Computer Physics Communications 283 (2023) 108567

Contents lists available at ScienceDirect

Computer Physics Communications

www.elsevier.com/locate/cpc

The Z1+ package: Shortest multiple disconnected path for the analysis of entanglements in macromolecular systems ☆,☆☆,☆☆☆

Martin Kröger [a,*], Joseph D. Dietz [b], Robert S. Hoy [b], Clarisse Luap [c]

[a] Polymer Physics, Department of Materials, ETH Zurich, CH-8093 Zurich, Switzerland
[b] Department of Physics, University of South Florida, Tampa, FL 33620, USA
[c] Independent researcher, CH-8049 Zurich, Switzerland

ARTICLE INFO

Article history:
Received 22 June 2022
Received in revised form 4 October 2022
Accepted 17 October 2022
Available online 24 October 2022

Keywords:
Entanglement network
Primitive path
Algorithm
Helical chains
Z code

ABSTRACT

This paper describes and provides Z1+, the successor of the Z- and Z1-codes for topological analyses of mono- and polydisperse entangled linear polymeric systems, in the presence or absence of confining surfaces or nano-inclusions. In contrast to its predecessors, Z1+ makes use of adaptive neighbor lists, and keeps the number of temporary nodes relatively large, yielding improved performance for large system sizes. Z1+ also includes several features its predecessors lacked, including several that are advantageous for analyses of semi-crystalline systems, brushes, nano-composites, and flowing liquids. It offers a graphical user interface that can be used to run Z1+ and visualize the results, and a PPA+ option that allows Z1+ to perform a primitive path analysis more efficiently than the standard procedure (PPA option). In addition to describing Z1+'s and PPA+'s implementation and computational performance in detail, we use it to show that it yields entanglement lengths that agree quantitatively with both a recently proposed unified analytic theory for flexible and semiflexible polymer-melt entanglement and with the available experimental data for these systems. Finally we show that the associated theoretical expressions, which express reduced entanglement-related quantities in terms of the scaled Kuhn segment density Λ, need not describe results for model polymer solutions of different "chemistries", i.e. different angular and dihedral interactions but the same Λ.

Program summary
Program title: Z1+
CPC Library link to program files: https://doi.org/10.17632/m425t6xtwr.1
Licensing provisions: Apache-2.0
Programming language: fortran 90, perl (standalone batch version) or java (interactive online version)
External requirements: perl 5+ (standalone batch version), jre 1.8.0+ (interactive online version)
Nature of problem: Starting from a multiple-chain configuration that consists of the coordinates of particles forming linear chains, as well as the dimensions of the rectangular or monoclinic (periodic or closed) cell containing the particles, the problem is to find the shortest multiple disconnected path obtained from the starting configuration upon (i) fixing chain ends at their original positions, (ii) disallowing bond crossings, and (iii) monotonically decreasing the path length.
Solution method: We use local geometric operations that fulfill the constraints (i)-(iii) and apply them as long as the path is still shrinking. This ultimately leads to disconnected polygons, i.e., paths characterized by nodes interconnected by straight lines. Each node has a corresponding 'entangled' node responsible for its existence. These pairs as well as the full configuration of the shortest path is reported.

☆ This work was supported by the Swiss National Science Foundation through grant 200021L_185052 and by the Swiss National Supercomputing Centre (CSCS) via project s987.
☆☆ The review of this paper was arranged by Prof. Weigel Martin.
☆☆☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).
* Corresponding author.
E-mail addresses: mk@mat.ethz.ch (M. Kröger), josephfox@usf.edu (J.D. Dietz), rshoy@usf.edu (R.S. Hoy), clarisse.luap@alumni.ethz.ch (C. Luap).
URL: https://www.complexfluids.ethz.ch (M. Kröger).

https://doi.org/10.1016/j.cpc.2022.108567
0010-4655/

Contents

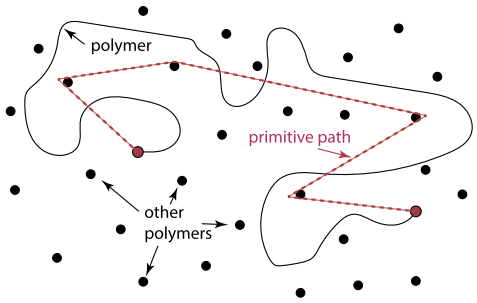

Fig. 1. Rubinstein and Helfand's interpretation [1] of Edwards' primitive path (PP) concept [2]. A single polymer (black curve) in a sea of polymers crossing the viewing plane (black circles). The PP (dashed red) is the series of connected line segments that shares its endpoints with the polymer and has minimum contour length. The entanglement network is obtained by reducing all chains to their PPs simultaneously [1,4,5]; for simplicity, the polymers forming the "sea" are depicted as immobile in this schematic drawing.

1. Introduction

Over the past 20 years, our understanding of polymer physics has been revolutionized by topological analyses (TA) of model polymeric systems. TA methods rely on Rubinstein and Helfand's interpretation [1] of Edwards' primitive path (PP) concept [2], which defines PPs as the axes of entangled polymers' tubes (as defined by the tube model of polymer melt rheology [3]). Specifically, TA methods aim to obtain PPs that are the shortest paths that chains fixed at their ends can contract into without crossing any other chains (Fig. 1). PPs obtained in this way have clarified the relationships between microscale intrachain and interchain structure and the entanglement of polymer melts [4–22], crystalline and semi-crystalline polymers [23–30], nanocomposites [31–45], thin films [46–50], networks and gels [51–56], brushes and polymers at interfaces [57–61], branched polymers [18,62–65], block copolymers [66–73], and other systems [74–77].

Generally, speaking, TA methods can be divided into two broad categories. Primitive path analysis (PPA) operates by disabling intrachain excluded-volume, angular and dihedral interactions while retaining interchain excluded-volume interactions, adjusting the covalent-bond interactions to set the equilibrium bond length to zero, and then allowing the system to contract into its corresponding PP network by integrating Newton's equations of motion at a low temperature [4,78]. The resulting PP network is a shortest multiple disconnected path (SMDP) in the sense that it is a state close to a local or global minimum of the total energy. Geometrical methods such as `Z` [5], `Z1` [79,80] and `CReTA` [7] aim to obtain a SMDP in its strict sense, without the need to introduce an energy functional and temperature. They treat polymer chains as volumeless and non-interpenetrable paths, and minimize their contour lengths subject to the constraints that (i) chains do not slip past each other and (ii) the contour length of each PP decreases monotonically.

Results obtained using these two categories of TA methods differ qualitatively from those obtained using earlier methods [81–84], but largely agree with each other. In particular, the statistical properties of the resulting networks are similar [79]. There are, however, important distinctions between them. Standard PPA, while it does an excellent job at obtaining SMDPs' statistical properties, performs poorly at identifying individual entanglements. Moreover, estimating the entanglement polymerization degree N_e using PPA, i.e., using the contour length of the SMDP, requires making assumptions about PPs' conformational statistics [4,12,85]. In contrast, geometrical methods accurately identify entanglement points' spatial locations and chemical positions [5,7], i.e. which monomer along a chain they correspond to. Since they allow estimation of N_e from the number of kinks along chains, they do not require making any such assumptions. This aspect makes geometrical algorithms preferable for anisotropic systems, e.g. analyses of entanglements in deformed glasses [31,32,86,87]. It is worth keeping in mind that the geometrical algorithms also yield PPs' conformational statistics.

The vast majority of the above-cited works employed Z, Z1, or standard PPA. In particular, since its original release in 2005 [5], the Z and Z1 algorithms have been employed in over 300 published studies. However, Z1 remained unpublished. Here, we deliver, test, and describe Z1+, the successor to the Z- and Z1-codes. There was a clear need for a robust update with additional features, motivated by the feedback received from Z1 users. To be specific, (i) the growing size of systems needed to be taken care of by a more suitable memory allocation and deallocation scheme, (ii) direct access to information about binary entanglements was not provided by Z and Z1, (iii) the calculation of the PPA network, using a separate software and other data formats, was time-consuming and prohibited the direct comparison of the outcome of two different TA methods, (iv) the input configuration format was restricted to so-called Z1-formatted files, (v) a simple integration with LAMMPS [88,89] was missing, (vi) the possibility of handling confining surfaces conveniently was missing. A further, important aspect of practical relevance is (vii) the speed of execution of the algorithms. While standard PPA is several orders of magnitude slower than Z or Z1, Z1+ provides a geometrical algorithm that is faster than Z1, and an implementation of PPA, so-called PPA+, that is only one order of magnitude slower than Z1+.

The outline of the rest of this paper is as follows. Section 2 provides instructions for installing and running Z1+, and a description of its output. Section 3 describes the geometrical Z1+ and dynamical PPA and PPA+ algorithms. Section 4 provides performance benchmarks, showing that Z1+ is much faster (computationally) than its predecessors, and also that it achieves faster convergence of N_e^{kink}-estimates with increasing N (the number of atoms or beads per linear chain), owing to its updated definition of kinks. Section 5 details applications of Z1+ to polymer nanocomposites, deformed systems, coarse-graining, semiflexible polymer melts, and polymeric systems with different microscopic interaction potentials. Finally, in Section 6 we summarize and conclude.

2. Installation, user manual and test runs

The public Z1+ distribution contains the following files, that should reside in your installation directory after successful download:

```
[Z1+] Z1+install.pl
[Z1+] Z1+template.pl
[Z1+] Z1+README.txt
[Z1+] module-shared.f90
[Z1+] module-output-formats.f90
[Z1+] module-messages.f90
[Z1+] module-folding.f90
[Z1+] module-license.f90
[Z1+] module-PPA.f90
[Z1+] module-CPPA.f90
[Z1+] module-main.f90
[Z1+] Z1+import-lammps.pl
[Z1+] Z1+rearrange.pl
[Z1+] Z1+.ex
[Z1+] LICENSE.txt
[Z1+] NOTICE.txt
[Z1+] Z1+.jar
[Z1+] .benchmark-X.Z1
```

where X extends over the range $\{01, 02, .., 14\}$ corresponding to 14 benchmark cases. Install Z1+ using the Z1+install.pl installer using the command

```
perl Z1+install.pl
```

This installer attempts to locate and report existing fortran compilers and perl versions, applies Z1+ to all benchmark configurations, and generates a script Z1+. If all tests have been successfully passed, the installation completes, and the user is asked to copy Z1+ to a location where it can be found, and to avoid running Z1+ in its installation directory. Entering the command Z1+ (calling Z1+ without command line arguments) is equivalent to entering Z1+ -help or Z1+ -h. If Z1+ is not found, your local path is missing in your path, and you can use ./Z1+ instead of Z1+ here, and in the following.

2.1. User manual

To make use of Z1+, you need to have access to a configuration of a polymeric system, e.g., generated by LAMMPS [88–90], genpol [91], gensaw [92], Simu-D [93], or any other academic or commercial software. To begin familiarizing yourself with the code, you can use one of the 14 benchmark-configurations available within the installation directory. To analyze a configuration, enter the command

```
Z1+ <configuration-file>
```

where <configuration-file> is the full path name of the configuration file, if it does not reside in the directory containing Z1+. Z1+ accepts the following (i) Z1-formatted snapshot or trajectory, (ii) lammps.data file, (iii) lammps.dump trajectory, and (iv) Z1+-formatted snapshot or trajectory (section 2.2). Other file formats like mol2, pdb, gromacs can be trivially converted to Z1-formatted or LAMMPS-formatted files.

There are a number of command-line options that do (with the exception of -selfZ and -PPA and -PPA+) not change the SMDP, but affect the amount of detail that is returned during execution:

```
-h or -help
-c or -clean
-l or -log
-s or -stats
-0 or -selfZ
-t or -PPA
-p or -PPA+
+  or -SP+
```

Use -help or -h to show the included documentation, -clean or -c to erase files from a previous run, -log or -l to create a logfile, -stats or -s to produce information about execution times of the various routines, -selfZ or -0 to take into account intrachain entanglements (the so-called self-entanglements), and + or -SP+ to create a SMDP with additional information about binary entanglements. The -PPA option performs a standard primitive path analysis (PPA) at finite temperature (section 3.2) instead of applying the main Z1+ algorithm (section 3.1). The -PPA+ option triggers an accelerated version of the standard PPA.

The following options can be used to pick selected snapshots from a trajectory:

```
-from=<frame-no>
-to=<frame-no>
```

The following command line options are available for lammps.data files containing bond information:

```
-ignore_H
-ignore_types=<type-no>[,<type-no>,<type-no>,..]
```

While -ignore_H ignores hydrogen atoms based on their mass, -ignore_types can be used to specify atom types that should be ignored, e.g., atoms belonging to side chains.

The Z1.jar file is a graphical user interface for the interactive online version of Z1+, written in java. It does not have the full functionality of the Z1+ code, and should not be used for configurations containing more than 10^5 nodes, but can be used to visualize a single configuration and its SMDP, and also to display statistical properties defined in this article. If the java runtime environment jre 1.8.0 or later is installed locally, one may just double-click the Z1.jar file to start the application.

2.2. Configuration file

Z1+ accepts the abovementioned four configuration file or trajectory formats. The format need not be specified by the user, since Z1+ recognizes it automatically. Here we provide details about these four possibilities.

Both Z1- and Z1+-formatted trajectory files are simply composed of their corresponding snapshots, one after the other, without any extra separators. A Z1-formatted snapshot contains 3 header lines followed by one line per atom (or bead). The first header line lists the number of chains C (including dumbbells, if present), the second line lists the (periodic or aperiodic) simulation cell edge lengths L_x, L_y, and L_z, and the third line has format $N_1\ N_2\ ...\ N_C$, where N_j is the number of beads in the jth chain; if the jth chain is a dumbbell, $N_j = 2$. Isolated or free beads should not be included in the configuration file, i.e., $N_j \geq 2$.

Thanks to the chosen programming language (fortran), for monodisperse systems (i.e., C chains with polymerization degree N) this sequence can also be encoded as C^*N instead of $N\,N\,..\,N$. The remaining $c = \sum_j N_j$ lines of the configuration file carry the x, y, and z-coordinates of the atoms, starting with the coordinates of the first atom of the first chain, followed by the 2nd atom of the first chain (the one bonded to the 1st atom), and terminating with the coordinates of the last atom of chain C.

A Z1+-formatted snapshot is identical with a Z1-formatted file, but can optionally have three additional lines to specify non-cubic periodic boxes, and to leave a integer-valued label (such as frame number or time step) that is used in the output to label results. Systems whose orthorhombic simulation cell has been sheared in x-direction, with a shear gradient in y-direction, can be specified by adding three lines to a Z1-formatted input configuration. Lines $(c+4)$ to $(c+6)$ should respectively be "-1", a numeric label such as the simulation time step, and the displacement $x_{\text{top}} - x_{\text{bottom}}$ along the x-direction between the central simulation cell and its first periodic neighbor in the positive y-direction. Using LAMMPS notation, this number corresponds to `xlo_bound − xlo`. It is also known as the Lees-Edwards parameter [94].

The `lammps.dump`- and `lammps.data`-files are described on the LAMMPS website [95]. Z1+ retrieves information about the simulation cell geometry, number of chains, bond connectivity, and atom coordinates from the `lammps.data` snapshot file. As information about connectivity is not included in `lammps.dump` snapshot or trajectory files, such files have to be stored in so-called `dump_modify sort-id` mode, and each molecule must have its individual molecule number.

Z1+ is designed to handle polydisperse linear polymeric systems. Polymers composed of two atoms, denoted as dumbbells, are not counted amongst the "true" number of polymers since their contour lengths cannot decrease during the reduction of a system to its corresponding SMDP, but can instead be used to model surfaces, confinement, and nano-inclusions.

2.3. Test runs

We set up 19 very different benchmark configurations (01 to 19) to test the validity of Z1+. The first 14 of these configurations are included with the Z1+ distribution. Snapshots for benchmarks 01, 02, 04, 05, and 14 are provided in Figs. 2, 3, 4, and 5, respectively, along with their SMDP networks. In these figures, the PPs are represented by transparent cylinders, and entanglement points are marked by white spots. Quantitative descriptors for all benchmarks are collected in Table 1. Users do not need to repeat these benchmark analyses manually; they are all performed and tested during installation.

To describe a single test run in detail, we apply Z1+ to benchmark 06, a system composed of 100 chains, each with 350 beads. Start Z1+ via

```
Z1+ <installation-directory>/.benchmark-06.Z1
```

The shortened output generated is:

```
    _____  ___
   /__  / /__ |
     / /    | |  _| |_     (c) 2022 mk@mat.ethz.ch
    / /_    | | |_   _|
   /____/   |_|   |_|

working directory: /home/user/configurations
installation directory: /home/user/Z1+

assuming Z1-formatted configuration file

[Z1+] removing Z1+SP.dat
[Z1+] finished cleaning

Z1+.ex launched ..

                        Z1+ atoms (init)       35000
                        Z1+ nodes (init)       35000
                              Z1+ chains         100
                         Z1+ true chains         100
                                Z1+ box    85.14970
                     Z1+ number density     0.05669
                              Z1+ min box    85.14953
                                 Z1+ epc     0.00000
      Z1+ init mean bondl (true chains)     0.95937
     Z1+ init mean bondl (+extra nodes)     0.95937
               Z1+ init max bondl (all)     1.14504
                              Z1+ lmax     1.14504
                         Z1+ thickness     0.00200
                              Z1+ rcut     0.99164
                                Z1+ M           86
                                Z1+ Mi          -1
                                Z1+ Mf           1
                          Z1+ binsize     0.99011
_______________ legend

Z1+ (legend)    pro: progress
Z1+ (legend)   rcut: cutoff radius
Z1+ (legend)    vis: visits
Z1+ (legend)    o  : crossings
Z1+ (legend)    new: new nodes
Z1+ (legend)    ---: too long
Z1+ (legend)    ?  : ghost nodes
Z1+ (legend)    x  : erased
Z1+ (legend)  bondl: mean edge length
Z1+ (legend)  nodes: number of nodes
Z1+ (legend)  maxID: largest node ID
Z1+ (legend) handle: # crosspoints handled
Z1+ (legend)   mem%: percentage of allocated memory used
_______________ Z1+ on the fly

 <Lpp>   rcut    vis  o new bond nodes maxID handled mem%
334.820  0.992     0   0  0 0.96 35000     0       0  0.0
195.290  0.992 34800   0  0 0.74 26429     0   34800  0.0
86.824   2.834 26273  64 44 1.78  4969 33444   61073 79.6
48.914   4.947  4822 128 53 3.21  1623 33550   65895 79.9
35.380   7.183  1469 141 46 4.83   843 33550   67364 79.9
30.537   8.936   677 134 34 5.99   614 33550   68041 79.9
28.760   9.932   439 123 25 6.31   550 33550   68480 79.9
28.130  10.303   369 120 19 6.39   530 34569   68849 82.3
27.919  10.431   348 121 18 6.39   527 34569   69197 82.3
 27.862 10.451   342 121 15 6.41   525 34569   69539 82.3
27.691  10.464   340 116 15 6.46   520 34569   69879 82.3
27.661  10.497   334 117 14 6.47   519 34569   70213 82.3
27.659  10.504   329 117 10 6.47   519 34569   70542 82.3
27.658  10.504   327 117  8 6.47   519 34569   70869 82.3
identifying kinks
27.348  18.193     0   0  0 6.07   280 34569   70869 82.3

_______________ Z1+ results for true chains
                                   chains         100
                                    <Lpp>    27.34798
                                      <Z>     0.80000
```

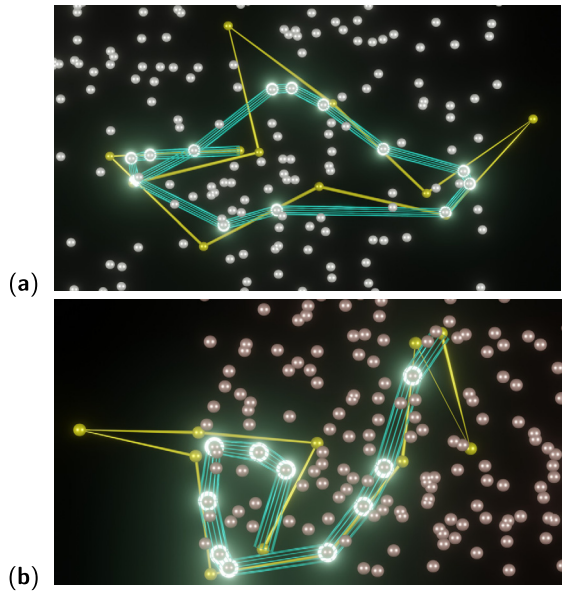

Fig. 2. Visualization of benchmark configurations 01-02 [panels (a-b)]. These configurations consist of single true chains plus many dumbbell obstacles that are aligned perpendicular to and cross the viewing plane. The true-chain configurations are shown in yellow while the shortest paths generated by `Z1+` are shown in green. Kinks contributing to $\langle Z \rangle$ are shown as white lamps. Relevant system properties are given in Table 1. All images were created using Blender 3.0 [96].

```
                                    Ree    24.72041
                                    app    22.34529
                                    bpp     0.07836
                                    <N>   350.00000
                    Ne_CK: classical kink   194.13541
                    Ne_MK:  modified kink   437.50000
                    Ne_CC: classical coil   285.15841
                    Ne_MC:  modified coil   796.13001
_________________ postprocessing Z1+ results
                            creating _line
                  creating Z1+summary.dat
                 creating Z1+summary.html
              analyzing+saving unfolded SP
_________________ generated files

 1) N_values.dat
 2) Ree_values.dat
 3) Lpp_values.dat
 4) Z_values.dat
 5) Z1+SP.dat
 6) Z1+summary.dat
 7) Z1+summary.html
 8) Z1+initconfig.dat

Z1+ finished after 0.28 seconds
```

Dumbells' chain statistics are not included in quantities such as $\langle Z \rangle$, $\langle L_{\text{pp}} \rangle$ and $R_{\text{ee}} = \langle R_{\text{ee}}^2 \rangle^{1/2}$ since these chains only serve as obstacles for the true chains.

2.4. Resulting files and information

2.4.1. Files `X_values.txt`

These files, with `X` = `N`, `Ree`, `Lpp`, or `Z`, have as many lines as analyzed snapshots and as many columns as true (non-dumbbell) chains. The nth column reports the number of nodes (for `X=N`), end-to-end distance (for `X=Ree`), contour length of the primitive path (for `X=Lpp`), and number of kinks (for `X=Z`) of the nth chain. Here nodes are either chain ends or kinks; only the latter contribute to $\langle Z \rangle$. If the number of chains varies between snapshots, so does the number of columns. All four files are generated for the user's convenience, but the information contained in these files can also be retrieved from `Z1+SP.dat`.

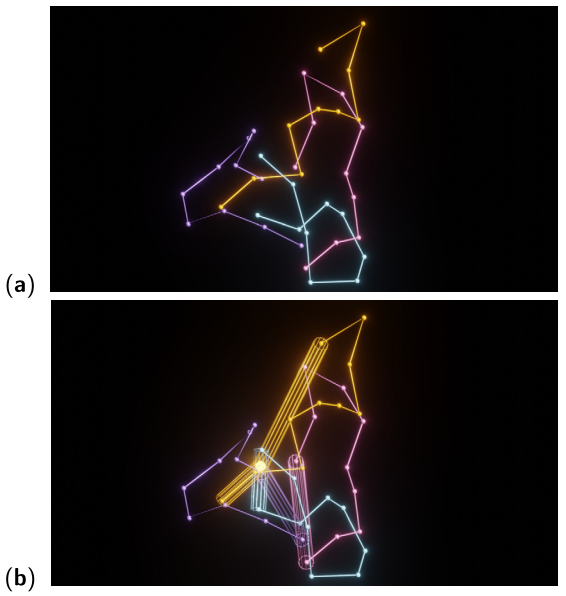

Fig. 3. (**a**) Benchmark 04 configuration and (**b**) the same configuration, together with its entanglement network. Relevant system properties are given in Table 1. Only 4 out of the 5 chains are shown; the 5th resides outside the displayed volume.

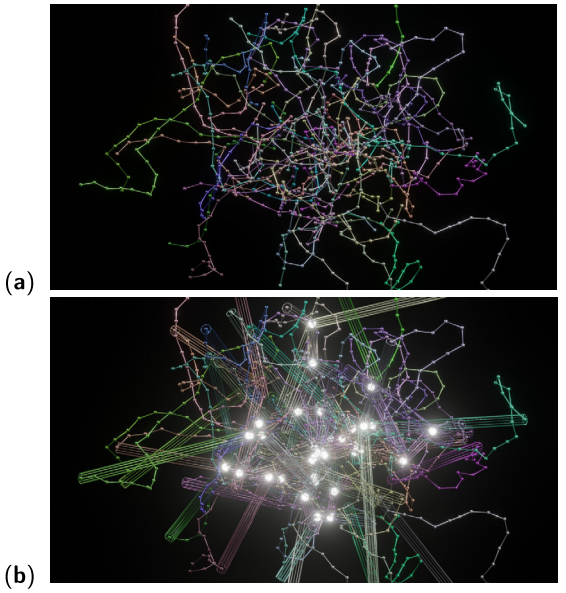

Fig. 4. (**a**) Unfolded benchmark 05 configuration and (**b**) the same configuration, together with its entanglement network. Relevant system properties are given in Table 1.

2.4.2. File containing the SMDP configuration

The `Z1+SP.dat` contains the full configuration of the SMDP, as well as additional information. For each of the analyzed snapshots, the format is as follows:

```
C
boxx boxy boxz
nodes of chain #1
xu yu zu node-position kink other-chain other-node
..
nodes of chain #2
...
```

where `C` is the total number of chains (including dumbbells), `boxx` the linear box size in x-direction, `xu` the unfolded x–coordinate of a node of the shortest path, `node-position` the continuous "atom number" of the original chain that can be assigned to the

Table 1
Quantities reported by `Z1+` in `Z1+summary.dat` for benchmark configurations nos. 01 to 19 (nos. 01 to 14 are included in the `Z1+` distribution), containing C chains with polymerization degree N, mean bond length $\langle \ell_0 \rangle$, at bead number density ρ. The directly measured quantities are the mean squared end-to-end distance $\langle R^2 \rangle$, the mean contour length of the PPs, $\langle L_{\rm pp} \rangle$, and the mean number of kinks per chain, $\langle Z \rangle$. All remaining quantities are derived quantities; $a_{\rm pp}$ and $b_{\rm pp}$ are defined as $a_{\rm pp} = \langle R^2 \rangle / \langle L_{\rm pp} \rangle$ and $b_{\rm pp} = \langle L_{\rm pp} \rangle / (N-1)$, and the various estimators $\mathcal{N}_e$ of the entanglement polymerization degree are defined in Appendix A. `PPA` and `PPA+` values of these quantities are not reported here. Benchmarks no. 18 and 19 have two snapshots each. Some of the configurations are visualized above, along with their entanglement networks; see Figs. 2-5 for benchmarks 01-02, 04, 05, and 14, respectively. σ denotes the length unit.

no	C	N	$\sqrt{\langle R^2 \rangle}$ $[\sigma]$	$\langle L_{\rm pp} \rangle$ $[\sigma]$	$\langle Z \rangle$	$a_{\rm pp}$ $[\sigma]$	$b_{\rm pp}$ $[\sigma]$	$\sqrt{\langle L_{\rm pp}^2 \rangle}$ $[\sigma]$	$\mathcal{N}_e^{\rm CK}$	$\mathcal{N}_e^{\rm MK}$	$\mathcal{N}_e^{\rm CC}$	$\mathcal{N}_e^{\rm MC}$	$\langle \ell_0 \rangle$ $[\sigma]$	ρ $[\sigma^{-3}]$
01	1	11	0.86	6.83	13.00	0.11	205.58	6.83	0.78	0.846	0.2	0.2	1.00	5.75
02	1	11	2.35	5.34	10.00	1.04	160.68	5.34	0.99	1.100	1.9	2.4	1.00	0.49
03	1	11	4.86	4.97	4.00	4.75	149.54	4.97	2.16	2.750	9.6	222.6	1.00	1.10
04	5	10	4.57	4.23	0.20	4.93	0.47	4.60	7.63	50.00	10.5	641.6	1.00	0.05
05	50	20	9.47	10.25	1.40	8.74	0.54	10.61	8.16	14.29	16.2	73.8	1.00	0.13
06	100	350	24.72	27.35	0.80	22.35	0.08	29.65	194.14	437.50	285.2	796.1	0.96	0.06
07	129	100	24.25	57.14	16.48	10.29	0.58	57.44	5.72	6.07	17.8	21.5	0.96	0.93
08	1032	100	24.15	60.92	19.00	9.57	0.62	61.21	5.00	5.26	15.6	18.3	0.95	0.93
09	8256	100	24.03	62.65	20.74	9.22	0.63	62.86	4.60	4.82	14.6	16.9	0.94	0.93
10	100	200	13.32	25.27	4.98	7.02	0.13	26.52	33.42	40.16	55.3	67.1	0.95	0.85
11	1044	100	91.95	91.94	0.05	91.97	0.93	91.96	94.52	> 100	99.0	> 100	0.97	0.85
12	37	819	93.88	249.47	13.78	35.33	0.31	252.32	55.40	59.42	115.8	131.4	1.45	0.05
13	64	1024	65.54	271.82	36.33	15.80	0.27	273.57	27.43	28.19	59.5	62.3	0.70	0.28
14	155	183	18.44	34.00	6.96	10.00	0.19	36.61	22.97	26.29	53.5	61.8	1.24	0.77
15	1044	100	12.74	29.98	10.62	5.41	0.30	30.58	8.60	9.42	17.9	20.8	0.97	0.85
16	100	400	26.52	60.22	10.30	11.68	0.15	61.59	35.39	38.84	77.4	90.9	0.97	0.85
17	984	100	35.99	71.32	15.46	18.16	0.72	71.51	6.07	6.47	25.2	33.6	0.97	0.80
18_1	1000	400	30.07	91.95	19.84	9.83	0.23	92.98	19.20	20.17	42.7	46.6	0.96	0.85
18_2	1000	400	30.13	92.05	19.88	9.86	0.23	93.16	19.16	20.12	42.7	46.6	0.96	0.85
19_1	1000	400	32.70	114.08	27.03	9.37	0.29	115.01	14.27	14.80	32.7	35.1	0.96	0.85
19_2	1000	400	32.79	114.40	27.11	9.40	0.29	115.35	14.23	14.76	32.8	35.1	0.96	0.85

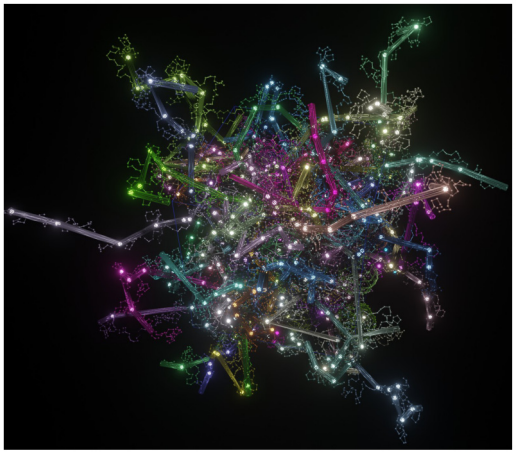

Fig. 5. Unfolded benchmark 14 configuration and its entanglement network. Relevant system properties are given in Table 1.

present node of the shortest path, `kink=1` for a kink, and `kink=0` for a terminal node. The last two columns are absent from this file if `Z1+` is called without the `-SP+` option. By default, each kink (interior node) of the shortest path has a corresponding node on the mutually entangled chain, provided by `other-chain` and `other-node`.

Here is an example for `Z1+SP.dat` generated using a single snapshot, benchmark 01 [Fig. 2a]. The program is called via `Z1+ + .benchmark-01.Z1`, and the execution time is 0.02 seconds on a single Intel Xeon Gold 6140R CPU core running at 2.3 GHz.

```
 301
 4.7372 4.7372 4.7372
 15
 0.000000   0.000000   0.000000     1.00 0
-0.689214  -0.033976   0.000000     1.40 1      95   1
-0.834315  -0.058901   0.000000     1.80 1      20   1
-0.803419  -0.231565   0.000000     2.18 1      80   1
-0.123370  -0.572624   0.000000     3.03 1     283   1
 0.284466  -0.457563   0.000000     3.50 1     128   1
 1.580799  -0.477470   0.000000     4.31 1     134   1
 1.767540  -0.255367   0.000000     5.02 1     275   1
 1.721692  -0.158411   0.000000     5.89 1      87   1
 1.103794   0.011219   0.000000     6.72 1     208   1
 0.645231   0.353517   0.000000     7.47 1     132   1
 0.399972   0.481420   0.000000     8.19 1     140   1
 0.249788   0.472715   0.000000     9.08 1      97   1
-0.350497  -0.000037   0.000000    10.04 1      36   1
-0.822600  -0.250800   0.000000    11.00 0
 2
 1.796600   1.274800  -1.000000     1.00 0
 1.796600   1.274800   1.000000     2.00 0
 2
 ...
```

The system is contained in a cubic box of box size 4.7372. The first chain is a true 11-atom chain, whose primitive path has 15 nodes. Its first node is located at the origin and its last node has `xu= -0.8226`. Because this chain lies in the $z = 0$ plane, its unfolded z-coordinates vanish. The 300 remaining chains are dumbbells with 2 nodes each.

The sixth column indicates the id (j) of the chain the given chain entangled with at that node; For example, the first kink has $j = 95$, indicating this entanglement is associated with chain 95. The last (seventh) column indicates the segment of chain j the node is entangled with; it has value i if it lies between nodes i and $i+1$ of chain j. Since dumbells have only two nodes, its value is always 1 in this example. Note that this benchmark configuration cannot be analyzed using the PPA algorithms, because the dumbbells have a bond length that is too large to prevent chain crossing.

If `Z1+` is called with the `-t` or `-p` options, the file containing the SMDP is named `PPA.dat` and `PPA+.dat`, respectively, to make sure the resulting file name carries information about the method chosen. Moreover, because there is no additional information like entanglement locations, these files are saved using the format of `Z1+initconfig.dat`, described next.

2.4.3. File *Z1+initconfig.dat*

For the user's convenience, and for visualization purposes, this file repeats the unfolded original configuration using the same coordinate system as for the shortest path. The file format is

```
C
boxx boxy boxz
atoms of chain #1
xu yu zu
..
atoms of chain #2
...
```

The terminal nodes of each chain should be identical in both `Z1+initconfig.dat` and `Z1+SP.dat`.

2.4.4. Summary files

The summary file `Z1+summary.dat` contains one line per snapshot. `Z1+summary.html` is an HTML-formatted version of this file that can be viewed in a browser. If `Z1+` is called with the `-t` or `-p` options, the summary files are named PPA-summary.dat and PPA+summary.dat, respectively, to make sure users can keep all three files conveniently in parallel. Each line of these files includes 15 numbers, in the following order:

```
label C N Ree Lpp Z app bpp LPP NeCK NeMK NeCC NeMC b0 n
```

Here `label` is a label retrieved from the trajectory, such as the time step for a `lammps.dump` trajectory, or the snapshot number for a `Z1+`-formatted trajectory. `C` is the number of chains, `N` is the mean number of nodes per true chain, `Ree` is the square root of the mean squared end-to-end distance of true chains, `Lpp` is the mean contour length of true chains' PPs, `Z` is the mean number of kinks per true chain (note that this value remains undefined, and is set to -1, if `Z1+` is called with the `-t` or `-p` options), `app` and `bpp` are obtained from `Ree`, `Lpp` and `N` as specified in the caption of Table 1, and `LPP` is the rms PP contour length. The `NeX` with `X` $\in$ {`CK`,`MK`,`CC`,`MC`} are the N_e estimators known as classical S-kink (CK), modified S-kink (MK), classical S-coil (CC), and modified S-coil (MC) (Appendix A). Finally, `b0` is the mean bond length of the original (true) chains, and `n` is the atom number density. Note that while `Z1+` reports single configuration estimators (S-estimators), various "M-estimators" [12] can be directly evaluated using the numbers contained in the `Z1+summary.dat` files, upon applying `Z1+` or `PPA` or `PPA+` to a homologous series of configurations; see Appendix A for further details.

3. Algorithms

In this section, we describe the geometrical `Z1+` and dynamical `PPA`/`PPA+` algorithms. The difference between the two approaches is visualized in Fig. 6, using a very simple example. The system consists of just two chains with fixed bond length $\ell_0 = \sigma$: a bent green chain with 14 beads, and a straight yellow chain with 10 beads. Fig. 6a shows the shortest disconnected path resulting from `Z1+`. The volumeless PPs are depicted as two series of transparent cylinders. There is a single kink on the green PP, giving rise to the entanglement point shared by both PPs, visualized as a white lamp. Fig. 6b shows the same shortest paths depicted in panel (a), and also the PPA resulting from `PPA` and `PPA+`, depicted as two series of solid cylinders. During energy minimization, the yellow chain actually increases its contour length. Due to chain sliding that is typical for PPA [78], the point of closest approach of the two PPs no longer lies in the plane spanned by the green chain. While the entanglement network is obviously quite different, the mean contour length of the PPs remains similar. Specifically, `Z1+`, `PPA` and `PPA+` respectively yield $\langle L_{\rm pp}\rangle \approx 10.09\sigma$, $\langle L_{\rm pp}\rangle \approx 10.09\sigma$, and $\langle L_{\rm pp}\rangle \approx 10.06\sigma$.

3.1. `Z1+` algorithm

The program reads one configuration at a time from the trajectory file provided by the user. Depending on the input file format, `Z1+` does the conversion to the internal file format used to store the PP network. The internal format uses a simulation box centered at the origin, and unfolded chain coordinates. As mentioned above, chains with more than two beads are denoted as true chains; dumbbells serve as obstacles. All chain ends remain fixed in space throughout the process. Each chain is represented by a varying number of nodes, originally coinciding with the beads. At each step of the sequential minimization procedure, three adjacent nodes of some chain A are considered. The three nodes form a triangle that defines an area enclosed by the triangle via its convex hull. Using adaptive neighbor lists, `Z1+` searches for all segments belonging to other chains (B, C, ...) that cross this area, and selects a segment from this list that allows it to maximally displace one of the nodes of the triangle continuously without producing overlap between chains during the continuous displacement along a straight line (Fig. 7). Depending on all the angles and distances in-

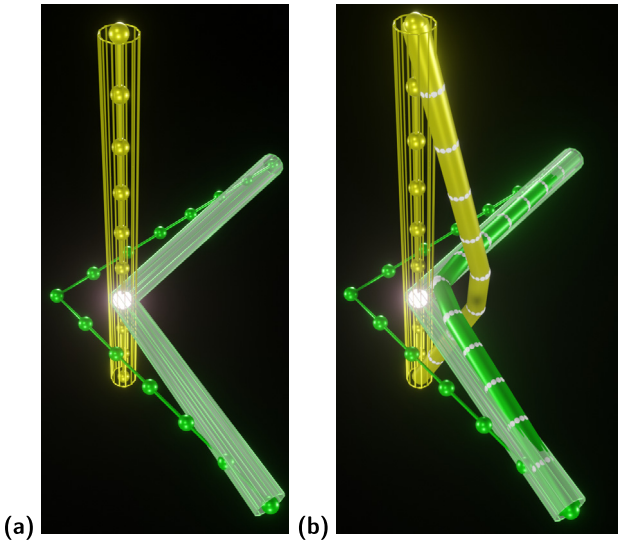

Fig. 6. Qualitative difference between **(a)** `Z1+` and **(b)** PPA, when applied to a configuration with two multibead chains (yellow and green). The white kink on otherwise straight PPs (transparent cylinders) created by `Z1+` represents an entanglement. The contour lengths of the curved PPs (opaque cylinders) resulting from PPA (either `PPA` or `PPA+`) contain information about the number of entanglements estimated using random walk statistics.

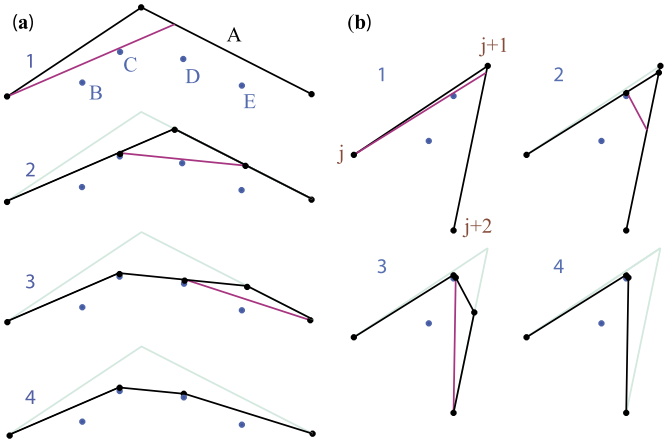

Fig. 7. Illustration of the steps performed during `Z1+`'s contour length minimization. In both panels, three nodes connected by two (black) bonds transform into four nodes connected by three (black) bonds over the course of steps 1-4. Two scenarios are shown in (**a**) and (**b**) for different opening angles of the triangle spanned by the three nodes. The blue solid circles mark other chains currently passing through the triangle. In contrast with `Z`, `Z1+` considers one single shortest (red) segment at a time; this segment begins on the jth node and terminates on the straight segment connecting nodes $j+1$ and $j+2$.

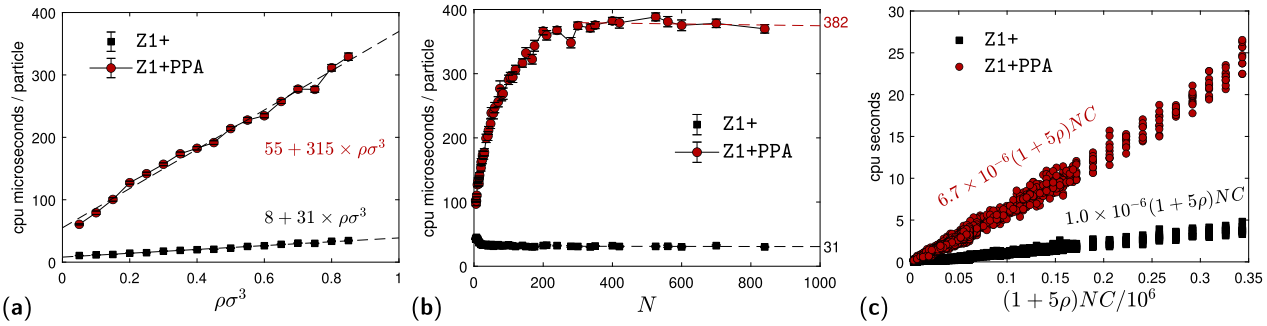

Fig. 8. Comparison of runtimes for `Z1+` and `PPA+`. All times are for a single Intel Xeon Gold 6140R CPU core running at 2.3 GHz. All results are averaged over either one thousand [panels (a-b)] or two hundred [panel (c)] independent configurations, generated using `genpol`. (**a**) Microseconds per particle versus particle number density $\rho\sigma^3$ in $NC = 10^4$ systems with $C = 100$ and $N = 100$. Runtimes are nearly linear in ρ; linear fits are shown with dashed lines. (**b**) Microseconds per particle versus N in $\rho = 0.4\sigma^{-3}$ systems with $NC = 8400$. While the speed tends to increase with increasing N for `Z1+`, it decreases with N for `PPA+`. However, both versions' runtimes approach a constant in the large-N limit; the asymptotic values are given on the right margin of the panel. (**c**) Runtimes vs. NC for a wide variety of N, C, and ρ. The choice of horizontal axis illustrates the fact that runtimes are roughly linear in $(1+5\rho)NC$.

volved in this operation, `Z1+` eventually inserts additional 'ghost' nodes. These nodes improve the performance of the neighbor-list routine, but do not affect the final results.

Once the contour length of the disconnected path network has reached a minimum, the network is reexamined, ghost nodes on straight partial chains are erased, and information about pairs of entanglement points is collected and stored in `Z1+SP.dat`.

Because `Z1+` is the successor of the published `Z` code and the online `Z1` application, we should mention the key differences here: (i) `Z` and `Z1` do not make use of neighbor lists, but instead reduce the number of nodes during the minimization process; (ii) `Z` does not report the atom number of the original chain assigned to the present node (4th column in `Z1+SP.dat`); (iii) `Z` does not properly handle periodic systems that contain any chains whose sizes exceed the box size; (iv) `Z` treats dumbbells as true chains; (v) `Z` and `Z1` do not report pairs of entangled nodes (columns 6 and 7 in `Z1+SP.dat`). Such node pairs (i, k) [or more precisely, node-point pairs] are identified by `Z1+` while analyzing a version of the final SMDP that still includes its ghost nodes. The distance between an entangled node i and one of the points on the entangled segment between nodes j and $j+1$ from another chain is zero. In the rare case where more than a single pair can be assigned to an entanglement, only one of them is specified via the last two columns in `Z1+SP.dat`.

3.2. *PPA and accelerated PPA+ algorithms*

If `Z1+` is started with the `-PPA` or `-PPA+` option, it assumes that the initial configuration is similar to a Kremer-Grest polymer melt, i.e., that all particles (mass m) interacted via the Lennard-Jones/WCA potential with well depth ϵ, that bonded particles were connected by a finitely extensible nonlinear elastic (FENE) potential, that the maximum bond length is well below 1.5σ, and that the Lennard-Jones monomer diameter σ was used as the unit of length in reporting the dimensionless particle coordinates. The base version (option `-PPA` or `-t`) of the present code follows the standard PPA procedure [78]. It takes the configuration, sets the FENE spring coefficient to $100\epsilon/\sigma^2$, removes all nonbonded WCA interactions, sets the timestep to $\Delta t = 0.008\sqrt{m\sigma^2/\epsilon}$, the temperature to $T = 10^{-3}\epsilon/k_B$, fixes all terminal beads of chains in space, and then performs a classical molecular dynamics in the NVT ensemble, using a velocity rescaling thermostat, over a duration of $t = 10^3\sqrt{m\sigma^2/\epsilon}$. Initial velocities of mobile beads are set to zero. The resulting PP configuration is returned in `PPA.dat`, and the derived quantities are collected in the files already known from the `Z1+` analysis, with the exception of `Z_values.dat`, which remains empty. For reasons explained below, the base version may be primarily useful for exactly reproducing existing results and doublechecking results obtained using the accelerated version, denoted as `PPA+`.

`PPA+`, which differs from the base version mainly in its speed of execution, is invoked if the `-PPA+` or `-p` option is used. It has five stages (I-V). Each stage is characterized by four parameters: maximum duration, temperature, integration time step, and a neighbor list skin thickness, all specified in Lennard-Jones units: Stage I (10,0.1,0.008,0.7), stage II (10,0.05,0.008,0.7), stage III (10,0.005,0.01,0.6), stage IV (10,0.001,0.01,0.3), and stage V (10,0.0001,0.01,0.2). Each of the five stages terminates prior to its maximum duration if the change of the contour length of the PPA network during a single time remains below a threshold. The use of this threshold ensures that the code proceeds to the next stage as soon as the current one becomes inefficient. All details are available in `module-PPA.f90`. The higher efficiency of `PPA+` arises from its temperature-adapted time step and dynamical neighbor list skin thickness, encoded by the abovementioned parameter values, that take into account the dynamical slowing down during the energy minimization process.

4. Performance benchmarks

Because `PPA+` is orders of magnitude faster than the standard `PPA`, we skip a quantitative comparison between PPA implementations. Similarly, since the `Z`-code suffers from finite size effects and is substantially less computationally efficient than `Z1+`, we do not report a quantitative comparison of `Z1+`'s performance to `Z`'s here. `Z1` was used during the last decade mainly in its online version, for which speed is very relevant, but we emphasize here that `Z1+` is even faster. The advantage of `Z1+` increases with increasing system size. Below, we focus on how `Z1+`'s performance scales with $\rho\sigma^3$, N and NC, how `Z1+`'s performance compares to that of `PPA+`, and how entanglement-related quantities obtained using `Z1+` compare to those obtained using `Z1` and `PPA+` analyses of the same systems.

Speed of execution — First we test the performance of `Z1+` as well as `PPA+` upon varying the particle number density $\rho = NC/V$ at constant number C of chains and polymerization degree and N, or the polymerization degree N at constant number density ρ and total number of particles NC. This is done in Figs. 8a, b. Varying all system parameters ρ, N, and C almost randomly leads to the results shown in Fig. 8c. The measured CPU times for the execution of `Z1+` are well captured by $t \simeq 10^{-6} \times (1+5\rho)NC$ seconds, where ρ enters in units of σ^{-3}. The CPU times for the execution of `PPA+` scale similarly, but are a factor of ~ 7 slower. These results are compatible with those presented in Figs. 8a, b.

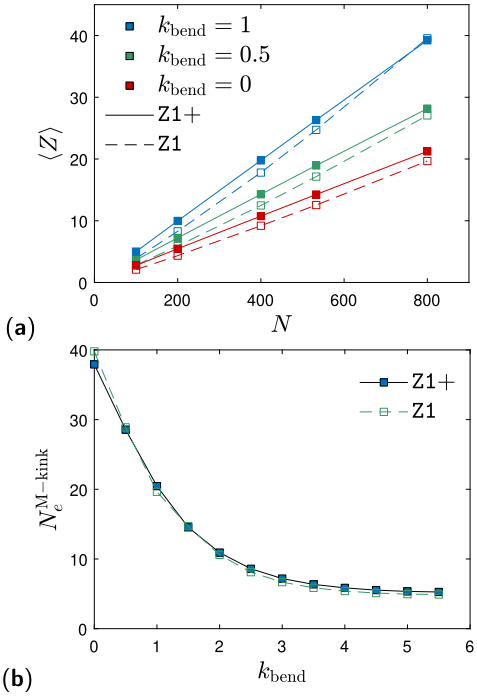

Fig. 9. Comparison between `Z1+` and its online predecessor, `Z1`, using weakly semi-flexible and fully relaxed Kremer-Grest monodisperse polymer melt configurations with $N \in [10, 800]$ at $T = \epsilon/k_B T$ and $\rho = 0.85\sigma^{-3}$, available from Ref. [22]. (**a**) Mean number of kinks $\langle Z \rangle$ versus N for systems with three different dimensionless bending stiffness values. (**b**) Estimator $N_e^{\rm M-kink}$ (Appendix A) versus $k_{\rm bend}$. The relative deviation is below 8% over the whole stiffness range.

`Z1` versus `Z1+` — Next we compare the entanglement networks produced by `Z1` and `Z1+` analyses of Kremer-Grest melts with a wide range of chain stiffnesses and chain lengths. Fig. 9a displays the mean number of kinks $\langle Z \rangle$ versus polymerization degree N for systems with three different $k_{\rm bend}$; cf. Eq. (3). Although results from `Z1` and `Z1+` are comparable, $\mathcal{N}_e^{\rm M-kink}(N) \equiv d\langle Z \rangle / dN$ converges slightly faster with increasing N when `Z1+` is employed. Since this faster convergence is a healthy feature [12], `Z1+` should be used instead of `Z1` in future studies, and the online version of `Z1` is going to be replaced by `Z1+`.

Differences between `Z1` and `Z1+` are even smaller if quantities related to the contour length $\langle L_{\rm pp} \rangle$ are compared. One reason for the abovementioned discrepancy between `Z1` and `Z1+` results is their slightly different definition of kinks; for details see the source codes. Since quantities like $\langle L_{\rm pp} \rangle$ and $\langle Z \rangle$ are all obtained straightforwardly from the shortest path configurations output by `Z1+`, users are encouraged to implement their own definitions of kinks, binary, ternary and higher-order contacts, etc. While the absolute $\langle Z \rangle$ values for any given system can depend significantly on the kink definition, the ratios of these values for systems with different N, deformation histories, electromagnetic field strengths, etc. should not if the definition is robust.

`PPA+` versus `Z1+` — We calculated the $N_e^{\rm SE}$ and $N_e^{\rm M-coil}$ of the abovementioned set of Kremer-Grest melts from their mean squared radii of gyration $\langle R^2 \rangle$, mean bond lengths $\langle \ell_0 \rangle$ and Kuhn lengths ℓ_K, as described in Ref. [22] and Appendix A. Results are shown in Fig. 10b. Because the $\langle L_{\rm pp} \rangle$ returned by `Z1+` are typically smaller than $\langle L_{\rm pp} \rangle$ returned by PPA algorithms (Fig. 6 and Fig. 10a), the N_e-estimators differ; the fractional differences increase with increasing chain stiffness. Fig. 10c displays the relevant ratios. A ratio close to unity signals good agreement between the SE- and M-coil estimators, which is the expected behavior. This expectation is slightly better-fulfilled by `Z1+` values of $\langle L_{\rm pp} \rangle$.

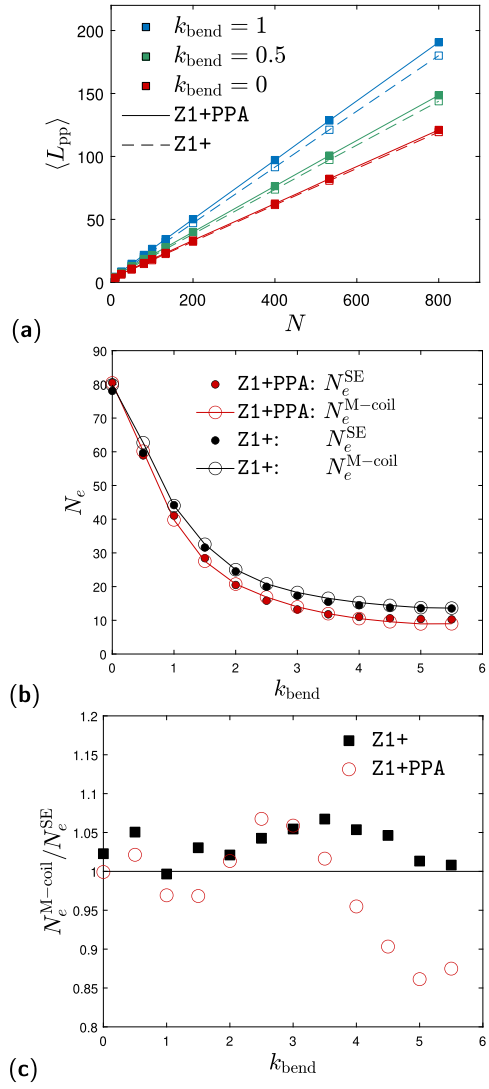

Fig. 10. Comparison of `Z1+` and `PPA+` results for mean PP lengths and two N_e-estimators for semiflexible Kremer-Grest chain melts with $N \in \{10, 800\}$, $\rho = 0.85\sigma^{-3}$ and temperature $T = \epsilon/k_B$ [22,97]. (**a**) shows $\langle L_{\rm pp} \rangle$ versus N for three selected $k_{\rm bend}$ values. (**b-c**) respectively show the absolute values of $N_e^{\rm SE}$ and $N_e^{\rm M\text{-}coil}$ (Appendix A) and their ratio.

5. Selected applications

5.1. Polymer nano-composites

`Z1+` can be used to study the entanglement network in the presence of rigid obstacles like nanoparticles [98–100] or planar boundaries [101–104]. To do so, the user has to mesh the confining surfaces. As any mesh consists of a number of straight edges, every edge has to just be re-interpreted as a dumbbell (polymer with $N = 2$ beads). Because `Z1+` does not move the terminal beads of a polymer, it will keep the dumbbells, and thus the confinement, in place. The shortest path produced by `Z1+` then contains the unmodified obstacles, and the PPs are not able to enter them because chains do not cross during the minimization process. The same principle can be applied for `PPA` and `PPA+`. In that case the largest distance between neighboring nodes of the mesh must remain below 1.5σ to make the obstacles' surfaces uncrossable for chains. As for `Z1+`, the `PPA` and `PPA+` keep dumbbells in place during the minimization process. All benchmark configuration featuring chains with $N = 2$ contain nanoparticles or other types of confinement.

5.2. Deformation-induced variation of the entanglement network

Since `Z1+` works with non-cubic (including sheared) simulation cells, it is readily applicable to studies of flowing/deformed/sheared/elongated systems [86,87,105–109]. Comparing a series of SMDPs obtained over a range of strains allows one to follow the trajectories of entanglement points and study the loss or gain of entanglements, as was done in Refs. [87,109]. LAMMPS [88,89] users may invoke the built-in command `shell Z1+ <datafile>` or `shell Z1+ <dumpfile>` after a `writedata` or `dump` command to call `Z1+` on the fly. Alternatively, users may save the configuration in a `Z1+`-formatted file and afterwards call `Z1+` from within their simulation program.

5.3. Coarse-graining

Primitive-path statistics of entangled polymers have been used to map multi-chain simulations onto single-chain mean-field theory [110]. Shortest-path analysis of coarse-grained molecular simulations was applied to improve physical understanding and molecular design via machine learning algorithms [111]. It was also applied to atomistic models [8,112] to rate the performance of coarse graining in estimating polymer properties. Hierarchical coarse-graining schemes have been developed and tested [107] for both equilibrium and non-equilibrium situations. The road from chemistry to rheology has also been explored via systematic coarse-graining of the dynamics of entangled polymer melts [113,114], including DNA [104]. All these works employed the unpublished `Z1`. The additional features provided by `Z1+`, its `PPA` and `PPA+` options, the added information about entangled nodes, speed of execution, additional data input formats, LAMMPS compatibility, etc., should help to make such studies not only more convenient, but also facilitate new types of entanglement-network analyses.

5.4. Semiflexible, monodisperse FENE chain melts

Developing a quantitatively accurate, unified analytic model to describe the entanglement of both flexible and semiflexible polymer melts [12,85,94,115–118] starting from their local intrachain and interchain structure was a long standing challenge. Very recently, as described in Refs. [21,22], we showed that PPA estimates of N_e, obtained using the Svaneborg-Everaers estimator N_e^{SE} (Appendix A), for bead-spring melts spanning the entire range of chain stiffnesses over which systems remain isotropic, yield the following prediction for the reduced plateau modulus $G\ell_K^3/k_BT$ [4]:

$$\frac{G\ell_K^3}{k_BT} = \frac{4}{5}\left[c_1\Lambda^{-3} + c_2\Lambda^{-2} + c_3\Lambda^{-7/5}\right]^{-1}, \qquad (1)$$

with $c_1 = 257$, $c_2 = 10.8$, and $c_3 = 4.75$. Here G is the melt plateau modulus estimated from $G = 4\rho k_BT/5N_e$, $\ell_K = C_\infty\langle\ell_0\rangle$ is the Kuhn length, and $\Lambda = \rho_K\ell_K^3$ is the scaled Kuhn segment density, where ρ_K is the number of Kuhn segments per unit volume. This expression unifies the Lin-Noolandi, Edwards-de Gennes/Milner, and Morse scaling theories of entanglement in flexible-, semiflexible-, and stiff-chain melts [119–124]. As shown in Fig. 11, it quantitatively matches all bead-spring data, semiquantitatively matches the entire available experimental dataset over several orders of magnitude in both Λ and $G\ell_K^3/k_BT$.

In general, TA results can be highly sensitive to both the TA method employed and the mathematical formula used to estimate N_e [12,85]. We showed in Refs. [21,22] that although the functional form of Eq. (1) is insensitive to these factors, the c-coefficients are rather sensitive to them. However, as we will now demonstrate, one can obtain an equally good fit to the available bead-spring and

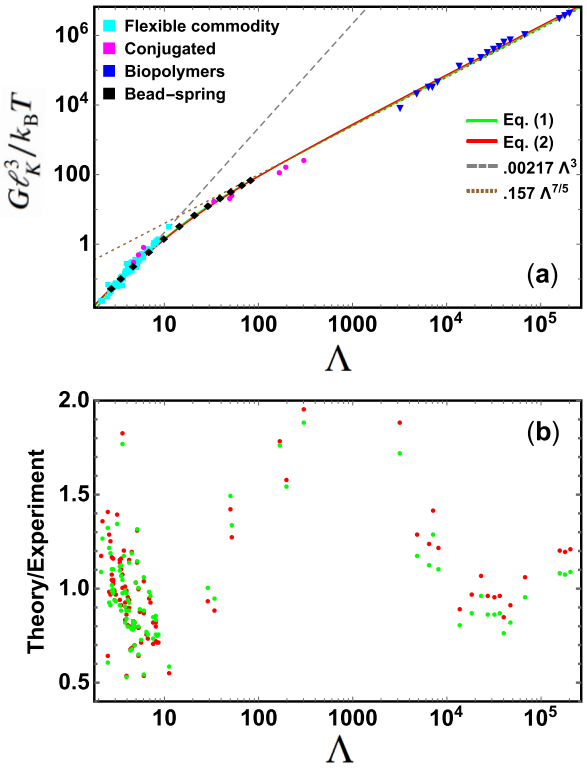

Fig. 11. (**a**) A comparison of theoretical expressions for $G\ell_K^3/k_BT$ to experimental results for flexible commodity-, semiflexible conjugated-, and stiff bio-polymer melts [126–131]. Solid curves show the unified analytic formulas discussed above, while the dashed and dotted lines respectively show fits to Lin-Noolandi and Morse scaling. (**b**) Ratios of these theoretical predictions to experimental values of $G\ell_K^3/k_BT$ for systems at the same Λ. Comparisons to other unified expressions for $G\ell_K^3/k_BT$ [11,131] are given in Ref. [22].

experimental data using topological N_e-estimates obtained using `Z1+` and the $N_e^{\mathrm{M\text{-}kink}}$ estimator. Fitting $N_e^{\mathrm{M-kink}}(C_\infty)$ to $\alpha C_\infty^{-3} + \beta C_\infty^{-2} + \gamma C_\infty^{1/5}$ as described in Refs. [21,22] yields $\alpha = 376$, $\beta = 26.5$, and $\gamma = 4.75$. Combining the relations $c_1 = \alpha(\rho\ell_0^3)^2$, $c_2 = \beta(\rho\ell_0^3)$, $c_3 = \gamma(\rho\ell_0^3)^{2/5}$ [22] with Everaers' rationalization of our observation that $N_e \approx N_e^{\mathrm{M\text{-}coil}} \approx 2N_e^{\mathrm{M\text{-}kink}}$ [12,16], and hence $G = 2\rho k_BT/5N_e^{\mathrm{M-kink}}$, yields the alternative prediction

$$G\ell_K^3/k_BT = \frac{4}{5}\left[218\Lambda^{-3} + 20.1\Lambda^{-2} + 4.26\Lambda^{-7/5}\right]^{-1}. \qquad (2)$$

This expression is compared to Eq. (1) and to the available experimental data in Fig. 11. The root mean squared error is $\sim 27\%$, which is slightly larger than that of Eq. (1) (25%), but the mean error is smaller (-0.5% rather than 2.3%). Note that obtaining the c-coefficients in Eqs. (1)-(2) required employing an N_e-estimator (such as $N_e^{\mathrm{M-coil}}$ or N_e^{SE}; cf. Appendix A) that properly accounts for PPs' non-Gaussian structure, extrapolating all results to the infinite-chain-length ($N \to \infty$) limit, obtaining excellent statistics to minimize statistical errors, and eliminating other known sources of systematic error such as poor sample equilibration [97,125].

5.5. Effect of microscopic interaction parameters on entanglement network

As described above, the reduced plateau modulus $G\ell_K^3/k_BT$ of dense semiflexible Kremer-Grest melts for fixed monomer number density ρ has been found to depend on a single dimensionless parameter, the scaled Kuhn segment density Λ. Similarly, for fixed ρ, N_e was found to depend only on chains' characteristic ratio $C_\infty = \ell_K/\ell_0$. Experiments on a wide variety of polymers also find

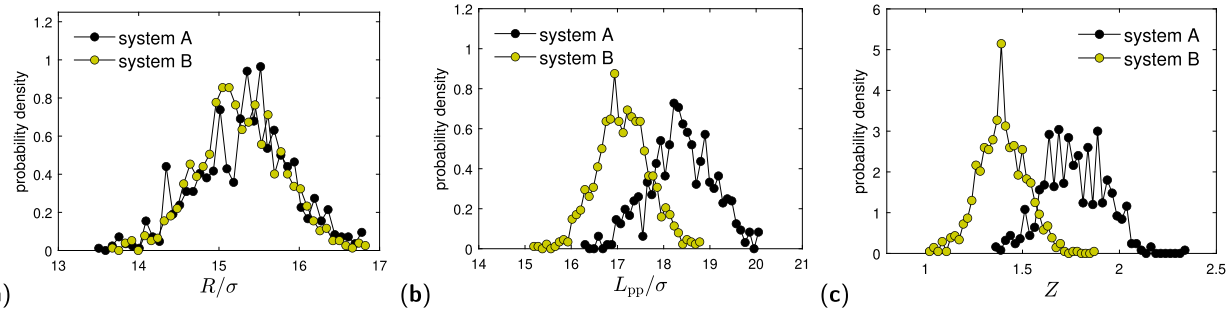

Fig. 12. Analysis of a series of systems A and B that exhibit the same number of chains ($C = 100$), polymerization degree ($N = 100$), bead number density ($0.4\sigma^{-3}$), Kuhn length ℓ_K, and thus identical Λ, but differ in the bending and torsion potentials. (**a**) Distribution of end-to-end distances, (**a**) distribution of PP contour lengths L_{pp} obtained via `Z1+`, (**a**) distribution of number Z of entanglements. The result remains qualitatively unchanged if we replace `Z1+` by `PPA+` results in panel (b).

that, at least to first order, dense melts' $G\ell_K^3/k_BT$ depends only on Λ [126–131]. However, this one-to-one relationship of $G\ell_K^3/k_BT$ to Λ is known to break down as ρ decreases into the semidilute regime [22,123,132]. Thus a natural followup question to ask is under which conditions less-concentrated melts with identical ρ and Λ, but different microscale structure, have the same N_e and hence the same G/k_BT.

To answer this question, we prepared two systems (A and B) that have different bending and torsion potentials, but the same ρ, ℓ_0, ℓ_K, N, and thus also the same Λ. The existence of such systems was demonstrated in Ref. [92]. A and B are monodisperse hard-sphere-chain systems with $N = 101$, sphere diameter $d = 0.9\sigma$, bond length $\ell_0 = \sigma$, and intramolecular bending and torsion potentials given by

$$\frac{U_{\text{bend}}(\theta)}{k_BT} = k_{\text{bend}}(1 - \cos\theta), \tag{3}$$

$$\frac{U_{\text{tors}}(\phi)}{k_BT} = -k_{\text{tors}}\phi^2. \tag{4}$$

Here $\theta \in [0, \pi]$ and $\phi \in (-\pi, \pi]$ denote bending and torsion angles, respectively, as in [92]. More precisely, if $\mathbf{u}_j = (\mathbf{r}_j - \mathbf{r}_j)/|\mathbf{r}_j - \mathbf{r}_j|$ denotes the jth unit bond vector along a polymer chain, $\cos\theta_j = \mathbf{u}_j \cdot \mathbf{u}_{j+1}$ and $\phi_j = \text{atan2}[\mathbf{u}_{j-1} \cdot (\mathbf{u}_j \times \mathbf{u}_{j+1}), (\mathbf{u}_{j-1} \times \mathbf{u}_j) \cdot (\mathbf{u}_j \times \mathbf{u}_{j+1})]$.

System A, with $k_{\text{tors}} = 0.21$ and $-k_{\text{bend}} = 2.0$, and system B, with $k_{\text{tors}} = 0$ and $-k_{\text{bend}} = 4.0$, have identical ℓ_K and Λ at $\rho = 0.4\sigma^{-3}$, which is a density small enough to ensure that the intramolecular interaction potentials fully determine the conformational statistical properties. As shown in Fig. 12, we found that these systems' L_{pp}- and Z-distributions are significantly different. This finding demonstrates a scenario where $G\ell_K^3/k_BT$ does not depend only on Λ and ρ. It should be relevant for the study of polymer solutions, and may help explain deviations between theoretical expressions and experimental data [22].

6. Conclusions

Inspired by the ever-increasing demands of the computational physics, biology and chemistry communities to analyze and characterize larger systems as well as polydisperse, confined, doped, deformed, ionic, or composite systems containing linear polymeric or macromolecular strands, the ongoing developments in compilers and memory-allocation schemes, requests for different file formats, and the apparent non–availability of an efficient classical PPA solver, we delivered a completely revised version of the `Z`-code, the so-called `Z1+` code and showed that it addresses all of the above-mentioned demands. Like its predecessors, `Z1+` is easy to use and parameter-free. It returns the SMDP and related results in a structured fashion that allows for direct postprocessing, and can handle multiframe trajectories (as opposed to being limited to single-frame snapshots). The `Z1+` and `PPA+` packages are designed for linear polymers, but may be extended to nonlinear polymer architectures, e.g., branched polymers with mobile junctions. Systems composed of cyclic chains might be better characterized using knot algorithms.

We also developed and provided `PPA+`, an accelerated version of the standard PPA algorithm. We considered several options for further accelerating `PPA+`, but doing so proved to be difficult; for example, accelerating it by employing classical unconstrained energy minimizers instead of its dynamic (force-based) energy minimization is inappropriate because these algorithms do not prevent chain disentanglement, chain slippage, and chain crossing. We recommend using `PPA+` (and PPA analyses, more generally) with caution, if the system contains dumbbells. While `Z1+` allows the user to analyze composite or confined systems using its built-in option to mesh the corresponding surfaces, `PPA+` results may depend strongly on the resolution of the chosen mesh and thus may be less physically relevant. Also note that the assumptions underlying the coil-based N_e-estimators (Appendix A) can fail dramatically in nonequilibrium systems such as deformed glasses, whereas kinks can still be extracted to accurately characterize the entanglement network [87].

To leave readers with an impression about the potential uses of `Z1+` and `PPA+`, we described a number of applications that either profited from `Z` extensions we provided during the last several years (including for example `Z1+` and its online version), or can benefit from other improvements included in the stock `Z1+` code. For example, we used `Z1+` and `PPA+` (as well as `Z1` and `PPA` for comparison) to analyze a large set of semiflexible bead-spring melt configurations [22,97]. This allowed us to discuss differences between standard PPA and `Z1+` and the convergence behavior of commonly employed N_e-estimators. We also generated and analyzed tangent-hard-sphere polymer solutions with two different types of angular/dihedral interactions, aiming to shed some light on the possible relevance of microscale intrachain and interchain structure for entanglement in polymer solutions.

Declaration of competing interest

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

Data availability

All data is included with this article and related computer program.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. DMR-1555242. We thank Achim Kröger for his help with the graphical user interface. We greatly acknowledge efforts made by an anonymous referee, who performed a thorough independent test of our code on an impressive variety of physical systems including freely-jointed chains of hard spheres [133], polymer solutions, polymer nanocomposites with cylindrical or spherical nanofillers, and polymers under confinement.

Appendix A. N_e-estimators

As described in Section 2.1, the `Z1+` software returns the complete SMDP, pairs of entangled nodes, contour lengths, gyration radii, etc. As a service for users, some of the quantities trivially derived from the data are reported as well. These include several estimates of the chemical entanglement length N_e. Here we summarize the definitions of these estimators. In every case we denote the N-dependent estimator by $\mathcal{N}_e$, so that [12]

$$N_e = \lim_{N\to\infty} \mathcal{N}_e(N) \tag{5}$$

There are two categories of estimators. "S"-estimators use data from a single configuration with a single chain length N, while "M"-estimators require a series of configurations with a range of N [12], or quantities extrapolated to $N \to \infty$, and thus cannot be returned without making any assumptions relying on the coordinates of a single configuration. `Z1+` automatically calculates several S-estimators, and M-estimators can be calculated by post-processing `Z1+`-generated output. "Coil" estimators that make use of $\langle L_{\rm pp}\rangle$ and/or $\langle L_{\rm pp}^2\rangle$ come in two flavors. One can either use the $\langle L_{\rm pp}\rangle$ value returned by `Z1+` in its regular mode, or use the $\langle L_{\rm pp}\rangle$ and $\langle L_{\rm pp}^2\rangle$ values returned upon starting `Z1+` using the `-PPA+` option. Estimators making use of the PPA values will be labeled by an additional 'PPA'.

The S-estimators `Z1+` automatically calculates are

- $\mathcal{N}_e^{\rm S-coil} = (N-1)\langle R^2\rangle/\langle L_{\rm pp}\rangle^2$,
- $\mathcal{N}_e^{\rm S-kink} = N(N-1)/[N+(N-1)\langle Z\rangle]$,
- $\mathcal{N}_e^{\rm mod-S-coil} = (N-1)/[\langle L_{\rm pp}^2\rangle/\langle R^2\rangle - 1]$, and
- $\mathcal{N}_e^{\rm mod-S-kink} = N/\langle Z\rangle$.

Differences between these estimators were discussed extensively in Ref. [12].

M-estimators are more accurate than S-estimators because they take into account PPs' non-Gaussian chain statistics, encoded in the variation of the quantities like $\langle R^2\rangle$, $\langle Z\rangle$ and $\langle L_{\rm pp}\rangle$ with N. M-estimators can be directly evaluated using the numbers contained in the summary files (section 2.4.4) if `Z1+` is applied to a homologous series of configurations. The three commonly employed M-estimators [12,85] are

- $\mathcal{N}_e^{\rm M-kink} = dN/d\langle Z\rangle$,
- $\mathcal{N}_e^{\rm M-coil} = z$, where z is the solution to

$$\left.\frac{\langle R^2\rangle}{L^2}\right|_{N=z} = \left.\frac{d}{dN}\left(\frac{\langle L_{\rm pp}\rangle^2}{L\langle \ell_0\rangle}\right)\right|_{N=z}, \tag{6}$$

  with $L = (N-1)\langle\ell_0\rangle$ being the chains' contour length, and
- $\mathcal{N}_e^{\rm SE} = x\ell_K/\langle\ell_0\rangle$, where x (denoted as $N_{\rm eK}$ in the original paper by Svaneborg and Everaers [85]) is the solution to

$$\frac{2x + e^{-2x} - 1}{2x^2} = \left(\frac{\langle L_{\rm pp}\rangle}{L}\right)^2, \tag{7}$$

  where $\ell_K = \lim_{N\to\infty}\langle R^2\rangle/L$ denotes the Kuhn length of the original chain.

Upon adding assumptions about the N-dependency of $\langle R^2\rangle$, $\langle L_{\rm pp}\rangle$ and/or $\langle Z\rangle$, or the value of N, M-estimators turn into S-estimators. A simple example of this is that when N is sufficiently large that $\langle Z\rangle \simeq cN$, $\mathcal{N}_e^{\rm M-kink} \simeq \mathcal{N}_e^{\rm mod-S-kink}$.

A subtler example is: assuming (i) that the statistics of the original chains are captured by the continuous wormlike chain model, $\langle R^2\rangle$ is related to contour length L and ℓ_K as follows:

$$\frac{\langle R^2\rangle}{L^2} = \frac{\ell_K}{L}\left[1 - \frac{\ell_K\left(1 - e^{-2L/\ell_K}\right)}{2L}\right]. \tag{8}$$

Using Eq. (8), the left hand sides of Eqs. (6) and (7) are identical, and one can estimate ℓ_K and thus $\mathcal{N}_e^{\rm SE}$ from a single configuration. For melts composed of chains that are sufficiently long for their statistics to be well described by the continuous wormlike chain model, $\mathcal{N}_e^{\rm SE}$ has been proven [22,85] to be more accurate than S-estimators. Assuming (ii) that N is large enough so that $\langle L_{\rm pp}\rangle$ is well approximated by $\langle L_{\rm pp}\rangle = \alpha N$ with an N-independent coefficient α (see also Fig. 10a), the right hand sides of Eqs. (6) and (7) are identical. Hence, if both assumptions are met, $\mathcal{N}_e^{\rm M-coil}$ and $\mathcal{N}_e^{\rm SE}$ are identical.

While assumption (i) is required to calculate $\mathcal{N}_e^{\rm SE}$ from a single configuration, both assumptions are needed to calculate $\mathcal{N}_e^{\rm M\text{-}coil}$ from a single configuration. While $\mathcal{N}_e^{\rm M\text{-}coil}$ does not rely on either of the two assumptions if implemented as an M-estimator, $\mathcal{N}_e^{\rm SE}$ includes finite N errors due to the failure of assumption (ii) at small N that cause it to predict $x = 0$ for unentangled chains with $\langle L_{\rm pp}\rangle = L$. As shown in Ref. [22], wormlike chain statistics may be replaced by other functional forms more suitable for model polymer melts that yield faster-converging estimates of ℓ_K for use in evaluating $\mathcal{N}_e^{\rm SE}$. Note, however, that since `Z1+` is applicable to all sorts of polymeric systems, it does not rely on assumptions about chain statistics, or sufficiently large N, and therefore does not report M-estimators based on individual configurations.

References

[1] M. Rubinstein, E. Helfand, J. Chem. Phys. 82 (1985) 2477.
[2] S.F. Edwards, Br. Polym. J. 9 (1977) 140.
[3] M. Doi, S.F. Edwards, The Theory of Polymer Dynamics, Clarendon Press, Oxford, United Kingdom, 1988.
[4] R. Everaers, S.K. Sukumaran, G.S. Grest, C. Svaneborg, A. Sivasubramanian, K. Kremer, Science 303 (2004) 823.
[5] M. Kröger, Comput. Phys. Commun. 168 (2005) 209.
[6] S.K. Sukumaran, G.S. Grest, K. Kremer, R. Everaers, J. Polym. Sci. B 43 (2005) 917–933.
[7] C. Tzoumanekas, D.N. Theodorou, Macromolecules 39 (2006) 4592.
[8] K. Foteinopoulou, N.C. Karayiannis, V.G. Mavrantzas, M. Kröger, Macromolecules 39 (2006) 4207–4216.
[9] S. Shanbhag, R. Larson, Macromolecules 39 (2006) 2413–2417.
[10] M. Vladkov, J.-L. Barrat, Macromolecules 40 (2007) 3797–3804.
[11] N. Uchida, G.S. Grest, R. Everaers, J. Chem. Phys. 128 (2008).
[12] R.S. Hoy, K. Foteinopoulou, M. Kröger, Phys. Rev. E 80 (2009) 031803.
[13] C. Tzoumanekas, F. Lahmar, B. Rousseau, D.N. Theodorou, Macromolecules 42 (2009) 7474–7484.
[14] J.-X. Hou, C. Svaneborg, R. Everaers, G.S. Grest, Phys. Rev. Lett. 105 (2010) 068301.
[15] P.S. Stephanou, C. Baig, G. Tsolou, V.G. Mavrantzas, M. Kröger, J. Chem. Phys. 132 (2010).
[16] R. Everaers, Phys. Rev. E 86 (2012) 022801.
[17] J. Chin, S.T. Milner, Macromolecules 47 (2014) 6077.
[18] S.H. Jeong, J.M. Kim, J. Yoon, C. Tzoumanekas, M. Kröger, C. Baig, Soft Matter 12 (2016) 3770–3786.
[19] H.-P. Hsu, K. Kremer, ACS Macro Lett. 7 (2018) 107–111.
[20] S.V. Bobbili, S.T. Milner, Macromolecules 53 (2020) 3861.
[21] R.S. Hoy, M. Kröger, Phys. Rev. Lett. 124 (2020) 147801.
[22] J.D. Dietz, M. Kröger, R.S. Hoy, Macromolecules 55 (2022) 3613.
[23] H. Xiao, C. Luo, D. Yan, J.-U. Sommer, Macromolecules 50 (2017) 9796–9806.
[24] Y. Nie, Z. Gu, Y. Wei, T. Hao, Z. Zhou, Polym. J. 49 (2017) 309–317.
[25] C. Luo, M. Kröger, J.-U. Sommer, Polymer 109 (2017) 71–84.
[26] C. Luo, M. Kröger, J.-U. Sommer, Macromolecules 49 (2016) 9017–9025.
[27] C. Luo, J.-U. Sommer, ACS Macro Lett. 5 (2016) 35–39.

[28] I.-C. Yeh, J.W. Andzelm, G.C. Rutledge, Macromolecules 48 (2015) 4228–4239.
[29] J.M. Kim, R. Locker, G.C. Rutledge, Macromolecules 47 (2014) 2515–2528.
[30] N.C. Karayiannis, K. Foteinopoulou, M. Laso, Philos. Mag. 93 (2013) 4108–4131.
[31] G.N. Toepperwein, J.J. de Pablo, Macromolecules 44 (2011) 5498.
[32] G.N. Toepperwein, N.C. Karayiannis, R.A. Riggleman, M. Kröger, J.J. de Pablo, Macromolecules 44 (2011) 1034.
[33] A.V. Karatrantos, N. Clarke, R.J. Composto, K.I. Winey, Soft Matter 9 (2013) 3877–3884.
[34] P. Bacova, F. Lo Verso, A. Arbe, J. Colmenero, J.A. Pomposo, A.J. Moreno, Macromolecules 50 (2017) 1719–1731.
[35] R.A. Riggleman, G. Toepperwein, G.J. Papakonstantopoulos, J.L. Barrat, J.J. de Pablo, J. Chem. Phys. 130 (2009) 244903.
[36] Y. Li, M. Kröger, W.K. Liu, Macromolecules 45 (2012) 2099–2112.
[37] A.V. Karatrantos, N. Clarke, R.J. Composto, K.I. Winey, Soft Matter 12 (2016) 2567–2574.
[38] A.V. Karatrantos, R.J. Composto, K.I. Winey, N. Clarke, J. Chem. Phys. 146 (2017) 203331.
[39] J.G. Ethier, L.M. Hall, Macromolecules 51 (2018) 9878.
[40] G.G. Vogiatzis, D.N. Theodorou, Arch. Comput. Methods Eng. 25 (2018) 591–645.
[41] A.V. Karatrantos, R.J. Composto, K.I. Winey, N. Clarke, Macromolecules 52 (2019) 2513–2520.
[42] A. Moghimikheirabadi, C. Mugemana, M. Kröger, A.V. Karatrantos, Polymers 12 (2020) 2591.
[43] H. Li, H. Wu, W. Zhang, X. Zhao, L. Zhang, Y. Gao, Polymer 231 (2021).
[44] Y.R. Sliozberg, M. Kröger, T.C. Henry, S. Datta, B.D. Lawrence, A.J. Hall, A. Chattopadhyay, Polymer 217 (2021).
[45] G. Hou, S. Li, J. Liu, Y. Weng, L. Zhang, Phys. Chem. Chem. Phys. 24 (2022) 2813–2825.
[46] D.M. Sussman, W.-S. Tung, K.I. Winey, K.S. Schweizer, R.A. Riggleman, Macromolecules 47 (2014) 6462.
[47] H.-P. Hsu, K. Kremer, J. Chem. Phys. 152 (2020) 154902.
[48] A.L. Bowman, E.P. Chan, W.B. Lawrimore, J.K. Newman, Nano Lett. 21 (2021) 5991–5997.
[49] J.G. Ethier, L.F. Drummy, R.A. Vaia, L.M. Hall, ACS Nano 13 (2019) 12816–12829.
[50] N.A. Garcia, J.-L. Barrat, Macromolecules 51 (2018) 9850–9860.
[51] S.D. Anogiannakis, C. Tzoumanekas, D.N. Theodorou, Macromolecules 45 (2012) 9475–9492.
[52] S. Li, J. Chen, D. Xu, T. Shi, J. Chem. Phys. 143 (2015).
[53] Y.R. Sliozberg, R.A. Mrozek, J.D. Schieber, M. Kröger, J.L. Lenhart, J.W. Andzelm, Polymer 54 (2013) 2555–2564.
[54] Y.R. Sliozberg, R.S. Hoy, R.A. Mrozek, J.L. Lenhart, J.W. Andzelm, Polymer 55 (2014) 2543–2551.
[55] V. Lenzi, M.M.D. Ramos, L.S.A. Marques, Mol. Simul. 47 (2021) 27–36.
[56] Y. Li, M. Kröger, W.K. Liu, Polymer 52 (2011) 5867–5878.
[57] R.S. Hoy, G.S. Grest, Macromolecules 40 (2007) 8389.
[58] J. Kalb, D. Dukes, S.K. Kumar, R.S. Hoy, G.S. Grest, Soft Matter 7 (2011) 1418.
[59] T. Ge, G.S. Grest, M.O. Robbins, ACS Macro Lett. 2 (2013) 882–886.
[60] M.A.G. Cunha, M.O. Robbins, Macromolecules 53 (2020) 8417–8427.
[61] P. Chen, Y. Yang, B. Dong, Z. Huang, G. Zhu, Y. Cao, L.-T. Yan, Macromolecules 50 (2017) 2078–2091.
[62] K. Moorthi, K. Kamio, J. Ramos, D.N. Theodorou, Macromolecules 45 (2012) 8453–8466.
[63] K. Moorthi, K. Kamio, J. Ramos, D.N. Theodorou, in: M. Tokuyama, I. Oppenheim (Eds.), 4th International Symposium on Slow Dynamics in Complex Systems: Keep Going Tohoku, in: AIP Conf. Proc., vol. 1518, World Ctr. Educ. & Res. Trans-Disciplinary Flow Dynam., 2013, pp. 455–458.
[64] A. Moyassari, T. Gkourmpis, M.S. Hedenqvist, U.W. Gedde, Macromolecules 52 (2019) 807–818.
[65] Y.R. Sliozberg, J.W. Andzelm, Chem. Phys. Lett. 523 (2012) 139–143.
[66] A. Ramirez-Hernandez, B.L. Peters, L. Schneider, M. Andreev, J.D. Schieber, M. Mueller, M. Kröger, J.J. de Pablo, Macromolecules 51 (2018) 2110–2124.
[67] G.P. Baeza, J. Polym. Sci. 59 (2021) 2405–2433.
[68] T.L. Chantawansri, T.W. Sirk, R. Mrozek, J.L. Lenhart, M. Kröger, Y.R. Sliozberg, Chem. Phys. Lett. 612 (2014) 157–161.
[69] P. Chen, Y. Yang, B. Dong, Z. Huang, G. Zhu, Y. Cao, L.-T. Yan, Macromolecules 50 (2017) 2078–2091.
[70] M. Nebouy, J. Morthomas, C. Fusco, G.P. Baeza, L. Chazeau, Macromolecules 53 (2020) 3847–3860.
[71] A.J. Parker, J. Rottler, ACS Macro Lett. 6 (2017) 786–790.
[72] A.J. Parker, J. Rottler, Macromolecules 51 (2018) 10021–10027.
[73] V. Sethuraman, D. Kipp, V. Ganesan, Macromolecules 48 (2015) 6321–6328.
[74] M. Laso, N.C. Karayiannis, K. Foteinopoulou, M.L. Mansfield, M. Kröger, Soft Matter 5 (2009).
[75] S.E. Root, S. Savagatrup, A.D. Printz, D. Rodriquez, D.J. Lipomi, Chem. Rev. 117 (2017) 6467–6499.
[76] H. Sun, C.P. Kabb, M.B. Sims, B.S. Sumerlin, Prog. Polym. Sci. 89 (2019) 61–75.
[77] J.D. Dietz, R.S. Hoy, Soft Matter 16 (2020) 6206.
[78] S.K. Sukumaran, G.S. Grest, K. Kremer, R. Everaers, J. Polym. Sci., Part B, Polym. Phys. 43 (2005) 917.
[79] S. Shanbhag, M. Kröger, Macromolecules 40 (2007) 2897–2903.
[80] N.C. Karayiannis, M. Kröger, Int. J. Mol. Sci. 10 (2009) 5054–5089.
[81] K. Kremer, G.S. Grest, J. Chem. Phys. 92 (1990) 5057.
[82] M. Kröger, H. Voigt, Macromol. Theory Simul. 3 (1994) 639–647.
[83] M. Kröger, S. Hess, Phys. Rev. Lett. 85 (2000) 1128–1131.
[84] K. Yashiro, T. Ito, Y. Tomita, Int. J. Mech. Sci. 45 (2003) 1863.
[85] C. Svaneborg, R. Everaers, Macromolecules 53 (2020) 1917.
[86] F. Leonforte, Phys. Rev. E 82 (2010) 041802.
[87] T. Ge, C. Tzoumanekas, S.D. Anogiannakis, R.S. Hoy, M.O. Robbins, Macromolecules 50 (2017) 459.
[88] S. Plimpton, J. Comput. Phys. 117 (1995) 1.
[89] A.P. Thompson, H.M. Aktulga, R. Berger, D.S. Bolintineanu, W.M. Brown, P.S. Crozier, P.J. in't Veld, A. Kohlmeyer, S.G. Moore, T.D. Nguyen, R. Shan, M.J. Stevens, J. Tranchida, C. Trott, S.J. Plimpton, Comput. Phys. Commun. 271 (2022) 108171.
[90] Y.R. Sliozberg, J.W. Andzelm, Polymer 52 (2012) 139.
[91] M. Kröger, Comput. Phys. Commun. 241 (2019) 178–179.
[92] O. Weismantel, A.A. Galata, M. Sadeghi, A. Kröger, M. Kröger, Comput. Phys. Commun. 270 (2022) 108176.
[93] M. Herranz, D. Martinez-Fernandez, P.M. Ramos, K. Foteinopoulou, N.C. Karayiannis, M. Laso, Int. J. Mol. Sci. 22 (2021) 12464.
[94] M. Kröger, Models for Polymeric and Anisotropic Liquids, Monography Within the Series Lecture Notes in Physics, vol. 675, Springer, New York, 2005.
[95] A. Kohlmeyer, LAMMPS Documentation, Sandia Corporation, Albuquerque, New Mexico, United States, 2022, https://www.lammps.org.
[96] Blender Online Community, Blender - a 3D Modelling and Rendering Package, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018, http://www.blender.org.
[97] J.D. Dietz, R.S. Hoy, J. Chem. Phys. 156 (2022) 014103.
[98] Y. Li, M. Kröger, W.K. Liu, Phys. Rev. Lett. 109 (2012) 118001.
[99] A. Moghimikheirabadi, M. Kröger, A.V. Karatrantos, Soft Matter 17 (2021) 6362–6373.
[100] Y. Li, B.C. Abberton, M. Kröger, W.K. Liu, Polymers 5 (2013) 751.
[101] J. Kirk, Z. Wang, P. Ilg, J. Chem. Phys. 150 (2019) 094906.
[102] J. Kirk, M. Kröger, P. Ilg, Macromolecules 51 (2018) 8996–9010.
[103] J. Kirk, P. Ilg, Macromolecules 50 (2017) 3703–3718.
[104] J.J. de Pablo, Annu. Rev. Phys. Chem. 62 (2011) 555–574.
[105] M.A. Galvani Cunha, P.D. Olmsted, M.O. Robbins, J. Rheol. 66 (2022) 619–637.
[106] Y. Ruan, Y. Lu, L. An, Z.-G. Wang, ACS Macro Lett. 10 (2021) 1517–1523.
[107] V.A. Harmandaris, Korea-Austr. Rheol. J. 26 (2014) 15–28.
[108] S. Jeong, S. Cho, J.M. Kim, C. Baig, J. Rheol. 61 (2017) 253–264.
[109] Y.R. Sliozberg, I.-C. Yeh, M. Kröger, K.A. Masser, J.L. Lenhart, J.W. Andzelm, Macromolecules 51 (2018) 9635–9648.
[110] R.J.A. Steenbakkers, C. Tzoumanekas, Y. Li, W.K. Liu, M. Kröger, J.D. Schieber, New J. Phys. 16 (2014) 015027.
[111] D. Nguyen, L. Tao, Y. Li, Front. Chem. 9 (2022) 820417.
[112] R. Miwatani, K.Z. Takahashi, N. Arai, Polymers 12 (2020) 382.
[113] J.T. Padding, W.J. Briels, J. Phys. Condens. Matter 23 (2011) 233101.
[114] Y.-L. Zhu, H. Liu, Z.-Y. Lu, J. Chem. Phys. 136 (2012) 144903.
[115] J. Hendricks, T. Kawakatsu, K. Kawasaki, W. Zimmermann, Phys. Rev. E 51 (1995) 2658–2661.
[116] F. Affouard, M. Kröger, S. Hess, Phys. Rev. E 54 (1996) 5178.
[117] R. Faller, A. Kolb, F. Müller-Plathe, Phys. Chem. Chem. Phys. 1 (1999) 2071.
[118] M. Kröger, Phys. Rep. 390 (2004) 453–551.
[119] Y.-H. Lin, Macromolecules 20 (1987) 3080–3083.
[120] T.A. Kavassalis, J. Noolandi, Phys. Rev. Lett. 59 (1987) 2674.
[121] S.F. Edwards, Proc. Phys. Soc. 92 (1967) 9.
[122] P.G. de Gennes, J. Phys. Lett. 35 (1974) L133.
[123] S.T. Milner, Macromolecules 53 (2020) 1314.
[124] D.C. Morse, Macromolecules 31 (1998) 7030–7043.
[125] R.S. Hoy, M.O. Robbins, Phys. Rev. E 72 (2005) 061802.
[126] L.J. Fetters, D.J. Lohse, D. Richter, T.A. Witten, A. Zirkel, Macromolecules 27 (1994) 4639.
[127] L.J. Fetters, D.J. Lohse, W.W. Graessley, J. Polym. Sci. B 37 (1999) 1023.
[128] L.J. Fetters, D.J. Lohse, S.T. Milner, W.W. Graessley, Macromolecules 32 (1999) 6847.
[129] B. Hinner, M. Tempel, E. Sackmann, K. Kroy, E. Frey, Phys. Rev. Lett. 81 (1998) 2614.
[130] F.G. Schmidt, B. Hinner, E. Sackmann, J.X. Tang, Phys. Rev. E 62 (2000) 5509.
[131] A.M. Fenton, R. Xie, M.P. Aplan, Y. Lee, M.G. Gill, R. Fair, F. Kempe, M. Sommer, C.R. Snyder, E.D. Gomez, R.H. Colby, ACS Central Sci. 8 (2022) 268.
[132] N. Uchida, G.S. Grest, R. Everaers, J. Chem. Phys. 128 (2008) 044902.
[133] K. Foteinopoulou, N.C. Karayiannis, M. Laso, M. Kröger, M.L. Mansfield, Phys. Rev. Lett. 101 (2008) 265702.