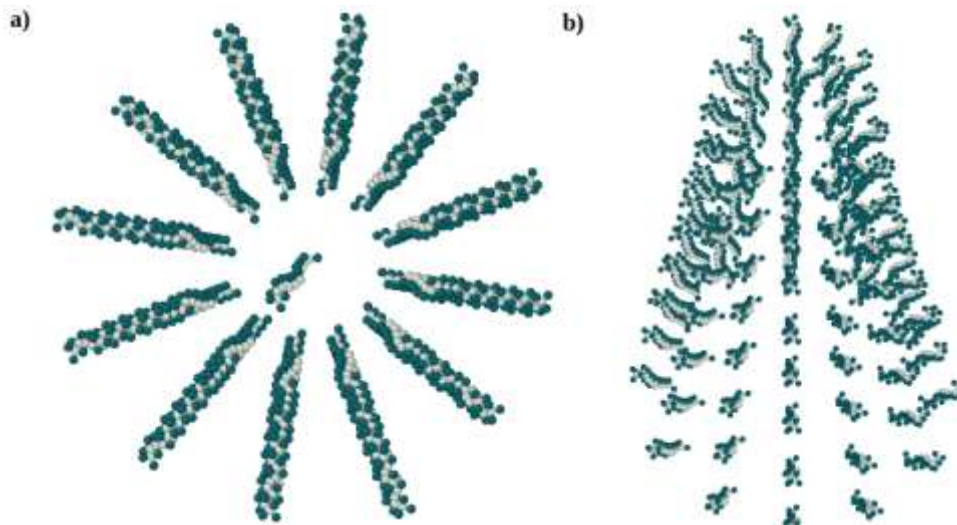
Figure 4. Representation of polyethylene chains in a system NC – 10/PD – 10 generated by MPolSys Modeler.



Source: Authors, generated with OVITO (Stukowski, 2009).

The systems were visually inspected to detect particle superposition, considering the rendering radius close to the atomic radii of carbon and hydrogen. Because of the helical character of the arrangement of the chains, a single inspection of the innermost chains is enough to guarantee the non-overlapping of the particles.

Figure 5. Representation of polyethylene chains in a system NC – 10/PD – 10 generated by MPolSys Modeler.

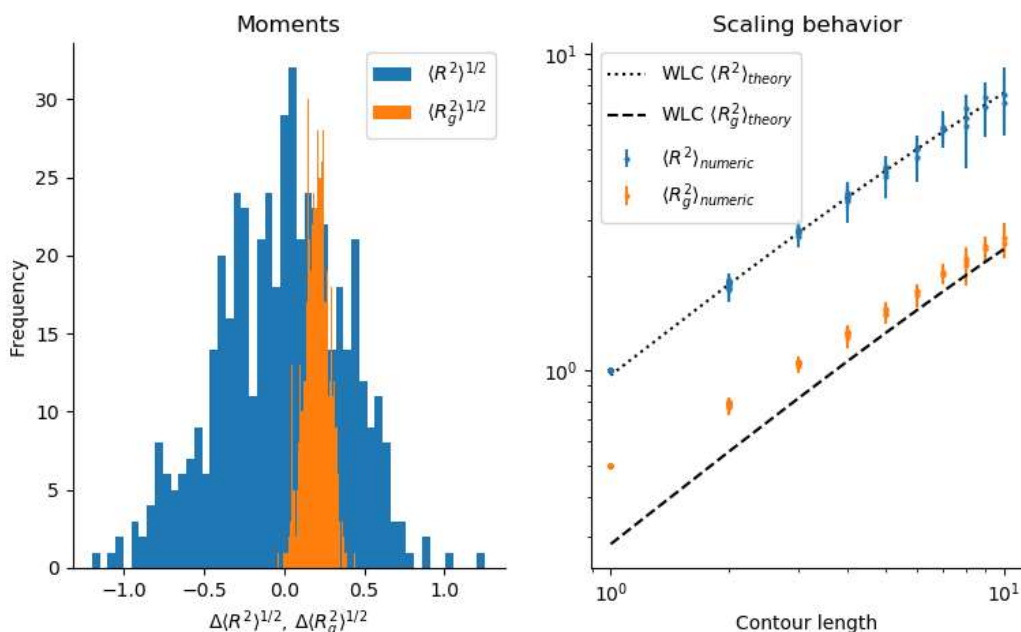


Source: Authors, generated with OVITO (Stukowski, 2009).

Due to the disproportion between the size of the chains and the NC = 500 system, these views were not placed here.

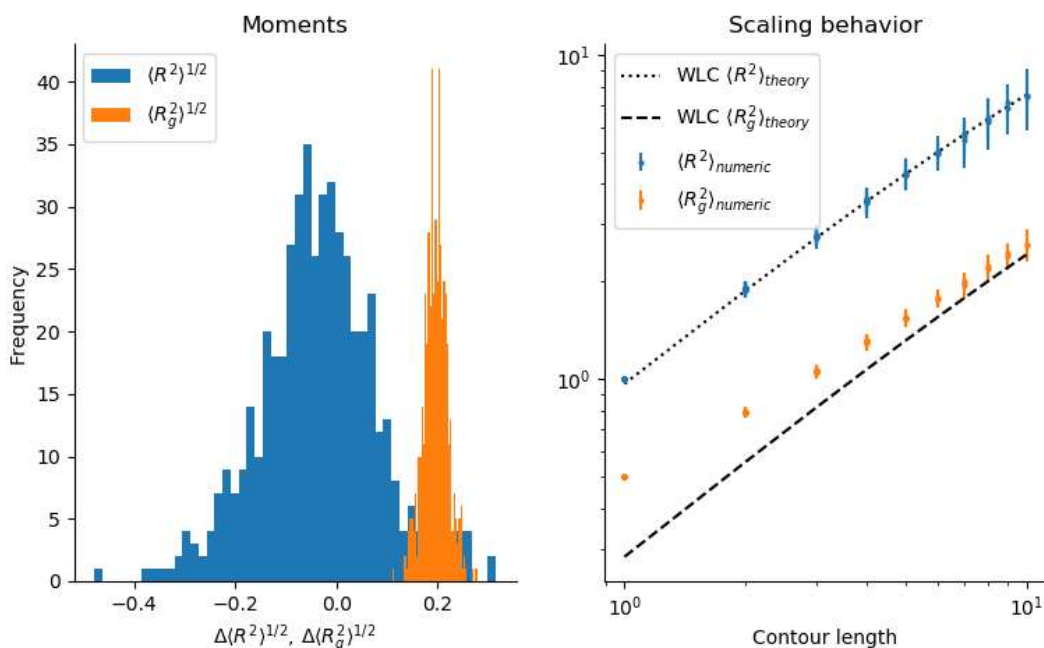
The plots of moments and scaling behavior for systems with PD = 10 described in Table 1 are presented in Figures (6), (7) and (8).

Figure 6. Histograms of $\Delta\langle R^2 \rangle^{1/2}$ and $\Delta\langle R_g^2 \rangle^{1/2}$ for a system NC – 10/PD – 10 and results of numerical experiments adherence with the theoretical prediction in a scalability condition



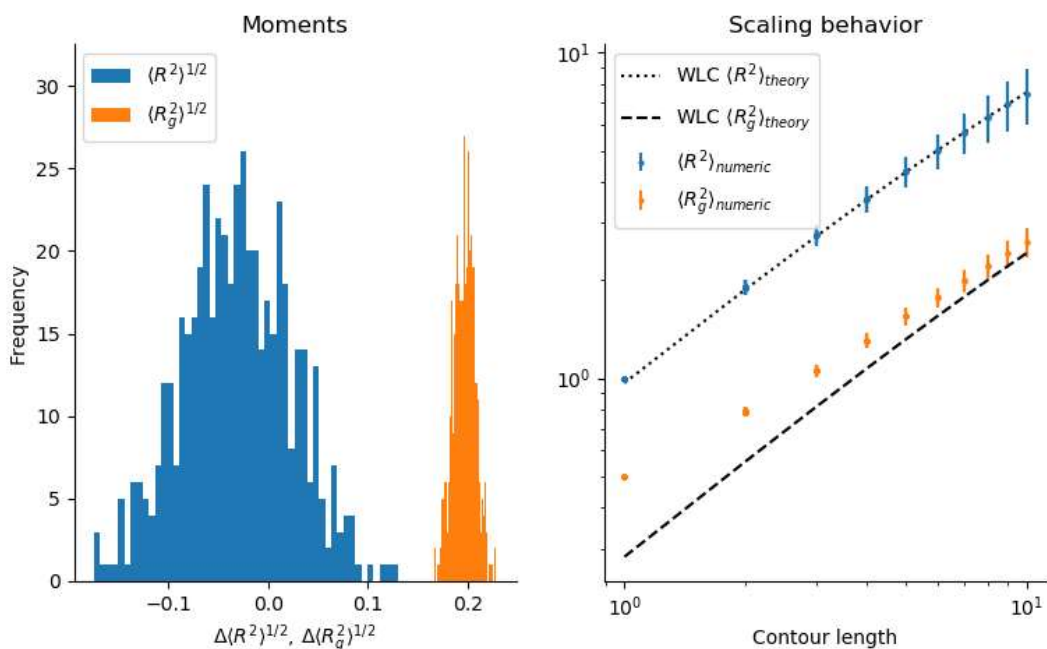
Source: Authors, generated with verifyWLC from PolymerCpp (Stefko, Douglass & Manley, 2020).

Figure 7. Histograms of $\Delta\langle R^2 \rangle^{1/2}$ and $\Delta\langle R_g^2 \rangle^{1/2}$ for a system NC – 100/PD – 10 and results of numerical experiments adherence with the theoretical prediction in a scalability condition.



Source: Authors, generated with verifyWLC from PolymerCpp (Stefko, Douglass & Manley, 2020).

Figure 8. Histograms of $\Delta\langle R^2 \rangle^{1/2}$ and $\Delta\langle R_g^2 \rangle^{1/2}$ for a system NC – 500/PD – 10 and results of numerical experiments adherence with the theoretical prediction in a scalability condition.

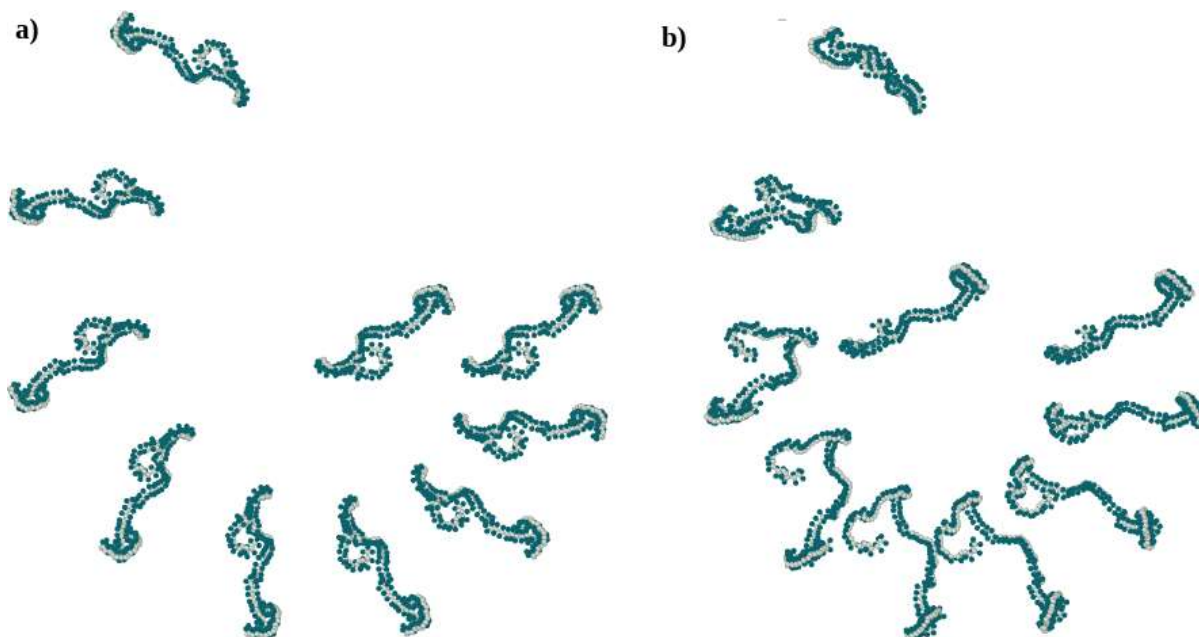


Source: Authors, generated with verifyWLC from PolymerCpp (Stefko, Douglass & Manley, 2020).

The adherence of $\Delta\langle R^2 \rangle^{1/2}$ values to the optimal WLC theoretical model, however, the same only occurs for $\Delta\langle R_g^2 \rangle^{1/2}$ values in longer chains, indicating a greater effectiveness of the system for chains with linear lengths greater than 6 Å.

In Figures (9) and (10) can be seen a two-view representation of the PD-100 polyethylene for a system with 10 and 100.

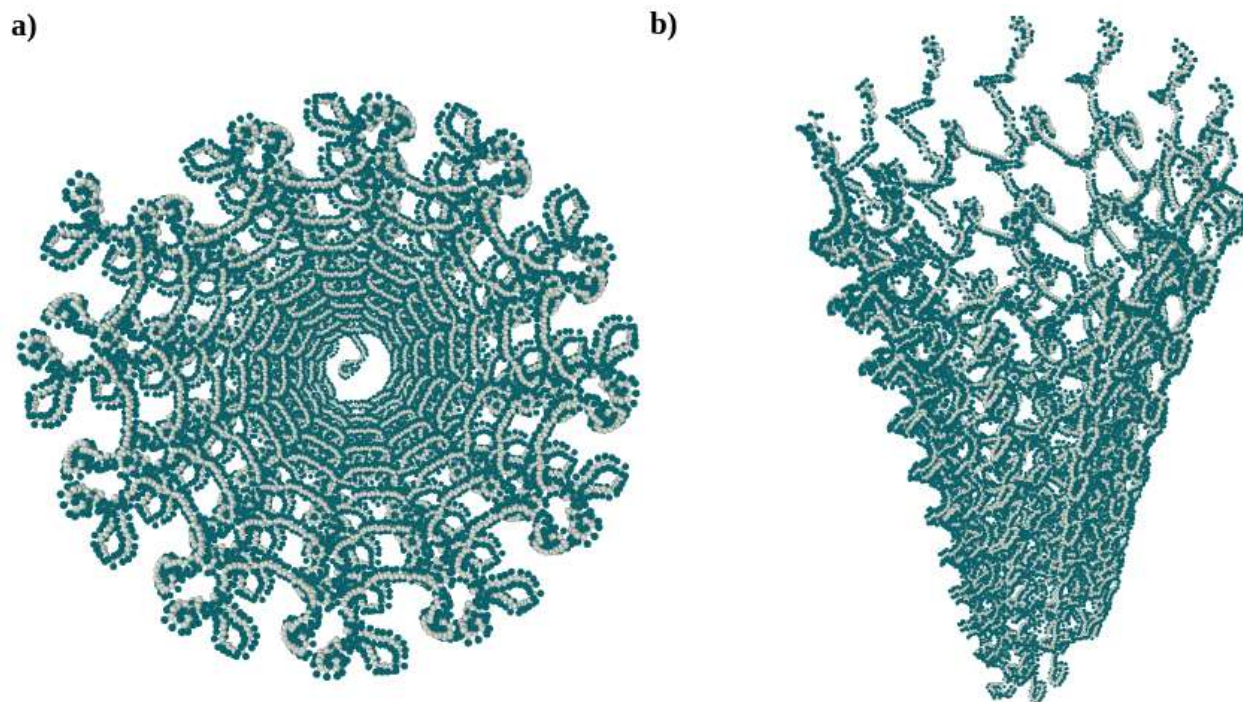
Figure 9. Representation of polyethylene chains in a system of NC – 10/PD – 100 generated by MPolSys Modeler..



Source: Authors, generated with OVITO (Stukowski, 2009).

Visual inspection of the innermost part of the system, as in the previous cases, also did not reveal any overlapping of particles, which indicates the efficiency of MPolSys Modeler in avoiding the phenomenon also in more complex systems.

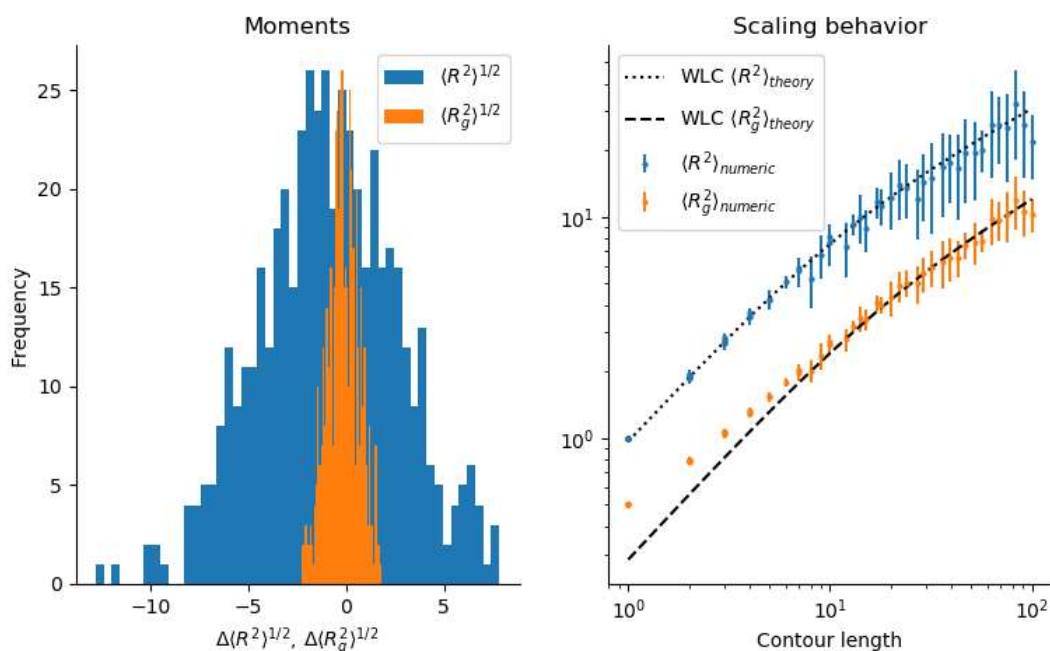
Figure 10. Representation of polyethylene chains in a system of NC – 10/PD – 100 generated by MPolSys Modeler..



Source: Authors, generated with OVITO (Stukowski, 2009).

The results of the numerical experiments for the NC-10/PD-100 and NC-100/PD-100 systems are shown in Figures (11) and (12).

Figure 11. Histograms of $\Delta\langle R^2 \rangle^{1/2}$ and $\Delta\langle R_g^2 \rangle^{1/2}$ for a system NC – 10/PD – 100 and results of numerical experiments adherence with the theoretical prediction in a scalability condition.

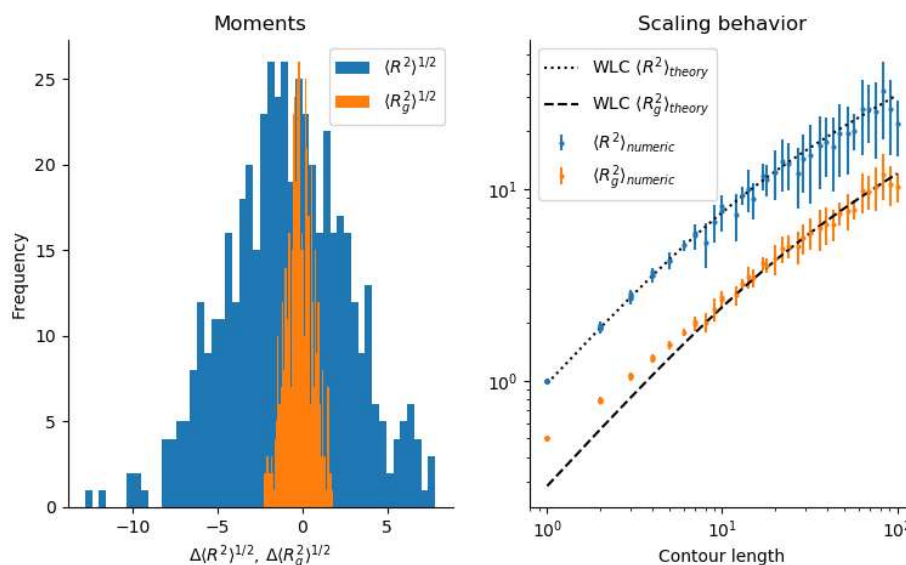


Source: Authors, generated with verifyWLC from PolymerCpp (Stefko, Douglass & Manley, 2020).

Histograms retain the same normal profile shown in simpler systems. With the increase in the radius of the chains, the

resolution of the mean values of $\Delta\langle R^2 \rangle^{1/2}$ and $\Delta\langle R_g^2 \rangle^{1/2}$ is lost, however, they still maintain a significant proximity to zero. The adherence of numerical data to the theoretical WLC model maintained a similar behavior, even with the increase in scale.

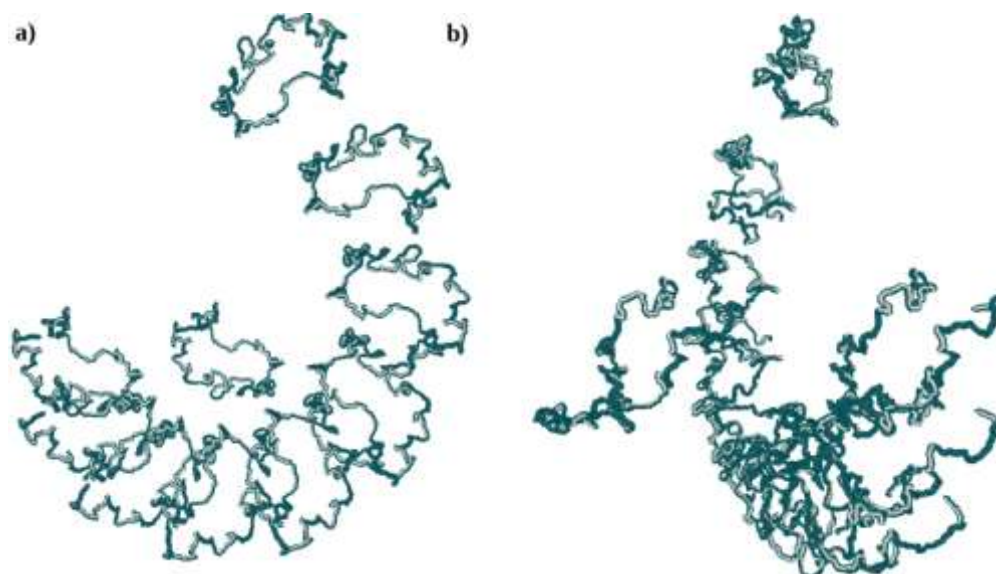
Figure 12. Histograms of $\Delta\langle R^2 \rangle^{1/2}$ and $\Delta\langle R_g^2 \rangle^{1/2}$ for a system NC – 100/PD – 100 and results of numerical experiments adherence with the theoretical prediction in a scalability condition.



Source: Authors, generated with verifyWLC from PolymerCpp (Stefko, Douglass & Manley, 2020).

The last simulated configuration system (NC-10/PD-1000) is shown in Figure (13), and the results of numerical experiments for this system in Figure (14).

Figure 13. Representation of polyethylene chains in a system of NC – 10/PD – 100 generated by MPolSys Modeler..



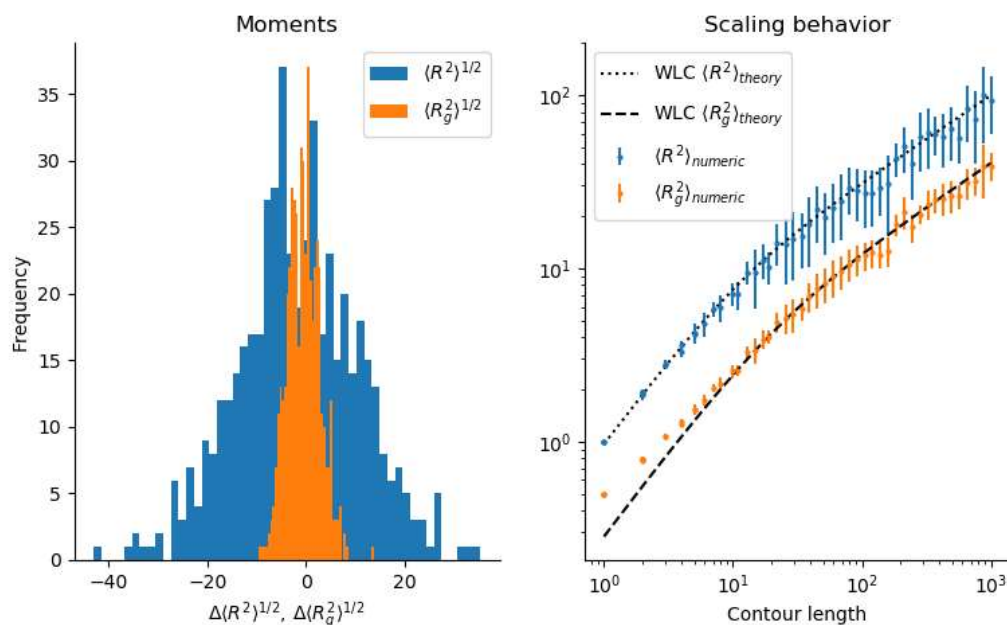
Source: Authors, generated with OVITO (Stukowski, 2009).

The global behavior was similar to the others, without identification of overlapping particles, a normal distribution of $\Delta\langle R^2 \rangle^{1/2}$ and $\Delta\langle R_g^2 \rangle^{1/2}$ values, approximately around zero. As well as an excellent adherence of the values of $\Delta\langle R^2 \rangle^{1/2}$ to the theoretical model of WLC and a similar adherence convergence for the values of $\Delta\langle R_g^2 \rangle^{1/2}$ for chains with lengths slightly less

than 10 Å.

The uniformity of results for systems of multiple complexities favors the software's behavioral stability for a wide range of use, both for polymerization degrees and number of chains.

Figure 14. Histograms of $\Delta\langle R^2 \rangle^{1/2}$ and $\Delta\langle R_g^2 \rangle^{1/2}$ for a system NC – 10/PD – 1000 and results of numerical experiments adherence with the theoretical prediction in a scalability condition.



Source: Authors, generated with verifyWLC from PolymerCpp (Stefko, Douglass & Manley, 2020).

4. Final Considerations

Based on the results obtained through the simulations carried out, it is possible to verify that MPolSys Modeler was successful in using the SAWLC's methodology to ensure that the polymer chains do not overlap. Thus, it can be used in simulation systems with a range of polymerization degrees and chains.

In parallel to this, it was also possible to notice, based on the generated adhesion graphs, that the tool performs better when it is used to generate systems with chain lengths greater than 6 Å. It should be noted that the tool can also be used for systems that have chains with lengths less than 6 Å, however adherence to the theoretical model will be lower. This is because of a limitation imposed by the SAWLC method.

After analyzing the weightings performed above, it is possible to suggest some increments that improved the performance of MPolSys Modeler. The first consideration is regarding the replacement of the chain overlapping check, which is done through visual localization, by an algorithm that performs this check. This would bring relevant improvements regarding the tool's reliability. Another gap for improvement is the replacement of the chain replication method by a methodology that uses the SAWLCs concept to perform the creation of unique chains. This will cause more realistic simulations, making the chains different from each other. Finally, there is also a possibility of optimization in incrementing the steps in the chain's composition arrangement helix. This optimization will make the system have a more cubic layout.

Acknowledgments

The authors are enormously grateful to Andrew Jewett, creator of Moltemplate, for his support in the modeling process, and the Federal Institute of Minas Gerais Campus Arcos for providing the computational infrastructure.

References

- Birta, L. G., & Arbez, G. (2013). *Modelling and simulation*. London: Springer.
- Calio, P. B., Hocky, G. M., & Voth, G. A. (2020). *Minimal Experimental Bias on the Hydrogen Bond Greatly Improves Ab Initio Molecular Dynamics Simulations of Water*. *Journal of chemical theory and computation*, 16(9), 5675-5684.
- Gartner III, T. E., & Jayaraman, A. (2019). *Modeling and simulations of polymers: a roadmap*. *Macromolecules*, 52(3), 755-786.
- Gujrati, P. D., & Leonov, A. I. (Eds.). (2010). *Modeling and simulation in polymers*. John Wiley & Sons.
- Guennebaud, G., & Jacob, B. (2010). Eigen. URL: <http://eigen.tuxfamily.org>, 3.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). *Array programming with NumPy*. *Nature*, 585(7825), 357-362.
- Hanwell, M. D., Curtis, D. E., Loni, D. C., Vandermeersch, T., Zurek, E., & Hutchison, G. R. (2012). *Avogadro: an advanced semantic chemical editor, visualization, and analysis platform*. *Journal of cheminformatics*, 4(1), 1-17.
- Humbert, M. T., Zhang, Y., & Maginn, E. J. (2019). *PyLAT: Python LAMMPS analysis tools*. *Journal of chemical information and modeling*, 59(4), 1301-1305.
- Jewett, A. I., Stelter, D., Lambert, J., Saladi, S. M., Roscioni, O. M., Ricci, M., ... & Goodsell, D. S. (2021). *Moltemplate: A Tool for Coarse-Grained Modeling of Complex Biological Matter and Soft Condensed Matter Physics*. *Journal of molecular biology*, 433(11), 166841.
- Kondratyuk, N. D., & Pisarev, V. V. (2019). *Calculation of viscosities of branched alkanes from 0.1 to 1000 MPa by molecular dynamics methods using COMPASS force field*. *Fluid Phase Equilibria*, 498, 151-159.
- Kratky, O., & Porod, G. (1949). *Röntgenuntersuchung gelöster fadenmoleküle*. *Recueil des Travaux Chimiques des Pays-Bas*, 68(12), 1106-1122.
- Martínez, L., Andrade, R., Birgin, E. G., & Martínez, J. M. (2009). *PACKMOL: a package for building initial configurations for molecular dynamics simulations*. *Journal of computational chemistry*, 30(13), 2157-2164.
- Ovchinnikov, V., & Karplus, M. (2012). *Analysis and elimination of a bias in targeted molecular dynamics simulations of conformational transitions: application to calmodulin*. *The Journal of Physical Chemistry B*, 116(29), 8584-8603.
- Plimpton, S., Kohlmeyer, A., Thompson, A., Moore, S., & Berger, R. (2020). *Lammps stable release 29 October 2020*. Zenodo. <https://doi.org/10.5281/zenodo.4157471>.
- Seabold, S., & Perktold, J. (2010, June). *Statsmodels: Econometric and statistical modeling with python*. In Proceedings of the 9th Python in Science Conference (Vol. 57, p. 61).
- Stefko, M.I, Douglass, K., & Manley, S. (2020). *PolymerCpp (0.1.3)*. Zenodo. <https://doi.org/10.5281/zenodo.3928659>
- Tribello, G. A., Ceriotti, M., & Parrinello, M. (2012). *Using sketch-map coordinates to analyze and bias molecular dynamics simulations*. *Proceedings of the National Academy of Sciences*, 109(14), 5196-5201.
- Stukowski, A. (2009). *Visualization and analysis of atomistic simulation data with OVITO—the Open Visualization Tool*. *Modelling and Simulation in Materials Science and Engineering*, 18(1), 015012.
- Sun, H. (1998). *COMPASS: an ab initio force-field optimized for condensed-phase applications overview with details on alkane and benzene compounds*. *The Journal of Physical Chemistry B*, 102(38), 7338-7364.
- Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). *The NumPy array: a structure for efficient numerical computation*. *Computing in science & engineering*, 13(2), 22-30.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., & Van Mulbregt, P. (2020). *SciPy 1.0: fundamental algorithms for scientific computing in Python*. *Nature methods*, 17(3), 261-272.