

Accepted Manuscript

Easy creation of polymeric systems for molecular dynamics with
Assemble!

Matteo T. Degiacomi, Valentina Erastova, Mark R. Wilson

PII: S0010-4655(16)00018-7

DOI: <http://dx.doi.org/10.1016/j.cpc.2015.12.026>

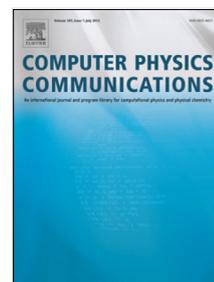
Reference: COMPHY 5845

To appear in: *Computer Physics Communications*

Received date: 28 May 2015

Revised date: 5 December 2015

Accepted date: 13 December 2015



Please cite this article as: M.T. Degiacomi, V. Erastova, M.R. Wilson, Easy creation of polymeric systems for molecular dynamics with *Assemble!*, *Computer Physics Communications* (2016), <http://dx.doi.org/10.1016/j.cpc.2015.12.026>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Easy creation of polymeric systems for Molecular Dynamics with *Assemble!*

Matteo T. Degiacomi^{a,*}, Valentina Erastova^{b,**}, Mark R. Wilson^c

^a*Department of Chemistry, Physical & Theoretical Chemistry Laboratory, University of Oxford, South Parks Road, Oxford, OX1 3QZ, U.K.*

^b*Department of Earth Sciences, Durham University, Lower Mountjoy, South Road, Durham, DH1 3LE, U.K.*

^c*Department of Chemistry, Durham University, Lower Mountjoy, South Road, Durham, DH1 3LE, U.K.*

Abstract

We present *Assemble!*, a program greatly simplifying the preparation of molecular dynamics simulations of polymeric systems. The program is controlled either *via* command line or an intuitive Graphical User Interface, and runs on all major operating systems. *Assemble!* allows the creation of a desired system of polymer chains from constituent monomers, packs the chains into a box according to the required concentration and returns all the files needed for simulation with Gromacs. We illustrate the capabilities of *Assemble!* by demonstrating the easy preparation of a 300 monomers-long polyisoprene in hexane, and an heterogeneous mixture of polybutadiene.

Keywords: Polymeric mixture, Molecular dynamics, Gromacs, Simulation preparation

PROGRAM SUMMARY

Program Title: Assemble!

Journal Reference:

Catalogue identifier:

Licensing provisions: GNU General Public License v2

*Corresponding author.

E-mail address: matteo.degiacomini@chem.ox.ac.uk. *phone number:* +44 (0)1865 275975

**Corresponding author.

E-mail address: valentina.erastova@durham.ac.uk. *phone number:* +44 (0)1913 341712

Preprint submitted to Computer Physics Communications

December 5, 2015

1
2
3
4
5
6
7
8 *Programming language:* Python 2.7.X

9 *Computer:* Workstations

10 *Operating system:* Windows, Linux, Mac OS X

11 *RAM:* depending on molecule database size and final system size: 10-20 Mb in
12 console mode, 30-150 Mb using the GUI

13 *Supplementary material:* User manual and tutorial with necessary input files are
14 distributed with the program

15 *Keywords:* Polymers, mixture, molecular dynamics

16 *Classification:* 16.11

17 *External routines/libraries:* numpy, wxpython (if using the GUI)

18 *Nature of problem:* preparation of complex polymeric systems for molecular dy-
19 namics simulations with Gromacs.

20
21
22
23 *Solution method:* creation of polymers collections using monomer units and a
24 Gromacs-compatible force field. Packing multiple polymer copies in a box accord-
25 ing to desired concentration.

26
27 *Running time:* system dependent, but typically less than 5 seconds.
28
29
30
31

32 **1. Introduction**

33
34 Polymers are ubiquitous and play fundamental roles in areas as diverse
35 as the oil industry, the pharmaceutical industry and materials development,
36 but their behaviour at a molecular level is in some cases still poorly under-
37 stood. Molecular Dynamics (MD) simulations can provide important insights
38 into molecular structure and motion at an atomistic level. In fact, polymers
39 were some of the first complex targets for this technique [1, 2]. The steady
40 increase in computational power and development of more effective simu-
41 lation methodologies and force fields has allowed the study of systems of
42 growing size and complexity. Examples include the study of interactions of
43 polymers with single-walled carbon nanotubes [3, 4], the prediction of solu-
44 bility of pharmaceutical compounds in polymeric carriers [5] and the study
45 of structure and mechanics of glassy polymers [6]. Moreover, by exploiting
46 coarse graining potentials, polymer simulation studies have been brought to
47 a mesoscale level [7, 8, 9, 10].
48
49
50
51

52
53 Despite these successes, polymers remain difficult to model. These molecules
54
55

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

can potentially be composed of very long chains, that can be of a single monomer type, in addition to di- or tri-block copolymer chains, random ordered multimer chains, stars and rings. Furthermore, polymers often exist as mixtures composed of different chain lengths, different branching and often different monomers. These features make the simulation of a realistic polymeric system extremely challenging, one of the primary challenges being the creation of an atomic arrangement representative of the polymeric system to be studied. Simplifications such as simulating only short regions, averaging chains lengths and simplifying compositions must sometimes be made in order to make the system tractable.

To date, several commercial and freely available software offer tools for polymer modelling. Traditionally, software would allow users to build an extended single polymer of predefined sequence. Some software implement strategies to generate more relaxed, possibly clash free chains. These strategies span from allowing user manipulation of a chain's dihedral angles (e.g. Pymol [11], Polymer Modeler [12]) up to exploiting Monte Carlo-based schemes to relax the chain (e.g. Enhanced Monte Carlo [13]). Some software is not only able to build a single chain, but also to assemble a full polymeric system (e.g. Moltemplate [14], Enhanced Monte Carlo [13], Materials Studio [15]) and sometimes allows for the assignment of parameters from an available force field. When the creation of a polymeric system is not possible, software like Packmol [16] can help in assembling individual molecules into a box. Majority of polymer modelling software exploits external engines by either running them internally, or creating input files for them. One of the most common choices is LAMMPS [17] (e.g. Polymer Modeler [12], Materials Studio [15], Scienomics MAPS [18], Moltemplate [14], Enhanced Monte Carlo [13]). Other codes, like HOOMD-blue [19], provide their own simulation engine. HOOMD-blue is controlled by Python scripting, allowing users to generate and simulate polymeric systems.

Although desired polymeric systems can be created using the software cited above, their relaxation in MD is often difficult when modelling chains of few hundred monomer units. These simulations usually require carefully designed simulation protocols. Hybrid simulation techniques have been developed as an alternative way to tackle this challenge. These range from relaxing a polymeric system at coarse grain level and subsequently reverse-mapping it into a higher resolution representation [20, 21], exploiting hy-

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

brid particle-continuum approaches [22], allowing the system to dynamically polymerize subunits into longer chains (Polymatic [23]), to exploiting Monte Carlo simulation techniques [24, 25].

We present *Assemble!*, a tool facilitating the simulation of polymeric systems in Gromacs [26]. Using either a GUI or console commands, *Assemble!* allows for the creation of polymers from monomer building blocks with a user-defined force field, and their assembly in a system ready for MD simulation. We demonstrate the effectiveness of *Assemble!* by modelling and simulating two complex polymeric systems: polyisoprene in hexane and a polybutadiene mixture. The main difficulty of the first resides in its length (300 subunits), whereas the main challenge in the second is represented by its heterogeneity.

2. Program Overview

Assemble! represents a polymer as a chain of molecular subunits. In order to generate the atomic arrangement associated with a given molecular sequence, *Assemble!* first determines the position of virtual hooking points (hereon "hooks") connected to the terminal atoms of every molecular subunit. Two consecutive subunits can be properly arranged by superimposing their terminal atoms with neighbor's hooks. The location of hooks with respect to molecules' terminal atoms are determined by extracting bond, angle and dihedral information from a given force field. When adding a new molecule to a polymer chain, *Assemble!* will make sure that no near clashes between atoms are generated. By default, atoms at less than 0.9 Å (less than an O-H bond) are considered as clashing, although the user can modify this parameter. If a clash is detected, the dihedral angles of both current and past hook will be alternatively perturbed, until the clash is solved. Our perturbation methodology, aimed at minimizing the displacement from equilibrium positions of both hooks, iteratively modifies their defining dihedral angles by $\pm 5^\circ$ increments. Since monomeric subunits are generated by exploiting equilibrium values of bonds, angles and dihedrals, the resulting polymeric chain will be already relaxed at a local scale, thus simplifying its equilibration in MD. Every generated polymer is aligned along its inertia tensor in order to minimise its bounding box. The final ensemble of generated clash-free polymers can be ultimately packed into a simulation box according to

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

desired concentrations. When inserting aligned polymers in a grid, it is likely that individual molecules will already be within non-bonded cutoff distance from their neighbours. Such interactions prevent the molecules to behave as in a full gas-phase simulation, i.e. self-interact and collapse into individual droplets. This contributes in reducing equilibration times.

To further simplify creation of polymeric assembly, the program can be controlled *via* the command line or a Graphical User interface (GUI, see Figure 1B).

The GUI is conceived to be used from top to bottom. Initially, a list of molecular subunits and an associated force field are provided by the user and assigned to a one-letter code. *Cis* and *trans* isoprene, *cis*, *trans* and *vinyl* butadiene and hydrogenated styrene subunits are distributed with *Assemble!* along with their associated TraPPE united atom parameters [27]. Additional, user-defined monomers can also be simply imported. One-letter codes corresponding to the loaded monomeric subunits can then be used to define a string associated with a desired polymeric sequence. Strings can be either user-defined, or randomly generated according to desired chain length and monomeric ratios. If desired, a box containing a polymeric mixture can be generated. The GUI allows concentrations in solution to be generated for all the defined polymers, using copies of the appropriate chains packed randomly in a box with the desired dimensions. At any moment, the state of the interface can be saved into a parameter file. Parameter files can be directly used to execute *Assemble!* *via* the command line, or loaded again in the GUI to continue working with it at a later time.

Assemble! produces all the necessary structure and topology files to run a molecular dynamics simulation with Gromacs. The system of interest can be represented at any granularity level supported by Gromacs: atomistic, united atom, or coarse grained. For every defined polymer, coordinate files in Gromacs (.gro) and protein database (.pdb) format, as well as the Gromacs topology file (.itp) are generated. The coordinate file (.gro) for the final system and associated topology file (.top) are also generated. Aiming for modularity, the system topology file (.top) only defines atomtypes, and imports individual polymer topology files (.itp) as the source of chain topology information. Furthermore, in order to help the analysis process, an index file (.ndx), allowing the selection of polymers by kind, is produced.

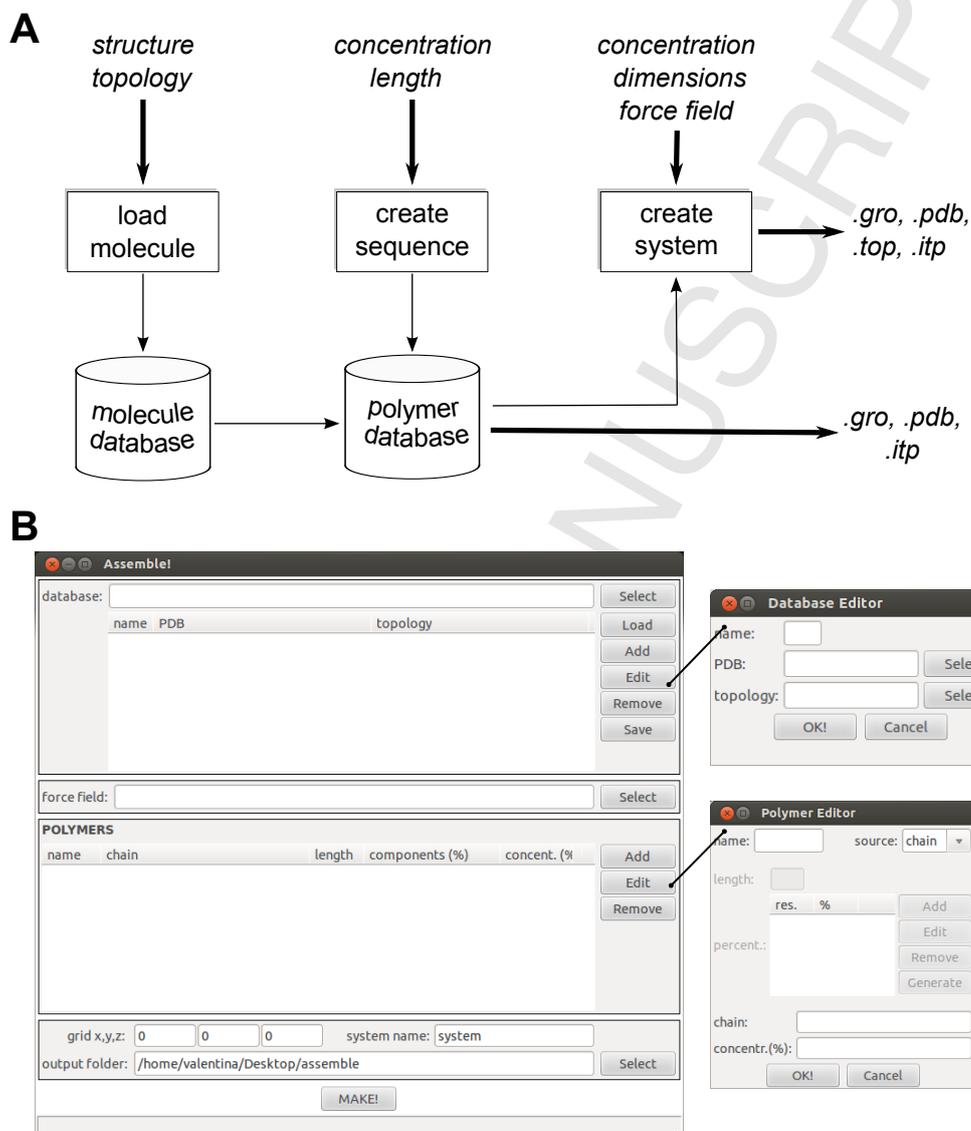


Figure 1: (A) Assemble! architecture. The user first provides an ensemble of monomer structures and topologies. An ensemble of different polymers can be subsequently generated by providing their chain length and monomer concentrations. Assemble! finally packs several instances of the defined polymeric chains into a simulation box according to desired concentrations. All files necessary to run a simulation with Gromacs are finally produced. In the schema, cylinders represent databases, and thick arrows input/output. (B) The GUI for the Assemble! program, showing the database editor and the polymer editor.

3. Usage Examples

In this section we demonstrate the use of *Assemble!* to create two systems: a 300 monomer unit-long polyisoprene molecule in hexane solvent and a polybutadiene mixture with a number-average degree of polymerisation equal to 25.4. In these examples we use the TraPPE united atom force field [27] and run simulations with Gromacs 4.6.2 [26].

3.1. Polyisoprene in hexane

We use a 300 monomer-long polyisoprene (PI) polymer chain with a 33% *cis* to 67% *trans* distribution. Using the GUI, we generated the polymer with a random distribution of monomers. The automatically generated input text file, featuring a 300 monomer-long random chain is:

```
mode gromacs
residue c ./database/cis-PI-monomer.pdb ./database/cis-PI-monomer.txt
residue t ./database/trans-PI-monomer.pdb ./database/trans-PI-monomer.txt
ForceField ./database/forcefield/trappe.ff.txt
output_folder ./
system_name mega-PI
box_grid_shape 0 0 0
chain PI ttcttttcttctctttctttccttctctttcttctcttttttcccttc\
tttctctctttttctttttcctcccctttttccttttttttcttttttttcttttt\
tctcctctttcttcttttttttcttttctttctttctttctttcctcctctttcttctt\
ttctctctttttcttcttcttccccctcttctttttctttttttctttcctcctctt\
ttctttcttcccttctttttctttttttctttttttcccctctttttttttctttttcctt
concentration PI 100
molecule PI
```

We solvated the produced polyisopropene in a pre-equilibrated 18^3 nm^3 cubic hexane box containing nearly 2500 hexane molecules (Figure 2A) using the Gromacs *genbox* tool. After energy minimisation, we ran two 200 ns simulations at 1 and 10 bar respectively. Simulations were carried out in the constant-*NPT* ensemble, using an isotropic Parrinello-Rahman barostat with the V-rescale thermostat set to 300 K, and a 0.001 ps time step (Figure 2B). For the analysis of both simulations, we verified that no self-interaction through periodic boundary conditions took place. The total volume reduction through the simulation was within 6% for both systems.

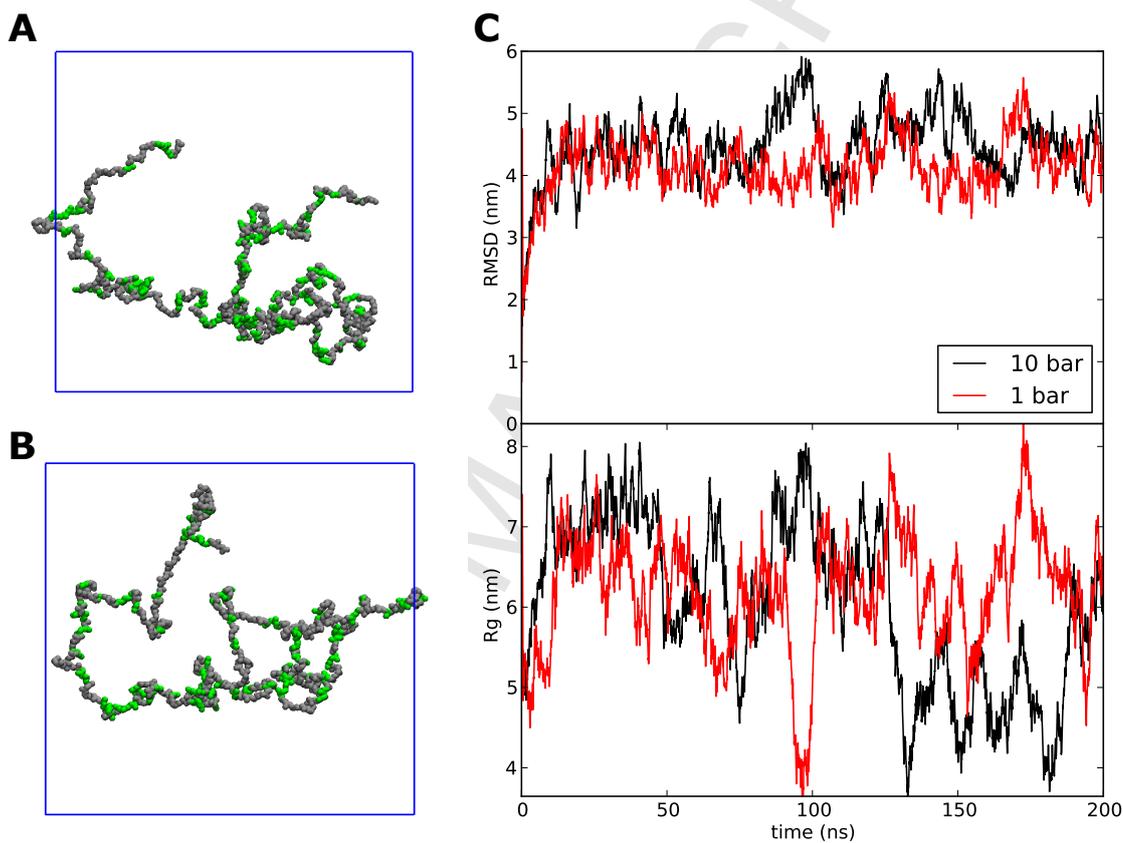


Figure 2: *polyisoprene initial (A) and final (B) arrangement after 200 ns molecular dynamics simulation at 10 bar. Cis and trans monomers are shown in grey and green, respectively, hexane molecules are not shown for clarity. (C) Polyisoprene root mean square deviation (top) and radius of gyration (bottom) at 1 and 10 bar.*

Analysis of the chain's root mean square deviation (comparing all polyisoprene atoms against the first frame's coordinates) and radius of gyration R_g , indicates that our straightforward setup leads to RMSD convergence after 10 ns (Figure 2C). At equilibrium, our average R_g of 6.18 ± 0.68 nm at 1 bar and 5.68 ± 1.01 nm at 10 bar are comparable with values reported by Fetter *et al* [28] in three different cases. Namely, for a 300 unit-long PI with molecular total mass, M , equal to 20436 g mol^{-1} (68.12 g mol^{-1} per monomer unit), R_g is $2.75 \times 10^{-2} M^{0.513} = 4.47$ nm in dioxane, $3.30 \times 10^{-2} M^{0.5} = 4.71$ nm according to the Baumann procedure, [29] and $1.26 \times 10^{-2} M^{0.61} = 5.37$ nm in cyclohexane. From 2C it can be seen that the measured radius of gyration undergoes large fluctuations. Importantly, though, the polyisoprene structure generated by *Assemble!* already features a radius of gyration comparable with mean values from the simulation. Directly starting with such a molecule, already relaxed at local level, provides an advantage and also requires a smaller hexane box for simulation.

3.2. Polybutadiene mixture

We use *Assemble!* to set up a polybutadiene system closely matched to an experimental composition of 40% *cis*, 50% *trans*, and 10% 1,2-*vinyl* units, with a number-average degree of polymerisation, $\langle X_n \rangle = 25.4$, as reported in a study by Smith *et al* [30]. To create a realistic representation for this system we set 10 different polymer chains, packed into a $6 \times 17 \times 22$ grid with a target concentration of 10% each, resulting in a $40.9 \times 40.3 \times 41.1$ nm³ box with composition of 37.5% *cis*, 53.7% *trans*, and 8.8% 1,2-*vinyl* units and $\langle X_n \rangle = 24.9$. The GUI-generated input file, featuring 10 random polybutadiene chains, is:

```

mode gromacs
residue C ./database/cis-PBD-monomer.pdb ./database/cis-PBD-monomer.txt
residue T ./database/trans-PBD-monomer.pdb ./database/trans-PBD-monomer.txt
residue V ./database/vinyl-PBD-monomer.pdb ./database/vinyl-PBD-monomer.txt
ForceField ./database/forcefield/trappe.ff.txt
output_folder ./
system_name PBD
box_grid_shape 6 17 22
chain PBD0 CCTVTCCCCCVCTTCCCTCCTTTTTC
concentration PBD0 10.0

```

```

1
2
3
4
5
6
7 chain PBD9 CCCTTTCTCTCCCTCTTCTTCTCCV
8 concentration PBD9 10.0
9
10 chain PBD8 CTCCTTCTCCTVCTTCVTCCTTT
11 concentration PBD8 10.0
12
13 chain PBD7 TTCCVTTTTTVCTCCVTTCTCTVT
14 concentration PBD7 10.0
15
16 chain PBD6 TTCCVTTTTTVCTCCVTTCTCTVT
17 concentration PBD6 10.0
18
19 chain PBD5 TTTCTCTCCCTTTTTTCTTCTTTCCCT
20 concentration PBD5 10.0
21
22 chain PBD4 VTCVTCTCCTCCTTVTTTTTTTTVC
23 concentration PBD4 10.0
24
25 chain PBD3 CCTCTCTTTTTTTTCVTCCTTCTTTCTV
26 concentration PBD3 10.0
27
28 chain PBD2 CCTCTCTTTTTTTTCVTCCTTCTTTCT
29 concentration PBD2 10.0
30
31 chain PBD1 TTTCVTTCTTTTVTCTTTCTVT
32 concentration PBD1 10.0
33
34 molecule PBD0 PBD9 PBD8 PBD7 PBD6 PBD5 PBD4 PBD3 PBD2 PBD1

```

This system is composed of long and flexible polymeric chains, which are prone to entanglement, and is therefore difficult to equilibrate. We devised a multi-step equilibration protocol aimed at minimising vacuum effects that can cause polymer chains to self interact and pack into droplets separated by free space:

- Energy minimisation (EM) of the generated structure with steepest descent, for 5000 steps with 0.01 nm maximum step size and force tolerance of 100 kJ mol⁻¹ nm⁻¹.
- 1 ns MD simulation of the energy minimised structure with a 0.001 ps time step: temperature kept low at 15 K (Velocity rescale thermostat) to prevent the molecules to collapse on themselves, a high 500 bar pressure (Berendsen barostat) is applied to bring molecules together. A 1.2 nm non-bonded cutoff distance is imposed. In our case, this distance is sufficient for individual molecules to interact with their neighbours in the starting geometry. During this run the volume of the box decreased over 10 times (from 68x10³ nm³ to 5.5x10³ nm³), while the average radius of gyration decreased only slightly (from 1.39 nm to 0.85 nm).

- A further 1 ns MD simulation at a slightly increased temperature of 100 K, with pressure reduced to 50 bar.
- After a brief EM (same parameters as above) to help resolve any possible close contacts, a 2 ns MD simulation at 200 K and 5 bar.
- Finally, before further changing simulation conditions, a short EM was carried out and the system was then simulated for 2 ns at 300 K and 1 bar, i.e. the realistic conditions for studying the system of interest.

After system equilibration, a 15 ns production phase at 300 K (V-rescale thermostat) and 10 bar (Parinello-Rahman barostat) was performed. RMSD and R_g stabilise, indicating that the system is equilibrated (Figure 3C). For simulation analysis, we considered only the last 5 ns simulation. Furthermore, in order to compare our results with data from Smith *et al.*, [30] we also performed a shorter 5 ns MD simulation matching the experimental conditions of 353 K and 1 bar (using the last 1 ns for analysis).

Our predicted R_g for individual components and system average R_g are reported in Table 1. The average R_g values of 1.32 nm (300 K, 10 bar) and 1.31 nm (353 K, 1 bar) are in good agreement with the measurements and simulations by Smith *et al* [30]: experimental R_g of 1.54 nm and calculated R_g of 1.32 nm. Our system features a number average polymerisation equal to 24.9, slightly smaller than in Smith *et al.* For comparison, according to Fetter *et al.*, [28] R_g in dioxane is $3.7910^{-2}\bar{M}^{0.5} = 1.39$ nm, and in cyclohexane $1.2910^{-2}\bar{M}^{0.609} = 1.038$ nm. Simulations predict densities of 0.866 g cm^{-3} (300 K, 10 bar) and 0.828 g cm^{-3} (353 K, 1 bar). The latter value is comparable with the experimental 0.843 g cm^{-3} (353 K, 1 bar) [30]. Results in Table 1 and Figure 3 indicate that the radius of gyration does not only depend on the length of the molecule, but also on the composition and the order of the monomer units.

4. Conclusion

This article introduces *Assemble!*, a program dramatically simplifying the setup of realistic polymeric systems. Given a desired polymeric sequence or composition and an appropriate force field, Gromacs-ready files for linear polymeric systems of virtually any complexity can be simply generated on

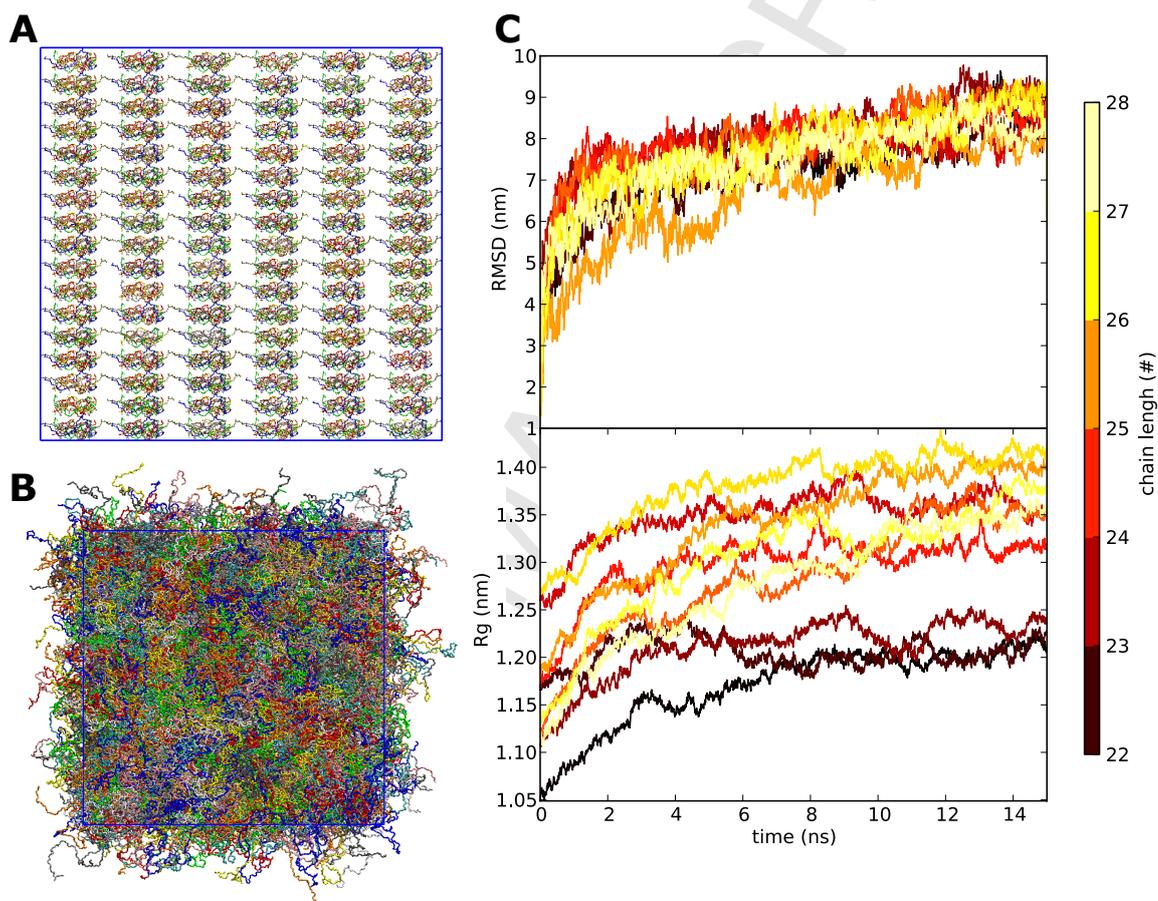


Figure 3: (A) polybutadiene system initial arrangement after generation of chains in Assemble!. (B) after 5 ns molecular dynamics simulation, the polybutadiene system is fully mixed. (C) Root mean square deviation (top) and radius of gyration (bottom) per polybutadiene molecule type. Lines are coloured as a function of chain length.

Table 1: radius of gyration for chains in a polybutadiene mixture. Mean and standard deviation are calculated using all chains extracted from the last 5 ns of simulation.

Polymer	$\langle Xn \rangle$	R_g , nm (300 K, 10 bar)	R_g , nm (353 K, 1 bar)
PBD0	27	1.416 ± 0.005	1.389 ± 0.011
PBD1	22	1.214 ± 0.005	1.194 ± 0.008
PBD2	26	1.352 ± 0.005	1.335 ± 0.011
PBD3	27	1.380 ± 0.005	1.361 ± 0.008
PBD4	26	1.399 ± 0.004	1.364 ± 0.011
PBD5	25	1.353 ± 0.007	1.356 ± 0.006
PBD6	24	1.234 ± 0.008	1.213 ± 0.010
PBD7	23	1.215 ± 0.010	1.212 ± 0.013
PBD8	27	1.353 ± 0.008	1.338 ± 0.011
PBD9	25	1.318 ± 0.003	1.309 ± 0.010
mean	24.9	1.316 ± 0.073	1.306 ± 0.070

the timescale of minutes.

Lyulin *et al.* reported that polymeric systems simulated from an initial gas-phase state require microsecond timescales in order to equilibrate microscopic quantities such as radius of gyration, whereas macroscopic ones such as density equilibrate faster [31]. *Assemble!* generates locally pre-relaxed polymeric chains, already interacting with each other at the beginning of the simulation. This contributes in speeding up the system's equilibration times. Here we demonstrate *Assemble!* usage by setting up and performing molecular dynamics on two different polymeric systems. Although our simulations show good agreement with experimental data, it is expected that even longer simulation might improve our radius of gyration predictions.

We foresee that *Assemble!* will be of great use in future polymer simulations. In the current state of development *Assemble!* allows for straightforward creation of linear chains from monomer units represented at any resolution, from atomistic to coarse grain. These linear chains can be also exploited, in a second step, as building blocks for branched polymers. Setting the dihedral angle between such repeat units allows for a desired tacticity or chirality in the resulting comb-polymer.

Assemble! future improvements will include simple protocols for the cre-

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

ation of branched, star and circular polymers. Updated source code, user manual and tutorial are freely available for academia at <http://degiacom.github.io/assemble>.

5. Acknowledgment

The authors wish to thank Jos Tasche for being *Assemble!*'s first user, and designing the program's logo. The authors would like to acknowledge the use of Durham University's high-performance computing services in carrying out this work.

- [1] D. Rapaport, Molecular dynamics study of a polymer chain in solution, *The Journal of Chemical Physics* 71 (8) (1979) 3299–3303.
- [2] G. S. Grest, K. Kremer, Molecular dynamics simulation for polymers in the presence of a heat bath, *Physical Review A* 33 (5) (1986) 3628.
- [3] S. S. Tallury, M. A. Pasquinelli, Molecular dynamics simulations of flexible polymer chains wrapping single-walled carbon nanotubes, *The Journal of Physical Chemistry B* 114 (12) (2010) 4122–4129.
- [4] W. Liu, C. L. Yang, Y. T. Zhu, M. S. Wang, Interactions between single-walled carbon nanotubes and polyethylene/polypropylene/polystyrene/poly (phenylacetylene)/poly (p-phenylenevinylene) considering repeat unit arrangements and conformations: a molecular dynamics simulation study, *The Journal of Physical Chemistry C* 112 (6) (2008) 1803–1811.
- [5] J. Gupta, C. Nunes, S. Vyas, S. Jonnalagadda, Prediction of solubility parameters and miscibility of pharmaceutical compounds by molecular dynamics simulations, *The Journal of Physical Chemistry B* 115 (9) (2011) 2014–2023.
- [6] J.-L. Barrat, J. Baschnagel, A. Lyulin, Molecular dynamics simulations of glassy polymers, *Soft Matter* 6 (15) (2010) 3430–3446.
- [7] G. Rossi, L. Monticelli, S. R. Puisto, I. Vattulainen, T. Ala-Nissila, Coarse-graining polymers with the martini force-field: polystyrene as a benchmark case, *Soft Matter* 7 (2) (2011) 698–708.

- 1
2
3
4
5
6
7
8 [8] K. Prasitnok, M. R. Wilson, A coarse-grained model for polyethylene
9 glycol in bulk water and at a water/air interface, *Physical Chemistry*
10 *Chemical Physics* 15 (40) (2013) 17093–17104.
- 11
12 [9] F. Müller-Plathe, Coarse-graining in polymer simulation: From the
13 atomistic to the mesoscopic scale and back, *ChemPhysChem* 3 (9) (2002)
14 754–769.
- 15
16 [10] Q. Sun, R. Faller, Systematic coarse-graining of atomistic models for
17 simulation of polymeric systems, *Computers & chemical engineering*
18 29 (11) (2005) 2380–2385.
- 19
20 [11] Schrödinger, LLC, The PyMOL molecular graphics system, version 1.3r1
21 (August 2010).
- 22
23 [12] B. P. Haley, N. Wilson, C. Li, A. Arguelles, E. Jaramillo, A. Strachan,
24 Polymer modeler (Aug 2010). doi:doi:10.4231/D3ST7DZ16.
25 URL <https://nanohub.org/resources/9230>
- 26
27 [13] Enhanced monte carlo (emc), montecarlo.sourceforge.net.
- 28
29 [14] Moltemplate, www.moltemplate.org.
- 30
31 [15] Accelrys materials studio, accelrys.com.
- 32
33 [16] J. M. Martínez, L. Martínez, Packing optimization for automated gener-
34 ation of complex system's initial configurations for molecular dynamics
35 and docking, *Journal of computational chemistry* 24 (7) (2003) 819–825.
- 36
37 [17] S. Plimpton, Fast parallel algorithms for short-range molecular dynam-
38 ics, *Journal of computational physics* 117 (1) (1995) 1–19.
- 39
40 [18] scienomics materials processes and simulations (maps), [http://](http://scienomics.com)
41 scienomics.com.
- 42
43 [19] J. Glaser, T. D. Nguyen, J. A. Anderson, P. Lui, F. Spiga, J. A. Millan,
44 D. C. Morse, S. C. Glotzer, Strong scaling of general-purpose molecular
45 dynamics simulations on gpus, *Computer Physics Communications* 192
46 (2015) 97–107.
- 47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

- 1
2
3
4
5
6
7
8 [20] G. Zhang, L. A. Moreira, T. Stuehn, K. C. Daoulas, K. Kremer, Equilibra-
9 tion of high molecular weight polymer melts: a hierarchical strategy,
10 ACS Macro Letters 3 (2) (2014) 198–203.
11
12 [21] T. Spyriouni, C. Tzoumanekas, D. Theodorou, F. Müller-Plathe, G. Mi-
13 lano, Coarse-grained and reverse-mapped united-atom simulations of
14 long-chain atactic polystyrene melts: Structure, thermodynamic prop-
15 erties, chain conformation, and entanglements, Macromolecules 40 (10)
16 (2007) 3876–3885.
17
18 [22] A. De Nicola, T. Kawakatsu, G. Milano, Generation of well-relaxed
19 all-atom models of large molecular weight polymer melts: A hybrid
20 particle-continuum approach based on particle-field molecular dynam-
21 ics simulations, Journal of Chemical Theory and Computation 10 (12)
22 (2014) 5651–5667.
23
24 [23] L. J. Abbott, K. E. Hart, C. M. Colina, Polymatic: a generalized sim-
25 ulated polymerization algorithm for amorphous polymers, Theoretical
26 Chemistry Accounts 132 (3) (2013) 1–19.
27
28 [24] M. Laradji, H. Guo, M. J. Zuckermann, Off-lattice monte carlo simu-
29 lation of polymer brushes in good solvents, Physical Review E 49 (4)
30 (1994) 3199.
31
32 [25] R. Auhl, R. Everaers, G. S. Grest, K. Kremer, S. J. Plimpton, Equilibra-
33 tion of long chain polymer melts in computer simulations, The Journal
34 of chemical physics 119 (24) (2003) 12718–12728.
35
36 [26] B. Hess, C. Kutzner, D. Van Der Spoel, E. Lindahl, Gromacs 4: Algo-
37 rithms for highly efficient, load-balanced, and scalable molecular simula-
38 tion, Journal of chemical theory and computation 4 (3) (2008) 435–447.
39
40 [27] C. D. Wick, M. G. Martin, J. I. Siepmann, Transferable potentials
41 for phase equilibria. 4. united-atom description of linear and branched
42 alkenes and alkylbenzenes, The Journal of Physical Chemistry B
43 104 (33) (2000) 8008–8016.
44
45 [28] V. Raju, E. Menezes, G. Marin, W. Graessley, L. Fetters, Concentration
46 and molecular weight dependence of viscoelastic properties in linear and
47 star polymers, Macromolecules 14 (6) (1981) 1668–1676.
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

- 1
2
3
4
5
6
7
8 [29] H. Baumann, On the estimation of unperturbed dimensions of long-
9 chain molecules in solution, *Journal of Polymer Science Part B: Polymer*
10 *Letters* 3 (12) (1965) 1069–1073.
11
12 [30] G. Smith, W. Paul, M. Monkenbusch, L. Willner, D. Richter, X. Qiu,
13 M. Ediger, Molecular dynamics of a 1, 4-polybutadiene melt. compari-
14 son of experiment and simulation, *Macromolecules* 32 (26) (1999) 8857–
15 8865.
16
17 [31] S. Lyulin, A. Gurtovenko, S. Larin, V. Nazarychev, A. Lyulin, Microsec-
18 ond atomic-scale molecular dynamics simulations of polyimides, *Macro-*
19 *molecules* 46 (15) (2013) 6357–6363.
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

Program title: Assemble!

Catalogue identifier: AEZE_v1_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEZE_v1_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: GNU General Public License, version 3

No. of lines in distributed program, including test data, etc.: 504242

No. of bytes in distributed program, including test data, etc.: 55078538

Distribution format: tar.gz

Programming language: Python 2.7.X.

Computer: Workstations.

Operating system: Windows, Linux, Mac OS X.

RAM: Depending on molecule database size and final system size: 10-20 Mb in console mode, 30-150 Mb using the GUI

Classification: 16.11.

External routines: numpy, wxpython (if using the GUI)

Nature of problem:

Preparation of complex polymeric systems for molecular dynamics simulations with Gromacs.

Solution method:

Creation of polymers collections using monomer units and a Gromacs-compatible force field. Packing multiple polymer copies in a box according to desired concentration.

Additional comments:

User manual and tutorial with necessary input files are distributed with the program

!!!! The distribution file for this program is over 55 Mbytes and therefore is not delivered directly when download or Email is requested. Instead a html file giving details of how the program can be obtained is sent. **!!!!**

Running time:

System dependent, but typically less than 5 seconds