Worksheet 3: Properties of Coarse-grained Polymers

Solutions

Johannes Zeman

June 15, 2016

Institute for Computational Physics, University of Stuttgart

Contents

Remarks

The Solutions provided here show possible approaches to solve the tasks from the corresponding worksheet and may not be exhaustive.

1 Introduction

In the first part of this worksheet, you will have to answer a few general questions about coarse-grained polymer models and solve a related mathematical task.

In the remainder of the worksheet, you will get to know our in-house software package ESPResSo (Extensible Simulation Package for Research on Soft matter). Using ESPResSo, you will perform several simulations involving coarse-grained polymers and analyze their properties.

All files required for this tutorial can be downloaded from the lecture's homepage.

2 Short Questions - Short Answers (6 points)

Task (6 points)

Answer the following questions:

- What is the persistence length of a polymer and how is it defined?
- Which real polymers can be described by the worm-like chain model?
- What are the differences between the ideal chain, the worm-like chain, the freely jointed chain and the self-avoiding chain?

Hint

- You might want to study literature to answer these questions. A good reference would be the book *Polymer Physics* by Rubinstein. From an ICP CIP-pool computer, you can find it under `/group/sm/2016/tutorial_03/Rubinstein.pdf`.

Solution

- The persistence length of a polymer chain is a measure for its stiffness. It is the expected value of the distance l along the polymer's contour at which correlations between its tangential vectors $\vec{e}(l)$ vanish:

$$l_p = \int_0^\infty \left(\int \vec{e}(l)\vec{e}(l+\Delta l)dl \right) d\Delta l \tag{1}$$

- The worm-like chain model can be used to describe very stiff polymers such as double-stranded DNA, RNA, or proteins.
- The ideal chain is a synonym for the freely jointed chain, where the bond angles between single monomers are uncorrelated, rendering the model to behave like a 3d random walk with fixed step size.
  In contrast to the ideal chain, the self-avoiding chain incorporates a repulsive potential between monomers, effectively incorporating an excluded volume.
  The worm-like chain model describes a continuosly flexible isotropic rod. It is a

special case of the freely rotating chain model with vanishing length of the Kuhn monomers restricted to small bond angles and constant persistence length.

---

3 Polymer Properties (6 points)

The mean-square radius of gyration of a polymer with N Kuhn monomers of length b is defined as

$$\left\langle R_g^2 \right\rangle \equiv \frac{1}{N} \sum_{i=1}^{N} \left\langle \left(\vec{R}_i - \vec{R}_{\text{com}} \right)^2 \right\rangle , \tag{2}$$

where $\vec{R}_i$ denotes the position of the i-th monomer in the chain, and $\vec{R}_{\text{com}}$ the chain's center of mass.

Task (6 points)

- Starting from equation (2), show that the mean-square radius of gyration of a *worm-like chain* is

$$\left\langle R_g^2 \right\rangle = \frac{1}{3} R_{\max} l_p - l_p^2 + \frac{2 l_p^3}{R_{\max}} - \frac{2 l_p^4}{R_{\max}^2} \left(1 - e^{-\frac{R_{\max}}{l_p}} \right) . \tag{3}$$

  Here, $R_{\max}$ is the maximum end-to-end distance of the polymer and l_p is its persistence length.

Hints

- Read chapter 2 of *Polymer Physics* by Rubinstein. Sections 2.3.1, 2.3.2, 2.4, and 2.4.1 are the most important ones for this task.
- Rewrite equation (2) so that it loses its dependence on $\vec{R}_{\text{com}}$ and depends solely on the distances $\left(\vec{R}_j - \vec{R}_i \right)$.
- Sums may be transformed into integrals.
- Think about how equation (2.35) in section 2.3.2 is connected to this problem.

Solution

- First, we rewrite equation (2) according to chapter 2.4 in *Polymer Physics* by Rubinstein:

$$\begin{aligned}
R_g^2 &\equiv \frac{1}{N}\sum_{i=1}^{N}\left(\vec{R}_i - \vec{R}_{\text{com}}\right)^2 \\
&= \frac{1}{N}\sum_{i=1}^{N}\left(\vec{R}_i^2 - 2\vec{R}_i\vec{R}_{\text{com}} + \vec{R}_{\text{com}}^2\right) \\
&= \frac{1}{N}\sum_{i=1}^{N}\left[\vec{R}_i^2\frac{1}{N}\sum_{j=1}^{N}1 - 2\vec{R}_i\frac{1}{N}\sum_{j=1}^{N}\vec{R}_j + \left(\frac{1}{N}\sum_{j=1}^{N}\vec{R}_j\right)^2\right] \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\vec{R}_i^2 - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}2\vec{R}_i\vec{R}_j + \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{N}\sum_{j=1}^{N}\vec{R}_j\right)^2 \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\vec{R}_i^2 - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}2\vec{R}_i\vec{R}_j + \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\vec{R}_i\vec{R}_j \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\vec{R}_i^2 - 2\vec{R}_i\vec{R}_j + \vec{R}_i\vec{R}_j\right) \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\vec{R}_i^2 - \vec{R}_i\vec{R}_j\right)
\end{aligned}$$

  This can be written in a symmetric form:

$$\begin{aligned}
R_g^2 &= \frac{1}{2}\left[\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\vec{R}_i^2 - \vec{R}_i\vec{R}_j\right) + \frac{1}{N^2}\sum_{j=1}^{N}\sum_{i=1}^{N}\left(\vec{R}_j^2 - \vec{R}_j\vec{R}_i\right)\right] \\
&= \frac{1}{2N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\vec{R}_i^2 - 2\vec{R}_i\vec{R}_j + \vec{R}_j^2\right) \\
&= \frac{1}{2N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\vec{R}_i - \vec{R}_j\right)^2 \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=i}^{N}\left(\vec{R}_i - \vec{R}_j\right)^2
\end{aligned}$$

  The average value is thus

$$\left\langle R_g^2\right\rangle = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=i}^{N}\left\langle\left(\vec{R}_i - \vec{R}_j\right)^2\right\rangle . \tag{4}$$

- We can now transform the sums into integrals by switching from the discrete summation indices i, j to continuous coordinates u, v. The summation indices go from $i = 1$ to N ($j = i$ to N) over monomers with Kuhn length b. The continuous coordinates must thus go from $u = 0$ to Nb ($v = u$ to Nb). Compared to a sum, we accumulate a factor of b per integral, and we have to correct for this using a prefactor of $\frac{1}{b^2}$:

$$\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=i}^{N}\left\langle\left(\vec{R}_i-\vec{R}_j\right)^2\right\rangle \to \frac{1}{N^2b^2}\int_0^{Nb}\int_u^{Nb}\left\langle\left(\vec{R}(u)-\vec{R}(v)\right)^2\right\rangle dv\,du$$

  The maximum end-to-end distance of a worm-like chain is that of a rod, which is obviously $R_{\max} = Nb$. Substituting this into the previous expression yields

$$\left\langle R_g^2\right\rangle = \frac{1}{R_{\max}^2}\int_0^{R_{\max}}\int_u^{R_{\max}}\left\langle\left(\vec{R}(u)-\vec{R}(v)\right)^2\right\rangle dv\,du\,. \tag{5}$$

- The integrant in equation (5) can be interpreted as the mean end-to-end distance of a worm-like chain with contour length $|u-v|$. According to the derivation given in section 2.3.2 of the abovementioned book, the mean end-to-end distance of a worm-like chain is given as

$$\left\langle R^2\right\rangle = 2l_pR_{\max} - 2l_p^2\left(1-e^{-\frac{R_{\max}}{l_p}}\right). \tag{6}$$

  In the case of a worm-like chain with contour length $|u-v|$, the maximum posibble end-to-end distance is $R_{\max} = |u-v| \overset{v\geq u}{=} v-u$. Inserting equation (6) with this condition into equation (5) yields

$$\left\langle R_g^2\right\rangle = \frac{1}{R_{\max}^2}\int_0^{R_{\max}}\int_u^{R_{\max}}\left[2l_p(v-u) - 2l_p^2\left(1-e^{-\frac{v-u}{l_p}}\right)\right]dv\,du\,. \tag{7}$$

  Now we can proceed solving the integrals:

$$\begin{aligned}
\left\langle R_g^2\right\rangle &= \frac{1}{R_{\max}^2}\int_0^{R_{\max}}\left[2l_p\int_u^{R_{\max}}(v-u)\,dv \;-2l_p^2\int_u^{R_{\max}}dv \;+2l_p^2\int_u^{R_{\max}}e^{-\frac{v-u}{l_p}}\,dv\right]du\\
&= \frac{1}{R_{\max}^2}\left[\left(2l_p^3-2l_p^2R_{\max}+l_pR_{\max}^2\right)\int_0^{R_{\max}}du \;+\left(2l_p^2-2l_pR_{\max}\right)\int_0^{R_{\max}}u\,du\right.\\
&\quad \left.+l_p\int_0^{R_{\max}}u^2\,du \;+2l_p^3\int_0^{R_{\max}}e^{\frac{u-R_{\max}}{l_p}}\,du\right]\\
&= \frac{1}{3}R_{\max}l_p - l_p^2 + \frac{2l_p^3}{R_{\max}} - \frac{2l_p^4}{R_{\max}^2}\left(1-e^{-\frac{R_{\max}}{l_p}}\right) \quad \blacksquare
\end{aligned} \tag{8}$$

4 Static Properties of Coarse-grained Polymers with ESPResSo (8 points)

4.1 The Software Packege ESPResSo

The software package ESPResSo is developed and maintained at the Institute for Computational Physics and is mainly intended to perform coarse-grained simulations with Lattice-Boltzmann (LB), Dissipative Particle Dynamics (DPD) and Langevin Dynamics (LD). It offers a broad variety of electrostatic algorithms, analysis tools and various other features such as the support of massively parallelized hardware architectures or GPU platforms.

- The package can be obtained free of charge under `http://espressomd.org/wordpress/download/`.
- Be advised to also have a look at the ESPResSo manual to understand how it operates: `http://espressomd.org/jenkins/job/master-doc/lastSuccessfulBuild/artifact/doc/ug/ug.pdf`

In the following, you will conduct coarse-grained simulations of polymers with LD to learn how to work with ESPResSo. The simulations focus on the ideal chain model and the chain with excluded volume interactions. You can either use the computers in the ICP CIP pool or install ESPResSo on your own computer.

4.2 Installing ESPResSo

Download and unpack the ESPResSo package version 3.2.0 (`espresso-3.2.0.tar.gz`). Follow the build procedure as given in the manual on pp. 27ff.

The `configure` script should be run with the option `--without-cuda` in order to avoid problems during compilation. After using `./configure` but *before* compiling with `make`, please uncomment the macros in `myconfig-sample.h` for `LENNARD_JONES` and rename it to `myconfig.h`.

4.3 Setting up and Running the Simulations

Download the template Tcl script `template.tcl` from the lecture website.

Examine the template script and also have a look at the manual (and perhaps on the test cases, too) to understand how to set up a polymer with Langevin Dynamics. You need harmonic springs with the spring constant $k = 10$ to connect the monomers. The temperature should be set to $T = 1$ and the friction coefficient of the Langevin thermostat to $\gamma = 1$.

Once the Tcl script is prepared, you can run the simulation with

```
$> /<install_dir>/Espresso template.tcl
```

4.4 Ideal Chain

Task (4 points)

- Perform simulations of an ideal coarse-grained polymer with Langevin Dynamics for different chain lengths $N \in \{10, 20, 30, 40, 50, 100, 200\}$ and determine the average radii of gyration $R_g(N)$.
- Determine the parameter ν in the relation $R_g(N) \propto N^{\nu}$.

Solution

N	10	20	30	40	50	100	200
$\langle R_g \rangle$	1.308	1.857	2.272	2.598	2.913	4.248	5.627

Table 1: Radii of gyration $\langle R_g \rangle$ of an ideal (freely jointed) chain for different chain lengths N simulated with ESPResSo 3.2.0.

- The average radii of gyration are listed in table 1.
- Fitting a function $f(N) = aN^{\nu}$ to the simulation data yields an exponent $\nu \approx 0.478$, which is close to the theoretical prediction of $\nu = 0.5$:

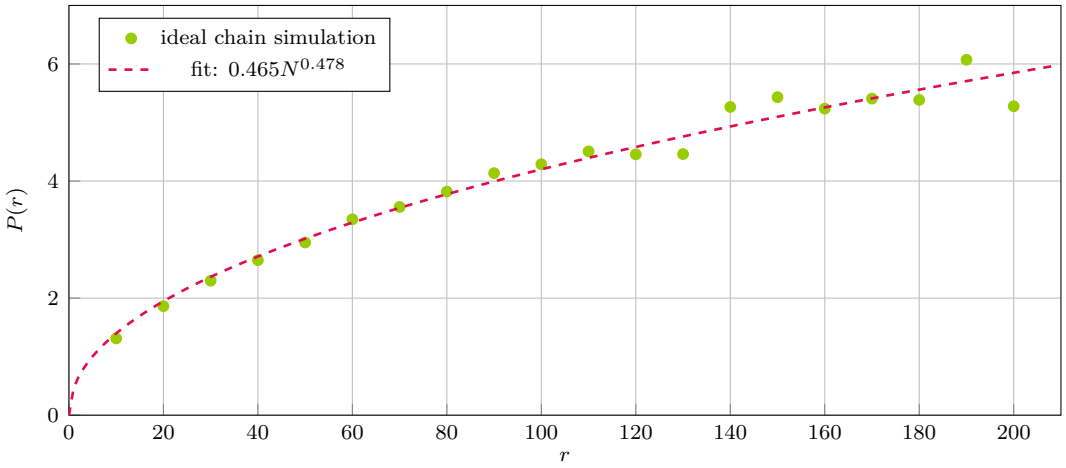

Figure 1: Radii of gyration $\langle R_g \rangle$ of an ideal (freely jointed) chain for different chain lengths N simulated with ESPResSo 3.2.0. Dots (●): Simulation data. Line (- - -): fit.

4.5 Chain with Excluded Volume Interactions

Task (4 points)

- Simulate a coarse-grained polymer with the same interactions and parameters as given above. In addition, apply Lennard-Jones interactions to the monomers with $\epsilon = 1$, $\sigma = 1$, and cutoff radius $r_c = 2^{\frac{1}{6}}$. Shift the Lennard-Jones function such that the force is zero for $r = r_c$.
- Repeat the simulations for the different values of N .
- Determine ν as in the previous task.

Solution

$\boldsymbol{N}$	10	20	30	40	50	100	200
$\langle \boldsymbol{R_g} \rangle$	1.798	2.812	3.636	4.526	5.081	8.284	11.672

Table 2: Radii of gyration $\langle R_g \rangle$ of a self-avoiding chain for different chain lengths N simulated with ESPResSo 3.2.0.

- The average radii of gyration are listed in table 2.
- Fitting a function $f(N) = aN^{\nu}$ to the simulation data yields an exponent $\nu \approx 0.646$, which is – as expected – greater than 0.5:

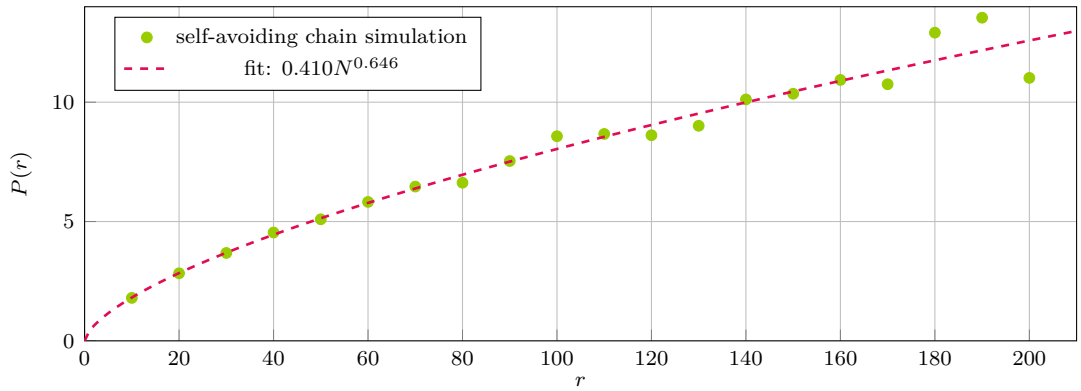

Figure 2: Radii of gyration $\langle R_g \rangle$ of a self-avoiding chain for different chain lengths N simulated with ESPResSo 3.2.0. Dots (●): Simulation data. Line (- - -): fit.
