universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

Hybrid hydrodynamic simulations of ring polymers

verfasst von / submitted by

Lisa Sappl, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2022 / Vienna, 2022

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 876
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Physik UG2002
Betreut von / Supervisor:	Univ.-Prof. Dipl.-Ing. Dr. Christos N. Likos

Abstract

The simulation of polymer solutions requires the development of methods that accurately include hydrodynamic interactions. Resolution on the atomistic scale is too computationally expensive to cover mesoscopic time and length scales on which the interesting polymer phenomena are observed. Therefore, coarse-graining methods have to be applied. In this work the solvent is simulated using Multi-Particle Collision Dynamics (MPCD) which is a well-established model, and for the polymer different coarse-graining methods are employed and compared against the monomer resolved Kremer–Grest model by their resulting diffusion coefficients. This thesis builds on the work by Ruiz-Franco et al. (2019) in which they simulated star polymers and linear chains in a solvent and developed two different coarse-graining methods in order to increase computational efficiency. This work extends their research to ring polymers and seeks to refine one of the authors' proposed model: the Penetrable Soft Colloid (PSC) model. It was found that both proposed models were not well suited to ring polymers, however the introduction of a factor to the PSC model delivered satisfying results for the diffusion behaviour by regulating the interaction intensity with the solvent.

Zusammenfassung

Die Simulation von Polymerlösungen erfordert die Entwicklung von Methoden, die die hydrodynamischen Interaktionen realitätsgetreu abbilden. Dabei übersteigt eine Auflösung in der atomistischen Größenordnung die Möglichkeiten heutiger Computer um die mesoskopischen Zeit- und Längsskalen abzudecken. Diese sind jedoch jene Größenordnungen an denen sich die zu beobachtenden Phänomene für Polymere abspielen. Aus diesem Grund ist es erforderlich grobkörnige Modelle anzuwenden, die die atomistischen Vorgänge vereinfacht darstellen. Eines dieser Modelle, welches auch in dieser Arbeit zur Modellierung des Lösungsmittels angewandt wird, ist die Multi-Particle Collision Dynamics (MPCD). Zur Simulation des Polymers werden in dieser Arbeit verschiedenste grobkörnige Modelle getestet und mit dem etablierten Kremer–Grest-Modell, welches das Polymer in einzelnen Monomeren darstellt, anhand ihres Diffusionsverhaltens verglichen. Dabei baut diese Arbeit auf den Forschungsergebnissen von Ruiz-Franco et al. (2019) auf, in welchen die Autoren bereits zwei grobkörnige Modelle entwickelten, und diese für lineare Kettenpolymere und Sternpolymere in Lösung testeten. In dieser Arbeit wird die Forschung auf Ringpolymere ausgeweitet, und außerdem eines der von den Autoren entwickelten Modelle versucht zu verfeinern: das Penetrable Soft Colloid (PSC) Modell. Es stellte sich heraus, dass die beiden ursprünglich vorgestellten Modelle sich zur Anwendung auf Ringpolymere nicht sehr gut eignen, jedoch die Einführung eines Faktors als Ergänzung zum PSC-Modell zufriedenstellende Ergebnisse für die Diffusion zeigte, indem sich damit die Intensität der Interaktion zwischen Polymer und Lösungsmittel regulieren ließ.

Acknowledgements

First and foremost I want to express enormous gratitude to my supervisor of this work: Univ.-Prof. Dipl.-Ing. Dr. Christos N. Likos. He provided me with the opportunity to work on a fascinating project and contribute to a topic that interests me personally. Not only was he always helpful with his professional opinions and scientific insight, but he has also supported and encouraged me during the whole process. Without him this work would not have been possible. I also want to give a special thanks to his team, and especially our colleague Dr. Andreas Zöttl who contributed his original ideas to this work.

Next I want to thank my family, my friends, and my partner Daniel for supporting me and giving me their valuable time to help me mentally overcome this challenging time. It means a lot that they believed in me and helped me not to give up. I am truly blessed to have so many kind companions who provide me with so much strength, wisdom and love.

Lastly I am grateful to the Vienna Scientific Cluster (VSC) for providing me with the computational resources that were necessary to perform most of my simulations, and I am also thankful to the University of Vienna for giving me the opportunity to study Physics and gifting me a beautiful memory of all the years to which I will always happily look back.

Contents

1 Introduction to polymer solutions **5**

2 The MPCD-Solvent **8**

2.1 The MPCD algorithm 9

2.2 Statistics of the solvent 12

2.3 Solvent diffusion 13

3 The monomer resolved model (MRM) **18**

3.1 The MRM algorithm 18

3.2 Spatial properties of the MRM ring polymer 23

3.3 MRM ring polymer diffusion 28

4 The effective monomer model (EMM) **32**

4.1 The EMM algorithm 32

4.2 EMM ring polymer diffusion 34

5 The penetrable soft colloid model (PSC) **36**

5.1 The PSC algorithm 36

5.2 Solvent density distribution 41

5.3 Penetrable soft colloid diffusion 43

6 The ellipsoidal PSC **44**

6.1 The ellipsoidal PSC algorithm 44

6.2 Ellipsoidal PSC diffusion 51

7 The spherical PSC with increased interaction factor **53**

7.1 The algorithm of the PSC with factor 53

7.2 Solvent density distribution of the PSC with factor 59

7.3 Diffusion of the PSC with factor 60

8 Conclusions and Outlook **65**

Chapter 1

Introduction to polymer solutions

Polymers are large macromolecules consisting of repeating monomers that are connected through covalent bonds [1]. The chemical identity of the monomers is one of the main distinctions between polymers. Another major classification is the polymer architecture, that is the configuration in which the monomers are connected to one another. The simplest architecture are linear chains, but there are also rings, star polymers and other more complex types of architectures like cross-linked networks (fig. 1.1). The type of architecture majorly influences a polymer's properties.

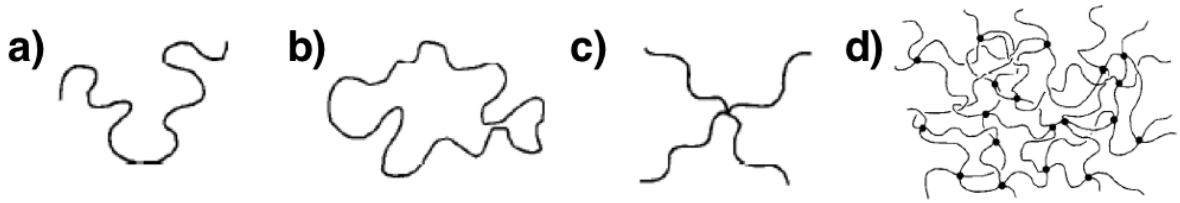


Figure 1.1: Different types of polymer architectures among many: a) linear chain, b) ring, c) star, d) polymer network. [1]

A polymer's architecture is fixed and can not be changed without breaking the covalent chemical bonds. Depending on their flexibility, the monomers' interaction among each other, or the environment however, polymers can freely change their spatial structure by rotating around their bonds and bending them. The different spatial shapes that a polymer can take are called conformations.

Polymers in solution experience Brownian motion which leads to incessant changes in their conformation [2]. One phenomenon that is dominated by this effect is diffusion. A phenomenological description of diffusion is offered by Fick's law, which says that for an inhomogeneous concentration $c(\vec{x},t)$, there is a flux $\vec{j}(\vec{x},t)$ that is proportional to the concentration gradient

$$\vec{j}(\vec{x},t) = -D \cdot \nabla\ c(\vec{x},t), \tag{1.1}$$

with a constant D that is called the diffusion constant or diffusion coefficient. Together with the continuity equation

$$\frac{\partial c(\vec{x},t)}{\partial t} = -\nabla \cdot \vec{j}(\vec{x},t), \tag{1.2}$$

these two equations result in Fick's second law of diffusion

$$\frac{\partial c(\vec{x},t)}{\partial t} = D \cdot \Delta\ c(\vec{x},t). \tag{1.3}$$

The diffusion coefficient therefore describes how quickly one substance can diffuse through another. It depends on the properties of both the solute and the solvent.

When a particle in a liquid experiences a constant force $\vec{F}$, it will be pulled through the liquid, reaching a constant velocity $\vec{V}$ [1]. This frictional force and the velocity are related to each other by the frictional coefficient ζ, such that

$$\vec{F} = \zeta \vec{V}. \tag{1.4}$$

The diffusion coefficient is directly related to the frictional coefficient via

$$D = \frac{k_B T}{\zeta}. \tag{1.5}$$

The first model to successfully describe polymer dynamics was the Rouse model. It describes monomers as beads that are connected to each other via springs. As a result, the total friction coefficient of the polymer ζ_R is the sum of all monomer friction coefficients ζ, and the Rouse diffusion D_R predicts

$$D_R = \frac{k_B T}{\zeta_R} = \frac{k_B T}{N\zeta}. \tag{1.6}$$

The Rouse model assumes that the monomer beads only interact with one another through the springs that connect them. This is a reasonable assumption when dealing with polymer melts. In dilute solutions on the other hand, the solvent plays an important role as a medium that transfers momentum between solutes. In polymer solutions momentum is transferred from a monomer to surrounding solvent particles from which it can be transferred again to another monomer [3]. This interaction is called hydrodynamic interaction. It only affects dynamic properties such as diffusion, but has no effect on the static properties of a solute. The influence that hydrodynamic interaction has on dynamic properties is strong such that this phenomenon can not be neglected usually. The model used to describe the dynamics of polymer solutions is the Zimm model, which states that the diffusing particle drags some of the surrounding solvent with it due to hydrodynamic interaction [1]. The Zimm diffusion D_Z predicts

$$D_Z \approx \frac{k_B T}{\eta_{sol} b N^{\nu}}, \tag{1.7}$$

using the solvent's dynamic viscosity η_{sol}, average monomer distance b and Flory exponent ν. The deviation of the Zimm diffusion from the Rouse diffusion shows that hydrodynamic interaction has a significant, nontrivial effect on polymer diffusion. Hydrodynamic interaction over distance decays only in the order of r^{-1} which classifies it as long ranged. Within a polymer, all monomers affect all other monomers and in more concentrated solutions there exist not only intramolecular but also intermolecular hydrodynamic interactions [4].

Diffusion plays a substantial role in biology for the proper functioning of cells [5]. It is because of diffusive motion, that molecules are able to find one another in a cell, as for example membrane proteins need to find the membrane. It has been shown that a protein's diffusion coefficient changes depending on whether the protein is disordered or folded. Understanding the dynamical behaviour of macromolecules is therefore a crucial part of understanding life. Ring polymers, such as circular DNA which is found in bacteria and viruses, play an important part in nature as well as in biotechnological applications [6]. However, compared to linear chain polymers their topological constraints decrease their attainable phase space dramatically. As a result, their dynamic and static properties differ significantly from linear chain polymers and their non-trivial behaviour is of great interest.

The interesting phenomena that complex fluids, such as polymer solutions, exhibit exist on mesoscopic time and length scales much larger than the typical atomic scale, but also much smaller than the visible scale [7]. If such complex fluids are modeled using computer simulation, it can be quite challenging to bridge the difference in time and length scales between the detailed molecular interactions and the behaviour that the system exhibits. To quantify the dynamic behaviour of a complex fluid it is necessary to develop suitable coarse-grained models that manage to retain the correct overall physical behaviour of smaller scales, while being efficient enough to cover the mesoscopic scales. Using full molecular dynamics is computationally too expensive for mesoscopic time and length scales. This work therefore uses an established model that offers a coarse-grained description of the solvent: The Multi-Particle Collision Dynamics (MPCD). However, when dealing with dense suspensions of polymers or polymers with complex architecture such as dendrimers or micellae, a coarse-grained description of the solvent alone is often not sufficient. For this reason, this work tests different coarse-grained models for their applicability to ring polymers against the well-established monomer resolved model (MRM). The focus is laid on the long-time dynamics of the center of mass and the diffusion coefficient. One of these models could hopefully help provide additional understanding to the rheology and dynamics of dense suspensions and large ring polymers in the future.

The thesis is organized as follows. In Chapter 2, the MPCD method is described and applied to a generic solvent. Chapter 3 introduces the monomer resolved model (MRM) for ring polymers and analyzes the polymer's static and dynamic properties such as radius of gyration and diffusion within this model. In Chapter 4 and 5 we apply the effective monomer model (EMM) and the penetrable soft colloid model (PSC) to ring polymers and compare the diffusion properties with the monomer resolved model. The PSC is extended to a general ellipsoidal shape in chapter 6, and in chapter 7 it is modified to regulate interaction with the solvent via a factor. All models are compared in terms of diffusive behaviour. Chapter 8 summarises our findings.

Chapter 2

The MPCD-Solvent

In a polymer solution the solvent greatly influences a polymer's structure and dynamics [8]. For any specific polymer there are good solvents which cause the polymer to expand, caused by the stronger interaction between monomers and solvent particles as compared to the interaction among monomers themselves, and there are bad solvents in which the opposite is true and the polymer exists in a compact conformation. As a result polymers' properties change significantly depending on the solvent in which they are embedded.

In a simulation of a polymer solution it is often impractical to employ molecular dynamics (MD) on both solvent particles and monomers. Especially for large polymers such simulations can become very time-consuming, and therefore limiting to system size and time scales, which is why several mesoscopic models that simplify solvent dynamics have been proposed. One of these mesoscopic models is the Multi-Particle Collision Dynamics (MPCD) simulation method which was first introduced by Malevanets and Kapral (1999) [9] and has since been widely used for the numerical simulation of complex fluids. Successful applications of an MPCD-solvent algorithm include simulations of polymers [10][8], swarm behaviour of active particles [11][12] and biological cells [13]. The original MPCD-method that uses a random rotation of the solvent relative velocities, is sometimes specifically referred to as "stochastic rotation dynamics" (SRD), whereas "MPCD" is occasionally used to refer to algorithms with different collision rules too.

MPCD is a particle-based algorithm which makes it easy to incorporate collisional coupling to other microscopic degrees of freedom, and straightforwardly applicable to complex system geometries [9]. Further, it satisfies the local conservation of mass, linear momentum and energy, which means that hydrodynamic interactions are respected. Moreover, it captures thermal fluctuations which play an important role in many applications. Malevanets and Kapral (1999) were also able to show that the MPCD-model has an H-theorem, that the equilibrium distribution of velocities is a Maxwell-Boltzmann distribution, and that it yields the correct hydrodynamic equations with an ideal-gas equation of state.

Naturally the model also has its limitations. For one, long-range hydrodynamic interactions are not sufficiently represented by local algorithms such as MPCD if Reynolds numbers are low [14]. This can be attributed to the compressibility and the low speed of sound of the solvent, which leads to the inaccuracy that the propagation of information takes too much time compared to the time scale of measurements over longer distances. This poses limitations to the applicability of MPCD for systems with low Reynolds numbers. Also, the standard MPCD algorithm does not conserve

angular momentum, which can lead to incorrect results when dealing with boundary conditions on walls that are given by forces, fluids with different viscosities in contact with each other, or finite-sized objects rotating in fluids [15]. For that reason, many have developed different variants of the original MPCD-method, such as a non-ideal equation of state [16], angular momentum conservation [17], or an isotropic collision rule [18].

2.1 The MPCD algorithm

The standard MPCD algorithm that was used for this work is described in detail by several sources. This section relies on the works of Gompper et al. (2008) [19] and Winkler (2012) [20] if not stated otherwise.

2.1.1 Initialisation of the solvent

Since the MPCD algorithm is a particle based approach, a number n of point-like solvent particles each with mass m are initialised. In most cases, it makes sense to initialise positions $\vec{r}_i(0)$ and velocities $\vec{v}_i(0)$ of the solvent particles $i = 1, ..., n$ in an equilibrium state, that is a uniform distribution of the positions in space, and a Maxwell-Boltzmann distribution of the velocities w.r.t. a temperature (precisely - a thermal energy) $k_B T$ of choice. The velocities are chosen such that the center of mass velocity for the box is 0. For spatial boundaries a cube of side length l was chosen and periodic boundary conditions were applied.

2.1.2 Dynamics of the solvent particles

The dynamics consist of 2 alternating steps: Streaming and Collision. During the streaming step the particles move ballistically for the duration of one time step h:

$$\vec{r}_i(t+h) = \vec{r}_i(t) + h \cdot \vec{v}_i(t). \tag{2.1}$$

For the collision step the simulation box is divided into cubic collision cells of side length a. Particles that share the same collision cell then collide with each other. The collision exchange of momentum is realised by rotating the particles' relative velocity vectors $\vec{w}_i$. The relative velocities are defined using the center of mass velocity of the collision cell $\vec{v}_{cm}$ via

$$\vec{w}_i := \vec{v}_i - \vec{v}_{cm}, \tag{2.2}$$

$$\vec{v}_{cm} := \frac{1}{n_c} \sum_{i=1}^{n_c} \vec{v}_i, \tag{2.3}$$

where n_c is the number of solvent particles in the respective cell. The velocities after collision can then be calculated using a standard universal rotation matrix $\hat{D}(\vec{\mathcal{R}}, \alpha)$

$$\vec{v}_i(t+h) = \vec{v}_{cm}(t) + \hat{D}(\vec{\mathcal{R}}, \alpha) \cdot (\vec{v}_i(t) - \vec{v}_{cm}(t)) \,, \tag{2.4}$$

$$\vec{w}_i(t+h) = \hat{D}(\vec{\mathcal{R}}, \alpha) \cdot \vec{w}_i(t). \tag{2.5}$$

As a result, the center of mass velocity $\vec{v}_{cm}$ of each collision cell remains the same before and after the collision. Only the relative velocities $\vec{w}_i$ are rotated and linear momentum is conserved locally. The angle of rotation α remains fixed during the entire simulation, as it is directly related to the liquid's viscosity (see section 2.3.3) and is as such an intrinsic property of it. The unit rotation vector $\vec{\mathcal{R}}$, i.e. the rotation axis, on the other hand is oriented randomly for each collision cell and time step. As a result, the rotation matrix $\hat{D}(\vec{\mathcal{R}}, \alpha)$ is calculated as follows:

$$\hat{D}(\vec{\mathcal{R}}, \alpha) = \begin{pmatrix} \mathcal{R}_x^2 + (1-\mathcal{R}_x^2)c & \mathcal{R}_x\mathcal{R}_y(1-c) - \mathcal{R}_z s & \mathcal{R}_x\mathcal{R}_z(1-c) + \mathcal{R}_y s \\ \mathcal{R}_x\mathcal{R}_y(1-c) + \mathcal{R}_z s & \mathcal{R}_y^2 + (1-\mathcal{R}_y^2)c & \mathcal{R}_y\mathcal{R}_z(1-c) - \mathcal{R}_x s \\ \mathcal{R}_x\mathcal{R}_z(1-c) - \mathcal{R}_y s & \mathcal{R}_y\mathcal{R}_z(1-c) + \mathcal{R}_x s & \mathcal{R}_z^2 + (1-\mathcal{R}_z^2)c \end{pmatrix}, \tag{2.6}$$

$s = \sin\alpha,\ c = \cos\alpha,$ and

$$\vec{\mathcal{R}} = \begin{pmatrix} \sin\vartheta\cos\varphi \\ \sin\vartheta\sin\varphi \\ \cos\vartheta \end{pmatrix}, \quad \varphi \sim U(0, 2\pi),\ \cos\vartheta \sim U(-1, 1). \tag{2.7}$$

While the collision cell's side length a remains fixed during the entire simulation (it is an intrinsic property of the liquid since it is directly related to the viscosity, see eq. 2.29), their center points have to be shifted by a uniformly picked random vector with components $\sim U(-a/2, a/2)$ before each collision step. This is done as to not violate Galilean invariance. Especially if the mean free path length $\lambda = h\sqrt{k_B T/m}$ is smaller than the collision cell length a, not implementing this random grid shift would repeatedly group together the same particles for collision. As a result particles that share a collision cell would remain correlated over several time steps leading to unrealistic results for the hydrodynamic equations and transport coefficients. In practise the random grid shift is usually done by temporarily shifting all particles' positions by this random vector, and shifting them back after the collision.

2.1.3 Cell-level canonical thermostating

If the simulation is desired to yield a canonical ensemble, some sort of thermostating is necessary. This is especially true for non-equilibrium conditions. A basic requirement for any thermostat is that it does not violate local momentum conservation, smear out local flow profiles, or distort the velocity distribution too much. One thermostat that meets these requirements and works well with the MPCD algorithm is the cell-level canonical thermostat. It starts by drawing a

random kinetic energy for each collision cell and time step from its probability density function in a canonical ensemble $p(E_{kin}|n_c)$ at a given n_c:

$$p(E_{kin}|n_c) = \frac{1}{E_{kin}\Gamma(f/2)} \left(\frac{E_{kin}}{k_B T}\right)^{f/2} \exp\left(-\frac{E_{kin}}{k_B T}\right), \quad f = 3(n_c - 1). \tag{2.8}$$

This distribution function is a gamma distribution. f denotes the relative velocities' $\{\vec{w}_i\}$ degrees of freedom of the considered system and $\Gamma(x)$ is the gamma function. The next step is to calculate a scaling factor κ using the kinetic energy E_{kin} obtained for each cell. At the end, the relative velocities are scaled by κ:

$$\kappa = \sqrt{\frac{2E_{kin}}{\sum_{i=1}^{n_c} m\vec{w}_i^2}}, \tag{2.9}$$

$$\vec{w}_i' = \kappa \cdot \vec{w}_i \ . \tag{2.10}$$

This thermostating method leaves the center of mass velocity of the respective collision cell $\vec{v}_{cm}$ unchanged. Additionally, scaling and rotating the relative velocities are commutative, which makes it irrelevant whether thermostating is implemented before or after the collision step:

$$\kappa \cdot \left(\hat{D}(\vec{\mathcal{R}}, \alpha) \cdot \vec{w}_i'\right) = \hat{D}(\vec{\mathcal{R}}, \alpha) \cdot (\kappa \cdot \vec{w}_i)\,. \tag{2.11}$$

2.1.4 Choice of MPCD parameters

The dimensionless parameters of MPCD are generally set as: $m = 1$, $k_B T = 1$, $a = 1$. The rotation angle $\alpha \in (\pi/2, 3\pi/4)$, and the average particle number per collision box $\langle n_c \rangle \in (3, 20)$ can be varied within a sensible range [21]. All numbers are given in reduced units. The reference units for length, mass, energy and time are a, m, $k_B T$, and $\tau = \sqrt{\frac{ma^2}{k_B T}}$ respectively.

There is a certain freedom of choice for the parameters, however in order to yield significant hydrodynamic interaction in the simulation, the mean free path length $\lambda = h\sqrt{k_B T/m}$ should be small enough, recommendedly around $\lambda = 0.1a$ [10]. As a result the time step size is set as $h = 0.1\tau$.

Table 2.1 depicted below shows the numerical values that were chosen for the various parameters in all subsequent simulations if not explicitly stated otherwise.

$n = 10^6$	number of solvent particles in the simulation box
$m = 1$	solvent particle mass
$k_B T = 1$	temperature (thermal energy) of the thermostated simulation
$l = 46$	side length of simulation box (cube), also "system size"
$h = 0.1$	MPCD time step
$a = 1$	side length of collision cell (cube)
$\langle n_c \rangle = n \cdot (a/l)^3 = 10.3$	average number of solvent particles in one collision cell
$\alpha = 3\pi/5$	rotation angle

Table 2.1: Summary of MPCD parameters

2.2 Statistics of the solvent

In equilibrium the solvent velocities $\vec{v}_i$ are Maxwell-Boltzmann distributed in the limit $n \to \infty$. Considering only one dimension that is a Gaussian distribution with the following parameters:

$$p(v_{i,x}) = \left(\frac{m}{2\pi k_B T}\right)^{1/2} \exp\left(-\frac{m}{2k_B T} v_{i,x}^2\right) , \quad \mu = \langle v_{i,x} \rangle = 0 , \quad \sigma^2 = \left\langle v_{i,x}^2 \right\rangle = \frac{k_B T}{m}. \quad (2.12)$$

The relative velocities $w_{i,x} = v_{i,x} - v_{cm,x}$ therefore also follow a Gaussian distribution, but with different parameters

$$\mu = \langle w_{i,x} \rangle = \langle v_{i,x} - v_{cm,x} \rangle = \langle v_{i,x} \rangle - \langle v_{cm,x} \rangle = 0 ,$$

$$\begin{aligned}
\sigma^2 &= \left\langle w_{i,x}^2 \right\rangle \\
&= \left\langle (v_{i,x} - v_{cm,x})^2 \right\rangle \\
&= \left\langle v_{i,x}^2 \right\rangle - \left\langle 2 v_{i,x} \cdot \frac{1}{n_c} \sum_j^{n_c} v_{j,x} \right\rangle + \left\langle v_{cm,x}^2 \right\rangle \\
&= \frac{k_B T}{m} - \frac{2}{n_c} \left(\left\langle v_{i,x}^2 \right\rangle + \langle v_{i,x} \rangle \cdot \sum_{j \neq i}^{n_c} \langle v_{j,x} \rangle \right) + \frac{k_B T}{n_c m} \\
&= \frac{k_B T}{m} - \frac{2}{n_c} \left(\frac{k_B T}{m} + 0 \right) + \frac{k_B T}{n_c m} \\
&= \frac{k_B T}{m} \left(1 - \frac{1}{n_c} \right) .
\end{aligned}$$

Applied to 3 dimensions, for one cell with a given n_c follows the relative velocity distribution

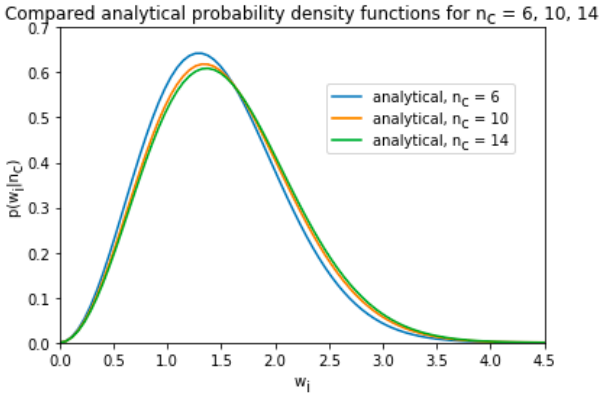
$$p(\vec{w}_i | n_c) = \left(\frac{m}{2\pi k_B T \left(1 - \frac{1}{n_c}\right)} \right)^{3/2} \exp\left(-\frac{m}{2k_B T \left(1 - \frac{1}{n_c}\right)} \vec{w}_i^2 \right) . \quad (2.13)$$

Considering that the cell particle number n_c fluctuates over time, and for a chosen cell is given by the Poisson distribution [20], the actual distribution function of the relative velocities $\vec{w}_i$ is obtained by averaging Eq.2.13 over the distribution of n_c (2.14)

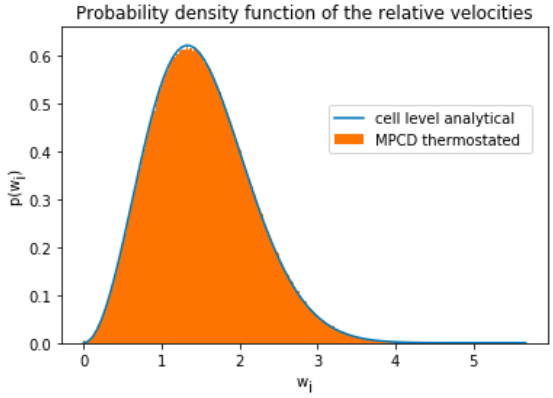
$$p(n_c) = e^{-\langle n_c \rangle} \cdot \frac{\langle n_c \rangle^{n_c}}{n_c!} \ , \tag{2.14}$$

$$p(\vec{w}_i) = \frac{1}{1 - (\langle n_c \rangle + 1)\, e^{-\langle n_c \rangle}} \cdot \sum_{n_c=2}^{\infty} e^{-\langle n_c \rangle} \cdot \frac{\langle n_c \rangle^{n_c}}{n_c!} \cdot p(\vec{w}_i | n_c) \ . \tag{2.15}$$

Comparing these theoretical findings with simulation results of an MPCD-solvent, it shows that the results agree well with each other. The following figure (2.1) shows the solvent relative velocity distribution a) analytically for different cell particle numbers n_c and b) comparing the analytical expression from eq. 2.15 with simulation results.



(a) Theoretical distribution of the solvent relative velocities w_i for given cell particle numbers $n_c = 6$ (blue), $n_c = 10$ (orange), and $n_c = 14$ (green).



(b) Comparison between the simulation (orange histogram data) and the theoretical distribution (blue) of the solvent relative velocities w_i averaged over all cell particle numbers n_c.

Figure 2.1: Solvent velocity distribution

2.3 Solvent diffusion

2.3.1 Analytic estimate for the solvent diffusion constant D

The self-diffusion coefficient is one of the main important properties of a solvent. Its calculation can be obtained from the Green-Kubo relation [20]

$$D = \frac{h}{6} \left\langle \vec{v}_i^2(0) \right\rangle + \frac{h}{3} \sum_{n=1}^{\infty} \left\langle \vec{v}_i(nh) \vec{v}_i(0) \right\rangle , \tag{2.16}$$

using the velocity auto-correlation function at discrete time steps $t_n = nh$. In order to analytically estimate the velocity auto-correlation function in this equation, we start by using 2.4

$$\langle \vec{v}_i(t+h)\vec{v}_i(t)\rangle = \langle \vec{v}_{cm}(t)\vec{v}_i(t)\rangle + \left\langle \hat{D}(\vec{\mathcal{R}},\alpha)\ \vec{v}_i(t)\cdot\vec{v}_i(t)\right\rangle - \left\langle \hat{D}(\vec{\mathcal{R}},\alpha)\ \vec{v}_{cm}(t)\cdot\vec{v}_i(t)\right\rangle . \tag{2.17}$$

The rotation matrix $\hat{D}(\vec{\mathcal{R}},\alpha)$ is independent of the particle velocities, which means it can be inspected isolatedly. For its expected value it is only necessary to look at the rotation vector $\vec{\mathcal{R}}$, since α is a constant. The rotation vector $\vec{\mathcal{R}}$ is sampled to be isotropic in space which means that all its components vanish when averaged. As a result the off-diagonal elements of $\hat{D}(\vec{\mathcal{R}},\alpha)$ are 0. This is different for the quadratic components $\mathcal{R}_i$, which are $\langle \mathcal{R}_i^2\rangle = \frac{1}{3}$ when averaged. For the rotation matrix we get as a result:

$$\left\langle D_{ij}(\vec{\mathcal{R}},\alpha)\right\rangle = \frac{1}{3}\left(1+2\cos\alpha\right)\cdot\delta_{ij}. \tag{2.18}$$

This simplifies (2.17) to

$$\langle \vec{v}_i(t+h)\vec{v}_i(t)\rangle = \frac{1}{3}(1+2\cos\alpha)\left\langle \vec{v}_i^2(t)\right\rangle + \frac{2}{3}(1-\cos\alpha)\left\langle \vec{v}_{cm}(t)\vec{v}_i(t)\right\rangle . \tag{2.19}$$

The average of the product between particle velocity and center of mass velocity will now be estimated by applying the molecular chaos assumption, which means that different particles are assumed to be independent. This is only an approximation, as due to hydrodynamic interactions they are actually correlated.

$$\begin{aligned}\langle \vec{v}_{cm}(t)\vec{v}_i(t)\rangle &= \left\langle \vec{v}_i(t)\frac{1}{n_c}\sum_j^{n_c}\vec{v}_j(t)\right\rangle = \left\langle\frac{1}{n_c}\right\rangle\left\langle \vec{v}_i(t)\sum_{j\neq i}^{n_c}\vec{v}_j(t)\right\rangle + \left\langle\frac{1}{n_c}\right\rangle\left\langle\vec{v}_i^2(t)\right\rangle \\ &= \left\langle\frac{1}{n_c}\right\rangle\left\langle\vec{v}_i^2(t)\right\rangle .\end{aligned} \tag{2.20}$$

To calculate the expectation value of $\frac{1}{n_c}$ the expression is averaged over the Poisson distribution, but since the particular particle i in the cell is fixed we need the probability distribution of finding $n_c - 1$ other particles in the cell. As a result we obtain

$$\left\langle\frac{1}{n_c}\right\rangle = \sum_{n_c-1=0}^{\infty}\frac{1}{n_c}\ e^{-\langle n_c\rangle}\ \frac{\langle n_c\rangle^{n_c-1}}{(n_c-1)!} = \frac{1}{\langle n_c\rangle}\sum_{n_c=1}^{\infty}e^{-\langle n_c\rangle}\ \frac{\langle n_c\rangle^{n_c}}{n_c!} = \frac{1}{\langle n_c\rangle}\ (1-e^{-\langle n_c\rangle}). \tag{2.21}$$

At last, after putting the results together and iterating, an analytical estimate for the velocity auto-correlation function can be found:

$$\langle \vec{v}_i(t+h)\cdot\vec{v}_i(t)\rangle = (1-\gamma)\cdot\left\langle\vec{v}_i^2(t)\right\rangle \ ,\ \text{with}\ \gamma = \frac{2(1-\cos\alpha)}{3\left\langle n_c\right\rangle}\left(\langle n_c\rangle - 1 + e^{-\langle n_c\rangle}\right), \tag{2.22}$$

$$\langle \vec{v}_i(nh) \cdot \vec{v}_i(0) \rangle = (1-\gamma)^n \cdot \left\langle \vec{v}_i^2(0) \right\rangle \ . \tag{2.23}$$

The diffusion coefficient D follows after realizing that the expected squared particle velocity is $\langle \vec{v}_i^2 \rangle = \frac{3k_BT}{m}$ and therefore

$$D = \frac{hk_BT}{2m} + \frac{h}{3}\sum_{n=1}^{\infty}(1-\gamma)^n \ \frac{3k_BT}{m} = \frac{hk_BT}{m}\left(\frac{1}{\gamma} - \frac{1}{2}\right) \tag{2.24}$$

within the molecular chaos assumption. We expect that including hydrodynamic interactions, the diffusion coefficient will be higher than in (2.24) as shown in [22] and in the following section (2.3.2).

2.3.2 Simulation results for the solvent diffusion constant D

The diffusion coefficient D from eq. 2.16 can also be expressed in terms of particle displacements instead of velocities if the sum is executed [19]

$$D := \lim_{t\to\infty} \frac{1}{6t} \left\langle \left(\vec{r}_i(t) - \vec{r}_i(0)\right)^2 \right\rangle = \lim_{t\to\infty} \frac{1}{6t} \left\langle \Delta\vec{r}_i^2(t) \right\rangle . \tag{2.25}$$

In simulation it is often determined by making a linear fit of the mean square displacement points $\left\langle \left(\vec{r}_i(t) - \vec{r}_i(0)\right)^2 \right\rangle$ at times t after the curve has stabilized. The diffusion coefficient is then simply $\frac{1}{6}$ of the linear fit's slope:

$$\left\langle \Delta\vec{r}_i^2(t) \right\rangle = 6D \cdot t. \tag{2.26}$$

We now evaluate the diffusion coefficient for different rotation angles α and compare the results with the analytical derivation in (2.24), keeping the rest of the parameters as summarized in table 2.1. The results are shown in figure 2.2 for the analytical curve (black) and simulation result values (red). As predicted, the simulation results are slightly higher than the theoretical estimate. This is especially true for larger angles α, as with increasing α the solvent behaves more like a liquid and less like a gas, which means that the role that hydrodynamic interactions play becomes more prominent.

2.3.3 Solvent viscosity and Schmidt number

The kinematic viscosity $\nu = \eta/\varrho$ ($\varrho = nm/l^3$ being the mass density of the fluid) consists of two components, which are the kinetic viscosity ν_{kin} and the collisional viscosity ν_{coll} [22]:

$$\nu = \nu_{kin} + \nu_{coll}, \tag{2.27}$$

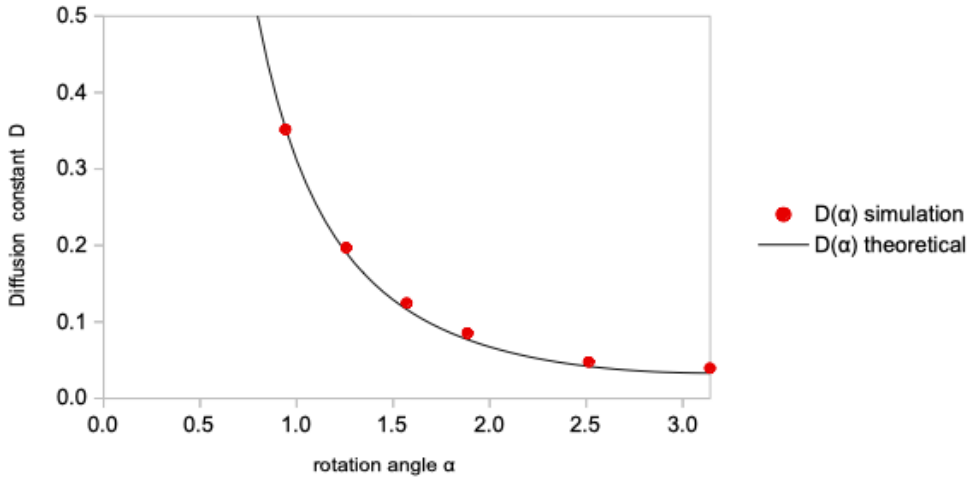


Figure 2.2: Self-diffusion coefficient D for the MPCD-solvent at different rotation angles α. The simulation results (red dots) are comparably higher than the theoretical approximation (black line) according to eq. 2.24.

$$\nu_{kin} = \frac{hk_BT}{m}\left(\frac{5\,\langle n_c\rangle}{(\langle n_c\rangle - 1)\,(4 - 2\cos\alpha - 2\cos(2\alpha))} - \frac{1}{2}\right), \tag{2.28}$$

$$\nu_{coll} = \frac{a^2}{18h}\,(1-\cos\alpha)\left(1 - \frac{1}{\langle n_c\rangle}\right). \tag{2.29}$$

As the diffusion coefficient can be interpreted as a measure of mass transport, the viscosity can be seen as a measure of momentum transport. In that way, the kinetic viscosity refers to the momentum transport by the particles' movement themselves, while the collisional viscosity describes transport that occurs via exchange during the particles' collisions. The ratio between the kinetic viscosity ν_{kin} and the collisional viscosity ν_{coll} varies strongly depending on the choice of MPCD parameters. Using eq. 2.28 and eq. 2.29 with the parameters of our choice (see section 2.1.4) we get:

$$\nu_{kin} = 0.039 \text{ and } \nu_{coll} = 0.656\ .$$

The collisional viscosity is the dominant component, which means that the MPCD-solvent of our choice behaves more like a liquid where the transport of momentum is mainly due to collisions and hydrodynamic interactions play a significant role. The opposite case would be a more gas-like state where momentum transport is mainly kinetic.

Another important measure to estimate the importance of hydrodynamics is the Schmidt number $Sc = \nu/D$ that describes the ratio between momentum transport (kinematic viscosity) and mass transport (diffusion coefficient). For gases $Sc \sim 1$ it is mostly around unity, whereas for liquids it is typically on the order of $Sc \sim 10^2$ to $\sim 10^3$. Using our calculated results for the viscosity and the simulation result for the diffusion coefficient we get:

$$Sc = 8.17$$

which is somewhere in between the typical gaseous and liquid states. In conclusion we expect hydrodynamics to play an important role with the chosen solvent parameters, but it could be even more significant if we chose smaller time steps h and larger collision angles α.

Chapter 3

The monomer resolved model (MRM)

In a next step to simulate a complex fluid, it is necessary to embed solute particles and couple their dynamics to the MPCD-solvent. Shortly after Malevanets and Kapral (1999) [9] presented their MPCD algorithm for a solvent, Malevanets and Yeomans (2000) [23] suggested to do the coupling by including the solute particles in the MP-collision step. More precisely, they coupled the solvent particles to monomer beads of short polymer chains. Monomer-monomer interaction would be handled via MD simulation using a velocity Verlet algorithm with forces directed by a harmonic potential. This approach has since been used often and successfully by many others. Ripoll et al. (2004) [24] used this method to study the influence of hydrodynamic interactions on short polymer chains, and Mussawisade et al. (2005) [10] used this method with different potentials for the monomer-monomer interaction. Both show that the simulation results agree well with Zimm theory for an appropriate choice of simulation parameters.

A different approach for the coupling between solute and solvent particles would be to use an additional force to describe their interaction. This approach was introduced by Malevanets and Kapral (2000) [25] and has also since been widely used. It is easy to implement, however it requires a very small time step for the molecular dynamics algorithm h_{MD} which means that computation can be costly for certain applications [19].

3.1 The MRM algorithm

In order to distinguish well between solvent particles and monomers in notation, their respective properties are written with small letters for solvent related quantities and capital letters for monomer related quantities.

3.1.1 Initialisation of the ring polymer

For the monomer resolved ring polymer we need to initialize N monomers each with mass M. Each of them has an individual initial position $\vec{R}_i(0)$ and velocity $\vec{V}_i(0)$ for monomer $i = 1, ..., N$. In general, the polymer can be initialised in any physically feasible state as long as the topology

matches the desired polymer's, which is a ring shape for this work. If the initial state is not a typical equilibrium state, the configuration will relax over time. The probably simplest and hereby selected way to initialise a ring polymer, albeit not at all a typical equilibrium conformation, is in the shape of a regular polygon in a chosen 2D-plane, with a side length equal to a typical monomer-monomer bond length R_{opt}. The velocities were initialised following a Maxwell-Boltzmann distribution w.r.t. a temperature k_BT, which is also the equilibrium velocity distribution conveniently. The velocities were centered, as was done with solvent velocities, such that the the average velocity is 0. This is to ensure that the total system has no global flow.

3.1.2 Internal molecular dynamics

The monomer-monomer interactions are simulated using the purely repulsive Lennard-Jones potential $U_{LJ}(R)$, also known as the Weeks-Chandler-Andersen potential, for excluded-volume interactions as well as the finitely extensible nonlinear elastic potential $U_{FENE}(R)$ for bonds as used by Chubak et al. (2020) [26]. They also added a bending potential $U_{angle}(\theta)$ for the angle between neighbouring bonds. For simplicity reasons this will not be done here and the bonds are kept fully flexible. The molecular dynamic was performed by integrating Newton's equations of motion numerically with the velocity Verlet algorithm at discrete time steps h_{MD}.

The purely repulsive Lennard-Jones potential

$$U_{LJ}(R) := \left(4\varepsilon\left[\left(\frac{\sigma}{R}\right)^{12} - \left(\frac{\sigma}{R}\right)^{6}\right] + \varepsilon\right) \cdot \theta(2^{1/6}\sigma - R) \tag{3.1}$$

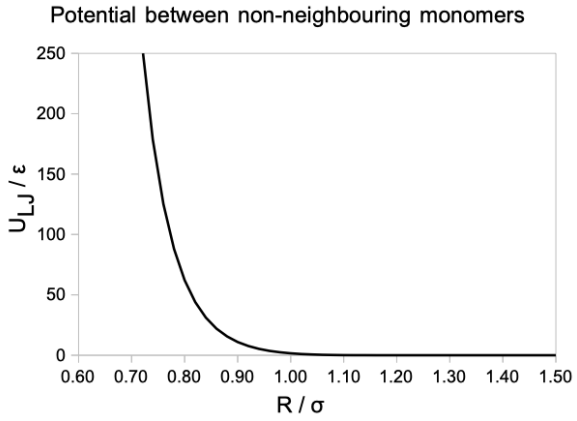
is applied between all monomer pairs. θ denotes the Heavyside step function, R is the absolute distance between the 2 monomers, σ can be interpreted as monomer diameter of soft spheres, and ε sets the energy scale.

The finitely extensible nonlinear elastic potential

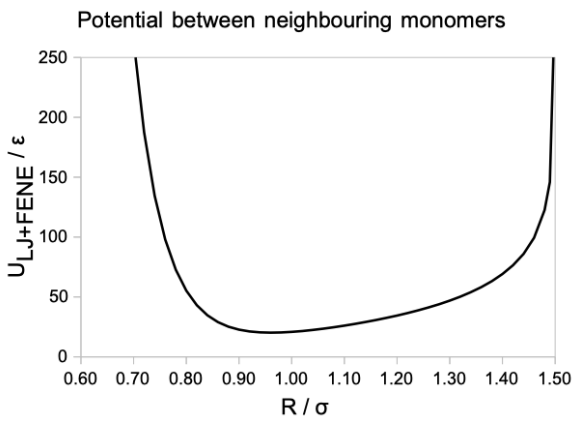
$$U_{FENE}(R) := -\frac{K}{2}R_{max}^2 \cdot \ln\left[1 - \left(\frac{R}{R_{max}}\right)^2\right], \quad R < R_{max} \tag{3.2}$$

is only applied between neighbouring monomers. Also here, R represents the absolute distance between the 2 neighbouring monomers, K is a spring constant, and R_{max} is the maximum allowed distance between 2 neighbouring monomers and is also a constant. It should be noted that the function for $U_{FENE}(R)$ is not defined for $R \geq R_{max}$. This state represents the breaking of a bond and is something that is not supposed to happen in our application. While the interaction between 2 non-neighbouring monomers is only governed by the Lennard-Jones potential, the interaction between 2 neighbouring monomers is governed by a sum of $U_{LJ}(R)$ and $U_{FENE}(R)$ (see also figure 3.1):

$$U_{i,i+1}(R) = U_{LJ}(R) + U_{FENE}(R) \tag{3.3}$$



(a) The pair potential between non-neighbouring monomers is represented by the purely repulsive Lennard-Jones potential.



(b) The pair potential between neighbouring monomers is a sum of the repulsive Lennard-Jones potential and the FENE potential.

Figure 3.1: Monomer-monomer pair potential

Section 3.1.1 mentions that for the initialization of monomer space coordinates a typical monomer-monomer distance was chosen R_{opt}. The most typical distance would be where the potential is minimal. But it is also possible to select the mean distance or any distance that is good within the potential well.

In order to use the velocity Verlet integration scheme, it is necessary to calculate the forces, which are defined as the negative gradient of the potential:

$$\vec{F}(\vec{R}) := -\nabla U(\vec{R}). \tag{3.4}$$

The total force acting on monomer i is derived from the sum of all potentials affecting it

$$\vec{F}_i(t) = \vec{F}_{LJ,i}(t) + \vec{F}_{FENE,i}(t), \tag{3.5}$$

with the force from the Lennard-Jones potential induced by all other monomers $j \neq i$

$$\vec{F}_{LJ,i}(t) = 24\varepsilon \sum_{j \neq i}^{N} \left(\frac{2\sigma^{12}}{R_{ij}^{14}(t)} - \frac{2\sigma^{6}}{R_{ij}^{8}(t)} \right) \theta(2^{1/6}\sigma - R_{ij}(t)) \cdot \vec{R}_{ij}(t), \tag{3.6}$$

and the force from the FENE potential that is only induced by neighbouring monomers $j = i \pm 1$

$$\vec{F}_{FENE,i}(t) = -K \sum_{j=i\pm1} \frac{1}{1 - \frac{\vec{R}_{ij}^2(t)}{R_{max}^2}} \cdot \vec{R}_{ij}(t). \tag{3.7}$$

The forces use the monomer-monomer distance which is defined as

$$\vec{R}_{ij}(t) = \vec{R}_i(t) - \vec{R}_j(t), \quad R_{ij}(t) = |\vec{R}_i(t) - \vec{R}_j(t)|.$$

The force is indirectly time dependent since it is a function of the particle positions which are themselves a function of time. The monomer positions $\vec{R}_i(t)$ and velocities $\vec{V}_i(t)$ can then be updated continually with the velocity Verlet algorithm [27]

$$\vec{R}_i(t + h_{MD}) = \vec{R}_i(t) + \vec{V}_i(t)h_{MD} + \frac{\vec{F}_i(t)}{2M}h_{MD}^2, \tag{3.8}$$

$$\vec{V}_i(t + h_{MD}) = \vec{V}_i(t) + \frac{\vec{F}_i(t) + \vec{F}_i(t + h_{MD})}{2M}h_{MD}. \tag{3.9}$$

3.1.3 Coupling between the MRM-polymer and the MPCD-solvent

A simple yet effective way to couple solute particles to the MPCD-solvent is to let them participate in the multi-particle collision [19] [20]. The monomers participate only in the collision step of the MPCD. They do not follow the streaming step, as their displacement is handled by the MD instead. Analogous to the solvent, monomers are treated as point-like particles for the collision step with respective mass M and velocity $\vec{V}_i$. The center of mass velocity of each cell with n_c solvent particles and N_c monomer particles is then calculated through

$$\vec{v}_{cm} = \frac{1}{mn_c + MN_c}\left(\sum_{i=1}^{n_c} m\vec{v}_i + \sum_{k=1}^{N_c} M\vec{V}_k\right), \tag{3.10}$$

replacing eq. 2.3 in that regard. Eq. 2.2 and 2.4 to calculate the relative velocities and the new velocities after collision do not have to be changed when adding solute particles to the collision. They only need to use the joint center of mass velocity, and they are applicable to both solvent particles and monomers.

Including the polymer in the cell-level thermostating process is very straightforward. In equilibrium, light particles and heavy particles have the same kinetic energy, which means the random choice of E_{kin} from (2.8) remains unchanged [22]. To calculate the scaling factor according to (2.9) the only thing to consider is to include the monomers in the denominator. The thermostating procedure does not have to be changed in any other way. However, the objective of this thesis is to compare different approximation models, and it is not always trivial what the thermostating process could look like. Therefore we chose to not include the polymer or any subsequent embedded object in the thermostating step, but rather let it be thermostated indirectly through the solvent. Simulation results show that thermal equilibrium is reached very quickly and therefore the indirect thermostating is sufficient (see section 3.2.1).

In the following there are instances when polymer simulation results with MPCD-solvent are compared to the results with no background solvent. In those cases the polymer was thermostated

using the same cell-level canonical thermostat as in section 2.1.3 for the monomers, but considering the whole polymer to be in one cell. In the end, this thermostat is a simple canonical velocity scaling thermostat as described in [28]. It was only used for simulations without solvent.

3.1.4 Choice of MRM parameters

The parameters of the polymer and for the MD should be in accordance with the solvent and MPCD specific values. In the collisional coupling scheme the monomer diameter σ should be of the order of the cell length a [29]. This is because in real systems the hydrodynamic radius of a solute particle is close to its structural radius $\sim \sigma/2$, and in the collisional coupling scheme the solute influences the solvent at the length scale of the collision cell a only. Hence we set $\sigma = a$. Also the monomer mass should be roughly equal to the total solvent mass per cell $M = m \langle n_c \rangle$ [22]. This ensures that both the embedded object and the solvent influence each other equally. The monomer number N can be varied as long as it is in accordance with the system size. For this work $N = 100$ was chosen for all simulations.

The combination of the purely repulsive Lennard-Jones potential with the finitely extensible nonlinear elastic potential to simulate polymer bonds is a well-established model by Grest and Kremer (1986) [30]. In the LJ-potential the energy and length scale are typically set to $\varepsilon = 1$ and $\sigma = 1$ respectively. The constants in the FENE-potential are generally chosen as $K = 30\varepsilon/\sigma^2$ and $R_{max} = 1.5\sigma$, both in relation to the LJ parameters. These numbers prevent the chains from crossing and have since been well established and used by many authors.

Using these parameters, time steps of $h_{MD} = 0.01\tau$ and $h_{MD} = 0.001\tau$ both have been employed successfully [31]. For this thesis a time step of $h_{MD} = 0.002\tau$ was selected. Remembering that the unit of time $\tau = \sqrt{ma^2/k_BT} = 1$ necessarily, and that the MPCD time step size $h = 0.1\tau$, MD steps and MPCD steps have to be alternated correctly such that the time is synchronised. Since the time step for the MPCD h is a multiple of the time step for the MD h_{MD}, the MD algorithm has to be performed $h/h_{MD} = 50$ times for each MPCD step.

Table 3.1 depicted below summarizes the numerical values that were chosen for the various parameters in all subsequent simulations if not explicitly stated otherwise. All numbers are given in reduced units.

$N = 100$	number of monomers the ring polymer has
$M = 10$	monomer mass
$h_{MD} = 0.002$	MD time step size
$h/h_{MD} = 50$	MD steps per MPCD step
$\sigma = 1$	monomer diameter and length scale for $U_{LJ}(R)$
$\varepsilon = 1$	energy scale for $U_{LJ}(R)$
$K = 30$	spring constant for $U_{FENE}(R)$
$R_{max} = 1.5$	maximum distance between neighbouring monomers for $U_{FENE}(R)$
$\langle N_c \rangle = N \cdot (a/l)^3 = 0.001$	average number of monomers in one collision cell

Table 3.1: Summary of MRM parameters

3.2 Spatial properties of the MRM ring polymer

3.2.1 Polymer gyration tensor

To quantify the polymer's conformation the gyration tensor is considered [32]

$$G_{\alpha\beta} = \frac{1}{N} \sum_{i=1}^{N} \langle S_{i,\alpha} S_{i,\beta} \rangle \,, \tag{3.11}$$

where $S_{i,\alpha} = R_{i,\alpha} - R_{CM,\alpha}$ is the α-coordinate of the relative position vector of monomer i with respect to the polymer center of mass $\vec{R}_{CM}$. The expectation value refers to the average over time. Of interest are especially the gyration tensors' eigenvalues $\{\Lambda_j\}$, $i = 1, 2, 3$ and the radius of gyration R_G which is connected to the eigenvalues via

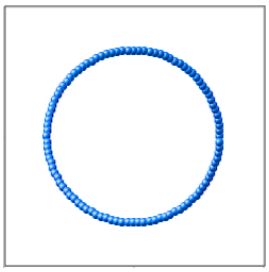
$$R_G^2 = \Lambda_1 + \Lambda_2 + \Lambda_3. \tag{3.12}$$

Subsequently, especially the square-roots of the eigenvalues $\lambda_j = \sqrt{\Lambda_j}$ are of importance, since they are a measure of length and can directly be related to the polymer's extension. In this work they are hereby called "root-eigenvalues". The radius of gyration is used to track the polymer's extension on average, and the (from here on sorted) root-eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$ give a deeper understanding about the polymer's proportions as they represent the polymer's 3 principal radii of gyration along the 3 principal axes [33].

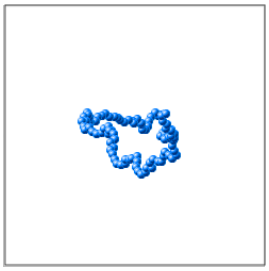
In order to obtain unbiased simulation results, the system has to be well equilibrated and be observed over a long enough time frame. The state in which the polymer was initialized, a regular polygon, is not an equilibrium conformation and has to be relaxed (see figure 3.2a,b). The equilibration progress can be estimated by observing the development of the gyration tensor root-eigenvalues $\{\lambda_j\}$ over time (as seen in figure 3.2c). When the fluctuations for all λ_j have stabilized the system's spatial properties can be seen as equilibrated. The remaining oscillatory behaviour of the λ_j that can be seen in fig. 3.2c can be explained by the fact that the first equilibration step was done without thermostating the polymer. Because this behaviour is not desired, all subsequent MD-only simulations of the polymer were thermostated like stated in sec. 3.1.3 to dampen these oscillations.

After the first equilibration step that resolves the polymer's isolated development, the resulting conformation is embedded in an MPCD solvent. Observing the time development of the root eigenvalues again, as well as the polymer's diffusion and temperature, the configuration was equilibrated for another $\Delta t = 5000\tau$. Over this span of time no significant change in the root eigenvalues or polymer temperature could be observed (fig. 3.3).

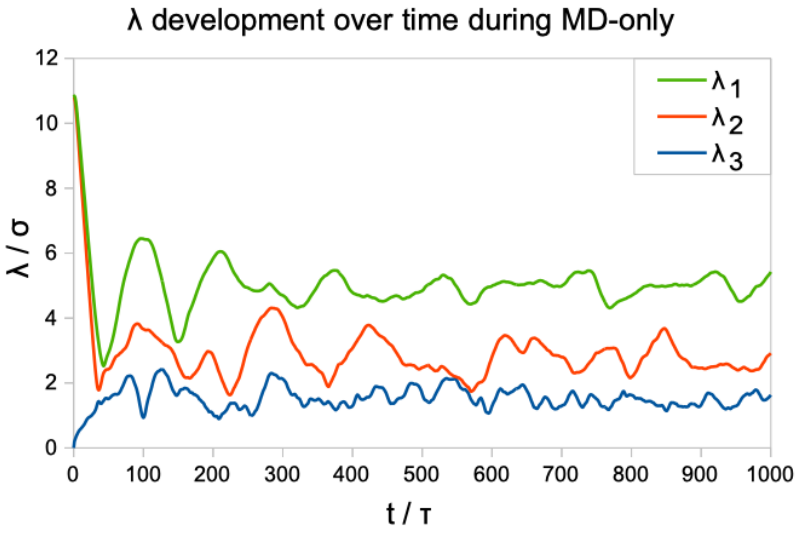
The mean square displacement, from which the diffusion coefficient is evaluated (eq. 2.26), was subject to heavy fluctuations. That means in order to obtain an accurate value for the diffusion coefficient it is necessary to average over multiple simulations. For a meaningful sample, the conformations have to be statistically independent. Because the diffusion coefficient is heavily influenced by the solute's size, creating a sample with independent conformations is especially



(a) Initial ring polymer conformation (t=0) projected on the xy-plane.



(b) Relaxed ring polymer conformation (t=1000) projected on the xy-plane.



(c) Time development of the root-eigenvalues λ. At around $t = 500$ all 3 values start to appear static, aside from remaining oscillatory fluctuations due to no thermostating. We consider the system to be relaxed at $t = 1000$.

Figure 3.2: Ring polymer equilibration using only MD and no background solvent

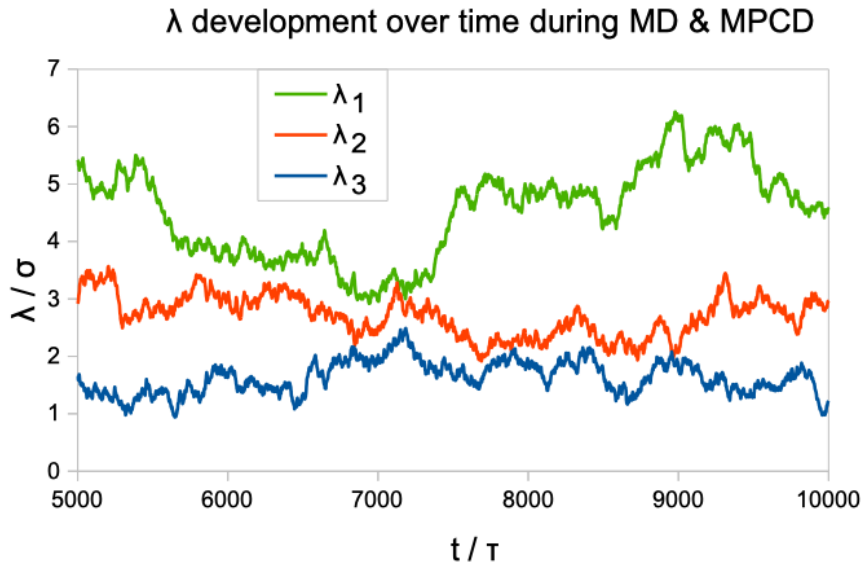


Figure 3.3: Ring polymer equilibration (second step) using MD and MPCD. Compared to the first equilibration step without solvent, there are many small fluctuations due to collisions with the solvent, and the oscillatory behaviour has vanished.

important. To judge conformational independence we consider the time development of the normalized diameter vector auto correlation function [34]

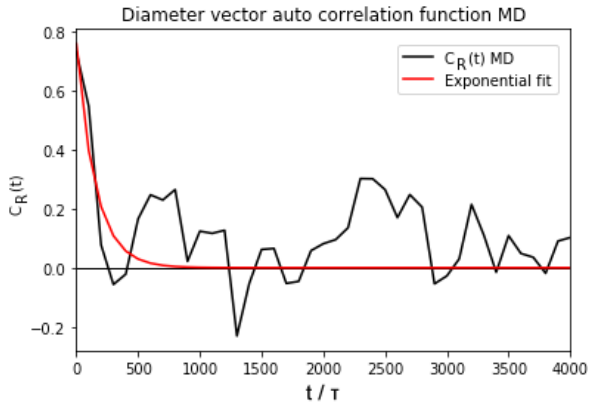
$$C_R(t) = \frac{\left\langle \vec{R}_d(0) \cdot \vec{R}_d(t) \right\rangle}{\left\langle \vec{R}_d^2 \right\rangle} \tag{3.13}$$

with the diameter vector $\vec{R}_d = \vec{R}_{i+N/2} - \vec{R}_i$ that denotes the distance between 2 opposing monomers in the ring polymer. The auto correlation function is expected to decay exponentially

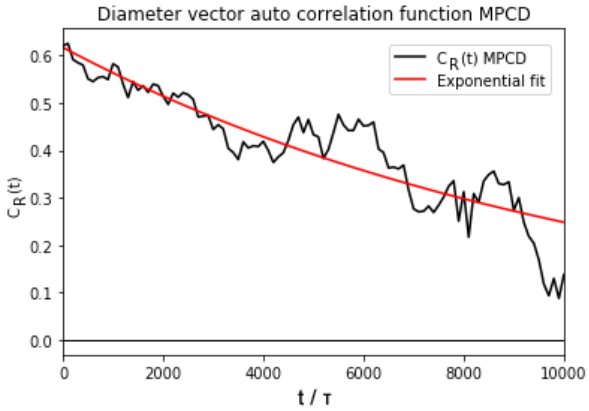
$$C_R(t) \propto \exp\left(-\frac{t}{\tau_{corr}}\right) \tag{3.14}$$

and the correlation time τ_{corr} provides a measure of how long it takes until the conformation is uncorrelated and hence statistically independent from the starting conformation. For this work 2 conformations are considered decorrelated if they are separated by at least $5\tau_{corr}$.

Figure 3.4 shows the diameter vector auto correlation function $C_R(t)$ for the ring polymer with and without MPCD solvent present. For simulations without embedding solvent the correlation time obtained by the exponential fit results in $\tau_{corr} = 154\tau$ whereas for simulations with an MPCD solvent the correlation time is $\tau_{corr} = 10981\tau$. Therefore, in order to obtain statistically independent samples, each simulation was independently equilibrated for 5000τ without solvent and another 5000τ with MPCD-solvent.



(a) The normalized diameter vector auto correlation function $C_R(t)$ for a ring polymer without solvent and its exponential fit. Already at $t \approx 1000\tau$ the correlations are close to 0.



(b) The normalized diameter vector auto correlation function $C_R(t)$ for a ring polymer in an MPCD-solvent and its exponential fit. Even after $t = 10000\tau$ the conformation is still visibly correlated.

Figure 3.4: Comparison between the normalized diameter vector auto correlation functions $C_R(t)$ of a ring polymer with and without embedding solvent. Without solvent the polymer conformation decorrelates much faster that with a solvent present.

It can be concluded that the equilibration time is different depending on several conditions including whether there is solvent present or not, whether thermostating is employed, and how artificial the initial configuration is. Therefore, it can be wise to create conditions that are favourable towards quick equilibration and change them only later when necessary.

The monomer velocity distribution is initialized in its equilibrium state and does not have to be explicitly monitored during the equilibration process. However, since the polymer itself is not thermostated directly but only indirectly through the solvent, it would be interesting to evaluate the efficiency of this indirect thermostating process. For that purpose, a simulation in which the ring polymer is initialized at $k_BT = 0$, meaning all monomers have a velocity of $\vec{V}_i(0) = 0$, was started and the temperature development over time observed. The temperature of the polymer can be estimated through its kinetic energy from classical mechanics

$$E_{kin} = \frac{1}{2} M \sum_{i=1}^{N} \vec{V}_i^2, \tag{3.15}$$

and from kinetic gas theory

$$E_{kin} = \frac{f}{2} k_B T, \quad f = 3N. \tag{3.16}$$

Equating the two delivers the relationship between polymer temperature and monomer velocities

$$k_B T = \frac{M}{3N} \sum_{i=1}^{N} \vec{V}_i^2. \tag{3.17}$$

The result of the thermostating process is shown by figure 3.5, and as can be seen the temperature adjusts very quickly. That concludes that indirect thermostating of the polymer via the solvent is a sufficiently efficient option.

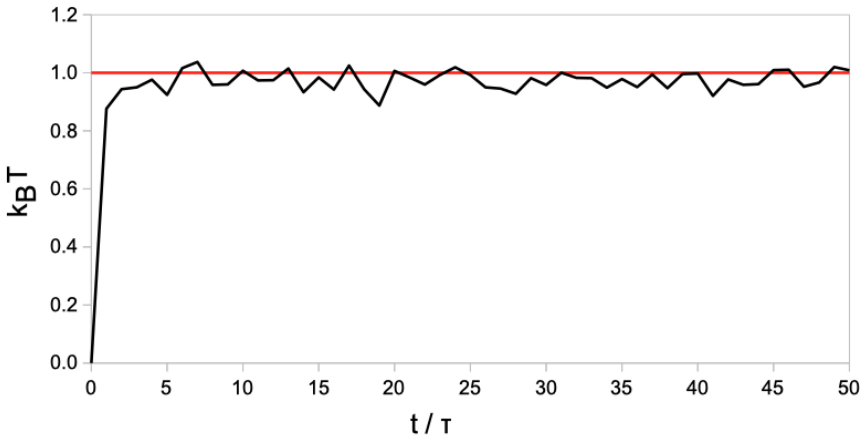


Figure 3.5: Temperature development of the monomer resolved ring polymer initialized at $k_BT = 0$ embedded in a thermostated MPCD solvent with $k_BT = 1$. The polymer's temperature quickly adjusts to its target value.

In order to determine the radius of gyration R_G and the root-eigenvalues of the gyration tensor $\{\lambda_j\}$ in equilibrium, a sample of 5 simulations each with and without the presence of an MPCD solvent was set up. All simulations used an equilibrated initial conformation and were evaluated over a time span of $\Delta t = 3000\tau$.

The simulation results for the gyration of a ring polymer with $N = 100$ in an MPCD-solvent are:

$$\lambda_{1,inS}/\sigma = 4.22 \pm 0.12, \;\; \lambda_{2,inS}/\sigma = 2.87 \pm 0.15, \;\; \lambda_{3,inS}/\sigma = 1.52 \pm 0.05,$$
$$R_{G,inS}/\sigma = 5.35 \pm 0.15.$$

The uncertainty indication refers to one standard deviation of the mean ($\bar{s} = s/\sqrt{n}$). Compared to that the same polymer without solvent (MD only) has the following gyration parameters:

$$\lambda_{1,noS}/\sigma = 4.27 \pm 0.06, \;\; \lambda_{2,noS}/\sigma = 2.73 \pm 0.02, \;\; \lambda_{3,noS}/\sigma = 1.60 \pm 0.01,$$
$$R_{G,noS}/\sigma = 5.35 \pm 0.06,$$

which is not distinguishable from the result with solvent included if the standard error is considered as the uncertainty (double standard error only for λ_3). This is an expected outcome because the equilibrium properties of a polymer are not affected by hydrodynamic interactions [19]. Therefore, quantities such as the radius of gyration are not influenced by the presence of an MPCD-solvent. This further justifies the use of isolated MD to generate a set of independent equilibrium conformations, adding the solvent only at a later stage.

3.2.2 Monomer density distribution

Another valuable spatial property is the radial monomer density distribution $\rho_{mon}(r)$ as it constitutes a useful basis for a coarse-grained model. The radial monomer density fulfills the normalization condition [35]

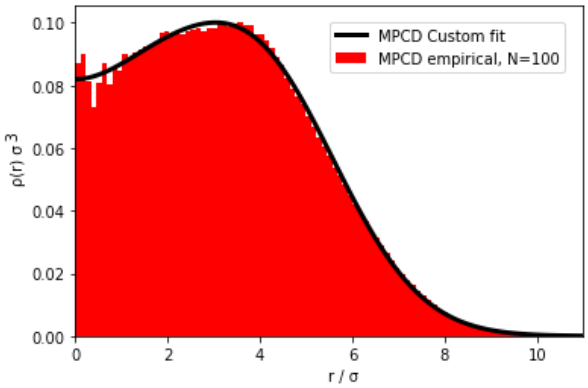
$$\iiint\limits_{V(r)} \rho_{mon}(r') \; d^3r' = 4\pi \int\limits_0^r \rho_{mon}(r') \; r'^2 \; dr' \stackrel{!}{=} N_{mon}(r) \tag{3.18}$$

where $N_{mon}(r)$ is the cumulative particle number from the center of mass at $r' = 0$ up to $r' = r$. If we define $R_{colloid}$ as the polymer's extension radius then $N_{mon}(R_{colloid}) = N$ is the total monomer number accordingly.

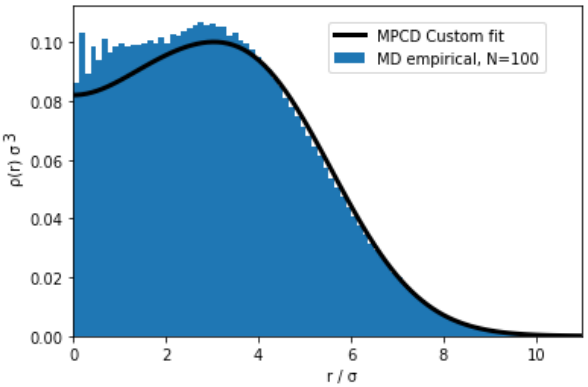
The radial monomer density $\rho_{mon}(r)$ can be assessed through simulation or theoretical considerations (as done for linear chains with a lattice self-avoiding random walk in [36]), and in order to obtain a smooth analytical function the discrete data is then approximated using an empirically created fit function. Figure 3.6 shows the simulation data for a ring polymer organized into a histogram and their empirical fit.

The analytical fit is based on the following function

$$\rho_{fit}(r) = A \cdot \left(r^5 + br^2 + d\right) \cdot e^{-\frac{r^2}{2s^2}} \tag{3.19}$$



(a) Radial monomer density distribution and empirically determined fit function of a ring polymer embedded in an MPCD solvent.



(b) Radial monomer density distribution of a ring polymer without an MPCD solvent present vs. the fit function obtained with MPCD solvent. The distribution is almost identical.

Figure 3.6: The radial monomer density distribution of a ring polymer

with fit parameters A, b, d, s. The fit function type was not derived from a theoretical background, but solely estimated from the histogram's appearance. Nevertheless, it provides a very good fit and is hence used as the basis for the coarse-grained models that follow. The fit parameters for the ring polymer with $N = 100$ result in $A = 2.246 \cdot 10^{-4}\sigma^{-8}, b = 66.16\sigma^3, d = 365.2\sigma^5, s = 2.128\sigma$.

If we compare the monomer density obtained from the ring polymer in an MPCD solvent (fig. 3.6 a) to the monomer density from the polymer without solvent (fig. 3.6 b), we can see that the results are almost identical. This is not surprising, since the other spatial properties R_G and $\{\lambda_j\}$ also show no dependence on whether an MPCD solvent is present or not.

3.3 MRM ring polymer diffusion

According to eq. 2.26 the diffusion coefficient D can be estimated from the mean squared-displacement $\langle \Delta \vec{r}_i^2(t) \rangle$. This is usually done by fitting the curve of the mean squared-displacement vs. time with a linear function. The slope of this fit then equals $6D$. The mean squared-displacement undergoes strong fluctuations and often needs some time at the beginning before the typical diffusive behaviour starts. Due to the fluctuations, it is important to observe the system over a sufficiently long time frame, and also to average over a substantial sample amount. For this work the time frame was $\Delta t = 5000\tau$ and a sample size of 20 independent simulations was evaluated, of which each had been well equilibrated. The averaged curve from all samples is shown in the following figure 3.7.

Hydrodynamic interactions are of long-range nature and velocity correlations persist over a long time [37]. Consequently, for a box with periodic boundaries the diffusion coefficient is dependent on the system size. For this reason, the diffusion coefficient obtained by the simulation is only the finite-size diffusion coefficient D_l of system size l. As shown by Singh et al. [38] the infinite-system-size diffusion coefficient D can be obtained from the finite-size diffusion coefficient by solving for the hydrodynamic radius R_H in

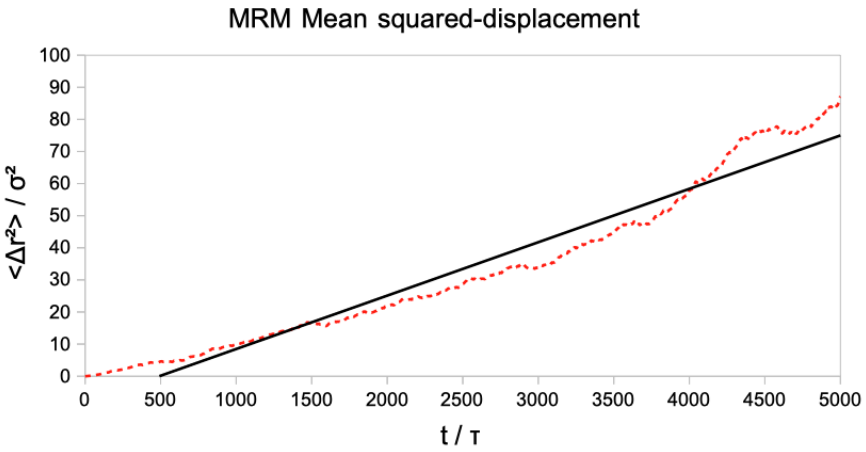


Figure 3.7: The diffusion coefficient is obtained by fitting the mean squared-displacement curve over time. In this graph, it is shown for the monomer resolved model. Simulation data is represented by the red dashed line, the linear fit is shown by the continuous black line.

$$D_l = \frac{k_B T}{6\pi\eta_{sol} R_H}\left[1 - \frac{R_H}{l}\left(2.837 - \frac{4\pi}{3}\frac{R_H^2}{l^2}\right)\right], \tag{3.20}$$

with the system size l and the dynamic viscosity of the solvent $\eta_{sol} = \varrho\nu$. This hydrodynamic radius R_H (or Stokes radius) is defined as the radius that a hard sphere would need in order to exhibit the same diffusion coefficient as the object at hand.

The infinite-system-size diffusion coefficient can subsequently be calculated using the Stokes-Einstein relation

$$D = \frac{k_B T}{6\pi\eta_{sol} R_H}. \tag{3.21}$$

For the finite-size diffusion coefficient of the MRM ring polymer we obtained:

$$D_l/D_0 = 0.00277 \pm 0.00045.$$

The unit of the diffusion coefficient is $D_0 = a^2/\tau = \sqrt{k_B T a^2/m}$. Considering the system size $l = 46$ and a solvent dynamic viscosity $\eta_{sol} = 7.143$ (as in section 2.3.3) this results in a hydrodynamic radius and hence an infinite-system-size diffusion coefficient of

$$R_H/a = 2.30 \quad \text{and} \quad D/D_0 = 0.00323.$$

3.3.1 Polymer diffusion in the Zimm model

Within the Zimm model it is possible to predict the diffusion coefficient, among other properties, of a chain polymer in a solvent [1]. The model was originally applied to ideal linear chains whereby the diffusion coefficient $D_{Z,l}$ is analytically calculated by

$$D_{Z,l} = \frac{8}{3\sqrt{6\pi^3}} \frac{k_B T}{\eta_{sol} b N^{\nu}} \tag{3.22}$$

with the expected bond length $b \approx \sigma$ and the Flory exponent ν which is $\nu = 1/2$ for ideal polymer chains. Based on this model, the Zimm diffusion coefficient using our parameters would be $D_{Z,l}/D_0 = 0.00274$, which is rather low compared to the infinite-system-size diffusion coefficient D from the simulation. However, the polymer in the simulation is cyclic and not linear. Moreover, it is self-avoiding and not ideal. Due to their topological constraints, ring polymers have a smaller hydrodynamic radius compared to linear chain polymers with the same monomer number N and monomer size σ. Therefore, also a higher diffusion coefficient is to be expected for them. Bloomfield and Zimm (1966) [39] expanded the Zimm model to ring polymers and they also introduced an expansion parameter ϵ to accommodate several effects contributing to polymer expansion or contraction including monomer excluded volume and solvent quality. The expansion parameter defines the value for the Flory exponent

$$\nu = \frac{1 + \epsilon}{2} \tag{3.23}$$

with the ideal polymer chain as a special case by setting $\epsilon = 0$. The diffusion coefficient for ring polymers within the Zimm model is calculated by

$$D_{Z,r} = \frac{2}{\sqrt{6\pi^3}} \frac{k_B T}{\eta_{sol} b N^{\nu}} \int_0^1 dq \; (1-q) \; \sqrt{(1-q)^{-2\nu} + q^{-2\nu}}, \tag{3.24}$$

and for the ideal ring polymer ($\nu = 1/2$) the integral can be solved analytically to

$$D_{Z,r} \; (\nu = 1/2) = \frac{1}{\sqrt{6\pi}} \frac{k_B T}{\eta_{sol} b N^{\nu}}. \tag{3.25}$$

Using our parameters this results in $D_{Z,r}/D_0 = 0.00322$ for the Zimm diffusion coefficient, which agrees very well with the simulation result.

For a polymer described by the Kremer–Grest model like in this work, the MPCD solvent is classified as a good solvent. That is because there are no excluded volume effects for monomers with solvent particles but there are repulsive forces between non-neighbouring monomers which leads to swelling of the polymer structure compared to a Gaussian chain. Therefore, a Flory exponent of $\nu = \frac{3}{5}$ as usual for self-avoiding chains [40] is more appropriate in this model. Eq. 3.24 can be solved numerically for the self-avoiding chain ($\nu = \frac{3}{5}$) which results in a diffusion coefficient of $D_{Z,r}/D_0 = 0.00265$. This value does not match the simulation result very well. The reason for that is most likely that with $N = 100$ the chain is rather short, such that excluded volume effects do not play a significant role.

Furthermore it is instructive to analyze the time scale for diffusive motion [1]. The Zimm relaxation time for an ideal linear chain

$$\tau_{Z,l} = \frac{1}{2\sqrt{3\pi}} \frac{\eta_{sol}}{k_B T} R^3 \propto \frac{R^2}{D_{Z,l}}, \quad R \approx bN^{\nu} \tag{3.26}$$

describes the characteristic time during which a chain polymer diffuses a distance of order of its own size. On time scales shorter than that $t < \tau_Z$ the chain exhibits viscoelastic modes and its motion is not purely diffusive. On time scales longer than that $t > \tau_Z$ the polymer moves diffusively. That means in order to measure the diffusion coefficient it is necessary to observe the displacement over a time frame longer than the Zimm time. For the polymer in this work the Zimm time amounts to $\tau_{Z,l} = 1163\tau$ if it were an ideal linear chain. Ring polymers of the same dimensions are always expected to have shorter relaxation times. This can be verified by comparing their diffusion coefficients in the Zimm model (eq. 3.22 and 3.25) which, albeit the same scaling behaviour, is slightly larger for rings than for chains. It has also been shown by simulation and experiments that this is the case [41]. Therefore the observed time frame of $\Delta t = 5000\tau$ is definitely long enough to succeed the polymer's Zimm relaxation time.

Chapter 4

The effective monomer model (EMM)

The first coarse grained model covered is the effective monomer model (EMM) which was introduced by Ruiz-Franco et al. (2019) [42]. The principle behind this model is to keep the polymer as monomer resolved for the interaction with the solvent, but cut the internal interaction between the monomers. The authors showed that this model produces the correct diffusion dynamics for linear chain polymers as compared to the monomer resolved model, but not so for star polymers. In the following section we will test its suitability for ring polymers.

4.1 The EMM algorithm

The EMM algorithm used in this section is based on the works of Ruiz-Franco et al. (2019) [42] if not explicitly stated otherwise.

4.1.1 Initialisation of the effective ring polymer

To initialize the effective polymer, the first step is to divide the simulation box into cells each with side length equal to the monomer diameter $a = \sigma$. We also need to define the cutoff radius $R_{colloid}$ for the monomer density function, on which we impose the condition $\rho_{mon}(R_{colloid})\sigma^3 < 10^{-3}$, to limit the polymer's extension. After that a uniform random number $\mathcal{R} \sim U(0,1)$ is drawn for each of the cells and compared to the value of the volume density $\rho_{mon}(r_{cell})\sigma^3$ at the distance r_{cell} of the cell's center to the simulation box center. If $\mathcal{R} \leq \rho_{mon}(r_{cell})\sigma^3$ we insert a monomer at the center of the respective cell. Cells whose center is located beyond the cutoff radius ($r_{cell} > R_{colloid}$) do not get a random number drawn for, as the volume density there is 0. At the end we have a number of monomers N_{eff} equal to the number of successful insertions with expectation value $\langle N_{eff} \rangle = N$. The following figure (fig. 4.1) shows an example configuration generated through this process. The initial velocities of the monomers are simply assigned to 0 since thermostating proceeds quickly.

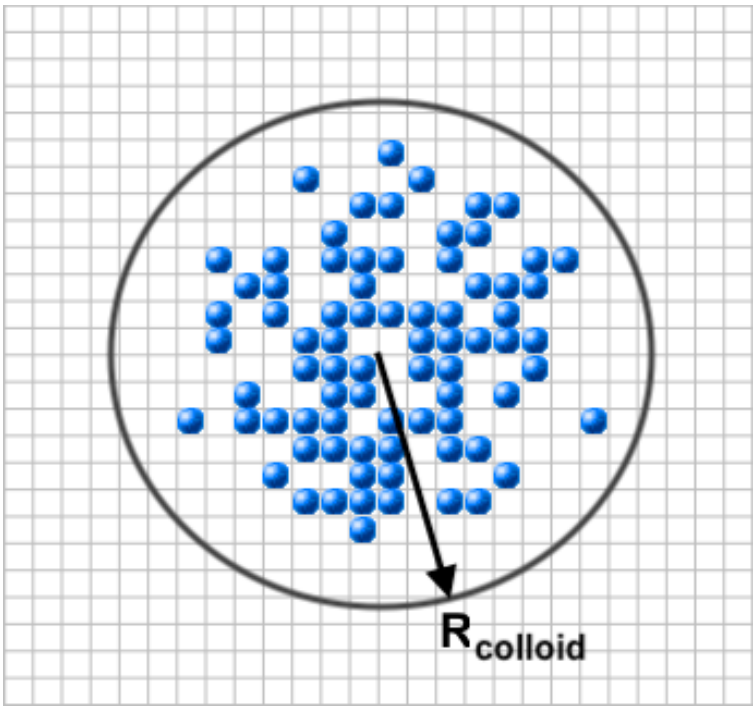


Figure 4.1: Example configuration after initialization for the effective monomer model EMM projected on the xy-plane

4.1.2 Internal interaction and coupling to the solvent

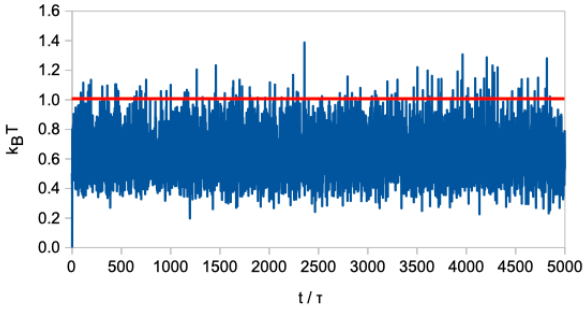
While the effective monomer model keeps depicting the polymer through a set of single effective monomers, the internal interaction between those is vastly simplified as compared to the monomer resolved model. Instead of undergoing molecular dynamics following the intermolecular potentials as discussed in section 3.1.2, the colloid moves like a rigid body and keeps its monomer configuration fixed over the course of the simulation. Consequently, the EMM neglects the angular momentum of the polymer. The colloid displacement, and with it all effective monomers, is ballistic in the same way as for the solvent particles following eq. 2.1 and uses its center of mass velocity $\vec{V}_{cm}$.

As for the interaction with the solvent we can simply let the effective monomers participate in the collision as described in section 3.1.3. After the collision the center of mass velocity $\vec{V}_{cm}$ needs to be updated using the new effective monomer velocities

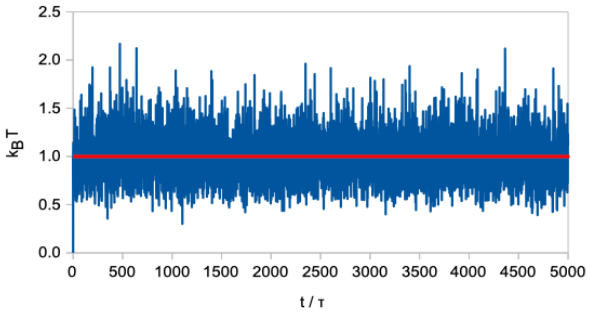
$$\vec{V}_{cm}(t+h) = \frac{1}{N_{eff}} \sum_{j}^{N_{eff}} \vec{V}_j(t+h). \tag{4.1}$$

Ruiz-Franco et al. (2019) [42] suggested that after calculating the new center of mass velocity, the effective monomer velocities should follow such that each monomer has the same velocity, which is the center of mass velocity ($\vec{V}_j(t+h) = \vec{V}_{cm}(t+h)$), before the next collision step. If we observe the colloid temperature however, we find that this balancing makes the colloid cooler compared to the thermostated solvent. This makes sense because the momentum exchange happens through the effective monomers with mass M_{mon} as opposed to the whole colloid with

mass $M = N_{eff} M_{mon}$. Through matching the velocities however, they are no longer Maxwell-Boltzmann distributed for the monomers and their temperature is comparably lower on average. Even after a collision with the thermostated solvent particles the temperature has not adjusted quickly enough before the next velocity averaging happens. For this reason, after a collision the individual monomer velocities were kept for the next collision in this work. The average velocity was only used for the colloid displacement. Figure 4.2 shows the difference in temperature for both methods.



(a) Temperature development of the EMM polymer if the individual effective monomer velocities $\vec{V}_j$ are averaged with each other before each collision step. The temperature fluctuates around an average value far lower than 1 ($k_B T < 1$).



(b) Temperature development of the EMM polymer if the individual effective monomer velocities $\vec{V}_j$ are kept between collision steps. The temperature fluctuates well around 1 ($k_B T = 1$).

Figure 4.2: Temperature development of the EMM polymer

Ruiz-Franco et al. (2019) [42] first tried to redraw the monomer configuration for each collision step. However the system showed unrealistic behaviour in the long term under those circumstances, which is why they decided to evaluate the dynamics by averaging over 14 different fixed rigid body configurations.

4.2 EMM ring polymer diffusion

The finite-size diffusion coefficient for the EMM ring polymer was evaluated by fitting the mean squared-displacement curve of 10 independent simulations over a time frame of $\Delta t = 5000\tau$. Compared to the monomer resolved model, the effective monomer model needed basically no equilibration time. The configuration remained fixed, and the temperature was equilibrated within the first few steps. The averaged curve from all samples is shown in the following figure 4.3.

For the finite-size diffusion coefficient of the EMM ring polymer we obtained:

$$D_l / D_0 = 0.00154 \pm 0.00030.$$

This results in a hydrodynamic radius and hence an infinite-system-size diffusion coefficient of

$$R_H / a = 3.72 \quad \text{and} \quad D / D_0 = 0.00200.$$

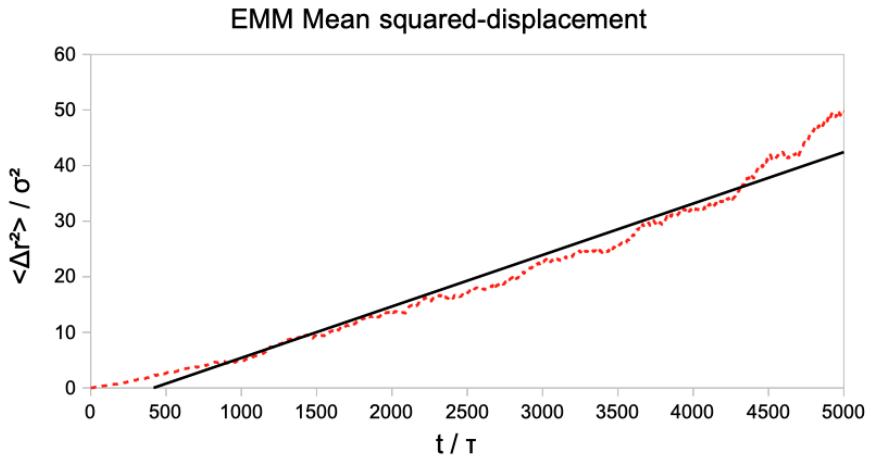


Figure 4.3: The mean squared-displacement of the effective monomer model over time (red dashed curve) vs. its linear fit (black continuous line).

Compared to the infinite-system-size diffusion coefficient in the monomer resolved model, this is significantly lower. Ruiz-Franco et al. (2019) [42] observed that the effective monomer model delivers satisfactory results for linear chains, but does not perform well for modeling star polymers. The simulation results for ring polymers in this work show that the effective monomer model is not well suited for ring polymers either. What stars and rings have in common, is that they possess nontrivial topologies, which evidently play an important role that is not captured by the EMM.

Chapter 5

The penetrable soft colloid model (PSC)

The second coarse grained model is the penetrable soft colloid model (PSC), which was also introduced by Ruiz-Franco et al. (2019) [42].

The term colloid refers to large molecules and particles, larger than the atomistic order of magnitude but small enough to experience significant Brownian motion [43]. Colloidal particles can be rigid entities, flexible macromolecules, or assemblies of small molecules in thermodynamic equilibrium with their environment such as water droplets in oil. Colloids have been studied a lot in the context of colloidal suspensions whose areas of application are manifold ranging from the sedimentation of clays, over biological systems such as cells or microorganisms, to nanotechnology and nano-manufacturing among many others [44].

In this model the monomer resolution is removed and instead the polymer is depicted as one single spherical colloid that is penetrable for solvent molecules. Movement of the solvent particles inside the colloid leads to interaction between the two, but the colloid itself acts as one rigid particle with no internal interaction. The authors showed that this approach replicates well the diffusion dynamics of star polymers as compared to the monomer resolved model, but not so for linear chain polymers. In this chapter the model is tested for its applicability to ring polymers.

5.1 The PSC algorithm

The PSC algorithm used in this section relies on the literature of Ruiz-Franco et al. (2019) [42] if not explicitly stated otherwise.

5.1.1 Initialisation of the soft penetrable colloid

The soft colloid consists of only one rigid body. Therefore it only needs one position vector $\vec{R}(t)$, indicating the position of its center, and one velocity vector $\vec{V}(t)$, which will be initialized at 0. Contrary to the previous coarse grained model, the PSC model also includes rotation which means that an angular velocity vector $\vec{\Omega}(t)$ needs to be initialized, also at 0. Both linear and

angular velocity equilibrate through the thermostated solvent over time.

The soft colloid can hardly be viewed as a point-like particle, instead it is modeled as a rigid body. Therefore, to govern angular momentum exchange with the solvent, we need the colloid's moment of inertia I additionally to its total mass $M = NM_{mon}$. For a radial symmetric object whose axis of rotation goes through the center of mass, the moment of inertia tensor can be simplified to a scalar. In this case it is calculated via [45]

$$I = \frac{2}{3} \iiint_V \varrho(r) r^2 dV = \frac{8\pi}{3} \int_0^{R_{colloid}} \varrho(r) r^4 dr \tag{5.1}$$

with the mass density $\varrho(r)$ of the rigid body, and the radial distance from its center r. The mass density $\varrho(r)$ and monomer density $\rho_{mon}(r)$ are connected through the monomer mass M_{mon}:

$$\varrho(r) = M_{mon} \cdot \rho_{mon}(r). \tag{5.2}$$

Using this relationship we obtain the following for the moment of inertia

$$I = \frac{8\pi}{3} M_{mon} \int_0^{R_{colloid}} \rho_{mon}(r) r^4 dr. \tag{5.3}$$

For a particle density function following (3.19) with the according fit parameters and a cutoff radius $R_{colloid}$ as defined in section 4.1.1 the value of the moment of inertia with which the colloid is initialized results in $I/I_0 = 19102$ with the unit $I_0 = ma^2$. The colloid's internal structure does not change over time, hence this value remains constant throughout the simulation.

5.1.2 Colloid movement and coupling to the solvent

The PSC is coarse-grained as such that it has no internal interactions. Its coupling to the solvent is implemented during the solvent particles' streaming step, contrary to the previous models where the coupling is during the multi-particle collision step. The PSC itself moves ballistically like a rigid body. Even though angular velocity is considered, enabling rotation of the colloid, its rotation angle does not need to be tracked due to its radial symmetry. In conclusion the algorithm consists of three subsequent steps: PSC ballistic movement, solvent particles ballistic movement with colloid interaction, and the multi-particle collision among the solvent particles.

The ballistic movement of the PSC works analogously to the solvent previously:

$$\vec{R}(t+h) = \vec{R}(t) + h \cdot \vec{V}(t). \tag{5.4}$$

As for the solvent particles, in this model they each have to pass a Monte-Carlo trial move in order to follow through with their ballistic movement according to (2.1). If the trial move is rejected,

they are scattered by the PSC instead. For this step, it is convenient to define the solvent particles position in reference to the PSC, such that r denotes the distance of the solvent particle relative to the colloid center $\vec{R}$. Trial moves for which both the initial and the final position lie outside the colloid are always accepted. If either one of the solvent particle's initial relative position $r_1 = |\vec{r}_1| = |\vec{r}_i(t) - \vec{R}(t+h)|$ or trial end relative position $r_2 = |\vec{r}_2| = |\vec{r}_i(t+h) - \vec{R}(t+h)|$ are inside the colloid ($r_1 \leq R_{colloid}$ or $r_2 \leq R_{colloid}$), the acceptance probability is calculated as follows:

$$p_{acc}(r_1 \to r_2) = \begin{cases} \min\left\{1; \frac{\rho_{sol}(r_2)}{\rho_{sol}(r_1)}\right\}, & \text{if } r_2 \leq r_1 \\ \min\left\{1; \frac{\rho_{sol}(r_1)}{\rho_{sol}(r_2)}\right\}, & \text{if } r_1 < r_2 \end{cases}. \tag{5.5}$$

The Monte-Carlo acceptance probability utilizes the solvent volume density $\rho_{sol}(r)$ which is the local volume fraction that is populated by the solvent after the monomer excluded volume is subtracted. Seeing that monomers are assumed to be sphere-shaped, the solvent density is calculated via

$$\rho_{sol}(r) = 1 - \frac{\pi}{6}\sigma^3 \cdot \rho_{mon}(r). \tag{5.6}$$

There are instances where the solvent density calculated this way is negative. This is especially true for the core of star polymers. In this case the solvent density should just be set to 0. With the monomer density function we obtained for ring polymers in this work however, this is not the case.

If the Monte-Carlo trial move is rejected, the solvent particle is scattered by the PSC. For that, we first define the scattering point $\vec{s}$. Dependent on whether the trial movement would have been fully or only partially inside the colloid, the particles are either scattered at the halfway point of that movement or at the colloid's surface respectively. In other words:

$$s = |\vec{s}| = \begin{cases} R_{colloid} & \text{if } r_1 < R_{colloid} < r_2 \text{ or } r_2 < R_{colloid} < r_1 \\ \left|\frac{\vec{r}_1 + \vec{r}_2}{2}\right| & \text{if } r_1, r_2 < R_{colloid} \end{cases}. \tag{5.7}$$

To show this more clearly, the two different cases are demonstrated in the following figure 5.1.

We can calculate the scattering point vector $\vec{s}$ by imposing the condition of its magnitude according to (5.7) and solving for constant k. To showcase this, we refer to fig. 5.2.

$$\vec{\Delta r} = \vec{r}_2 - \vec{r}_1 \tag{5.8}$$

$$\vec{s} = \vec{r}_1 + k \cdot \vec{\Delta r} = (1-k) \cdot \vec{r}_1 + k \cdot \vec{r}_2, \quad k \in (0,1) \tag{5.9}$$

$$|\vec{s}| = |\vec{r}_1 + k \cdot \vec{\Delta r}| \stackrel{!}{=} s \tag{5.10}$$

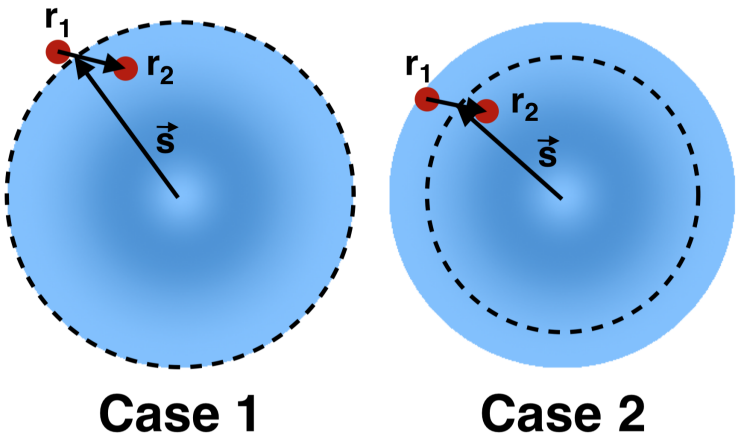


Figure 5.1: Scattering of the solvent particle by the PSC. Case 1: The solvent trial movement would be only partially inside the colloid, hence the particle is scattered at the colloid surface. Case 2: The solvent trial movement would be fully inside the colloid, therefore scattering happens at the halfway point.

For case 2, where scattering happens at the halfway point, this is straightforward as $k = 1/2$. In case 1, where scattering happens at the colloid surface, eq. 5.10 must be solved for $s = R_{colloid}$. Out of the 2 solutions that will present for k, only 1 fulfills the necessary condition $k \in (0, 1)$.

$$k_{1,2} = \frac{-\vec{r}_1 \cdot \vec{\Delta r} \pm \sqrt{(\vec{r}_1 \cdot \vec{\Delta r})^2 - \vec{\Delta r}^{\,2}(\vec{r}_1^{\,2} - R_{colloid}^2)}}{\vec{\Delta r}^{\,2}} \tag{5.11}$$

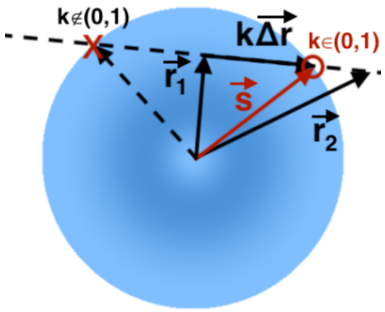


Figure 5.2: To calculate the scattering point vector, we solve for factor k. Only one solution is correct with $k \in (0, 1)$.

Naturally, on elastic collision we would impose energy conservation, linear momentum conservation, and angular momentum conservation. However, the PSC is technically not a continuous rigid body with a smooth surface, but more a construct of monomers, each with their individual momentum. Therefore instead of imposing energy conservation, the PSC should be viewed as a thermostated heat bath and the new solvent relative velocities are assigned randomly according to the distributions [47]

$$p(\nu_n) = \frac{m}{k_B T}\nu_n \cdot \exp\left(-\frac{m}{2k_B T}\nu_n^2\right), \tag{5.12}$$

$$p(\nu_t) = \sqrt{\frac{m}{2\pi k_B T}} \cdot \exp\left(-\frac{m}{2k_B T}\nu_t^2\right), \tag{5.13}$$

with the velocity component ν_n that is perpendicular to the colloid surface and the velocity component ν_t that is tangential to it. This is an approximation that assumes that the colloid surface is an infinitely extended plane, which is legitimate when the fluid mean free path length is small compared to the colloid diameter. Linear and angular momentum conservation remain upheld. The new absolute velocity of the solvent particle then calculates as

$$\vec{v}_i(t+h) = \vec{V}(t) + \vec{\Omega}(t) \times \vec{s}_i + \nu_n \hat{e}_n + \nu_t \hat{e}_t, \tag{5.14}$$

with the unit vector normal to the collision plane $\hat{e}_n$ and the unit vector tangential to it $\hat{e}_t$. Reviewing (5.5) we can see that this is not a usual Monte-Carlo acceptance probability, especially seeing this equation does not follow detailed balance [48]

$$\rho_{sol}(r_1)p(r_1 \to r_2) = \rho_{sol}(r_2)p(r_2 \to r_1), \tag{5.15}$$

$$\rho_{sol}(r_1)p_{gen}(r_1 \to r_2)p_{acc}(r_1 \to r_2) = \rho_{sol}(r_2)p_{gen}(r_2 \to r_1)p_{acc}(r_2 \to r_1). \tag{5.16}$$

The generation probabilities p_{gen} for both trial moves are the same

$$\begin{aligned} &p_{gen}(\vec{r}_1 \to \vec{r}_2) = p(\vec{v}_1) \\ &p_{gen}(\vec{r}_2 \to \vec{r}_1) = p(\vec{v}_2) = p(-\vec{v}_1) \end{aligned}$$

$p(\vec{v}_1) = p(-\vec{v}_1)$ ... Maxwell-Boltzmann distribution

$$\begin{aligned} \Rightarrow \quad & p_{gen}(\vec{r}_1 \to \vec{r}_2) = p_{gen}(\vec{r}_2 \to \vec{r}_1) \\ \Rightarrow \quad & p_{gen}(r_1 \to r_2) = p_{gen}(r_2 \to r_1) \end{aligned}$$

Therefore the equation for detailed balance reduces to

$$\rho_{sol}(r_1)p_{acc}(r_1 \to r_2) = \rho_{sol}(r_2)p_{acc}(r_2 \to r_1), \tag{5.17}$$

which is generally not fulfilled by (5.5). Therefore (5.5) alone would not yield the desired solvent density profile inside the colloid, but would lead to a uniform distribution. This is counteracted by the scattering procedure. In a usual Monte-Carlo scheme, a rejected move is simply not performed. In the PSC method here, a rejected move is replaced by scattering the particle

instead. By choosing the normal unit vector $\hat{e}_n$ as such that it always points away from the colloid center, independent of whether the solvent trial move constitutes penetration or escape, detailed balance is restored for the major part of the colloid (a detailed explanation of this can be found in section 7.1.1). As a result, the desired solvent density profile is achieved in the end (see section 5.2). Therefore we define the normal unit vector $\hat{e}_n$ with

$$\hat{e}_n = \frac{\vec{s}_i}{|\vec{s}_i|}. \tag{5.18}$$

The tangential unit vector $\hat{e}_t$ is chosen randomly under the condition that it is orthogonal to the normal unit vector: $\hat{e}_t \perp \hat{e}_n$.

After the collision, also the colloid's velocity and angular velocity have to be adapted:

$$\vec{V}(t+h) = \vec{V}(t) + \frac{m}{M} \cdot \left[\vec{v}_i(t) - \vec{v}_i(t+h)\right], \tag{5.19}$$

$$\vec{\Omega}(t+h) = \vec{\Omega}(t) + \frac{m}{I} \cdot \vec{s}_i \times \left[\vec{v}_i(t) - \vec{v}_i(t+h)\right]. \tag{5.20}$$

The collisions are resolved serially one solvent particle after the next, while always updating the colloid linear and angular velocity in between. The alternative would be resolving all solvent particles at once and updating the colloid linear and angular velocity only at the end, using the sum of all solvent momentum exchanges. While the first approach costs more computing time, especially if there are a lot of collisions to perform, the second approach can lead to unrealistic colloid behaviour. Since the solvent velocity expectation values are $\langle \vec{v}_i(t+h) \rangle = \vec{V}(t)$ and $\langle \vec{v}_i(t) \rangle \approx 0$, the colloid velocity can get very large if there are many scatterings per time step. This correlation does not occur in real systems, since collisions naturally happen more sequentially, rather than all at once. Therefore in this work the collisions are resolved one by one in the order of the particle's indexing, which is in no particular order in space.

Instead of ballistic movement, the alternative solvent displacement uses the new velocity due to scattering

$$\vec{r}_i(t+h) = \vec{r}_i(t) + \frac{h}{2}\vec{v}_i(t+h). \tag{5.21}$$

The third step, which is the multi-particle collision among solvent particles, is performed in the same way as if there was no solute present.

5.2 Solvent density distribution

After a certain amount of equilibration time the solvent particles inside the PSC are distributed according to the solvent density profile (eq. 5.6). This process is very quick, such that the equilibrium density profile is already achieved after only 1000 time steps, which is the equivalent

of a time span $\Delta t = 100\tau$ in our system with step size $h = 0.1$. Figure 5.3 below shows the solvent density distribution in equilibrium state for the PSC model compared to the theoretical solvent density function.

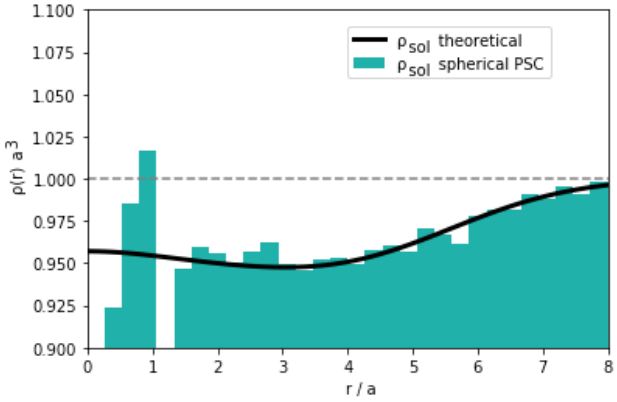
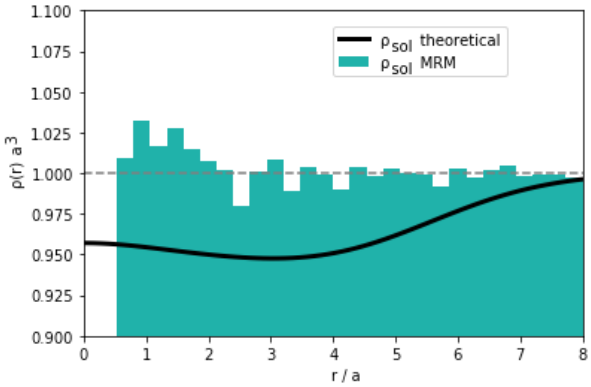
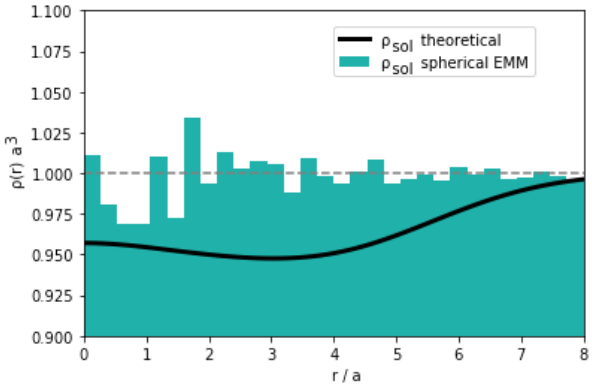


Figure 5.3: The radial solvent density distribution in the PSC model (green histogram) compared to the theoretical function (black line). At small r there are more fluctuations due to the smaller volume elements dV.

If we also analyze the solvent density distribution for the monomer resolved model MRM, and the effective monomer model EMM, we can see that these two do not yield the desired distribution as given by the theoretical solvent density function (fig. 5.4).



(a) The radial solvent density distribution in the monomer resolved model (MRM) compared to the theoretical function.



(b) The radial solvent density distribution in the effective monomer model (EMM) compared to the theoretical function.

Figure 5.4: In both the monomer resolved model and the effective monomer model solvent particles are not distributed according to the theoretical density function $\rho_{sol}(r)$.

This is because in the MPCD model there is no excluded volume for solvent particles [49]. They are point-like and experience no potential, neither in relation to each other nor in relation to the solute, similar to an ideal gas. As a result solvent particles penetrate the monomers freely and the density is uniform. This is a slightly unrealistic but intrinsic feature of the MPCD model. By introducing the Monte-Carlo acceptance probability for solvent movement in the PSC model this feature is explicitly rectified.

5.3 Penetrable soft colloid diffusion

To evaluate the diffusion coefficient for the PSC, a set of 10 independent simulations over a time span of $\Delta t = 5000\tau$ was considered. Like before this was done by applying a linear fit to the mean squared-displacement. Also for this coarse-grained model no significant equilibration time is necessary, since there is no change in the colloid's internal structure and the temperature adjusts quickly. The averaged curve from all 10 simulations is depicted by figure 5.5.

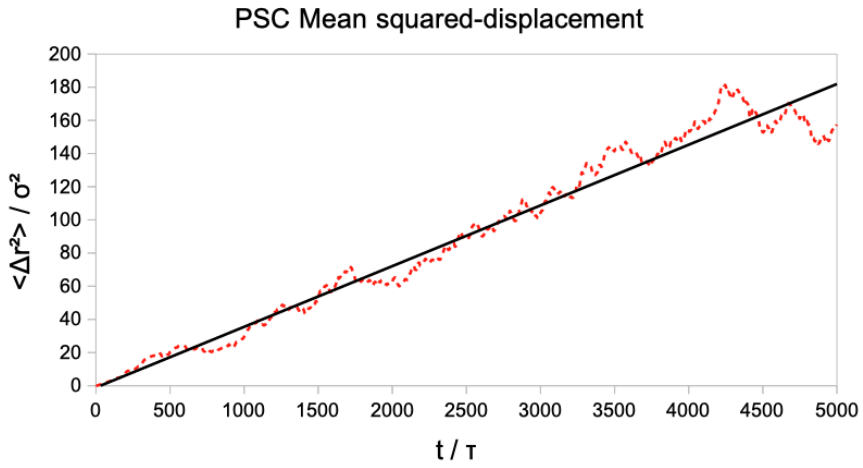


Figure 5.5: The mean squared-displacement of the penetrable soft colloid over time (red dashed line) and its linear fit (black continuous line).

For the finite-size diffusion coefficient of the PSC we obtained:

$$D_l/D_0 = 0.00610 \pm 0.00141.$$

This results in a hydrodynamic radius and hence an infinite-system-size diffusion coefficient of

$$R_H/a = 1.13 \quad \text{and} \quad D/D_0 = 0.00656.$$

The result for the diffusion coefficient in this model is significantly higher than the result from the MRM. In the research of Ruiz-Franco et al. (2019) [42] the PSC model delivers satisfactory results for star polymers, but overestimates the diffusion coefficient of linear chains significantly. The simulation results here show that for ring polymers the PSC model does not accurately estimate the diffusion coefficient which means that both the EMM and the PSC did not deliver the desired results for ring polymers. The discrepancy between the MRM and the PSC will be addressed via the introduction of modified acceptance rules in chapter 7.

Chapter 6

The ellipsoidal PSC

Since neither the EMM nor the PSC deliver satisfying results for the diffusion physics of ring polymers, in this chapter we introduce another coarse-grained model. This is a refined version of the PSC discussed in the previous chapter that does not limit the colloid's shape to a sphere. Rather it allows the colloid to be of an ellipsoidal shape with any set of 3 length parameters $\{a, b, c\}$ that represent its extension along the 3 principal axes.

It could be shown that the frictional coefficients ζ of ellipsoidal particles were larger than those of spheres with the same volume [50]. Additionally, the frictional coefficient is higher for prolate ellipsoids than for oblate ones with equal volume. Since the diffusion coefficient of a particle is indirectly proportional to its frictional coefficient (eq. 1.5), that means that the diffusion coefficient of ellipsoids is smaller than that of spheres with the same volume. This indicates that expanding the PSC model such that it supports not only a spherical colloid but also an arbitrary ellipsoidal colloid could close the gap between the diffusion discrepancies. Especially considering that the gyration tensor suggests an ellipsoid with an aspect ratio of $\lambda_1/\lambda_3 \approx 3$ for ring polymers, this can hardly be considered an approximately spherical shape. The ellipsoid proportions are illustrated by figure 6.1.

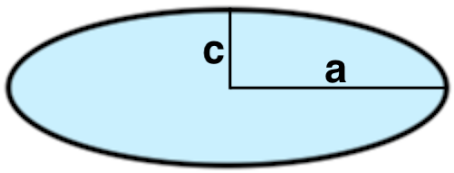


Figure 6.1: Accurate proportions of the ellipsoid as calculated by the gyration tensor root eigenvalues. The longest axis (a) is shown against the shortest axis (c).

6.1 The ellipsoidal PSC algorithm

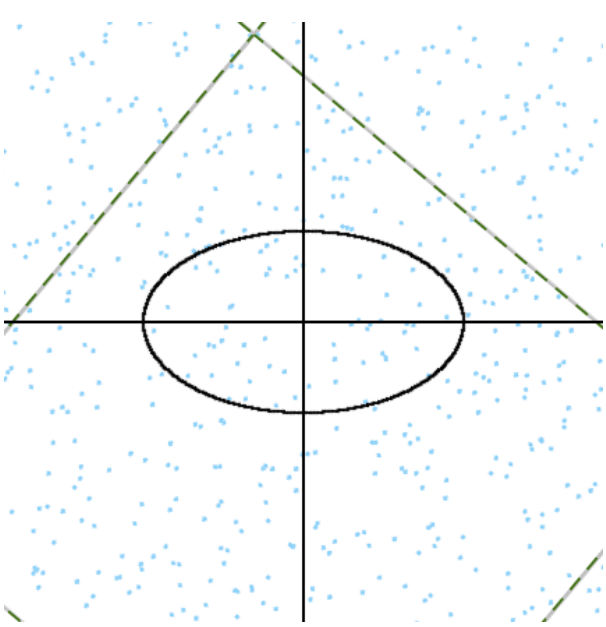
The algorithm for the ellipsoidal PSC is based on the spherical PSC in chapter 5, but is altered to accommodate an ellipsoidal shape of the colloid.

6.1.1 Initialisation of the ellipsoidal PSC

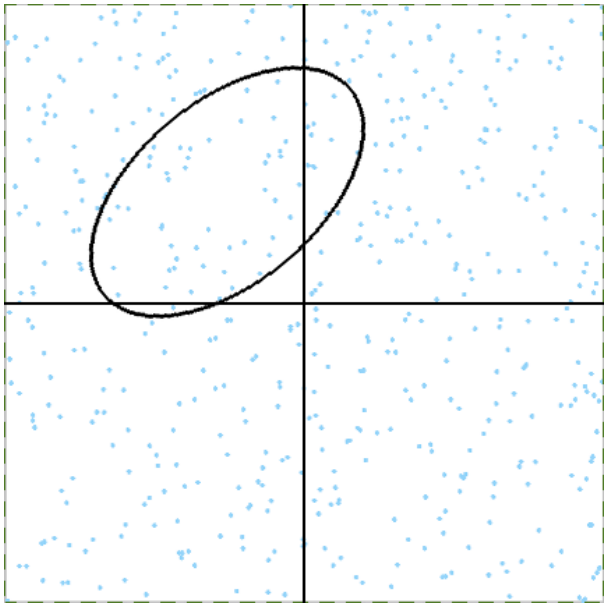
Just like the spherical PSC, also the ellipsoidal penetrable soft colloid needs to be initialized with a position vector $\vec{R}(t)$, a center of mass velocity vector $\vec{V}(t)$, and an angular velocity vector $\vec{\Omega}(t)$. It is convenient to initialize the position at the simulation box center, and both velocities at 0.

Due to the colloid's anisotropy, for this model we also need to track its orientation relative to the simulation box. Hence we define a rotation matrix $\hat{Q}(t)$ that we use to track the colloid's relative orientation over time. The colloid is initially oriented along the simulation box, meaning that its principal axes are the Cartesian axes x, y, and z, and as such, its rotation matrix equates the unit matrix: $\hat{Q}(0) = \mathbb{1}$. $\hat{Q}(t)$ is updated every step using the colloid's angular velocity $\vec{\Omega}(t)$.

The ellipsoidal PSC also needs a defined moment of inertia $\hat{I}$, and total mass $M = NM_{mon}$. Contrary to a sphere an ellipsoid is not isotropic. That means its moment of inertia $\hat{I}$ cannot be reduced to a scalar, but has to be represented by a tensor. However, there is still mirror symmetry along all 3 principal planes which means that all of the moment of inertia tensor's off diagonal elements vanish in the colloid's frame of reference. To show this, we first define the colloid's frame of reference as the frame whose axes are aligned with the colloid's principal axes, its origin is located at the colloid's center of mass, and it moves and rotates together with the colloid. In other words, for all times t in this frame we have $\vec{R}'(t) = 0$, $\vec{V}'(t) = 0$, $\vec{\Omega}'(t) = 0$, and $\hat{Q}'(t) = \mathbb{1}$. Contrary to that, we have the observer's frame of reference, which is the objective frame that has been used so far. Figure 6.2 visualizes the 2 defined frames of reference.



(a) In the colloid's frame of reference, the PSC sits still at the coordinate origin.



(b) In the observer's frame of reference the colloid moves and rotates following the equations of motion.

Figure 6.2: Comparison between the colloid's frame of reference and the observer's frame of reference. The small dots represent solvent particles.

In the colloid's frame of reference we parametrize the ellipsoidal PSC the same way that ellipsoids are usually parametrized with the parameters $\{a, b, c\}$ and resulting volume V:

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1, \quad V=\frac{4\pi}{3}abc. \tag{6.1}$$

To facilitate the calculations involving the ellipsoidal PSC we use stretched spherical coordinates (r',θ',ϕ') instead of Cartesian coordinates (x,y,z):

$$\begin{aligned}\frac{x}{a}&=r'\ \sin\theta'\ \cos\phi'\\ \frac{y}{b}&=r'\ \sin\theta'\ \sin\phi'\\ \frac{z}{c}&=r'\ \cos\theta'\end{aligned} \tag{6.2}$$

The stretched spherical coordinates work exactly like spherical coordinates after stretching or contracting the Cartesian axes such that the ellipsoid ends up as a sphere with radius $r'=1$.

Previously, we evaluated the monomer density function $\rho_{mon}(r)$ by treating the monomer resolved ring polymer as a radial symmetric object and averaging over all directions and a number of samples. For this model we need a monomer density function that accommodates the ellipsoidal shape. One way would be to evaluate the density function by averaging over the ellipsoid octants instead of all directions in order to keep the symmetry but not lose angular information. In that case the monomer density function would depend on all 3 parameters r, ϕ, and θ. In order to obtain a smooth and precise function this would require a large sample size, and all integrals would be 3-dimensional. Another way is to use the radial symmetric density function evaluated previously, but define it for the stretched spherical coordinates. That means we transform $r \to r'$ and then normalize again the monomer density $\rho_{mon}(r')$:

$$r \to r' = \frac{r}{R_{colloid}}, \tag{6.3}$$

$$\iiint\limits_{V'} \rho_{mon}(r')dV' = N, \quad \text{with } dV' = abc\ r'^2 \sin\theta\ dr'd\theta d\phi. \tag{6.4}$$

We can hereby calculate all elements of the moment of inertia tensor $\hat{I}$ [45]. The off diagonal elements vanish. To see this one only needs to use the transformed coordinates from (6.2) in

$$I_{\mu\nu} = I_{\nu\mu} = -\iiint\limits_{V'} x_\mu x_\nu\ \rho_{mon}(r')\ dV' = 0 \quad \text{for } \mu \neq \nu. \tag{6.5}$$

That leaves us with only diagonal elements for the moment of inertia tensor $\hat{I}$.

$$\hat{I} = \begin{pmatrix} I_{aa} & 0 & 0 \\ 0 & I_{bb} & 0 \\ 0 & 0 & I_{cc} \end{pmatrix}, \tag{6.6}$$

$$I_{aa} = \frac{4\pi}{3}\ abc\ (b^2 + c^2)\ M_{mon} \int_0^1 \rho_{mon}(r') r'^4\ dr', \tag{6.7}$$

$$I_{bb} = \frac{4\pi}{3}\ abc\ (a^2 + c^2)\ M_{mon} \int_0^1 \rho_{mon}(r') r'^4\ dr', \tag{6.8}$$

$$I_{cc} = \frac{4\pi}{3}\ abc\ (a^2 + b^2)\ M_{mon} \int_0^1 \rho_{mon}(r') r'^4\ dr'. \tag{6.9}$$

So far we have not yet defined values for the ellipsoid parameters $\{a, b, c\}$. By transitioning from a sphere to an ellipsoid it makes sense to impose the condition that the colloid's total volume does not change. This is a direct conclusion if the colloid's mass and overall density are supposed stay the same.

$$V_{sphere} = \frac{4\pi}{3} R_{colloid}^3 \stackrel{!}{=} V_{ellipsoid} = \frac{4\pi}{3} abc. \tag{6.10}$$

The ellipsoid parameters $\{a, b, c\}$ or ellipsoid semi-axis lengths have a direct relationship with the root-eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$. Precisely, the root-eigenvalues are equal to the semi-axis lengths of the ellipsoid of inertia [46]. In order to match a physical ellipsoid they can simply be scaled with a factor. Using this and (6.10) gives us the values for the ellipsoid parameters

$$a : b : c = \lambda_1 : \lambda_2 : \lambda_3, \tag{6.11}$$

$$a = \lambda_1 \cdot \frac{R_{colloid}}{\sqrt[3]{\lambda_1 \lambda_2 \lambda_3}}, \quad b = \lambda_2 \cdot \frac{R_{colloid}}{\sqrt[3]{\lambda_1 \lambda_2 \lambda_3}}, \quad c = \lambda_3 \cdot \frac{R_{colloid}}{\sqrt[3]{\lambda_1 \lambda_2 \lambda_3}}. \tag{6.12}$$

The ellipsoid parameters for the ring polymer in this work result in: $a = 15.14$, $b = 10.30$, and $c = 5.45$.

6.1.2 Ellipsoidal PSC movement and coupling to the solvent

In principal the ellipsoidal PSC is simulated in the same way as the spherical PSC. Each time step consists of 3 steps: PSC ballistic movement, solvent particles ballistic movement with colloid interaction, and the multi-particle collision among the solvent particles. In detail however, these steps have to be adjusted to work for the ellipsoidal shape of the colloid.

The ballistic movement of the PSC in this model encompasses both translation and rotation. The translational ballistic movement is the same as for the spherical PSC according to (5.4).

Additionally in this model we also track colloid orientation $\hat{Q}(t)$. The new colloid orientation matrix $\hat{Q}(t+h)$ is calculated with the help of a rotation matrix $\hat{D}(\vec{\mathcal{R}}(t), \alpha(t))$

$$\hat{Q}(t+h) = \hat{D}(\vec{\mathcal{R}}(t), \alpha(t)) \cdot \hat{Q}(t). \tag{6.13}$$

The similarity to the rotation matrix $\hat{D}(\vec{\mathcal{R}}, \alpha)$ that is used for the multi-particle collision of the solvent particles (eq. 2.4 - 2.7) is no coincidence. In fact, the rotation of solvent velocities and colloid orientation are mathematically equivalent. The rotation matrix $\hat{D}(\vec{\mathcal{R}}(t), \alpha(t))$ is defined through the rotation angle $\alpha(t)$ and the rotation axis $\vec{\mathcal{R}}(t)$, both of which can be calculated from the angular velocity $\vec{\Omega}(t)$ following

$$\alpha(t) = |\vec{\Omega}(t)| \cdot h, \tag{6.14}$$

$$\vec{\mathcal{R}}(t) = \frac{\vec{\Omega}(t)}{|\vec{\Omega}(t)|}. \tag{6.15}$$

The rotation matrix $\hat{D}(\vec{\mathcal{R}}(t), \alpha(t))$ is then calculated according to (2.6). Alternatively, we can define the matrix form $\hat{\mathcal{R}}$ of the unit rotation vector $\vec{\mathcal{R}}$. Then the calculation of the rotation matrix can be summarized to

$$\hat{D}(\hat{\mathcal{R}}(t), \alpha(t)) = \mathbb{1} + \sin\alpha(t) \cdot \hat{\mathcal{R}}(t) + (1 - \cos\alpha(t)) \cdot \hat{\mathcal{R}}^2(t), \tag{6.16}$$

$$\text{with } \hat{\mathcal{R}}(t) = \begin{pmatrix} 0 & -\mathcal{R}_z(t) & \mathcal{R}_y(t) \\ \mathcal{R}_z(t) & 0 & -\mathcal{R}_x(t) \\ -\mathcal{R}_y(t) & \mathcal{R}_x(t) & 0 \end{pmatrix}, \tag{6.17}$$

which is known as Argyris' form of the Euler–Rodrigues formula [51]. Translation and rotation are independent of each other and can therefore be performed in any order.

The ballistic movement of the solvent particles (2.1) is a trial move, that will be performed if accepted. In case of rejection, the solvent particle is scattered by the ellipsoidal PSC. This works analogously to the spherical PSC, but now using stretched spherical coordinates. That means solvent particle initial and trial end position vectors have to be transformed to the colloid's frame of reference first. This is done by translating and then rotating the position vector. Then the solvent particle's stretched relative position r', which is the radial distance from the stretched spherical coordinates (r', ϕ', θ'), is calculated. To obtain the stretched vectors we apply the stretching matrix $\hat{S}$

$$\hat{S} = \begin{pmatrix} \frac{1}{a} & 0 & 0 \\ 0 & \frac{1}{b} & 0 \\ 0 & 0 & \frac{1}{c} \end{pmatrix}, \tag{6.18}$$

$$\vec{r'}_1 = \begin{pmatrix} x_1/a \\ y_1/b \\ z_1/c \end{pmatrix} = \hat{S} \cdot \hat{Q}^T(t+h) \cdot \left(\vec{r}_i(t) - \vec{R}(t+h) \right), \tag{6.19}$$

$$\vec{r'}_2 = \begin{pmatrix} x_2/a \\ y_2/b \\ z_2/c \end{pmatrix} = \hat{S} \cdot \hat{Q}^T(t+h) \cdot \left(\vec{r}_i(t+h) - \vec{R}(t+h) \right). \tag{6.20}$$

Here we used the property of rotation matrices that the inverse is the transpose $\hat{Q}^{-1} = \hat{Q}^T$. The stretched relative positions $r'_1 = |\vec{r'}_1|$ and $r'_2 = |\vec{r'}_2|$ are the magnitudes of the stretched relative position vectors $\vec{r'}_1$ and $\vec{r'}_2$ respectively. To avoid confusion the notation is chosen such that primed variables are always stretched variables in this section.

If either one of the solvent particle's initial stretched relative position r'_1 or trial end stretched relative position r'_2 are inside the colloid ($r'_1 \leq 1$ or $r'_2 \leq 1$) the trial move could be rejected. The trial move is accepted with a probability $p_{acc}(r'_1 \to r'_2)$ according to (5.5) and (5.6) as defined for the spherical PSC, but using the stretched positions instead.

In case that the trial move is rejected, the solvent particle is scattered by the PSC at the scattering point $\vec{s}$. Dependent on whether the trial move would have been fully or only partially inside the ellipsoidal colloid, the particles are scattered at the halfway point of that movement or at the colloid's surface respectively. This follows the same principle as defined for the spherical colloid, but the scattering point has to be re-defined to accommodate the ellipsoidal surface

$$s' = |\hat{S} \cdot \vec{s}| = \begin{cases} 1 & \text{if } r'_1 < 1 < r'_2 \text{ or } r'_2 < 1 < r'_1 \\ |\hat{S} \cdot \frac{\vec{r}_1 + \vec{r}_2}{2}| & \text{if } r'_1, r'_2 < 1 \end{cases}. \tag{6.21}$$

The scattering point vector is again calculated by imposing the condition for its stretched magnitude according to (6.21) and solving for constant k

$$\vec{s} = (1-k) \cdot \vec{r}_1 + k \cdot \vec{r}_2, \quad k \in (0,1), \tag{6.22}$$

$$|\vec{s'}| = |\hat{S} \cdot \vec{s}| = |(1-k) \cdot \hat{S} \cdot \vec{r}_1 + k \cdot \hat{S} \cdot \vec{r}_2| = |(1-k) \cdot \vec{r'}_1 + k \cdot \vec{r'}_2| \stackrel{!}{=} s', \tag{6.23}$$

$$\vec{\Delta r'} = \vec{r'}_2 - \vec{r'}_1. \tag{6.24}$$

For case 2, where scattering happens at the halfway point, we get $k = 1/2$. In case 1, where scattering happens at the surface, eq. 6.23 needs to be solved for $s' = 1$. Out of the 2 solutions that will present for k, only one fulfills the necessary condition $k \in (0,1)$

$$k_{1,2} = \frac{-\vec{r'}_1 \cdot \vec{\Delta r'} \pm \sqrt{(\vec{r'}_1 \cdot \vec{\Delta r'})^2 - \vec{\Delta r'}^2 (\vec{r'}_1^2 - 1)}}{\vec{\Delta r'}^2}. \tag{6.25}$$

The calculation of the normal unit vector $\hat{e}_n$ changes for the ellipsoidal PSC as well. It is not simply parallel to the scattering vector anymore (see fig. 6.3).

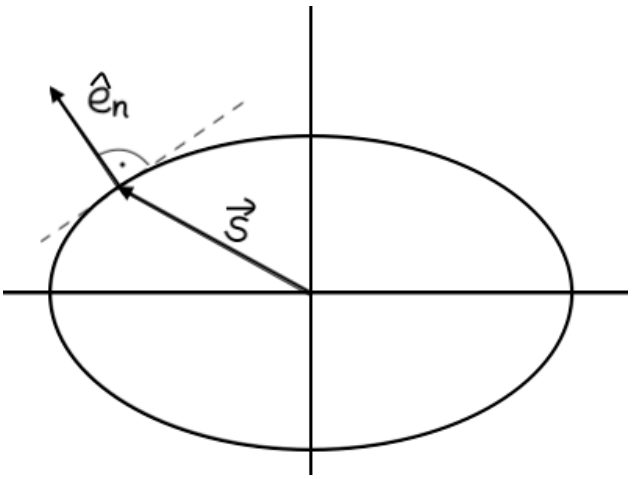


Figure 6.3: For the ellipsoid the normal unit vector $\hat{e}_n$ at the scattering point (x_s, y_s, z_s) is generally not parallel to the scattering vector $\vec{s} = (x_s, y_s, z_s)$.

The scattering point (x_s, y_s, z_s) lies on the surface of the ellipsoid (or one of its sub-ellipsoids inside) with the surface equation

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = R_s^2 = \text{const.} \tag{6.26}$$

If this is written in the form $f(x, y, z) = 0$ then the gradient of $f(x, y, z)$ defines the set of normal vectors $\hat{n}$

$$f(x, y, z) = \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - R_s^2 = 0, \tag{6.27}$$

$$\hat{n} = \text{grad } f(x, y, z) = \begin{pmatrix} 2x/a^2 \\ 2y/b^2 \\ 2z/c^2 \end{pmatrix}. \tag{6.28}$$

The normal unit vector $\hat{e}_n$ for (x_s, y_s, z_s) is then calculated by selecting the normal vector at the scattering point $\hat{n}_s$ and normalizing it

$$\hat{e}_n = \frac{\hat{n}_s}{|\hat{n}_s|} = \frac{1}{|\hat{n}_s|} \cdot \begin{pmatrix} 2x_s/a^2 \\ 2y_s/b^2 \\ 2z_s/c^2 \end{pmatrix}. \tag{6.29}$$

We can easily see that the normal unit vector $\hat{e}_n$ would be parallel to the scattering vector $\vec{s}$ for a sphere (with $a = b = c$). Additionally, also for the ellipsoidal PSC the normal unit vector should always point outwards. The tangential unit vector $\hat{e}_t$ is chosen randomly under the condition that it is orthogonal to the normal unit vector: $\hat{e}_t \perp \hat{e}_n$. The calculations of the new solvent

velocity (eq. 5.12 - 5.14) and the new colloid center of mass velocity (eq. 5.19) do not have to be adapted in another way for the ellipsoidal PSC. The new colloid angular velocity needs to be adapted using the inverse moment of inertia tensor $\hat{I}^{-1}$

$$\hat{I}^{-1} = \begin{pmatrix} \frac{1}{I_{aa}} & 0 & 0 \\ 0 & \frac{1}{I_{bb}} & 0 \\ 0 & 0 & \frac{1}{I_{cc}} \end{pmatrix}, \tag{6.30}$$

$$\vec{\Omega}(t+h) = \vec{\Omega}(t) + m\hat{I}^{-1} \cdot \vec{s}_i \times [\vec{v}_i(t) - \vec{v}_i(t+h)]\,. \tag{6.31}$$

The alternative solvent displacement following (5.21) and the ballistic movement following (2.1) are unaltered for the ellipsoidal PSC and they are calculated in the observer's frame of reference.

The third step, which is the multi-particle collision among solvent particles is performed the same way as if there was no solute present, and in the observer's frame of reference obviously. This step is exactly the same as for the spherical PSC.

6.2 Ellipsoidal PSC diffusion

For the evaluation of the diffusion coefficient in the ellipsoidal PSC model a set of 20 independent simulations were evaluated over a time span of $\Delta t = 5000\tau$. The mean squared-displacement curve of the 20 averaged simulations is shown by figure 6.4.

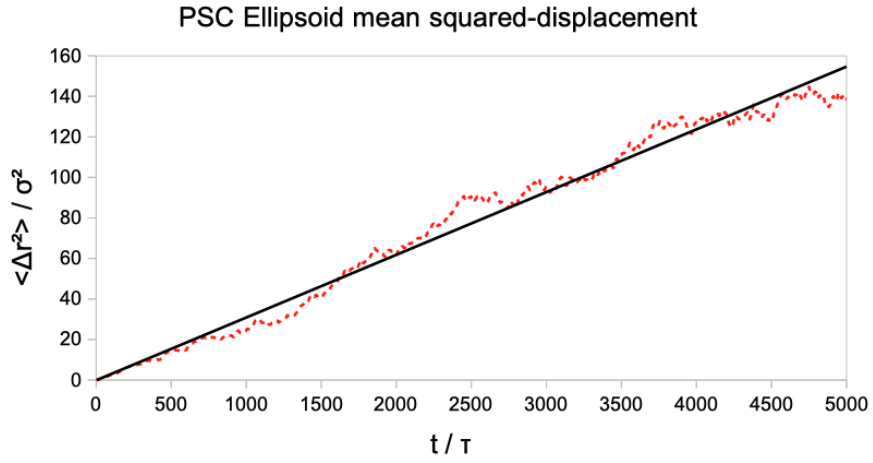


Figure 6.4: The mean squared-displacement of the ellipsoidal PSC model over time (red dashed line) vs. its linear fit (black continuous line).

For the finite-size diffusion coefficient of the ellipsoidal PSC the following value was obtained:

$$D_l/D_0 = 0.00516 \pm 0.00110.$$

This results in a hydrodynamic radius and hence an infinite-system-size diffusion coefficient of

$$R_H/a = 1.32 \quad \text{and} \quad D/D_0 = 0.00562.$$

While the ellipsoidal shape of the PSC did minimally lower the diffusion coefficient as compared to the spherical PSC, the effect is not sufficient when we compare the results to the monomer resolved model. Despite the high uncertainty, the diffusion coefficient is still significantly overestimated. As a result, the more complicated ellipsoidal shape is not effective enough to be considered a suitable alternative coarse-grained model, so we need a more drastic modification which will be introduced in the following chapter.

Chapter 7

The spherical PSC with increased interaction factor

Unfortunately the PSC's shape change from a sphere to a general ellipsoid did not deliver satisfying results for the diffusion coefficient. At that point, if the theoretical diffusion coefficient is known, and it can be acquired by a simulation following the MRM algorithm (as described in chapter 3), it is possible to adjust the acceptance probability $p_{acc}(r_1 \to r_2)$ of the PSC algorithm in such a way that the desired diffusion coefficient is enforced.

The principle behind this method is that Monte-Carlo acceptance probabilities can be modified using a factor f, and within limit this can be done while still following detailed balance. Upholding detailed balance is critical in order to obtain correct converging results for the local solvent density. The factor f only lowers the rate of this convergence. In the case of the PSC model it allows an adjustable increase of the interaction rate, which in turn decreases the diffusion coefficient. On the downside, this method does not work if the diffusion coefficient obtained by the PSC model needs to be increased to match a desired value, but fortunately this has not yet been the case. For typical Monte-Carlo applications this method is not useful, as most of them aim to produce an accordingly distributed sample without focusing on the changes that the system undergoes to get there. The increased interaction factor f presented in this chapter would only slow this process down, such that it takes longer to arrive at an equilibrated sample.

In this chapter the adaption is applied to and evaluated for the spherical PSC. It can be applied to the ellipsoidal PSC as well if colloid shape is important for some reason, but the spherical PSC model is generally more simple and efficient to implement.

7.1 The algorithm of the PSC with factor

The algorithm of the PSC with factor is based on the normal PSC algorithm. The only change is in the calculation of the Monte-Carlo acceptance probability $p_{acc}(r_1 \to r_2)$ and consequences arising from that.

7.1.1 Differences between the normal PSC algorithm and the PSC-with-factor algorithm

The precise formula to calculate the acceptance probability for a specific algorithm can be chosen freely as long as it follows detailed balance (see eq. 5.17). In the Metropolis algorithm it is usually defined as [48]

$$p_{acc}(r_1 \to r_2) = \min\left\{1; \frac{\rho(r_2)}{\rho(r_1)}\right\} \tag{7.1}$$

if $\rho(r_i)$ describes the probability density to find state r_i and the generation probability is symmetrical. If we introduce a factor $0 < f \leq 1$, detailed balance would be upheld for

$$p_{acc}(r_1 \to r_2) = \min\left\{f; f \cdot \frac{\rho(r_2)}{\rho(r_1)}\right\} \tag{7.2}$$

as well. That means for $0 < f \leq 1$ the ensemble will converge to its equilibrium state. This process just takes longer for smaller f. The PSC algorithm does not use the usual Metropolis algorithm, but the acceptance probability is conditional. Tailored to the PSC algorithm it results in

$$p_{acc}(r_1 \to r_2) = \begin{cases} \min\left\{f; f \cdot \frac{\rho_{sol}(r_2)}{\rho_{sol}(r_1)}\right\}, & \text{if } r_2 \leq r_1 \\ \min\left\{f; f \cdot \frac{\rho_{sol}(r_1)}{\rho_{sol}(r_2)}\right\}, & \text{if } r_1 < r_2 \end{cases}. \tag{7.3}$$

In order to follow detailed balance in the end, also the orientation of the normal unit vector $\hat{e}_n$ needs to be adapted. The modified acceptance probability ensures more interaction on average, but the ratio of solvent particles that are reflected to those that are only deflected or pass unhindered has to be constant regardless of the factor f. The first of the 2 cases in (7.3) is straightforward, as it is the same as for the original Metropolis algorithm and therefore needs no adaption. The second of the 2 cases was added by the authors of the original PSC model [42] in order to increase the interaction rate and slow down diffusion [55]. To get the same density profile as in the original PSC model, some of the interacting solvent particles for which case 2 applies have to be reflected instead of deflected. In total there are 4 possibilities for $p_{acc}(r_1 \to r_2)$ and the choice of $\hat{e}_n$:

- $r_2 \leq r_1$ and $\frac{\rho_{sol}(r_2)}{\rho_{sol}(r_1)} \geq 1 \quad \Rightarrow \quad \hat{e}_n$ always points outwards ,
- $r_2 \leq r_1$ and $\frac{\rho_{sol}(r_2)}{\rho_{sol}(r_1)} < 1 \quad \Rightarrow \quad \hat{e}_n$ always points outwards ,
- $r_1 < r_2$ and $\frac{\rho_{sol}(r_1)}{\rho_{sol}(r_2)} \geq 1 \quad \Rightarrow \quad \hat{e}_n$ always points inwards ,
- $r_1 < r_2$ and $\frac{\rho_{sol}(r_1)}{\rho_{sol}(r_2)} < 1 \quad \Rightarrow \quad \hat{e}_n$ points inwards with probability $\frac{1-f}{1-f \cdot \frac{\rho_{sol}(r_1)}{\rho_{sol}(r_2)}}$.

In order to understand especially the fourth of these possibilities for the normal vector $\hat{e}_n$, it is instructive to look at the ratio of trial moves that are effectively accepted for penetration vs. escape. A precise explanation of what "effectively accepted" means in this context follows promptly. In order to get the desired solvent density profile, this ratio should be exactly the ratio between the solvent densities of their respective end positions.

First, we look at the areas for which

$$\rho_{sol}(r_i) > \rho_{sol}(r_j) \ , \text{ if } r_i < r_j, \tag{7.4}$$

which is usually true only for a small core area. In the original PSC, the acceptance probability for both events penetration ($r_{outer} \rightarrow r_{inner}$) and escape ($r_{inner} \rightarrow r_{outer}$) is $p_{acc} = 1$. The choice of a normal vector $\hat{e}_n$ is redundant in that case, and the ratio between effectively accepted trial moves for penetration vs. escape is 1. As a result, the solvent density follows a uniform distribution in the small core area, which is a small inaccuracy of the original model, but does not have a significant impact on the solvent density profile overall. In the PSC with factor, the acceptance probability for both events penetration and escape is $p_{acc} = f$. The choice for the normal vector $\hat{e}_n$ can therefore either be backwards for both cases, or forwards for both cases such that the ratio between effectively accepted trial moves for penetration vs. escape is 1 in this model as well. The choice was arbitrarily made such that $\hat{e}_n$ points backwards, meaning it points outwards for penetration (possibility 1) and inwards for escape trial moves (possibility 3).

For the major parts of the colloid however, the solvent density function is increasing with r

$$\rho_{sol}(r_i) < \rho_{sol}(r_j) \ , \text{ if } r_i < r_j. \tag{7.5}$$

In the original PSC, the acceptance probability for penetration trial moves ($r_{outer} \rightarrow r_{inner}$) in this area equals $p_{acc} = \rho_{sol}(r_{inner})/\rho_{sol}(r_{outer})$. The choice of the normal unit vector $\hat{e}_n$ is such that it always points outwards, which means that rejected particles are reflected backwards. Therefore the fraction of effectively accepted particles is the same as the acceptance probability. In the event of escape ($r_{inner} \rightarrow r_{outer}$), the acceptance probability is $p_{acc} = \rho_{sol}(r_{inner})/\rho_{sol}(r_{outer})$. However, since the normal unit vector $\hat{e}_n$ points outwards for this event too, rejected particles are not reflected backwards, but they are only slightly deflected instead, and on average their movement is unhindered. As a result, all escape trial moves are effectively accepted. That means the ratio between effectively accepted trial moves for penetration events vs. escape events is $\rho_{sol}(r_{inner})/\rho_{sol}(r_{outer})$ as required to obtain the target density profile $\rho_{sol}(r)$. In the PSC with factor, we are required to arrive at the same ratio of effectively accepted moves in order for the two models to be equivalent in their static properties. In this new model, the acceptance probability for penetration trial moves ($r_{outer} \rightarrow r_{inner}$) equals $p_{acc} = f \cdot \rho_{sol}(r_{inner})/\rho_{sol}(r_{outer})$. We choose the normal unit vector $\hat{e}_n$ to be directed outwards, such that the fraction of effectively accepted particles is the same as the acceptance probability (possibility 2). As a result, the fraction of effectively accepted particles for escape events has to be equal to f. The fraction of effectively accepted particles $p_{acc,eff}$ can be calculated using the probability with which the normal unit vector points backwards $p(\hat{e}_n$ points backwards$)$

$$p_{acc,eff} = 1 - p_{rej,eff} = 1 - (1 - p_{acc}) \cdot p(\hat{e}_n \text{ points backwards}). \tag{7.6}$$

The probability with which the normal unit vector points backwards has either only been 1 or 0 so far. In the PSC with factor, the acceptance probability for escape trial moves ($r_{inner} \to r_{outer}$) equals $p_{acc} = f \cdot \rho_{sol}(r_{inner})/\rho_{sol}(r_{outer})$. We require $p_{acc,eff} \stackrel{!}{=} f$. Using these, we can calculate the probability with which the normal unit vector should point backwards (possibility 4), which in the case of escape is inwards:

$$p(\hat{e}_n \text{ points inwards}) = \frac{1 - p_{acc,eff}}{(1 - p_{acc})} = \frac{1 - f}{(1 - f \cdot \frac{\rho_{sol}(r_{inner})}{\rho_{sol}(r_{outer})})} \tag{7.7}$$

This algorithm is rather complex but was chosen that specific way to facilitate comparison between the original PSC and the PSC with factor. If desired, the PSC with factor model can easily be adapted and simplified by using the Metropolis algorithm with factor (eq. 7.11) unconditionally everywhere and have the normal unit vector $\hat{e}_n$ always point backwards compared to the trial direction. Naturally, this would lead to a different numerical value for the factor.

7.1.2 Calculation of the increased interaction factor

Suppose that the diffusion coefficient obtained using the PSC model is D_{PSC} and the diffusion coefficient obtained using the monomer resolved model, which is also the theoretically desired value, is D_{theo}. Then the 2 differ by a factor f_D

$$f_D = \frac{D_{PSC}}{D_{theo}}. \tag{7.8}$$

The objective is to relate the factor used in the Monte-Carlo scheme f to the factor between the diffusion coefficients f_D. The finite-size diffusion coefficient D_l that results from the simulations is dependent on the solvent's viscosity (eq. 3.20). It shows that they are inversely proportional to one another. As discussed in section 2.3.3, the total viscosity $\nu = \nu_{kin} + \nu_{coll}$ is the sum of kinetic viscosity and collisional viscosity. While the kinetic viscosity is directly proportional to the time step h (eq. 2.28), the collisional viscosity is inversely proportional to it (eq. 2.29). For the solvents that are of interest to these kind of simulations, usually the collisional viscosity dominates over the kinetic viscosity $\nu_{kin} \ll \nu_{coll}$ and therefore, the kinetic viscosity may be neglected and the viscosity is approximately inversely proportional to the time step h. The time step h on the other hand is inversely proportional to the collision rate, and the collision rate is directly proportional to the expected Monte-Carlo rejection probability $p_{rej}(r_1 \to r_2) = 1 - p_{acc}(r_1 \to r_2)$. In short

$$D_l \propto \frac{1}{\nu} \propto h \propto \frac{1}{N_{collisions}/t} \propto \frac{1}{\langle p_{rej}(r_1 \to r_2) \rangle} \ . \tag{7.9}$$

As a result we can express the factor between the diffusion coefficients f_D in terms of the expected Monte-Carlo rejection probabilities

$$f_D = \frac{D_{PSC}}{D_{theo}} = \frac{\langle p_{rej}(r_1 \to r_2)_{theo} \rangle}{\langle p_{rej}(r_1 \to r_2)_{PSC} \rangle}. \tag{7.10}$$

The increased interaction factor f used in the Monte-Carlo scheme directly changes the acceptance probability

$$\langle p_{acc}(r_1 \to r_2)_{theo} \rangle = f \cdot \langle p_{acc}(r_1 \to r_2)_{PSC} \rangle \,. \tag{7.11}$$

By combining (7.10) and (7.11) we get the relationship between the two factors

$$f = \frac{1 - f_D \cdot \langle p_{rej}(r_1 \to r_2)_{PSC} \rangle}{1 - \langle p_{rej}(r_1 \to r_2)_{PSC} \rangle}. \tag{7.12}$$

The last unknown before this model can be implemented is the expected Monte-Carlo rejection probability $\langle p_{rej}(r_1 \to r_2) \rangle$. It can obviously be tracked empirically during a simulation using the original PSC model. This is convenient if the solute's diffusion coefficient with that model is not yet known and the simulation has to be performed anyway. However, if the solute's diffusion coefficient with the PSC model is already known, it would be practical if the expected rejection probability can be calculated theoretically.

To derive an analytical expression for $\langle p_{rej}(r_1 \to r_2) \rangle$ it is necessary to know the joint probability density $f(r_1, r_2)$ for r_1 and r_2. The expectation value can then be calculated by

$$\langle p_{rej}(r_1 \to r_2) \rangle = \frac{\int_0^{R_{colloid}} \int_0^{R_{colloid}} p_{rej}(r_1 \to r_2) \cdot f(r_1, r_2) \; dr_1 dr_2}{\int_0^{R_{colloid}} \int_0^{R_{colloid}} f(r_1, r_2) \; dr_1 dr_2}. \tag{7.13}$$

In equilibrium the initial positions r_1 are distributed following $\rho_{sol}(r)$ as provided by the Monte-Carlo scheme. The trial end position $r_2 = |\vec{r}_1 + h\vec{v}|$ is dependent on the initial position and its probability density is also a function of r_1. The variable that connects r_1 and r_2, which is $\Delta r = |h\vec{v}|$, on the other hand is independent of r_1 and follows only the Maxwell-Boltzmann distribution. The joint probability density $f(r_1, \Delta r) = f_1(r_1) \cdot f_\Delta(\Delta r)$ is therefore separable and simply a product of the individual probability densities

$$f_1(r_1) \; d^3 r_1 \propto \rho_{sol}(r_1) \; d^3 r_1 = \rho_{sol}(r_1) r_1^2 \sin\theta_1 \; dr_1 d\theta_1 d\phi_1, \tag{7.14}$$

$$\begin{aligned} f_\Delta(\Delta r) \; d^3(\Delta r) &= \sqrt{\frac{m}{2\pi k_B T h^2}} \cdot e^{-\frac{m}{2k_B T h^2}(\Delta r)^2} \; d^3(\Delta r) \\ &= \sqrt{\frac{m}{2\pi k_B T h^2}} \cdot e^{-\frac{m}{2k_B T h^2}(\Delta r)^2} \cdot (\Delta r)^2 \sin\theta_\Delta \; d(\Delta r) d\theta_\Delta d\phi_\Delta. \end{aligned} \tag{7.15}$$

Both quantities were expressed using spherical coordinates, and for Δr they are oriented as such that θ_Δ denotes the angle between the vectors $\vec{r}_1$ and $\vec{\Delta r}$ (see figure 7.1). The rejection

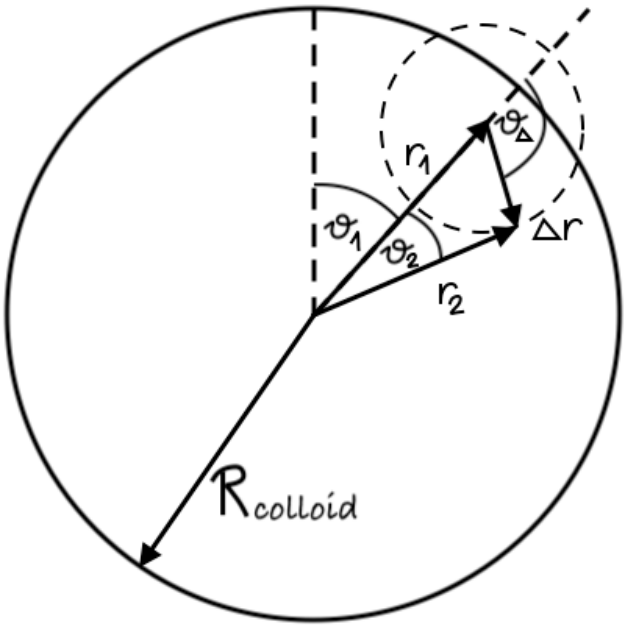


Figure 7.1: Visualization of the relationship between $\vec{r}_1$, $\vec{r}_2$, and $\vec{\Delta r}$.

probability $p_{rej}(r_1 \rightarrow r_2)$ uses the quantities r_1 and r_2. From geometrical considerations it is possible derive the transformation $(\Delta r, \theta_\Delta, \phi_\Delta) \rightarrow (r_2, \theta_2, \phi_2)$ with

$$(\Delta r)^2 = (\vec{r}_2 - \vec{r}_1)^2 = r_2^2 - 2r_2 r_1 \cos\theta_2 + r_1^2, \tag{7.16}$$

$$\cos\theta_\Delta = \frac{r_2 \cos\theta_2 - r_1}{\Delta r(r_2, \theta_2)}, \tag{7.17}$$

$$\phi_\Delta = \phi_2, \tag{7.18}$$

$$J = \left| \frac{\partial(\Delta r, \theta_\Delta, \phi_\Delta)}{\partial(r_2, \theta_2, \phi_2)} \right| = \frac{r_2^2 \sin\theta_2}{(\Delta r)^2 \sin\theta_\Delta}, \tag{7.19}$$

$$d(\Delta r) d\theta_\Delta d\phi_\Delta = J \; dr_2 d\theta_2 d\phi_2. \tag{7.20}$$

where θ_2 denotes the angle between the vectors $\vec{r}_1$ and $\vec{r}_2$. The Jacobi determinant J helps to transform the volume element. The joint probability density for r_1 and r_2 can now be calculated by transforming and subsequent integration over the other variables.

$$
\begin{aligned}
f(r_1, r_2)\ dr_1 dr_2 &\propto \int_0^{2\pi}\int_0^{\pi}\int_0^{2\pi}\int_0^{\pi} \rho_{sol}(r_1) r_1^2 \sin\theta_1\ e^{-\frac{m(r_1^2 - 2r_1 r_2 \cos\theta_2 + r_2^2)}{2k_B T h^2}} r_2^2 \sin\theta_2\ d\theta_1 d\phi_1 d\theta_2 d\phi_2 dr_1 dr_2 \\
&\propto \rho_{sol}(r_1) r_1^2 r_2^2\ e^{-\frac{m(r_1^2+r_2^2)}{2k_B T h^2}} \int_0^{\pi} e^{\frac{m r_1 r_2 \cos\theta_2}{k_B T h^2}} \sin\theta_2\ d\theta_2 dr_1 dr_2 \\
&\propto \rho_{sol}(r_1) r_1 r_2\ e^{-\frac{m(r_1^2+r_2^2)}{2k_B T h^2}} \sinh\left(\frac{m r_1 r_2}{k_B T h^2}\right) dr_1 dr_2.
\end{aligned}
\tag{7.21}
$$

For a general solvent density function $\rho_{sol}(r)$ the expression in eq. 7.13 can be calculated numerically, although it is wise to rearrange (7.21) for that by joining the hyperbolic sine and exponential function, as otherwise the hyperbolic sine becomes too large quickly. For the ring polymer in this work both the numerically integrated and the simulation based expected rejection probability $\langle p_{rej}(r_1 \to r_2)_{PSC} \rangle$ deliver the same result which is

$$\langle p_{rej}(r_1 \to r_2)_{PSC} \rangle = 0.00055.$$

That results in a Monte-Carlo increased interaction factor of

$$f = 0.99934.$$

The analytical expression only considers trial moves that happen fully inside the colloid and neglects all trial moves that occur when either one of the positions r_1 or r_2 is outside. Because of the asymptotic solvent density $\rho_{sol}(r)$ Those represent only a small fraction of both rejected trial moves and total trial moves however. In fact, most trial move rejections and therefore collisions happen at $r_1 = 6.24$ ($R_{colloid} = 9.48$). Figure 7.2 illustrates the density weighted rejection probability by trial end position r_2 for different fixed values of r_1.

7.2 Solvent density distribution of the PSC with factor

In the previous section (sec. 7.1) the PSC model with increased interaction factor was derived from theoretical principles. In order to verify the model's correctness the solvent density profile was tracked via a set of 20 simulations. Also for this model the equilibration process is very quick such that the equilibrium density profile is achieved after 1000 time steps, or $\Delta t = 100\tau$ in our system. The resulting density profile is depicted by figure 7.3 below and we can see that the target density profile is achieved.

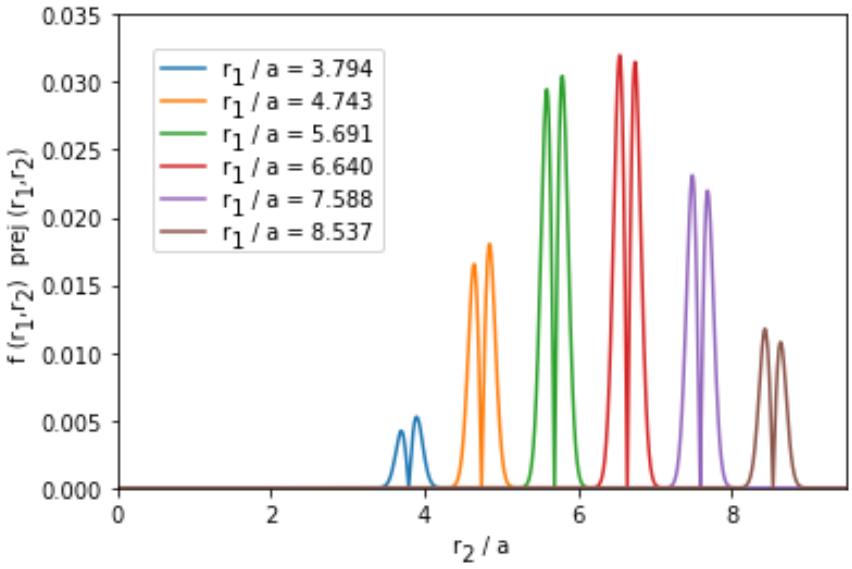


Figure 7.2: The density weighted rejection probability by trial end position r_2 for different values of r_1. As can be seen only a small fraction of rejections occur at the colloid surface. The dips at $r_2 \approx r_1$ appear because these trial moves are not rejected due to similar solvent densities. For $r_2 \ll r_1$ and $r_2 \gg r_1$ trial moves rarely get generated because the distance is too far for typical velocities.

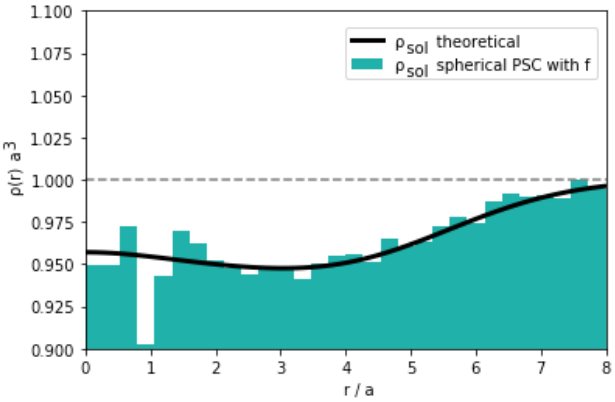


Figure 7.3: The radial solvent density distribution in the PSC model with factor follows the theoretical density function $\rho_{sol}(r)$ as required.

7.3 Diffusion of the PSC with factor

In the PSC model with factor the diffusion coefficient was evaluated using a set of 20 independent simulations over a time span of $\Delta t = 5000\tau$. Figure 7.4 below shows the resulting graph of the mean squared-displacement over this time span averaged over all 20 simulations and its linear fit.

The finite-size diffusion coefficient obtained from the mean squared-displacement curve of the PSC with factor results in:

$$D_l/D_0 = 0.00307 \pm 0.00044.$$

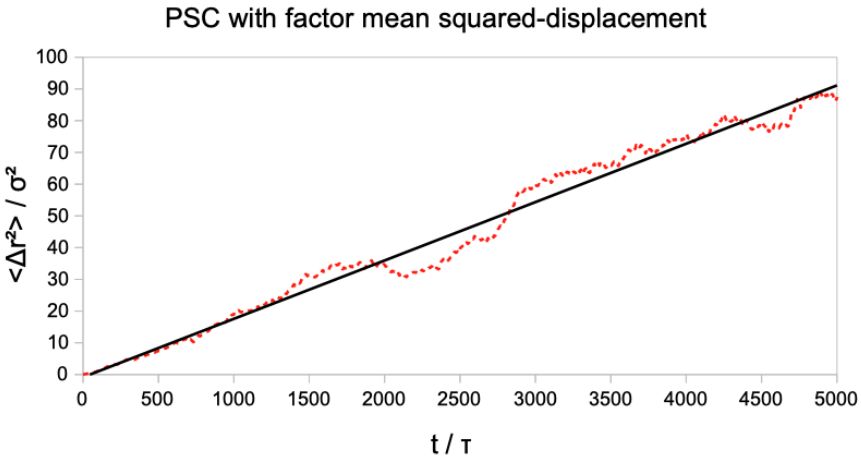


Figure 7.4: The mean squared-displacement of the PSC model with increased interaction factor over time (red dashed line) vs. its linear fit (black continuous line).

This delivers a hydrodynamic radius and hence an infinite-system-size diffusion coefficient of

$$R_H/a = 2.11 \quad \text{and} \quad D/D_0 = 0.00352.$$

The diffusion coefficient estimated by this model quite accurately reproduces the results from the monomer resolved model. Within uncertainty the results can be considered indistinguishable. Unfortunately the diffusion coefficient undergoes heavy fluctuations and it cannot be estimated very precisely with a set of only 10 or 20 simulations. It would be interesting whether this model performs satisfactory with a larger sample size as well, since then the uncertainty is reduced and the quality of this coarse-grained model can be assessed more accurately.

Nevertheless, the correct results for the density profile and diffusion coefficient suggest that the Monte Carlo scheme with increased interaction factor works well and provides a new flexible adaption to the PSC model proposed by Ruiz-Franco et al. (2019) [42].

7.3.1 Anisotropy of the diffusion

For this section the PSC model with factor was employed with an ellipsoidal shape as described in chapter 6.

Without external forces, in an objective frame the diffusion of an anisotropic object becomes isotropic in the long run [52]. This is not the case for a frame that is fixed to the solute's principal axes. Using MD simulation, it can be shown that an anisotropic solute's translational and rotational diffusion are different in magnitude along the object's principal axes [53]. This can also be observed experimentally in tangible reality [54]. Diffusion along the long axis of a solute is expected to be greater than the diffusion in a perpendicular direction.

Both the monomer resolved polymer and the ellipsoidal PSC are anisotropic objects that should exert this behaviour as a result. To analyze the anisotropy of the translational diffusion, the solutes orientation has to be quantified and its displacement vector transformed from the observer frame

to the colloid frame for each time step. For the ellipsoid this is simple since the ellipsoid is a rigid body and its orientation is tracked using the colloid orientation matrix $\hat{Q}(t)$. The displacement vector in the observer frame is calculated from the position vectors by

$$\delta\vec{R}(t) = \vec{R}(t) - \vec{R}(t-h), \quad \delta\vec{R}(0) = 0. \tag{7.22}$$

To calculate the displacement vector in the colloid frame, one has to transform its components using the orientation matrix $\hat{Q}(t)$ which represents the orthonormal basis of the colloid frame

$$\delta\vec{R}'(t) = \hat{Q}^T(t-h) \cdot \delta\vec{R}(t). \tag{7.23}$$

In practice, choosing $\hat{Q}^T(t-h)$ or $\hat{Q}^T(t)$ or a normalized average has little effect on the results because the orientation changes slowly. The total displacement in the colloid frame is then simply the cumulative sum

$$\Delta\vec{R}'(t) = \sum_{n=0}^{t/h} \delta\vec{R}'(nh), \tag{7.24}$$

and its components describe the displacement along the colloid's 3 principal axes instead of the 3 observer coordinates. At the end, the component-wise diffusion coefficient is calculated with the component-wise mean squared-displacement and follows the relationship [54]

$$\left\langle \Delta\vec{R}_i^2(t) \right\rangle = 2D_i \cdot t, \quad D = \frac{D_1 + D_2 + D_3}{3}. \tag{7.25}$$

For the monomer resolved polymer the orientation matrix $\hat{Q}(t)$ has to be constructed from the monomers' individual positions each time step. This is done by calculating the root-eigenvalues $\{\lambda_j\}$ and their respective eigenvectors $\{\hat{e}_j\}$ from the gyration tensor $G_{\alpha\beta}$. The orientation matrix is then built up from the sorted eigenvectors such that their matching root-eigenvalues are sorted descendingly as done previously:

$$\hat{Q}(t) = (\hat{e}_1\ \hat{e}_2\ \hat{e}_3.) \tag{7.26}$$

The problem here is however, that the monomers move individually and as a result the polymer exhibits significant shape changes over time. These lead to sudden jumps in the orientation matrix and a more irregular rotational behaviour for the MRM in general (see figure 7.5).

These shape changes can be elastic such that within short time they are reversed, or they can be long-lasting. Despite that, the eigenvectors were sorted in descending order of the root-eigenvalues, just as if the shape change was a normal rotation. This means that any anisotropic effects that occur in the polymer diffusion may be underestimated due to a possible delay in momentum orientation. However, the eigenvectors were multiplied by -1 in case of a jump that is more than $\pi/2$, since these do not have any physical meaning and the eigenvectors are simply coincidentally oriented in either direction when the orientation matrix is calculated each step.

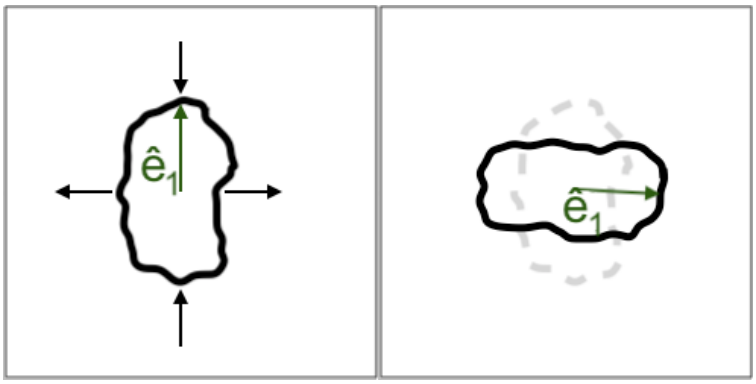
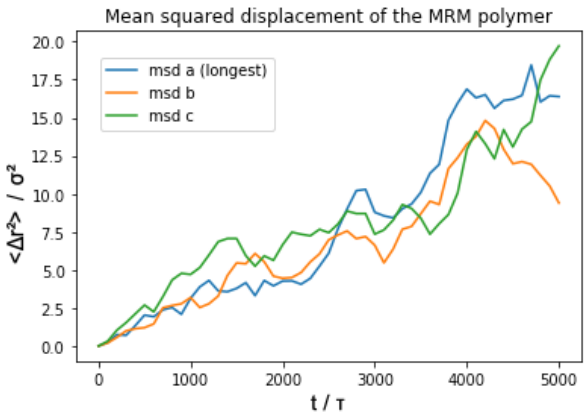


Figure 7.5: The monomer resolved polymer changes its shape. This causes jumps in the eigenvector $\{\hat{e}_j\}$ orientation.

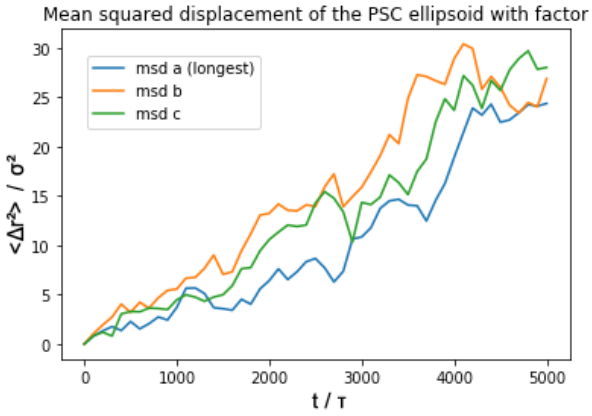
Figure 7.6 shows the results for the mean squared-displacement of the MRM polymer and the PSC ellipsoid with factor. Contrary to expectations, diffusion seems to be completely isotropic despite the solutes' anisotropic shapes. The value for each (finite-size) diffusion coefficient and its standard error are as follows:

	MRM polymer	ellipsoidal PSC with factor
$D_{l,1}/D_0$ (longest axis)	0.00186 ± 0.00065	0.00254 ± 0.00098
$D_{l,2}/D_0$ (medium axis)	0.00130 ± 0.00051	0.00297 ± 0.00182
$D_{l,3}/D_0$ (shortest axis)	0.00141 ± 0.00062	0.00301 ± 0.00107
D_l/D_0	0.00152 ± 0.00034	0.00284 ± 0.00071

Table 7.1: Component-wise diffusion coefficients and total diffusion in the colloid frame.



(a) The component-wise mean squared-displacement of the MRM polymer in the colloid frame. Diffusion seems to be isotropic.



(b) The component-wise mean squared-displacement of the ellipsoidal PSC with factor in the colloid frame. The colloid qualitatively replicates the isotropic behaviour regarding the component-wise diffusion.

Figure 7.6: Component-wise mean squared-displacement of the MRM polymer and the ellipsoidal PSC with factor.

If we compare the total diffusion in the colloid frame with the total diffusion in the system frame, it becomes apparent that for the PSC with factor they are the same within uncertainty. However,

the total diffusion coefficient for the MRM polymer in the colloid frame is significantly lower than in the system frame. Further, in the colloid frame the diffusion coefficients of the two models do not match well anymore. This is contradictory to the results in the system frame, where diffusion of the two models is indistinguishable within standard error. Both of these observations could be attributed to the shape changes that the MRM polymer undergoes, which lead to jumps in the eigenvector orientation. These eigenvector jumps are sudden which means that local momentum correlations are lost frequently in the colloid frame and therefore diffusion is slowed down.

The unexpected isotropic diffusive behaviour may be due to the fact that the colloid is penetrable. That way the amount of interacting solvent particles is not relative to the local surface area but the local volume. On the other hand the standard errors are rather large and may dominate over the anisotropy that would be too small to be visible in this case. Nevertheless, it is an interesting result that deserves deeper investigation in the future.

Chapter 8

Conclusions and Outlook

The mean squared-displacement curves of all employed models are summarized and compared by figure 8.1. Compared to the monomer resolved model (MRM) only the PSC model with factor exhibits a similar diffusion behaviour for ring polymers. In all other models the diffusion coefficient is either significantly underestimated (EMM) or overestimated (spherical and ellipsoidal PSC).

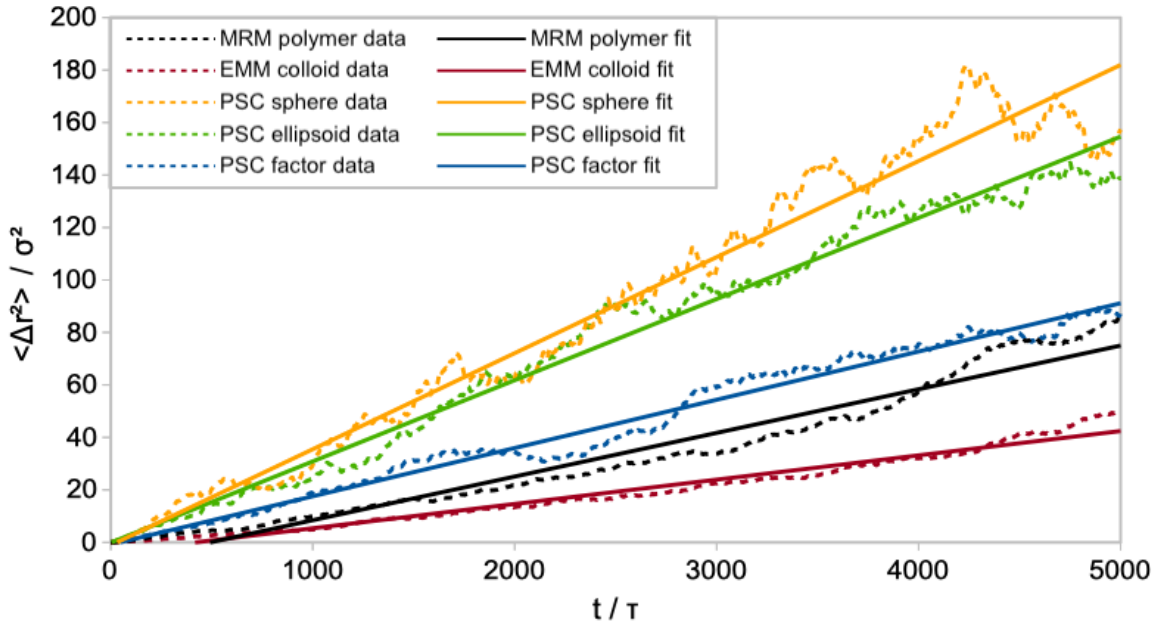


Figure 8.1: The mean squared-displacement (dashed lines) and their linear fits (continuous lines) of all presented models.

Adapting the PSC model by enabling not only spherically shaped colloids but general ellipsoids did not significantly improve the diffusion behaviour for ring polymers, even though they have rather large aspect ratios $a/c \approx 3$. On the other hand, introducing a factor to the PSC's Monte-Carlo acceptance probability, thus freely increasing interaction, provides the desired results and slows down the diffusion coefficient such that it matches the MRM. An advantage of this method is that it offers a lot of freedom since the factor can be chosen freely between $0 < f \leq 1$. As a result, the diffusion coefficient can be decreased arbitrarily within range. The greatest disadvantage of this method is however, that the target diffusion coefficient has to be known.

Unfortunately, the results for the diffusion coefficient are within a relatively large uncertainty. That is due to the fact that only one instance of the solute is simulated in the box and averaging has to be done using several independent simulations, which is computationally expensive. A bigger sample size of e.g. $n = 1000$ instead of $n = 10$ or $n = 20$ could reduce the uncertainty by a factor of ~ 10. To save computational resources, it could be considered to use shorter runs with less time steps each. This would allow for a better assessment of the differences between the presented models.

The reason why the PSC model as proposed by the original authors does not deliver consistent results could be, that it relies on the solvent density gradient to control the probability of interaction. Compared to that, interaction in the monomer resolved model is more resembling to a dependence on absolute monomer density. The reason why this model works for star polymers but not for linear chains and rings is probably that star polymers have a steeper monomer density function, and hence a higher solvent density gradient overall. As a result they possess higher interaction probabilities and consequently a slower diffusion.

By analyzing the component-wise diffusion coefficient in the colloid frame, it turned out that it is isotropic within uncertainty. It was expected that the diffusion parallel to the longest axis would exceed the diffusion along the other perpendicular axes, however this was not the case. The reason for that may be that the uncertainty is larger than the anisotropy which is why the effects are not visible, but it may also be that due to their penetrability, the soft colloids behave different to hard colloids in that regard. In any case this would be an interesting subject to investigate more deeply.

Bibliography

[1] M. Rubinstein, and R.H. Colby, *Polymer Physics* (Oxford University Press, 2003).

[2] M. Doi, and S.F. Edwards, *The Theory of Polymer Dynamics* (Oxford University Press, 1986).

[3] I. Teraoka, *Polymer Solutions: An Introduction to Physical Properties* (John Wiley & Sons: New York, 2002).

[4] H. Yamakawa, *Modern Theory of Polymer Solutions* (Harper & Row: New York, 1971).

[5] P.E. Schavemaker, A.J. Boersma, and B. Poolman, "How Important Is Protein Diffusion in Prokaryotes?", *Front. Mol. Biosci.* **5,** 93 (2018).

[6] Z.A. Dehaghani, I. Chubak, C.N. Likos, and M.R. Ejtehadi, "Effects of topological constraints on linked ring polymers in solvents of varying quality", *Soft Matter* **16,** 3029 (2020).

[7] U.D. Schiller, T. Krüger, and O. Henrich, "Mesoscopic Modelling and Simulation of Soft Matter", *Soft Matter* **14,** 9 (2018).

[8] S.H. Lee and R. Kapral, "Mesoscopic description of solvent effects on polymer dynamics", *J. Chem. Phys.* **124,** 214901 (2006).

[9] A. Malevanets and R. Kapral, "Mesoscopic model for solvent dynamics", *J. Chem. Phys.* **110,** 8605 (1999).

[10] K. Mussawisade, M. Ripoll, R.G. Winkler, et al., "Dynamics of polymers in a particle-based mesoscopic solvent", *J. Chem. Phys.* **123,** 144905 (2005).

[11] Y. Yang, V. Marceau, and G. Gompper, "Swarm behavior of self-propelled rods and swimming flagella", *Phys. Rev. E* **82,** 031904 (2010).

[12] A. Zöttl and H. Stark, "Hydrodynamics determines collective motion and phase behavior of active colloids in quasi-two-dimensional confinement", *Phys. Rev. Lett.* **112,** 118101 (2014).

[13] H. Noguchi and G. Gompper "Shape transitions of fluid vesicles and red blood cells in capillary flows", *PNAS* **102,** 14159 (2005).

[14] A. Shakeri, K.-W. Lee, and T. Pöschel "Limitation of stochastic rotation dynamics to represent hydrodynamic interaction between colloidal particles", *Phys. Fluids* **30,** 013603 (2018).

[15] I.O. Götze, H. Noguchi, and G. Gompper "Relevance of angular momentum conservation in mesoscale hydrodynamics simulations", *Phys. Rev. E* **76,** 046705 (2007).

[16] T. Ihle, E. Tüzel, and D.M. Kroll "Consistent particle-based algorithm with a nonideal equation of state", *Europhys. Lett.* **73,** 664 (2006).

[17] H. Noguchi and G. Gompper "Transport coefficients of off-lattice mesoscale-hydrodynamics simulation techniques", *Phys. Rev. E* **78,** 016706 (2008).

[18] S. Mühlbauer, S. Strobl, and T. Pöschel "Isotropic stochastic rotation dynamics", *Phys. Rev. Fluids* **2,** 124204 (2017).

[19] G. Gompper, T. Ihle, D.M. Kroll, and R.G. Winkler, "Multi-particle collision dynamics: A particle-based mesoscale simulation approach to the hydrodynamics of complex fluids", in *Advanced Computer Simulation Approaches for Soft Matter Sciences III,* edited by C. Holm and K. Kremer (Springer-Verlag Berlin Heidelberg, 2008), pp. 1–87.

[20] R.G. Winkler, "Flow Simulations with Multiparticle Collision Dynamics", in *Hierarchical Methods for Dynamics in Complex Molecular Systems - Lecture Notes,* edited by J. Grotendorst, G. Sutmann, G. Gompper, and D. Marx (Forschungszentrum Jülich GmbH, 2012), pp. 417-443.

[21] R. Wang, Z. Zhang, L. Li, and Z. Zhu "Preference Parameters for the Calculation of Thermal Conductivity by Multiparticle Collision Dynamics", *Entropy* **23,** 1325 (2021).

[22] M. Ripoll, K. Mussawisade, R.G. Winkler, and G. Gompper "Dynamic regimes of fluids simulated by multiparticle-collision dynamics", *Phys. Rev. E* **72,** 016701 (2005).

[23] A. Malevanets, and J.M. Yeomans "Dynamics of short polymer chains in solution", *Europhys. Lett.* **52,** 231 (2000).

[24] M. Ripoll, K. Mussawisade, R.G. Winkler and G. Gompper "Low-Reynolds-number hydrodynamics of complex fluids by multi-particle-collision dynamics", *Europhys. Lett.* **68,** 106 (2004).

[25] A. Malevanets, and R. Kapral "Solute molecular dynamics in a mesoscale solvent", *J. Chem. Phys.* **112,** 7260 (2000).

[26] I. Chubak, C.N. Likos, K. Kremer, and J. Smrek "Emergence of active topological glass through directed chain dynamics and nonequilibrium phase segregation", *Phys. Rev. Research* **2,** 043249 (2020).

[27] W.C. Swope, H.C. Andersen, P.H. Berens, and K.R. Wilson "A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters" *J. Chem. Phys.* **76,** 637 (1982).

[28] G. Bussi, D. Donadio, and M. Parrinello "Canonical sampling through velocity rescaling", *J. Chem. Phys.* **126,** 014101 (2007).

[29] V. Dahirel, X. Zhao, B. Couet, G. Batôt, and M. Jardat "Hydrodynamic interactions between solutes in multiparticle collision dynamics", *Phys. Rev. E* **98,** 053301 (2018).

[30] G. Grest, and K. Kremer "Molecular dynamics simulation for polymers in the presence of a heat bath", *Phys. Rev. A* **33,** 3628 (1986).

[31] G.A. Hegde, J. Chang, Y. Chen, and R. Khare "Conformation and diffusion behavior of ring polymers in solution: A comparison between molecular dynamics, multiparticle collision dynamics, and lattice Boltzmann simulations", *J. Chem. Phys.* **135,** 184901 (2011).

[32] M. Liebetreu, and C.N. Likos "Hydrodynamic inflation of ring polymers under shear", *Commun. Mater.* **1,** 4 (2020).

[33] J. Rudnick, and G. Gaspari, *Elements of the Random Walk: An introduction for Advanced Students and Researchers* (Cambridge University Press, 2004), pp. 130-134.

[34] A.F. Katsarou, A.J. Tsamopoulos, D.G. Tsalikis, and V.G. Mavrantzas "Dynamic Heterogeneity in Ring-Linear Polymer Blends" *Polymers* **12** 752 (2020).

[35] S. Huissmann, R. Blaak, and C.N. Likos "Star Polymers in Solvents of Varying Quality", *Macromolecules* **42** 2806 (2009).

[36] S. Redner "Distribution functions in the interior of polymer chains", *J. Phys. A: Math. Gen.* **13** 3525 (1980).

[37] J. Smiatek and F. Schmid "Mesoscopic Simulation Methods for Studying Flow and Transport in Electric Fields in Micro- and Nanochannels", in *Advances in Microfluidics,* edited by R.T. Kelly (InTech, 2012), pp. 97-126.

[38] S.P. Singh, C.-C. Huang, E. Westphal, G. Gompper, and R.G. Winkler "Hydrodynamic correlations and diffusion coefficient of star polymers in solution", *J. Chem. Phys.* **141,** 084901 (2014).

[39] V. Bloomfield, and B.H. Zimm "Viscosity, Sedimentation, et Cetera, of Ring and StraightChain Polymers in Dilute Solution", *J. Chem. Phys.* **44,** 315 (1966).

[40] S.M. Bhattacharjee, A. Giacometti, and A. Maritan "Flory theory for Polymers", *J. Phys.: Condens. Matter* **25,** 503101 (2013).

[41] K.-W. Hsiao, C.M. Schroeder, and C.E. Sing "Ring Polymer Dynamics Are Governed by a Coupling between Architecture and Hydrodynamic Interactions", *Macromolecules* **49,** 1961 (2016).

[42] J. Ruiz-Franco, D. Jaramillo-Cano, M. Camargo, C.N. Likos, and E. Zaccarelli "Multi-particle collision dynamics for a coarse-grained model of soft colloids", *J. Chem. Phys.* **151,** 074902 (2019).

[43] J.K.G. Dhont "An Introduction to Dynamics of Colloids", in *Studies in Interface Science* edited by D. Möbius and R. Miller (Elsevier Science B.V., 1996).

[44] D.S. Bolintineanu, G.S. Grest, J.B. Lechman, F. Pierce, S.J. Plimpton, and P.R. Schunk "Particle dynamics modeling methods for colloid suspensions", *Comp. Part. Mech.* **1,** 321 (2014).

[45] R.M. Dreizler, C.S. Lüdde *Theoretische Physik 1 - Theoretische Mechanik* (Springer-Verlag Berlin Heidelberg, 2008).

[46] E.J. Rawdon, "Physical Knot Theory: The Study of Sizes and Shapes of Polymers", in *Introductory Lectures on Knot Theory* edited by L.H. Kauffman, S. Lambropoulou, S. Jablan, and J.H. Przytycki (World Scientific, 2012), pp. 457-481.

[47] M. Hecht, J. Harting, T. Ihle, and H.J. Herrmann "Simulation of claylike colloids", *Phys. Rev. E* **72,** 011408 (2005).

[48] M.H. Kalos, and P.A. Whitlock *Monte Carlo Methods* (Wiley-VCH Verlag: Weinheim, 2008), pp. 35-76.

[49] M.P. Allen, and D.J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, 2017).

[50] J.T. Edward "Molecular Volumes and the Stokes-Einstein Equation", *J. Chem. Educ.* **47,** 261 (1970).

[51] R.M. Brannon, *Rotation, Reflection, and Frame Changes - Orthogonal tensors in computational engineering mechanics* (IOP Publishing, 2018).

[52] R. Vasanthi, S. Ravichandran, and B. Bagchi "Needlelike motion of prolate ellipsoids in the sea of spheres", *J. Chem. Phys.* **114,** 7989 (2001).

[53] Z.-T. Zhang, X. Zhao, and B.-Y. Cao "Diffusion Tensors of Arbitrary-Shaped Nanoparticles in Fluid by Molecular Dynamics Simulation", *Sci Rep* **9,** 18943 (2019).

[54] Y. Han, A.M. Alsayed, M. Nobili, J. Zhang, T.C. Lubensky, and A.G. Yodh "Brownian Motion of an Ellipsoid", *Science* **314,** 626 (2006).

[55] C.N. Likos, personal communication (2021).