Computational Characterisation of Mesogenes Containing Carbohydrate- and Cyclitol-Based Building Blocks

Inaugural-Dissertation

zur
Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Universität zu Köln
vorgelegt von

SEBASTIAN BREUERS

aus Erkelenz

Erstgutachter:	Priv.-Doz. Dr. Dirk Blunk
Zweitgutachter:	Prof. Dr. Hans-Günther Schmalz

Tag der mündlichen Prüfung:
Freitag, den 12.04.2013

Kurzzusammenfassung

Im Rahmen dieser Arbeit werden verschiedene Methoden vorgestellt, in Vorarbeiten experimentell charakterisierte, kohlenhydratbasierte Flüssigkristalle mit Hilfe von Computersimulationen zu untersuchen.

Flüssigkristalle spielen in vielen Anwendungsfällen wie der Displaytechnologie, den Mikro- und Nanowissenschaften, der Pharmakologie und der Kosmetik eine wichtige Rolle. Zuckerbasierte Mesogene zeigen neben ihrer Umweltverträglichkeit und strukturellen Vielfalt interessante Mesophasen. Die Zusammenhänge zwischen molekularen Eigenschaften und der Phasenstruktur sind bei Zuckermesogenen noch nicht vollständig erschlossen und eine detaillierte Vorhersage von Struktur-Eigenschafts-Beziehungen ist oftmals noch nicht möglich. Insbesondere der Einfluss von Wasserstoffbrückennetzwerken ist in diesen Verbindungen von Bedeutung und bis heute noch nicht, insbesondere in thermotropen Flüssigkristallen, mit Hilfe von Simulationsmethoden untersucht worden. In dieser Arbeit werden zwei Modellsysteme vorgestellt und Studien präsentiert, experimentelle Daten zu diesen Modellsystemen durch Simulationsrechnungen zu reproduzieren, den Einfluss von polaren Strukturmotiven zu verstehen und die Wirkung von enantiomerenreinen und racemischen Flüssigkristallsystemen auf die Phasenbildung aufzuklären.

Abstract

This work summarises different approaches to investigate carbohydrate-based liquid crystals by means of computer simulations. Liquid crystals play an important role in various use cases like display technology, nanosciences, pharmacology and cosmetics. Sugar-based mesogens exhibit, beside their biodegradability and structural diversity, interesting mesophases. The relations between molecular properties and supramolecular phase structures in sugar mesogens are not yet sufficiently understood and predictions of structure-property-relationships are often not yet possible. Especially, the influence of hydrogen bond networks is of major importance and has not been investigated for thermotropic liquid crystals with computational chemistry methods. Two model systems will be introduced and experiments and results presented to reproduce experimentally derived data. The influence of polar moieties will be explained and the effect of enantiopure as well as racemic mixtures of liquid crytal systems on the phase formation elucidated.

“The problem is reality, as usual.”

(*Mattia Felice Palermo,* **2012**)

“The problem is the expectation that an ideal can become reality.”

(*Isabella Miglioli*, **2012**)

Acknowledgement

This thesis would not have been possible without the work and existence of others. I am grateful in a variety of ways to their contribution and support.

I am very thankful to Priv.-Doz. Dirk Blunk for the interesting topic, for giving me the freedom to exploit this scientific field, and for motivating me to go abroad and find inspiration in a differently experienced working group. I want to thank Prof. Hans-Günther Schmalz for giving me the possibility to work under these excellent conditions in his working group. Additionally, I want to thank Prof. Ulrich Lang who equally provided to my financial support.

Without the computational resources of the Regional Computing Center Cologne (RRZK) this work would not have reached its current dimension. I am grateful to Viktor Achter and his team, especially Stefan Borowski, who both have been there equally with advice and recommendation on using my most important working tool, the CHEOPS cluster.

The work in the AK Blunk and AK Schmalz provided me with a lot of interesting personalities who made the last three years an enjoyable time and a period of learning, how different the world can be percepted and approached. I want to thank the members of the AK Schmalz, AK Blunk, and AK Giernoth, especially Shute Ye, Kai Wirz, Matthias Winter, Carina Hagemann, Andreas Bröhl, Andrea Kuchenbuch, Richard Meisenheimer, and Hanna Sebode, who all often gave me a big laugh in periods of scientific frustration.

A wonderful and beautiful experience was my stay in Bologna, at the University of Bologna and at the Computing Center *CINECA* funded by the HPCEuropa 2 project. I am very grateful for the possibility to go there. Without the great support of Prof. Claudio Zannoni and his willingness to host me in his group this would not have been possible. I want to thank especially Isabella Miglioli, Chiara Latini, Giulia Cantelli, and Giada Cotugno for their warm and welcoming, integrative and embracing attitude. I am specially thankful to Mattia Felice Palermo, Luca Muccioli, Gabriele D'Avigno, and Otello Maria Roscioni for their eagerness and readiness to explain and share scientifically demanding topics in a way that I could digest. The whole working group of Prof. Zannoni, in particular Domenico Summa, Antonio Pizzirusso, Lara Querciagrossa, Erica Benini, Matteo Ricci, Silvia Orlandi, and Roberto Berardi provided me with a pleasant working environment that I enjoyed a lot. All of the aforementioned made my several stays in Bologna a wonderful and enriching experience, both scientifically as well as personally.

Having a life beside university gave me often strength to cope with scientific obstacles and frustration. I am very grateful for my friends who gave and give support in good as well as critical times. Laura Ferreira Gonzalez became a special friend over the last eleven years. I thank her for being always there for me, whenever I need her. I thank my chosen family and dear friends Sandra Meid, Karin Komšić-Buchmann, Jutta Schütte, Mariko Ponczeck, Yvonne Bothur, Isabella Miglioli, Michael Hons, Torsten Leuchtenberg, Magnus Anschütz, and Jens Schulze, who created cheering moments and spent encouragement in less enjoyable times.

I want to dearly thank my parents, Karl-Heinz und Ursula Breuers, who gave me the possibility to live the life I live. I also want to thank my chosen grandmother Brigitte Gottwald, for her warm and embracing attentiveness. And I am very grateful to and for my brother, Frederique Breuers, who is in many, many situations my "go-to guy".

Contents

I. Introduction 1

1. Motivation 2

II. Theory 5

2. Liquid Crystals 6
    2.1. Thermotropic Liquid Crystals . . . 7
        2.1.1. Nematics . . . 7
        2.1.2. Smectics . . . 8
        2.1.3. Further Phase Types . . . 11
    2.2. Lyotropic Liquid Crystals . . . 11
        2.2.1. Amphiphilicity . . . 11
        2.2.2. Micelles . . . 12
        2.2.3. Phase Types . . . 13

3. Carbohydrate- and Cyclitol-Based LCs 15
    3.1. Carbohydrate-Derived Liquid Crystals . . . 15
    3.2. Inositol-Derived Liquid Crystals . . . 16

4. Simulation of Liquid Crystals 19
    4.1. Lattice Models . . . 20
    4.2. Coarse Grained Models . . . 20
        4.2.1. Hard Objects . . . 21
        4.2.2. Soft Objects . . . 22
        4.2.3. Multi-Site Representations . . . 22
    4.3. Atomistic Simulations . . . 24

5. Molecular Dynamics 27
    5.1. Integrating *Newton*'s Equations of Motion . . . 27
    5.2. Force Fields . . . 28
        5.2.1. Bonded Interactions . . . 28
        5.2.2. Non-Bonded Interactions . . . 28

5.2.3. Optimised Potentials for Liquid Simulations 29
5.2.4. Atomtypes 33
5.2.5. *United Atom* vs. *All Atom* 33
5.3. Simulation Parameters 33
5.3.1. Periodic Boundary Conditions 34
5.3.2. Temperature Coupling 36
5.3.3. Pressure Coupling 38

6. Observables 40
6.1. Hydrogen Bonds 40
6.2. Diffusion Coefficient 41
6.3. Order Parameters 45
6.3.1. Orientational Order Parameter 45
6.3.2. Positional Order Parameter 46
6.4. Correlation Functions 50

III. Results 53

7. Ressources and Applications 54
7.1. Hardware 54
7.2. Software 54

8. Bulk investigations of G_1C_{12} 56
8.1. Simulation Setup 56
8.2. Cooling I 57
8.2.1. Orientational Order 57
8.2.2. Self-diffusion 58
8.3. Cooling II 59
8.3.1. Orientational Order 60
8.3.2. Positional Order 60
8.3.3. Layerspacing 61
8.3.4. Lateral Self-diffusion 61
8.4. Summary 64

9. Cooling of UA-$C_{12}E_3I_1$ and UA-$E_3I_1C_{12}$ in a bulk 65
9.1. Simulation Setup 65

9.2. Formation of Order 66
 9.2.1. Orientational Order 68
 9.2.2. Spatial Correlation Function 70
9.3. Self-diffusion 71
9.4. Hydrogen Bonds 73
9.5. Summary 75

10. Heating UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$ 77
10.1. Thin Films 77
10.2. Simulation Setup 79
10.3. Orientational Order Parameter 80
10.4. Positional Order Parameter 80
10.5. Self-diffusion 81
10.6. Summary 83

11. Racemic and Enantiopure Samples 85
11.1. Simulation Setup 87
11.2. $C_{12}E_3$AA-I_1 and *rac*-$C_{12}E_3$AA-I_1 88
 11.2.1. Orientational Order 88
 11.2.2. Positional Order 88
 11.2.3. Hydrogen bonds 89
 11.2.4. Diffusion 90
 11.2.5. Summary 92
11.3. E_3AA-I_1C_{12} and *rac*-E_3AA-I_1C_{12} 93
 11.3.1. Orientational Order 95
 11.3.2. Positional Order 95
 11.3.3. Diffusion 97
 11.3.4. Summary 97
11.4. $C_{12}E_3$AA-I_1 and E_3AA-I_1C_{12} 98
11.5. Summary 99

IV. Outlook 101

12. Summary 102

13. Future Work 106

V. Appendix **V-1**

A. Topologies **V-2**

A.1. UA-$C_{12}E_3I_1$ . . . V-2
A.2. $C_{12}E_3$AA-I_1 . . . V-6
A.3. UA-$E_3I_1C_{12}$ . . . V-12
A.4. E_3AA-I_1C_{12} . . . V-16
A.5. G_1C_{12} . . . V-23

B. Analysis Tool **V-28**

B.1. `g_order_tensor.c` . . . V-28
B.2. `gmx_order_tensor.c` . . . V-28
B.3. `getPositionalAndOrientationalOrder.sh` . . . V-53
B.4. `getHBondsIntra-Inter-everyTimeStep.sh` . . . V-55

List of Figures

1. Structure and texture of cholesteryl acetate. 6
2. Nematic phase. 8
3. Texture of a nematic LC. 8
4. Smectic phases. 10
5. Smectic phases – layer interdigitation. 10
6. Amphiphile. 11
7. Micelles. 12
8. Lyotropic phases. 14
9. Structure of *n*-dodecyl-β-D-glucopyranoside. 16
10. Structure of *myo*-inositol. 17
11. The structure of phytic acid or *myo*-inositol hexakisphosphate. . 17
12. Building block of $C_{12}E_3AA\text{-}I_1$ and $E_3AA\text{-}I_1C_{12}$. 18
13. Overview of LC simulation approaches. 19
14. Lattice model (2D). 20
15. Examples of hard objects. 21
16. Examples of multi-site, coarse grained mesogens. 23
17. Atomistic description of *n*-dodecyl-β-D-glucopyranoside. 24
18. Structure of nCB. 25
19. Harmonic potential. 30
20. *Fourier* series for dihedral angle. 30
21. *Lennard-Jones* potential. 31
22. *Coulomb* potential. 32
23. *All atom* and *united atom* description of a sugar. 34
24. Periodic Boundary Conditions. 35
25. Spacefilling simulation boxes. 36
26. Minimum Image Convention. 36
27. Hydrogen bond. 40
28. Hydrogen bond – Geometrical description. 41
29. *Fick*'s first law of diffusion. 42
30. Smectic order parameter. 47
31. Structure of *n*-dodecyl-β-D-glucopyranoside. 56
32. Orientational order of G_1C_{12} (low temperature). 58
33. Self-diffusion G_1C_{12} (low temperature). 59

34. Orientational order of G_1C_{12} (high temperature). 60
35. Positional order of G_1C_{12} (high temperature). 61
36. Layerspacing of G_1C_{12} (high temperature). 62
37. Self-diffusion of G_1C_{12} (high temperature). 63
38. G_1C_{12} isotropic to SmA transition. 64
39. Structures of the inositol model systems. 65
40. Cooling of 1024 molecules of UA-$C_{12}E_3I_1$. 67
41. Cooling of 1024 molecules of UA-$E_3I_1C_{12}$. 67
42. Orientational order of 1024 molecules of UA-$C_{12}E_3I_1$. 68
43. Orientation of 1024 molecules of UA-$C_{12}E_3I_1$ and of UA-$E_3I_1C_{12}$. 69
44. Orientational order of 1024 molecules of UA-$E_3I_1C_{12}$. 69
45. Expected orientational of cooling simulation. 70
46. Spatial auto-correlation functions of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$. 72
47. Self-diffusion of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$. 73
48. Intermolecular inositol-inositol hydrogen bonds of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$. 74
49. Inter- and intramolecular inositol-ethoxy hydrogen bonds of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$. 75
50. Percentage of intramolecular hydrogen bonds of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$. 76
51. Workflow: Thin films to bulk. 78
52. Starting configurations for melting simulations. 79
53. Orientational order of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$ (heating). . . 81
54. Positional order of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$ (heating). 82
55. Self-diffusion of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$ (heating). 82
56. Lateral diffusion of UA-$C_{12}E_3I_1$ (heating). 84
57. Starting configurations for racemic and enantiopure simulations. 86
58. Orientational order of $C_{12}E_3$AA-I_1 and *rac*-$C_{12}E_3$AA-I_1. 89
59. Positional order of $C_{12}E_3$AA-I_1 and *rac*-$C_{12}E_3$AA-I_1. 90
60. Layerspacing of $C_{12}E_3$AA-I_1 and *rac*-$C_{12}E_3$AA-I_1. 91
61. Hydrogen bonds of $C_{12}E_3$AA-I_1 and *rac*-$C_{12}E_3$AA-I_1. 92
62. Diffusion coefficient of $C_{12}E_3$AA-I_1 and *rac*-$C_{12}E_3$AA-I_1. 93
63. Lateral diffusion of $C_{12}E_3$AA-I_1 and *rac*-$C_{12}E_3$AA-I_1. 94
64. Evolution of positional order of E_3AA-I_1C_{12}. 95
65. Orientational order of E_3AA-I_1C_{12} and *rac*-E_3AA-I_1C_{12}. 96

66. Positional order of E_3AA-I_1C_{12} and *rac*-E_3AA-I_1C_{12}. 96
67. Diffusion coefficient of E_3AA-I_1C_{12} and *rac*-E_3AA-I_1C_{12}. 97
68. Number of hydrogen bonds of E_3AA-I_1C_{12} and *rac*-E_3AA-I_1C_{12}. . 98
69. Investigated model systems. 102
70. Orientational & positional order parameter of G_1C_{12}. 103
71. Orientational & positional order parameter of UA-$C_{12}E_3I_1$ and UA-$E_3I_1C_{12}$. 105
72. Atomtypes UA-$C_{12}E_3I_1$. V-2
73. Atomtypes $C_{12}E_3$AA-I_1. . . . . V-6
74. Atom numbers $C_{12}E_3$AA-I_1. . . . . V-6
75. Atomtypes UA-$E_3I_1C_{12}$. V-12
76. Atomtypes E_3AA-I_1C_{12}. V-16
77. Atom numbers E_3AA-I_1C_{12}. V-17
78. Atomtypes G_1C_{12}. V-23
79. Charges G_1C_{12}. V-23

List of Tables

1. Phase transition temperatures of n-dodecyl-β-D-glucopyranoside 16
2. Phase transition temperatures of $C_{12}E_3$AA-I_1 and E_3AA-I_1C_{12}. . 19
3. Atomtypes and non-bonded parameters of UA-$C_{12}E_3I_1$. V-2
4. Bond definitions of UA-$C_{12}E_3I_1$. V-3
5. Angle definitions of UA-$C_{12}E_3I_1$. V-3
6. Dihedral definitions of UA-$C_{12}E_3I_1$. V-4
6. Dihedral definitions of UA-$C_{12}E_3I_1$. V-5
7. Atomtypes and non-bonded parameters of $C_{12}E_3$AA-I_1. . . . . . V-6
8. Point charges for $C_{12}E_3$AA-I_1. V-7
9. Bond definitions of $C_{12}E_3$AA-I_1. . . . . V-8
10. Angle definitions of $C_{12}E_3$AA-I_1. . . . . V-9
11. Dihedral definitions of $C_{12}E_3$AA-I_1. . . . . V-10
11. Dihedral definitions of $C_{12}E_3$AA-I_1. . . . . V-11
12. Atomtypes and non-bonded parameters of UA-$E_3I_1C_{12}$. V-12
13. Bond definitions of UA-$E_3I_1C_{12}$. V-13
14. Angle definitions of UA-$E_3I_1C_{12}$. V-13
15. Dihedral definitions of UA-$E_3I_1C_{12}$. V-14
15. Dihedral definitions of UA-$E_3I_1C_{12}$. V-15
16. Atomtypes and non-bonded parameters of E_3AA-I_1C_{12}. V-16
17. Atom point charges for E_3AA-I_1C_{12}. V-17
18. Bond definitions of E_3AA-I_1C_{12}. V-18
19. Angle definitions of E_3AA-I_1C_{12}. V-19
20. Dihedral definitions of E_3AA-I_1C_{12}. V-20
20. Dihedral definitions of E_3AA-I_1C_{12}. V-21
20. Dihedral definitions of E_3AA-I_1C_{12}. V-22
21. Atomtypes and non-bonded parameters of G_1C_{12}. V-23
22. Bond definitions of G_1C_{12}. V-24
23. Angle definitions of G_1C_{12}. V-25
24. Dihedral definitions of G_1C_{12}. V-26
24. Dihedral definitions of G_1C_{12}. V-27

List of Symbols and Abbreviations

D	Diffusion coefficient
J	Diffusive flux
c	Concentration
x	Spatial component
2D	Two-Dimensional
$\mathbf{a}$	Acceleration
AA	*All Atom*
β_{ij}	Angle between molecules i and j
BP	Blue Phase
$C_{12}E_3$AA-I_1	Combined *all atom/united atom* OPLS description of 1-*O*-[2′-[2″-[2‴-(dodecyloxy)ethoxy]ethoxy]ethyl]-*myo*-inositol C_{12}: *UA* dodecyl substructure; E_3: *UA* triethylene glycol substructure; AA-I_1: *AA* *myo*-inositol substructure
CG	Coarse Grained model
CHelpG	CHarges from Electrostatic Potentials using a Grid based method
Col	Columnar
COM	Center Of Mass
C	Coulomb
δ	Variable smectic layer distance
DFT	Density Functional Theory
DNA	DeoxyriboNucleic Acid
$D_{\parallel}$	Lateral diffusion in a layer
$D_{\perp}$	Diffusion perpendicular to a layer
DSC	Differential Scanning Calorimetry
d_{Sm}	Smectic layer distance; layerspacing
E	Energy
e	$1.602 \cdot 10^{-19}$ C; electronic charge

E_3AA-I_1C_{12}	Combined *all atom/united atom* OPLS description of 1-*O*-dodecyl-4-O-[2′-[2″-[2‴-(hydroxy)ethoxy]ethoxy]ethyl]-*myo*-inositol E_3: *UA* triethylene glycol substructure; AA-I_1: *AA* *myo*-inositol substructure; C_{12}: *UA* dodecyl substructure
ϵ_0	$8.854 \cdot 10^{-12}\,\mathrm{F\,m^{-1}}$; vacuum permittivity
ϵ_{ij}	Potential well depth of *Lennard-Jones* potential
$\mathbf{F}$	Force
F	Farad
FF	Force Field
f_{ij}	Fudge factor for *Lennard-Jones* potential
fs	Femtosecond
g_0	Radial distribution function
g_1	Spatial correlation function with $P_1(\mathbf{u}_i \cdot \mathbf{u}_j) = \cos(\beta_{ij})$
g_2	Spatial correlation function with $P_2(\mathbf{u}_i \cdot \mathbf{u}_j) = \frac{3}{2}\cos^2(\beta_{ij}) - \frac{1}{2}$
G_1C_{12}	Combined *all atom/united atom* description of *n*-dodecyl-β-D-glucopyranoside G_1: *AA* β-D-glucopyranoside substructure C_{12}: *UA* dodecyl substructure
$H_{I/II}$	Hexagonal (normal/inverted)
HPC	High Performance Computing
$I_{I/II}$	Discontinuous cubic (normal/inverted)
IP6	Inositolhexakisphosphate
Iso	Isotropic
IUPAC	International Union of Pure and Applied Chemistry
K	Kelvin
k_r	Bond spring constant
k_θ	Bond angle spring constant
L_α	Lamellar
λ_+	Biggest eigenvalue of order tensor Q
λ_-	Smallest eigenvalue of the order tensor Q
λ_0	Middle eigenvalue of order tensor Q

λ_T	Rescale factor *Berendsen* thermostat
LC	Liquid Crystal
LINCS	LINear Constraint Solver; algorithm to constrain bond lengths and angles.
m	Mass
m	Meter
MC	Monte Carlo
MD	Molecular Dynamics
µs	Microsecond
M	Mesophase
MSD	Mean Square Displacement
$\mathbf{n}$	Director
nCB	4-*n*-alkyl-4′-cyanobiphenyl
N*	Chiral nematic (formerly: cholesteric)
nm	Nanometer
N	Nematic
O(n)	O-notation: linearly scaling algorithm
OPLS	Optimised Potentials for Liquid Simulations
P	Pressure
P_0	Reference pressure
PBC	Periodic Boundary Conditions
ϕ	Dihedral angle
$\phi_{1\text{–}4}$	Phase shift
P_L	Legendre polynomial
ps	Picosecond
p_ξ	Heat bath function of *Nosé-Hoover* algorithm
$Q_{\alpha\beta}$	Order tensor
r	Bond length
r_0	Equilibrium bond length
ρ_0	Average density
σ_{ij}	Potential distance of *Lennard-Jones* potential

SmA*	Chiral smectic A phase
SmA	Smectic A
SmB	Smectic B
SmC	Smectic C
SmF	Smectic F
SmI	Smectic I
Sm	Smectic
T	Temperature
t	Time
T_0	Reference temperature
$\langle\tau_n\rangle$	Smectic order parameter
τ_n	Weighting factor
τ_{P_B}	Pressure damping factor in *Berendsen* barostat
τ_{T_B}	Temperature damping factor of *Berendsen* algorithm
$\tau_{T_{NH}}$	Oscillational time of *Nosé-Hoover* thermostat
$\mathrm{T_{CrSmA}}$	Transition temperature crystal → smectic A
θ	Bond angle
θ_0	Equilibrium bond angle
$\mathrm{T_{SmAIso}}$	Transition temperature smectic A → isotropic
UA	*United Atom*

UA-C$_{12}$E$_3$I$_1$	*United atom* description of 1-*O*-[2′-[2″-[2‴-(dodecyloxy)ethoxy]ethoxy]ethyl]-*myo*-inositol
	C$_{12}$: *UA* dodecyl substructure;
	E$_3$: *UA* triethylene glycol substructure;
	I$_1$: *UA* *myo*-inositolsubstructure

UA-E$_3$I$_1$C$_{12}$	*United atom* description of 1-*O*-dodecyl-4-O-[2′-[2″-[2‴-(hydroxy)ethoxy]ethoxy]ethyl]-*myo*-inositol
	E$_3$: *UA* triethylene glycol substructure;
	I$_1$: *UA* *myo*-inositolsubstructure;
	C$_{12}$: *UA* dodecyl substructure

$\mathbf{u}_i$	Longitudinal axis of molecule i
$\mathbf{v}$	Velocity

$V_{1\text{–}4}$	Force constants for *Fourier* potential
$\mathrm{V}_{\mathrm{I/II}}$	Bicontinuous cubic (normal/inverted)
`VMD`	Visual Molecular Dynamics; visualisation application
$\mathbf{x}$	Position
$\mathbf{Z}$	Versor; layer normal
z	Projection onto the versor

I

Introduction

The idea and motivation of this work will be described and discussed. A short overview is given of the definition, properties, and application fields of liquid crystals (LCs). Furthermore, the computational methods are described, which have been applied up to now to gain an understanding of the driving forces of LC arrangements. Additionally, the classes of carbohydrate- and cyclitol-based LCs are introduced, which are dealt with in this work.

1. Motivation

The liquid crystalline state of matter plays an important role in current materials science, where it is used in different areas of application like display technology,[1, 2] cosmetics,[3] nanoscience,[4, 5] pharmacy or pharmacology.[6–8] Moreover, it is also an important factor in nature where it can be observed in cell membranes[9] or in the DNA (deoxyribonucleic acid) in certain stages of the cell life cycle.[10] Even an essential influence of lyotropic liquid crystals on the development of early and primitive life forms by enabling the formation of compartments is postulated.[11]

Liquid crystals combine characteristics of liquids and crystalline solids. While liquid crystalline phases are in general fluids, they show anisotropic properties due to long-range orientational order and varying degrees of positional order – from none at all up to translational, repetitive patterns in three dimensions.[12, 13] The specific positional order and symmetry of the molecular ensemble define the type of the respective liquid crystalline phase. Consequentially, several types of mesophases are known which differ in their molecular order and symmetry. Furthermore, they are characterised by their thermodynamic stability which defines the transition points between the different phases. The appearance of several liquid crystalline phases (polymorphism) in different temperature and/or concentration ranges is an often observed fact.

Computer simulations of these remarkable phenomena are an important tool for the understanding of the underlying molecular processes and the development of new liquid crystalline compounds. But since these phenomena arise from a bulk material of condensed matter, respective calculations are quite complex and approximative computational methods have to be applied.

The prediction and description of supramolecular phenomena in liquid crystals by means of molecular dynamics or other simulation techniques have been in the focus of current research efforts.[14–16] In the last years, several different approaches have been proven to be applicable in order to estimate, predict, or understand the supramolecular behaviour of liquid crystals.[15] Many of these methods use – especially in the beginnings of LC simulation – quite rough approximations for the description of the LC entities like lattice models,[17] one-site coarse grained models with hard shapes (hard convex bodies, hard cut spheres)[18, 19] or with soft shapes (Gay-Berne potentials),[20–22] multi-site coarse grained models,[23] or *united atom* models.[24, 25] Monte Carlo (MC)[17, 26–33] as well as Molecular Dynamic (MD)[24, 34–40] techniques were used to calculate the dynamics and interactions in the bulk systems.

The most applicable approach for such computations today is MD, which gives a good compromise between resource and time requirements and precision of the results. In MD, the so called force field defines, with the help of empirical parameters, how the atoms, molecules and their assembly interact. Recent studies have shown that the force field has to be suitably adapted to the problem under investigation in order to achieve good results in reproducing liquid crystal phases. A quite promising approach is described by the working group of Zannoni.[24] In this publication, the authors showed that it is possible to reproduce experimental results from literature known liquid crystals by applying a *united atom* force field approach using the *AMBER* force field. The *Lennard-Jones* parameters of the force field were tuned in an iterative series to reproduce the correct phase behaviour and transition temperatures of the 4-*n*-alkyl-4′-cyanobiphenyl (nCB) molecule series.

One class of liquid crystalline compounds with special physical properties are sugar-, cyclitol-based and/or oligo-oxyethylene-based liquid crystals.[41] They tend to exhibit surface active properties and show, in solution with water, the formation of lyotropic phases and as pure substances thermotropic phase behaviour.[42–46] The chemical constitution of such compounds highly facilitates biodegradability, which assigns them an important role as surfactants. One of the aims of this work is to understand, how subtle changes in the substitution pattern at the sugar moiety affects the mesophase formation and sequence. Another aspect is the arrangement of single structural elements (sugar, alkyl, and oligo-oxyethylene parts) in the molecule and their influence on the existence of LC phases. This could lead to knowledge-driven design of tuned liquid crystals with desired properties. For instance, single substituted sugar moieties in thermotropic liquid crystals mainly lead to a direct transition from the isotropic liquid to smectic (Sm) mesophase.[47]

MD can be a promising tool to gain insight into the intra-, inter-, and supramolecular behaviour. Several studies with carbohydrate- and cyclitol-based, thermotropic liquid crystals will be introduced in this work, documenting approaches to solve the problem of liquid crystal simulations with substructures bearing multiple hydrogen bond donor and acceptor groups.

II

Theory

This chapter deals with the theoretical background of liquid crystallinity. It explains the terminology of thermotropic and lyotropic LCs. Additionally, carbohydrate- and cyclitol-based LCs and the two model systems dealt with in this thesis are introduced. Eventually, the theoretical basics of liquid crystal simulation in general and Molecular Dynamics in particular are summarised. It will be especially focussed on the methods used in this work to simulate the atomistic systems. The last section of this chapter will deal with observables that were investigated in this work, e. g. orientational and positional order parameters or hydrogen bonds.

2. Liquid Crystals

The first published identification of the liquid crystalline state of matter was in 1888 by Friedrich Reinitzer in cholesteryl benzoate (**1**) and cholesteryl acetate (**2**). He described a cloudy liquid phase between the solid state and the clearing point, while the substance was melted. Furthermore, by visual inspection vivid colours appeared. Under a polarisation microscope with crossed polarisers visible textures could be observed in the same phase (cf. Figure 1), indicating birefringence and hence, anisotropic properties.[48–50] Otto Lehmann introduced the terms of 'flüssige Krystalle' (fluid crystals), 'fliessende Krystalle' (flowing crystals), and 'krystallinische Flüssigkeiten' (crystal fluids), denoting different types of LCs.[51]

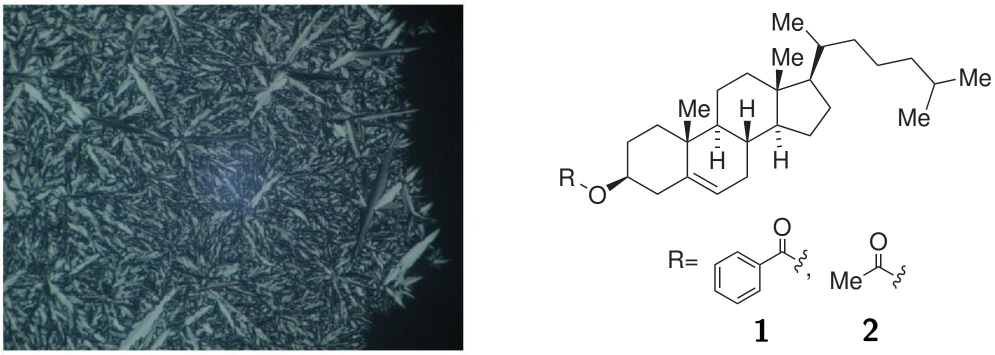


Figure 1: Texture of cholesteryl acetate in a polarisation microscope (left).[52] Structure of cholesteryl benzoate (**1**) and cholesteryl acetate (**2**) (right).

This terminology was updated by Georges Friedel, who termed the phases, appearing between the perfectly ordered periodic state and the completely disordered, amorphous or liquid structures, "mesomorphic".[53] Nowadays, this term also comprehends plastic crystals, which will not be further discussed in this work. Hence, this term will here only denote the liquid crystalline phase. The entities that form LC phases are called mesogens. Mesogens can either be single molecules or oligomeric aggregates, e. g. dimers.

Eventually, he also introduced the terminology for the two liquid crystal types dealt with in this work, nematics (cf. Section 2.1.1, page 7) and smectics (cf. Section 2.1.2, page 8).

Liquid crystalline or mesomorphic phases can be formed by a broad field of compounds. The characteristic property of such phases is that between the crystalline state and the common isotropic liquid one, new types of molecular order are formed, where the molecules possess a long-range orientational order but no or only a short range positional order.[12, 13] On the molecular level, the self assembly of the molecules is an entropy-driven process that is enhanced by intermolecular interactions.[54, 55]

Three broader classes of liquid crystalline states are defined according to the fact if they appear either in a certain temperature range (thermotropic liquid crystals) or in certain concentration/temperature ranges of dissolved compounds (lyotropic liquid crystals), or if additionally the pressure has an influence an the phase formation (barotropic liquid crystals). The latter will not be discussed in this work.

The abbreviations introduced in the following sections are in accordance to the publications of BARÓN[56] and its translation by TSCHIERSKE et al.,[57] in which the fundamental terminology and definition of liquid crystals is summarised.

2.1. Thermotropic Liquid Crystals

Thermotropic liquid crystals show mesomorphic structures in a suitable temperature range. The two most prominent and for this work most important mesophases are the nematic and the smectic phases, which are described in the following.

2.1.1. Nematics

The nematic phase is the simplest form of liquid crystals. Nematic phases (N) can be formed by calamitic, i. e. rod-shaped molecules,[58] or discotic, i. e. disk-shaped, molecules. The inherent property is the orientation of all molecules in a preferential direction. This preferential direction is denoted by a unit vector **n** called director (cf. Figure 2, page 8). While there is a preferential orientation of the molecules, the centers of mass of the molecules are distributed randomly, i. e. no translational order forms. The name nematic is derived from the greek word for thread (nema, $\nu\eta\mu\alpha$), since their phases exhibit characteristic, thread-like disclinations, i. e. disturbances in the orientational order,[59] in the polarisation microscope between crossed polarisers (cf. Figure 3).

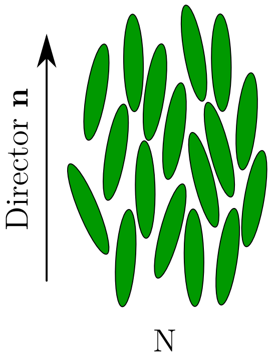


Figure 2: Schematic representation of rod-like molecules in the nematic phase. N: Nematic phase; **n**: Director, i. e. a unit vector indicating the main preferential direction.

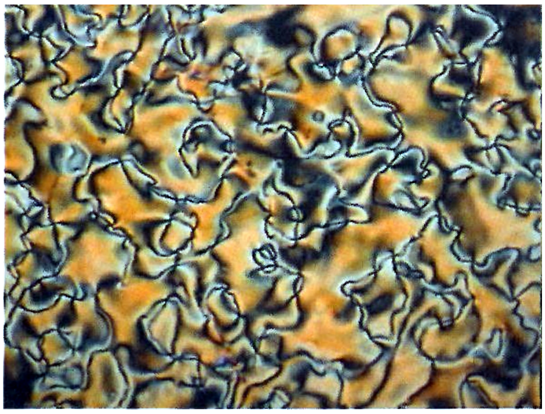


Figure 3: Schlieren texture of a nematic liquid crystal in a polarisation microscope, crossed polarisers.[60] Visible are the disclination lines, i. e. disturbances in the orientation.

2.1.2. Smectics

Like nematic phases also the smectic phases are formed by rod-shaped molecules. The characteristic property of smectic liquid crystals is that the molecular centers of mass are arranged in layers. This leads to the macroscopic property of a greasy or soapy consistency. Hence, the name was derived from the greek word for soap (smegma, $\sigma\mu\eta\gamma\mu\alpha$).

Three different smectic phases – denoted by A, B, and C – were first described by Sackmann and Demus in 1966.[61] In the following years, their supramolecular characteristica were revealed and clarified. The smectic A (SmA) phase consists of layers in which the molecules are parallel aligned and oriented in average parallel to the layer normal.[62] It can be understood as a two-dimensional fluid, since the molecules are randomly distributed and mobile within the layer (cf. Figure 4, page 10).

In this respect, smectic C (SmC) phases are quite comparable to SmA. The main difference is that in the SmC phase the molecules inside a layer are tilted by a tilt angle ϕ with respect to the layer normal.[63, 64] Insofar SmA phases can even be understood as a special case of SmC (tilt angle $\phi = 0$). In this case the versor, i. e. the layer normal, coincides with the main direction of the molecules or director (cf. Section 2.1.1, page 7). But the differences in the symmetry lead to significant differences on the macroscopic level.

de Vries proposed an established *diffuse cone model* that describes a rotation of the molecules around the layer normal with an in average fixed tilt angle.[65, 66] This rotation holds for the SmA phases as well as SmC phases. While in SmA phases there is a rotational symmetry, this symmetry vanishes in SmC phases. Two extreme examples of arrangement of molecules in a layer are the tilt in the opposite directions in neighbouring layers leading to a herringbone structure, or a tilt in the same direction. The *diffuse cone model* establishes the possibility of arrangements between this two extreme arrangements.

The Smectic B (SmB) phase has a higher degree of order than SmA or SmC phases. The additional degree of order lies in a hexagonal organisation within a layer.[67] If a phase sequence contains a SmC or a SmB phase as well as a SmA phase, SmC and SmB phases appear always at temperatures lower than the SmA phase.[68] In Figure 4 on page 10 these smectic molecular organisations are schematically demonstrated.

Even higher ordered smectics are possible like SmI and SmF, which are structurally comparable to SmB phases but the molecules are tilted towards the edge of the hexagon or the corner, respectively.[13] But such highly ordered smectic phases will not be discussed further at this point.

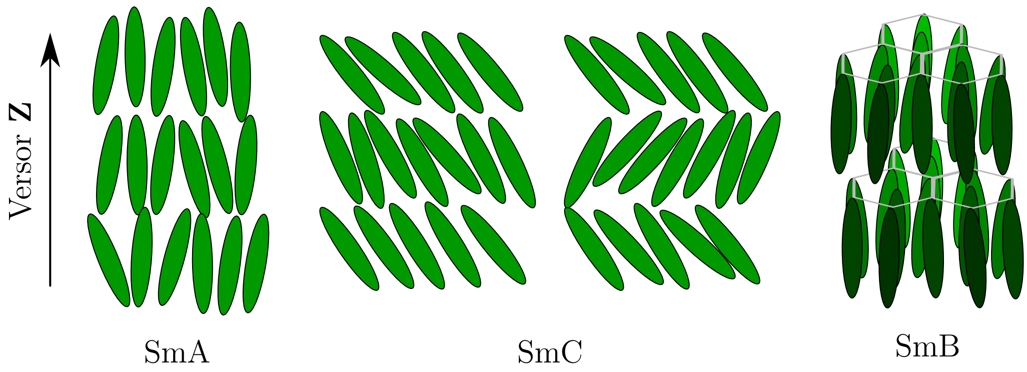


Figure 4: Schematic illustration of the three most frequent smectic phases, SmA, SmB, and SmC. The optical axis **Z** is parallel to the layer normal or versor. In the SmA phase the versor **Z** and director **n** coincide.

For molecules with an intrinsic structural polarity, three different variants of the molecular arrangement within the layers can be distinguished. Figure 5 illustrates these possibilities and their notation. Depending on how the repetitive unit, i. e. the layer, is composed $\mathrm{SmA_1}$, $\mathrm{SmA_2}$, and $\mathrm{SmA_d}$ are discriminated. If the molecules are arranged in a monolayer with undefined "upwards" or "downwards" orientation of the molecules, the phase is of type $\mathrm{SmA_1}$. A phase is denoted with $\mathrm{SmA_2}$, if the repetitive unit is a bilayer. If the bilayers are interdigitised, the phase is denoted as $\mathrm{SmA_d}$. These categories exist with $\mathrm{SmC_1}$, $\mathrm{SmC_2}$, and $\mathrm{SmC_d}$ for the SmC phases as well.[56, 57]

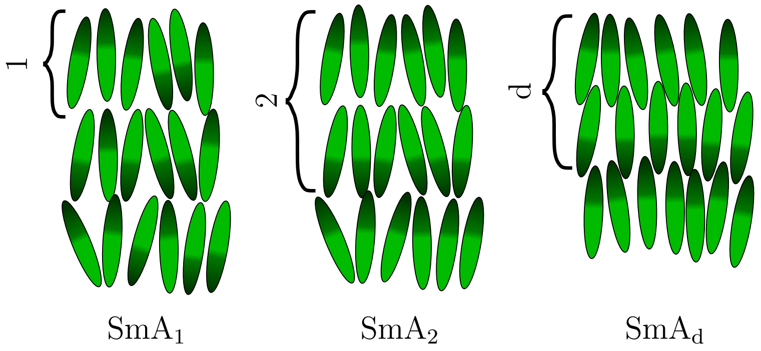


Figure 5: SmA subphases of polar molecules. From left to right: $\mathrm{SmA_1}$: Monolayer built up from randomly "upwards" or "downwards" oriented molecules; $\mathrm{SmA_2}$: bilayer consisting of two antiparallel layers; SmA_d: bilayer consisting of two interdigitised, antiparallel layers. 1, 2, and d denote the layer width based on multiples of the molecular length with $1 < d < 2$.

2.1.3. Further Phase Types

Beside the previously discussed thermotropic LC phases there are also chiral variations of the aforementioned, like the chiral nematic – formerly known as cholesteric – phase (N*).[69] Another type are the blue phases (BP), which have a three dimensional lattice structure and a helical orientational order.[13, 62, 70, 71] While the latter LC phases are all mainly based on calamitic molecules, there are other LC phases that are built up from discotic molecules.[72–76] Mostly, these molecules form columns by stacking on top of each other, hence these phases are denoted as columnar liquid crystals (Col).[77] Those will not be further discussed here.

2.2. Lyotropic Liquid Crystals

Lyotropic liquid crystalline phases are formed by compounds in solution. They can form with amphiphilic or amphiphob compounds, which form micellar structures.

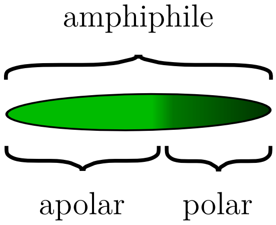


Figure 6: Schematic representation of a rod-like molecule with amphiphilic properties.

2.2.1. Amphiphilicity

Amphiphilic molecules consist of regions that interact differently with a solvent (cf. Figure 6). The lyophilic part, i. e. the part, which strongly interacts with the solvent, orients towards the solvent, while the other part orients away. Thus, amphiphilicity leads to a segregation into microdomains (microphase seperation), a property, which can also have a strong influence in thermotropic liquid crystals.[46, 78]

2.2.2. Micelles

Based on their amphiphilicity respective molecules can organise to form micellar structures. These micelles can have various shapes from spheres over cylinders to discs.[57, 78] In case of a nonpolar solvent, the micelles are organised *inverse* as opposed to conventional micelles formed in polar solvents like water (cf. Figure 7). In conventional micelles, interactions appear between the solvent and the polar head groups by hydrogen bonds or due to ionic interactions, while *van der Waals* interactions take place between the lipophilic alkyl chains.[41] The actual driving force to form micelles is the hydrophobic effect. This effect is driven by a gain in entropy. The hydrophobic parts of a molecule force the water molecules to reorient and rebuild a hydrogen bond network around the molecule without an actual interaction between solvent and solute. The entropy rises by avoiding this water reorganisation, hence less water molecules need to be locally ordered.[79]

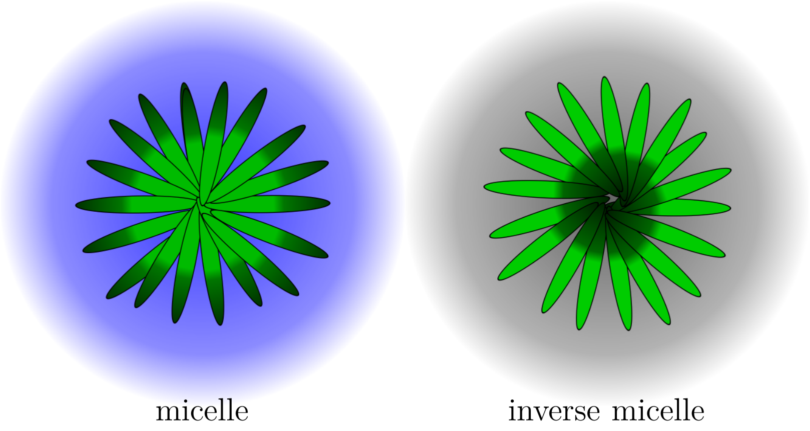


Figure 7: Schematic representation of rod-like molecules forming micelles. In a polar solvent (blue) the polar parts (dark) orient outwards; in an apolar solvent (grey) the apolar parts (light) orient outwards.

Micellar structures have an inherent orientational order and act as building blocks leading to positional organisation of the molecules in the phase.

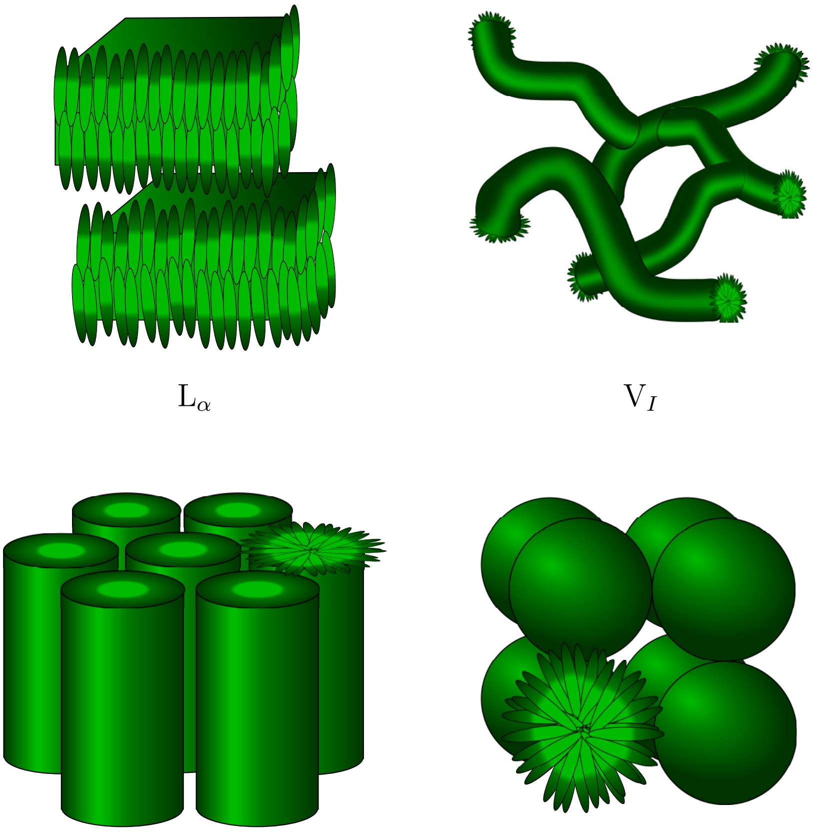
2.2.3. Phase Types

Environmental and intrinsic factors, which influence lyotropic LCs are the temperature and the degree of dilution, i. e. the concentration of the amphiphile. Depending on the molecular composition and the aforementioned factors, different phase types are possible. In the case of barotropic LCs the pressure is also a modifying factor.

Well known lyotropic liquid crystal phases are the lamellar (L_α), the bicontinuous cubic ($V_{I/II}$), the columnar or hexagonic ($H_{I/II}$), and the discontinuous cubic phase ($I_{I/II}$). The latter three exist in a normal (subscript I) or an inverted version (subscript II), depending on the ratio of the volume fraction of the polar parts of the molecules to the volume fraction of the apolar parts of the molecules.[78]

While L_α forms layered structures, $V_{I/II}$ shows a three-dimensional, web-like structure, in which the solvent on the one side and the solute on the other side form seperated, but for each species continuously connected domains. The $H_{I/II}$ phases consist of columns arranged in hexagons and the $I_{I/II}$ is built up from micelles surrounded by the solvent.[12] In Figure 8 on page 14 these four phases are schematically illustrated.

Additionally to the L_α, $V_{I/II}$, $H_{I/II}$, and $I_{I/II}$ intermediate phases are possible, which generate a broad suprastructural variety. They are subsumed under the terminology *ribbon* and *mesh* intermediate phases, depending on if the micelles form bands and ribbons or sieve and mesh like structures. Especially the L_α, the bicontinuous cubic phases and these intermediate phases have a strong relationship to biological membrane systems.[80]



L_α V_I

H_I I_I

Figure 8: Schematical illustration of the four conventional lyotropic phase types. L_α: lamellar; V_I: bicontinuous cubic; H_I: hexagonal; I_I: discontinuous cubic.[12, 78]

3. Carbohydrate- and Cyclitol-Based Liquid Crystals

Carbohydrate- and cyclitol-based liquid crystals share a common property for the formation of mesophases. They bear hydroxyl functions, which give rise to hydrogen bond network formations. This results in strong directed intermolecular interactions which can lead to highly ordered supramolecular structural arrangements.

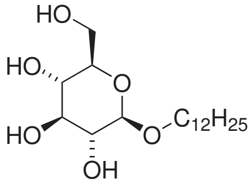
3.1. Carbohydrate-Derived Liquid Crystals

An interesting class of liquid crystalline compounds are carbohydrate-based LCs. Glycolipids can be amphotropic molecules, i. e. they can form thermotropic as well as lyotropic mesophases. A broad variety of features makes them a target for scientific analysis and applications in materials science, pharmacology, and cosmetics. They are easily accessible, non-toxic, and biodegradable. A high diversity of molecular structures is possible due to various substitution patterns at the sugar unit and due to different configurations at the chirality centers. Glycolipids mainly exhibit rather low melting temperatures and show often a good solubility in water and organic solvents which renders them applicable under physiological conditions and in standard human environments.[46]

In those amphiphiles the hydroxyl functions of the sugar moiety serve as hydrogen bond donors and acceptors and form hydrogen bond networks either with polar solvents and sugar molecules (lyotropic LCs) or in thermotropic LCs exclusively with other polar parts of the molecules. Linear sugar-based amphiphiles tend to exhibit SmA_d phases[45, 47, 81] or their chiral versions SmA_d*.[82]

A structurally quite simple alkyl-glycoside that exhibits liquid crystalline behaviour and forms a smectic A phase is *n*-dodecyl-β-D-glucopyranoside (**3**) (cf. Figure 9, page 16).

GOODBY thouroughly studied and described its phase sequence.[83] He could determine by Differential Scanning Calorimetry (DSC) that the alkyl-carbohydrate **3** has three crystal modifications, which appear at different



3

Figure 9: Structure of *n*-dodecyl-β-D-glucopyranoside.

temperatures and a SmA mesophase before melting into the isotropic liquid (cf. Table 1).

Table 1: Phase transition temperatures of *n*-dodecyl-β-D-glucopyranoside (**3**).[83]

	Cr_1		Cr_2		Cr_3		M		Iso
3	•	54.8 (328.0)	•	63.2 (336.3)	•	80.4 (353.6)	SmA	143.4 (416.5)	•

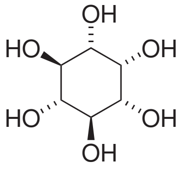
Cr_{1-3}: Crystal phases, M: Mesophase, Iso: Isotropic.
Units are given in °C (K).

Due to its structural simplicity and its comparably broad mesophase of 63 K compound **3** appeared as a suitable model system to approach the simulation of hydrogen bond-based liquid crystal.

3.2. Inositol-Derived Liquid Crystals

Besides traditional aldoses or ketoses like glucose or fructose also cyclic polyols like inositol (cf. Figure 10, page 17) offer the general advantages described before. Many different properties render inositol an interesting family of compounds for the synthesis of liquid crystals, among them biodegradability, bioavailability, and pharmaceutical importance.

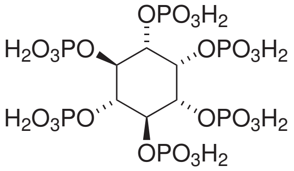
Inositol is a ubiquitous compound, i. e. it can be found in all life forms from *Archaea* bacteria to plants and animals, and in them in all tissues.[84] It often appears as phytic acid **5** (IP6), which serves as a storage and transport compound of organically bound inorganic phosphate (cf. Figure 11, page 17).[85] Phosphorylated derivatives of *myo*-inositol (**4**), have an important contribution as messengers that regulate the Ca^{2+} homeostasis in cells.[86–88] Additionally,



4

Figure 10: The structure of the most frequently occuring stereoisomer of inositol, *myo*-inositol (**4**).

they play a role in cell proliferation and gene regulation, either indirectly by releasing second messenger molecules[89] or directly by regulating the DNA replication or the chromatin degradation and organisation.[90] *Myo*-inositol derivatives are even involved in cytoskeletal organisation.[91] Due to their important influence as messenger and regulator of cellular structure and function they also appear in the context of several deseases like bipolar disorders or Alzheimer's desease, which are both based on Ca^{2+} dysregulation,[86, 92] or in cancer.[93] Thus they can become an important drug or druggable target.[86, 90, 94] Compound **5** even appears to reduce cell proliferation and has thus a beneficial contribution in cancer treatment.[93]



5

Figure 11: The structure of phytic acid or *myo*-inositol hexakisphosphate.

Structurally, inositol or hexahydroxycyclohexane is a cyclic molecule with six carbon atoms, each bearing a hydroxy function.[84] The free hydroxyl groups at this so called cyclitol give rise to the formation of hydrogen bonds. *myo*-inositol (**4**) is the most frequent occuring stereoisomer of inositol (cf. Figure 10).

In combination with alkyl chains and oligo-ethoxy groups Catanoiu built up a broad variety of inositol-based compounds with different organisation and substitution patterns at the cyclitol ring system **4**. In different synthesis sequences these molecules were built up from *myo*-inositol **4**, triethylene glycol **6**, and dodecylbromide **7** (cf. Figure 12).[42–44] The insertion of oligooxyethylene groups into a molecule introduces hydrogen bond acceptors. Hence, molecules containing ethoxy groups can show in protic solvents lytropic LC phases.[12]

Though consisting of the same molecular subunits the synthesised compounds show very different properties. An example are the racemic mixtures of compound *rac*-**8** and *rac*-**9** synthesised by CATANOIU.[44] Only *rac*-**8** shows a SmA phase as mesophase. The racemic compound *rac*-**9** melts directly from the crystal into an isotropic liquid (cf. Table 2, page 19).[44]

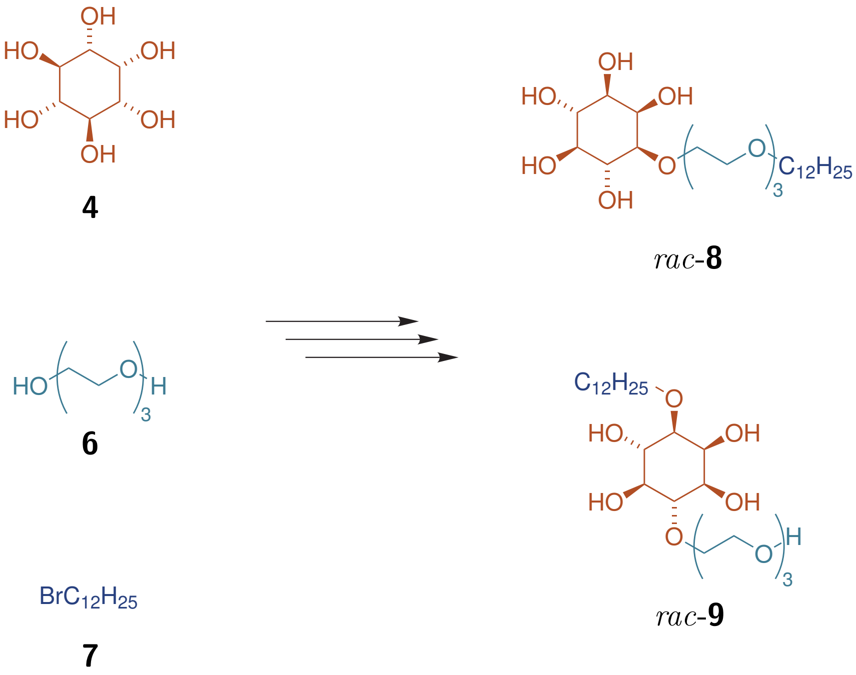


Figure 12: Conceptual generation of 1-*O*-[2′-[2″-[2‴-(dodecyloxy)ethoxy]-ethoxy]ethyl]-*myo*-inositol (*rac*-**8**) and 1-*O*-dodecyl-4-O-[2′-[2″-[2‴-(hydroxy)ethoxy]ethoxy]ethyl]-*myo*-inositol (*rac*-**9**) from *myo*-inositol (**4**), triethylene glycol (**6**), and dodecylbromide (**7**).[44]

The slightly increased complexity of *rac*-**8** and *rac*-**9** in comparison to the alkyl glucoside **3** and the finding of the different mesophase behaviour makes this two compounds interesting targets for further studies in a computational approach. To explain and to find reasons for this difference is one of the aims of this work and will be evaluated in the course of this thesis.

Table 2: Phase transition temperatures of *rac*-**8** and *rac*-**9**.[44]

	Cr		M		Iso
rac-**8**	●	80.1 (353.3)	SmA	153.1 (426.3)	●
rac-**9**	●	80.1 (353.3)	—	—	●

Cr: Crystal, M: Mesophase, Iso: Isotropic.
Units are given in °C (K).

4. Simulation of Liquid Crystals

A lot of different methods have been used to understand liquid crystal phase formation with computational methods, ranging from abstract lattice models over coarse grained molecular models to atomistic approaches (cf. Figure 13).

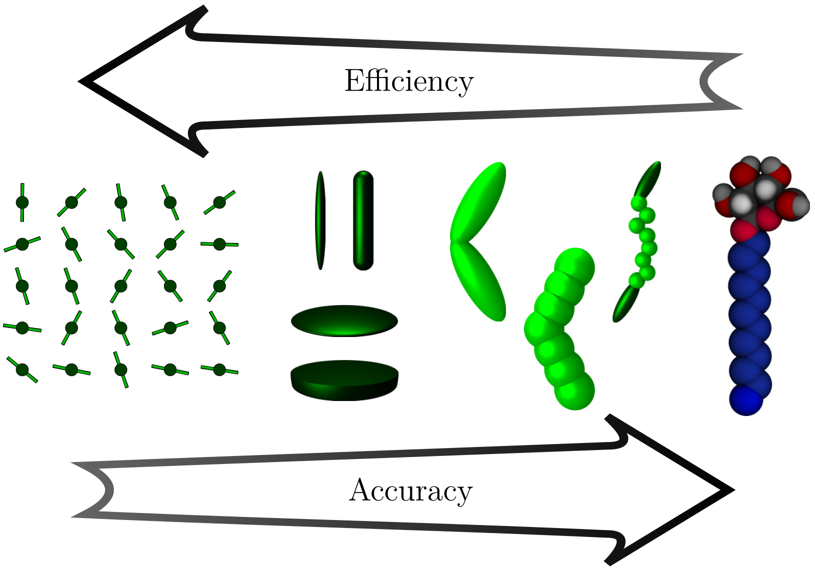


Figure 13: Different methods to approach LC simulation. From left to right: Lattice models, hard and soft one-site models, multi-site coarse grained approaches, atomistic representations. While the degree of accuracy rises, the computational efficiency falls and more computational power is needed.

4.1. Lattice Models

Lattice models are based on the mean field theory by Maier and Saupe, who described interactions in nematics and the formation of a nematic phase only by dipole-dipole dispersion interactions. That means that only the longitudinal axis is considered, while omitting the molecular shape and the positional movement of molecule.[95–97] Lebwohl and Lasher were the first who applied this theory in computational simulations.[17, 31] They organised the molecules, represented only by their longitudinal axes, on the vertices of a lattice, allowing only interactions between direct neighbours (cf. Figure 14). Especially this aspect models the local character of the mean field theory, since this theory also states that in a (nematic) LC phase a preferential orientation is present in every point of a phase.[97] *Monte Carlo* simulations with restricted[31] and unrestricted directional orientation[17] could properly reproduce the formation of the nematic phase. In this kind of simulations a positional organisation is neglected and only the orientation of molecules is in the focus. Lattice models are computationally the most efficient ones to address the problem of nematic phase formation.[98]

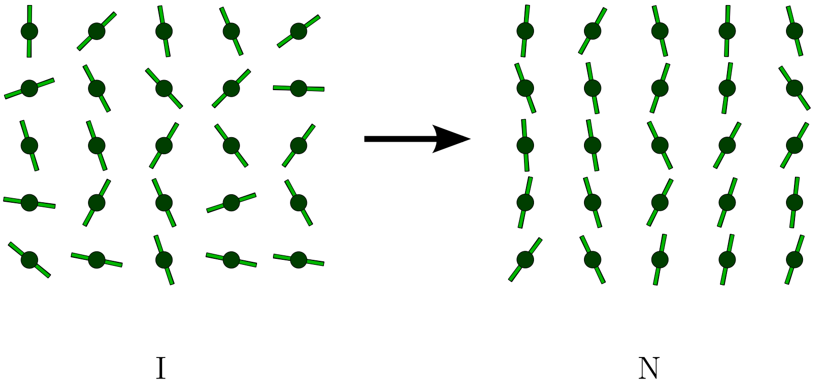


Figure 14: Two-dimensional representation of the lattice model showing the transition/reorientation from an isotropic orientational distribution into a nematic order.

4.2. Coarse Grained Models

Coarse grained (CG) models describe the molecular entities by rough approximations. They either represent the molecule as one object (one-site) or split it into reasonable units, which are treated differently (multi-site). Many degrees of coarse graining or accuracy, respectively, are possible, especially considering the multi-site approach.

4.2.1. Hard Objects

Hard objects are one-site coarse grained models. They are based on the theory of Onsager, who introduced a mathematical model to describe the influence of shape and excluded volume effects on the formation of LC phases.[54] He formulated that a high density solution of rod-shaped particles favours the formation of a nematic order over the isotropic distribution.[54] In this approach, the relation of the particle length L to the particle diameter D plays an important role. Furthermore, only the shape of these molecules is taken into account for the mathematical model as well as for the simulations (cf. Figure 15).

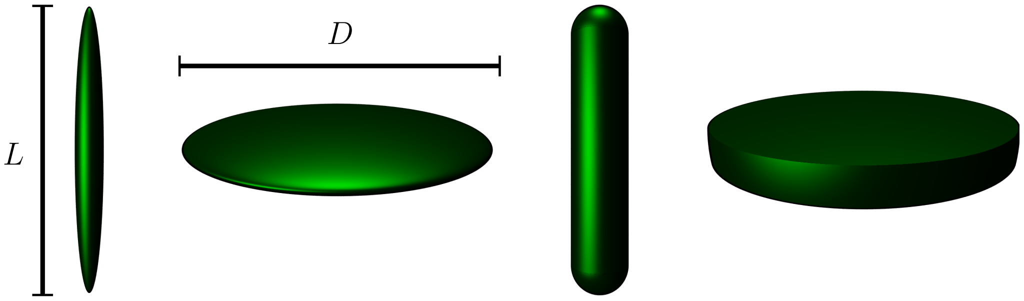


Figure 15: Examples for hard objects. From left to right: needle-like, prolate ellipsoid; platelet-like, oblate ellipsoid; spherocylinder; hard cut sphere. An important aspect is the relation of the length L to the diameter D.

Extensive work by Frenkel, Allen, Stroobants, and co-workers revealed that the shape of the molecule representing object influences the formation of different LC phases.[28–30, 99] This formation is strictly density dependent and can be simulated by MD and MC calculations, respectively.[99] Frenkel introduced the usage of ellipsoids with different length to diameter ratio.[30] Needle-[100] or platelet-like ellipsoids,[28, 29, 101] even with infinitely small thickness, are only able to show an isotropic to nematic transition. Two other geometric objects, spherocylinders and hard cut spheres, exhibit a richer variety of phases. While spherocylinders are able to reproduce smectic A phases, since the cylindrical part fills space more efficiently than the spherical caps,[102–104] even the formation of a columnar phase can be shown with hard cut sphere objects.[103, 105]

4.2.2. Soft Objects

Another approach, which aims in mimicking a more realistic behaviour of the interaction sites in a simulation system, is to model the aforementioned geometries with an interaction potential. In this case the interactions take place through a *Lennard-Jones* like potential ((10) on page 29) instead of elastic collisions. The most prominent variant is the *Gay-Berne* potential, which simplifies the mesogens to several or one interaction site, models an orientation dependence with a *Gaussian* function, and introduces strength and range parameters via the *Lennard-Jones* potential.[20, 106] *Gay-Berne* potentials were already successful in simulating nematic,[107] smectic A and B phases,[108, 109] and even columnar phases in LC mixtures.[110] Approaches to chiral molecules have also been performed with *Gay-Berne* ellipsoids. MEMMER and co-workers were able to simulate the formation of N*, SmA*, and blue phases.[26, 111, 112]

4.2.3. Multi-Site Representations

Multi-site representations, i. e. coarse grained representations with more than one unit representing a mesogenic object with specific properties, were successfully used by WILSON and others.[23, 113, 114] They are interesting, because they simplify the chemical structure while retaining essential features of the molecule, e. g. flexible parts or polar areas. Up to now, the approaches taken in the literature are based on the elements described in section 4.2.2. While *Lennard-Jones* potentials are used to model spherical parts, *Gay-Berne* potentials are used to model anisotropic, ellipsoidal structures. Additional potentials can be used to tune the interaction in a certain chemical aspect, like e. g. polarity. Figure 16 on page 23 shows some examples of successfully applied multi-site models. With a rigid V-shaped molecule formed from seven *Lennard-Jones* spheres, a simulation of the formation of smectic A and B phases was possible.[115] The formation of smectic phases could also be simulated in system of multi-site objects of two ellipsoids connected flexibly via eight *Lennard-Jones* spheres.[23] With a two-site *Gay-Berne* model a dependence of the phase formation to the bend angle between the two units was observed. A nematic phase (straight molecules), a smectic A phase (slightly bent), and even a chiral SmA (SmA*) phase (strongly bent) could be simulated.[116]

This approach is also used in polymer science in combination with LCs.[117] Hence, multi-site CGs are also beneficial in addressing larger molecular structures, e. g. like dendrimers, which consist of core elements and flexible attachments, like alkyl chains or siloxanes.[113, 114] A broad variety of lyotropic phases was also simulated with coarse grained approachs, in which water was represented by one particle and the LC mesogen by several interaction centers which interact via short-range *Lennard-Jones* potentials.[118]

In the last years, some methods aim at approximating multi-site potentials which reproduce results of atomistic simulations. These methods are called *multiscale approaches*, since with them a switching between atomistic simulations and coarse graining becomes possible. Thus, a better and faster sampling of the phase space is possible, the transitions into other phases can be accelerated (CG), and the reached phase states can then again be sampled with an atomistic approach.[119, 120] Computational models which work in a similar fashion aim in adopting the surface of a molecule and map appropriate potentials to resemble the molecular surface properties (surface interaction models).[121–124]

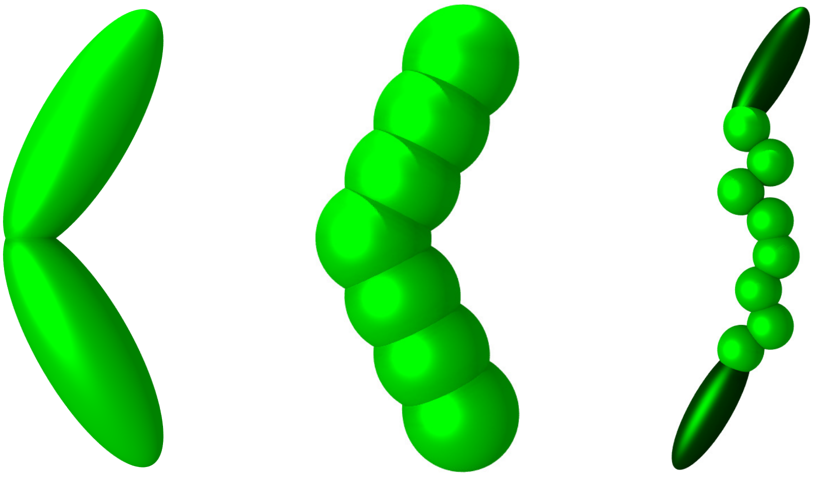


Figure 16: Examples of coarse grained, multi-site mesogen objects formed from *Gay-Berne* (ellipsoids) and *Lennard-Jones* potentials (spheres).[23, 115, 116]

4.3. Atomistic Simulations

Especially in the last decade, atomistic simulations are in the focus of research efforts. On the one hand computational resources become more accessible and more efficient, on the other hand atomistic descriptions of molecules could represent the experimental behaviour of molecules to a higher degree. Atomistic simulations consider the molecule as built-up from atoms or atomtypes (cf. Figure 17). Every atomtype represents an element in a certain chemical environment, e. g. oxygens in hydroxyl groups are treated differently from oxygens in an ether group.

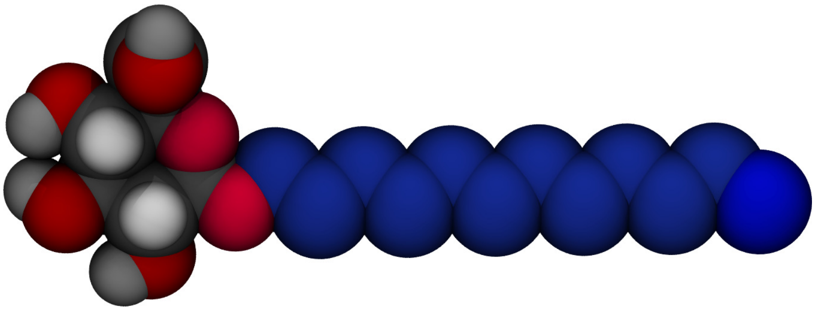
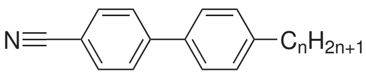


Figure 17: Example of an atomistic description of alkyl-glucose **3**. Every colour denotes a different atomtype. While the sugar moiety is described in an *all atom* fashion, the alkyl moiety is composed of *united atoms*, each representing a CH_2 or CH_3 group, respectively.

An approximation in atomistic treatments of molecules is realised in the handling of aliphatic carbon atoms. Two ways of treating the atoms are possible. Either every atom is considered, including the hydrogens, this is called *all atom* (*AA*) approach. Or the hydrogen atoms attached to an aliphatic carbon atom are treated as a combined unit together with the carbon atom, i. e. each CH, CH_2, or CH_3 group and CH_4 (methan) is represented as one unit, a *united atom* (*UA*).

One of the bigger problems of an atomistic description is the significant increase in internal degrees of freedom. Hence, especially in the beginning of atomistic simulations, rather rigid molecules which by definition have less internal degree of freedom were investigated. A group of compounds that has emerged as an intensively studied model system is the 4-*n*-alkyl-4′-cyanobiphenyl (nCB) series (**10**), where n denotes the length of the alkyl chain (cf. Figure 18, page 25).[24, 34, 125–134]



10

Figure 18: Structure of the 4-n-alkyl-4′-cyanobiphenyl homologuous series.

The nCB series consists of a quite rigid, rod-like core motif with a cyano-group and a flexible alkyl chain of varying length. The polar cyano-group has an important influence on the supramolecular arrangement of the molecules. Mainly based on *Coulomb* interactions it enables the formation of dimers, which is especially important for the stability of the $\mathrm{SmA_d}$ phase in 8CB. Computational experiments, in which the partial charges are removed, show a destabilisation of the smectic LC phase.[127]

Other systems that are described by simulations of an atomistic fashion are steroids with their rigid steran scaffold. Due to its conformational rigidity the molecules are reduced to a rod-like structure.[135] Various linear molecules with phenyl or other aromatic ring systems,[136–140] or disk-shaped molecules built up from extended aromatic ring systems were also studied.[141, 142] All those systems have in common that the internal degrees of freedom are reduced either by their configuration and their resulting rigid conformation (steran scaffold) or the aromaticity which keeps the rings planar.

Due to the high degree of translational, rotational, and vibrational motion in atomistic approaches, a lot of different configurations become accessible. Hence, the phase space, i. e. the sum of possible configurations, which the simulated system can adopt, becomes quite big. In this context the term configuration is used as the sum of all conformations at a certain time. This is why, especially in the early stages, only the melting of preordered phases was studied.[136, 141] It has been only in recent years that the simulation of the onset of order from disordered, i. e. isotropic, configurations is subject in atomistic computational studies.[125, 127, 137–139, 142]

A more thorough description how atomistic simulations are performed with Molecular Dynamics is given in Section 5 on page 27.

Summarizing, it can be stated that atomistic simulations of thermotropic LCs have focussed on rigid molecules with reduced internal degrees of freedom and moderately long alkyl chains. Furthermore, strong polarised or partially charged molecules as building block for thermotropic LCs has not yet been studied by simulation techniques, although it is known that polar aspects of a molecule are important. The simulation of carbohydrate or cyclitol derivatives especially with oligoethoxy- and alkyl moieties, which add flexibility, constitutes a new topic in computational LC science.

5. Molecular Dynamics

Molecular Dynamics aims at the prediction of molecular movement by solving *Newton*'s equations of classical motion. This technique works with reasonable approximations and simplifications, i. e. reducing the atoms and bonds to spheres connected via springs. A carefully designed function is used to derive the resulting forces in the system and thus the energy. This function is called the "Force Field" (FF).

5.1. Integrating *Newton*'s Equations of Motion

Newton's second law creates a relation between forces in a system and deceleration or acceleration:

$$\mathbf{F} = m\frac{d\mathbf{v}}{dt} = m\frac{d^2\mathbf{x}}{dt^2} = m\mathbf{a} \tag{1}$$

where $\mathbf{F}$ is the resulting force, m the mass of the observed object, and $\mathbf{v}$ its velocity.Using this law, new positions and velocities can be calculated. For MD it is essential to derive the forces acting on each atom. This can be accomplished by the so called force field, i. e. a functional description that defines how atoms interact. This potential function is derived for every atom with respect to its position and thus the forces can be determined as:

$$\mathbf{F}_i = -\frac{\delta E_{total}}{\delta \mathbf{x}_i} \tag{2}$$

where F_i is the force acting on the ith molecule with the position $\mathbf{x}_i$ and E_{total} as the additive force field or potential energy function. Using these forces and applying a proper, meaning small enough integration time step Δt the new positions and velocities can be calculated as:

$$\frac{\mathbf{F}_i}{m_i} = \frac{d\mathbf{v}_i}{dt} = \mathbf{a}_i \tag{3}$$

Several different algorithms exist to calculate the new positions. Since the computational implementation of molecular mechanics is not in the focus of this work, they will not be discussed here further, but can be read from literature.[143–145]

5.2. Force Fields

A force field is the functional description which defines the energy in the system. It consists of several mathematical terms dealing with different aspects of molecular interactions. These interactions can be roughly divided into bonded and non-bonded contributions:

$$E_{total} = E_{bonded} + E_{non-bonded} \tag{4}$$

Depending on the actual implementation of the various contributing terms, numerous force fields have been developed. Many of them like the *AMBER*[146, 147] or *OPLS*[148, 149] force field share common functional features and just differ in details, like the parameters of some atomtypes or the way how the point charges are assigned.

5.2.1. Bonded Interactions

Bonded interactions describe the behaviour of atoms that are connected with other atoms in the molecule. Thus, molecular bonds, bond angles, dihedral angles, and out-of-plane angles are covered by that:

$$E_{bonded} = E_{bonds} + E_{angles} + E_{dihedrals} \tag{5}$$

By definition, these contributions are only intramolecular.

5.2.2. Non-Bonded Interactions

Non-bonded contributions account for several inter- and intramolecular relations. They describe the intermolecular interplay between atoms in terms of *van der Waals* and *Coulomb* interactions:

$$E_{non-bonded} = E_{van\ der\ Waals} + E_{Coulomb} \tag{6}$$

Additionally, intramolecular interactions between atoms that are separated by three bonds, i. e. 1-4 interactions, are taken into account within this type of interactions.

5.2.3. Optimised Potentials for Liquid Simulations

Depending on the force field, different functional representations are used to describe bonds, angles, proper and improper dihedrals. The force field, which is used throughout this work is the "Optimised Potentials for Liquid Simulations" (*OPLS*). It was established by JORGENSEN ET AL. to fit the special requirements of liquids in MD systems.[32, 148–151] The bonded and non-bonded interactions in this FF are defined as follows:

$$E_{bonds} = \sum_{bonds} k_r \, (r - r_0)^2 \tag{7}$$

$$E_{angles} = \sum_{angles} k_\theta \, (\theta - \theta_0)^2 \tag{8}$$

$$\begin{aligned} E_{dihedrals} &= \frac{V_1}{2}\left[1 + \cos\,(\phi - \phi_0)\right] + \frac{V_2}{2}\left[1 - \cos 2(\phi - \phi_0)\right] \\ &\quad + \frac{V_3}{2}\left[1 + \cos 3(\phi - \phi_0)\right] + \frac{V_4}{2}\left[1 - \cos 4(\phi - \phi_0)\right] \end{aligned} \tag{9}$$

$$E_{nonbonded} = \sum_{i>j} f_{ij} \left(E_{van\ der\ Waals} + E_{Coulomb}\right) \tag{10}$$

$$\begin{aligned} E_{van\ der\ Waals} &= \frac{A_{ij}}{r_{ij}^{12}} - \frac{C_{ij}}{r_{ij}^{6}} \\ &= 4\epsilon_{ij}\left(\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right) \end{aligned} \tag{11}$$

$$E_{Coulomb} = \frac{q_i q_j e^2}{4\pi\epsilon_0 r_{ij}} \tag{12}$$

Bonds and angles are represented by harmonic potentials (compare (7) and (8)) with k_r and k_θ as the spring constants, r_0 and θ_0 as the equilibrium bond length and angle, respectively, while r or θ is the actual bond length or angle. This means that every displacement from the equilibrium length or angle leads to an increase in energy (cf. Figure 19, page 30).

The dihedral angles, i. e. the rotation around a bond, are described as a fourier series (9). In this equation ϕ is the actual angle between the two bond substituents, $\phi_{1–4}$ is the phase shift of the periodic contribution, and $V_{1–4}$ are constants to tune the shape of the potential. Figure 20 on page 30 shows an example of such a potential function.

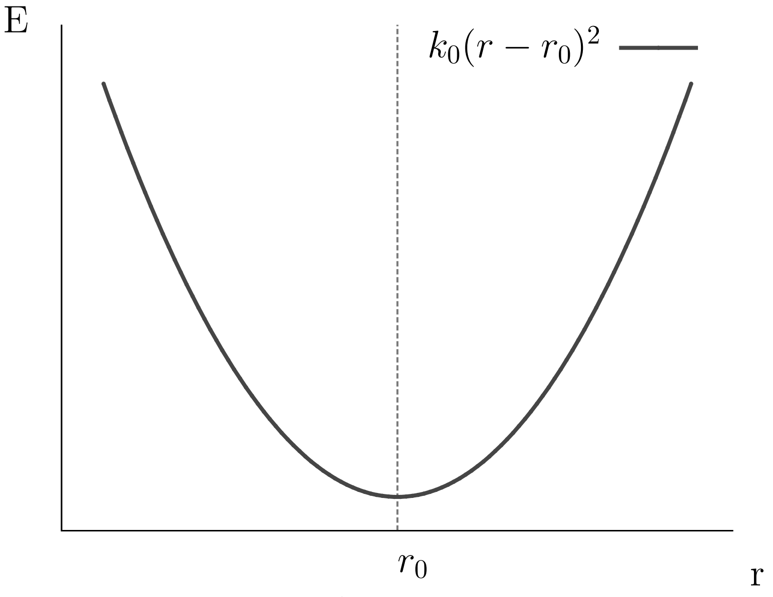


Figure 19: Harmonic potential for the bond compression and stretching.

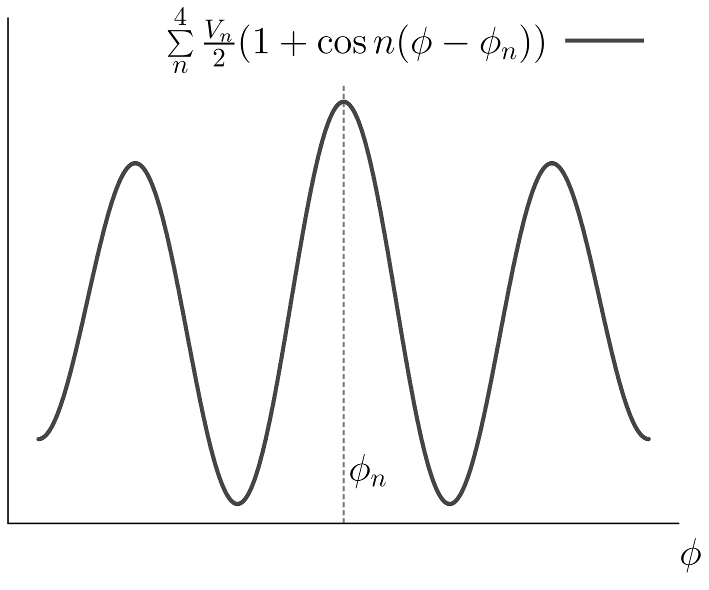


Figure 20: *Fourier series* describing the rotation around a bond. The potential curve was created with the parameters for the OH-C2-C2-OH bond (C2: sp3 carbon with two hydrogens (*united atom*); OH: alcohol oxygen).[146]

Van der Waals and *Coulomb* interactions are combined in a non-bonded term (10). While *van der Waals* interactions are calculated by a *Lennard-Jones* potential (11), the *Coulomb* ones are represented by a *Coulomb* potential (12). The so called fudge factor f_{ij} is set to 0.5 for 1-4 interactions and to 1 for every other.

Figure 21 exemplifies how the parameters ϵ_{ij} and σ_{ij} define the shape of the potential. While σ modifies at which distance the potential becomes zero, ϵ determines the depth of the potential well. Those constants are dependent on the atoms that are engaged in this interplay. The geometric average is applied as a combining rule in the OPLS force field to derive these constants:

$$\sigma_{ij} = \sqrt{\sigma_{ii}\sigma_{jj}} \tag{13}$$

$$\epsilon_{ij} = \sqrt{\epsilon_{ii}\epsilon_{jj}} \tag{14}$$

While the first term $4\epsilon_{ij}(\sigma_{ij}/r_{ij})^{12}$ describes the short range repulsive forces, the second term $-4\epsilon_{ij}(\sigma_{ij}/r_{ij})^{6}$ takes care of the medium range attraction.

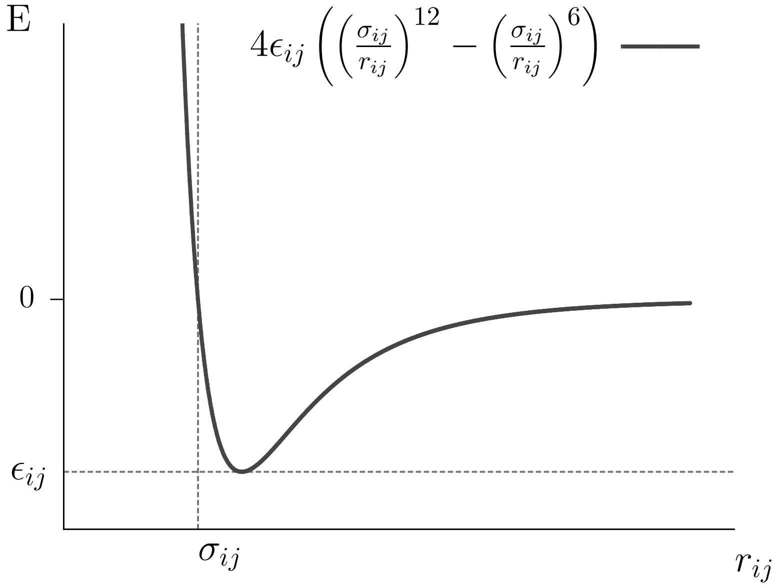


Figure 21: *Lennard-Jones* potential describing *van der Waals* interactions between atoms i and j. The influence of the characteristic constants ϵ_{ij} and σ_{ij} is depicted by dashed lines.

The *Coulomb* potential ((12) on page 29) describes an attractive or repelling force (depending on the sign of the charges) of the participating atoms. The closer the atoms get the stronger the effect becomes (cf. Figure 22, page 32). In this function e is the electronic charge, ϵ_0 is the dielectric constant or vacuum permittivity, q_i and q_j are the atomic charges of the participating atoms, and r_{ij} is the distance between the charge centers.

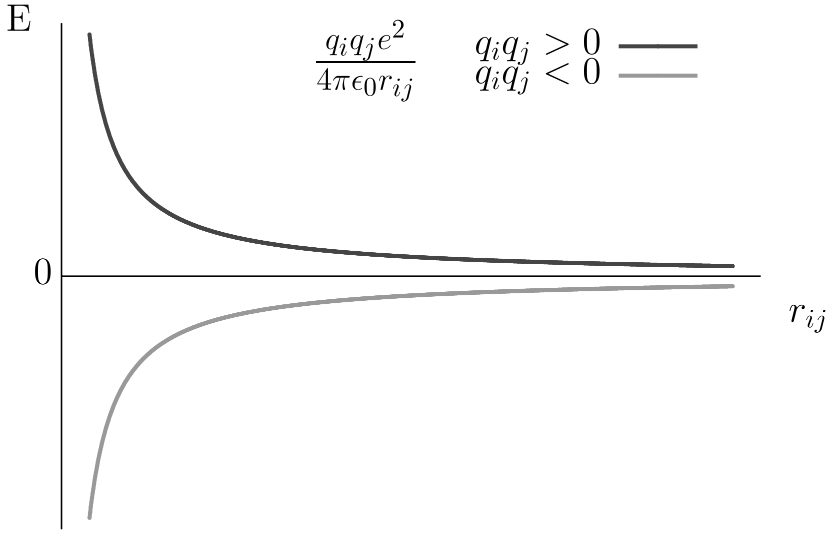


Figure 22: The *Coulomb* potential acting on charged atoms in the system.

Atomic charges There are different ways to acquire the atomic charges for a force field. The appropriate method depends strongly on how the other parameters in the force field are derived. For example one way is a quantum chemical optimisation of the molecule and a subsequent population analysis with a suitable population algorithm, e. g. *CHelpG*.[152] Another way is to derive the atomic charges from experimental values. In the OPLS force field the atomic charges are fit in such a way that the force field reproduces liquid properties like viscosity and density correctly.[148, 149]

Besides, specific molecular subgroups like the amino group (NH_2) or carboxyl group (COOH) are defined, in which the charges of the atoms, building this functional entity, add up to zero. This is necessary for the sake of energy conservation, since those charge groups are used to increase performance in MD algorithms. In parallel computations with *domain decomposition*[153] charge groups can be separated. This would lead to charge separation if the charge groups are not neutral.

Due to these reasons, every atomtype has always the same charge in OPLS force field. Thus, possibly neccessary deviations from this rule have to result in a definition of new atomtypes, if needed.

5.2.4. Atomtypes

To take into account that atoms behave in dependence of their chemical environment, force fields deal with atomtypes instead of just elements. Hence, more than one atomtype exists to describe for example the element carbon. A carbon atom behaves differently if it is part of e. g. a saturated C-C bond or instead of an unsaturated one.

5.2.5. *United Atom* vs. *All Atom*

Several approximations that describe a molecule quite roughly, e. g. the aforementioned *Gay-Berne* potentials[21] or the *Martini* force field[154] description, have been used in MD techniques. There are in the field of atomistic modeling two ways of representing a molecule. The straight forward approach is to describe every atom as it is (*all atom*; *AA*). A computational less demanding but still accurate way is to deal with hydrogens attached to carbon atoms as one entity (*united atom*; UA). Figure 23 on page 34 shows an example of a sugar in these two descriptions. Note that hydrogens participating in polar bonds must not be substituted, since they can participate in hydrogen bond networks.

Especially, when dealing with macromolecules, like proteins, or with large supramolecular assemblies, this approach can save reasonable amounts of computational time.

5.3. Simulation Parameters

Besides the force field, which defines the behaviour of a molecule, a variety of options exist to specify the simulation conditions. With these parameters the

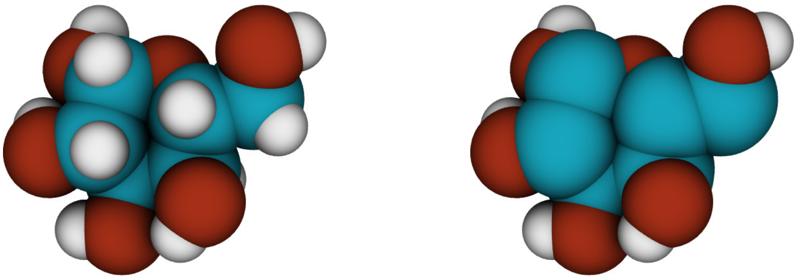


Figure 23: A sugar (Galactose) described in *all atom* (left) and *united atom* (right) fashion. Mind that the polar hydrogens remain unchanged and that the united carbon-hydrogen entities are slightly enlarged.

temperature or pressure can be kept constant, the type of simulation ensemble can be defined, or it can be determined how the electrostatic interactions are evaluated.

5.3.1. Periodic Boundary Conditions

Simulating a material as a bulk with MD techniques has the obstacle that, although thousands of molecules can be simulated at once, the borders of the simulation ensemble influence the conditions in the simulation to a great extend. For example borders can superimpose an order on the system because the molecules tend to align at the surfaces. A "real world bulk material" has a huge number of molecules between two borders or surfaces, far beyond today's simulation possibilities. This makes it necessary to introduce the concept of "Periodic Boundary Conditions" (PBC). This concept is illustrated in a two-dimensional example in Figure 24 on page 35. Every molecule leaving one face of the simulation box enters on the opposite side into the same box. Hence, every face of the simulation sees the image of its opposite face. This means that the size of the simulation box has to be chosen carefully according to the size of the molecules simulated and the cutoff distances of the *van der Waals* and electrostatics potentials to avoid that molecules interact with themselves.

Boxtype For obvious reasons periodic boundary conditions can only be used with simulation boxes that allow a gap free stacking. Three examples of such

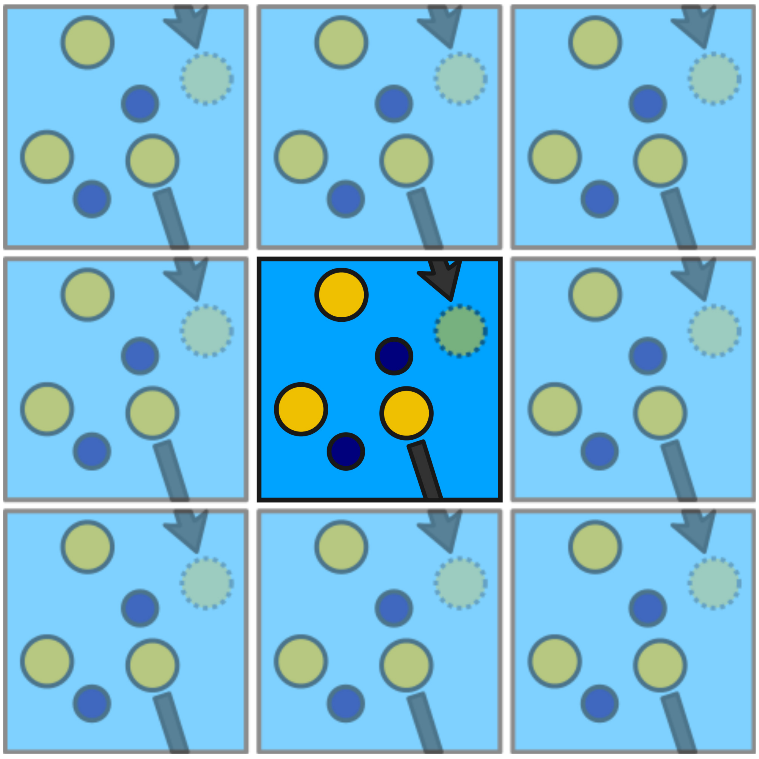


Figure 24: Periodic boundary conditions: objects that leave the simulation box at one side enter the same box again from the opposite face.

suitable cells, a cube or rectangular, a truncated octahedron, and a dodecahedron, are shown in Figure 25.

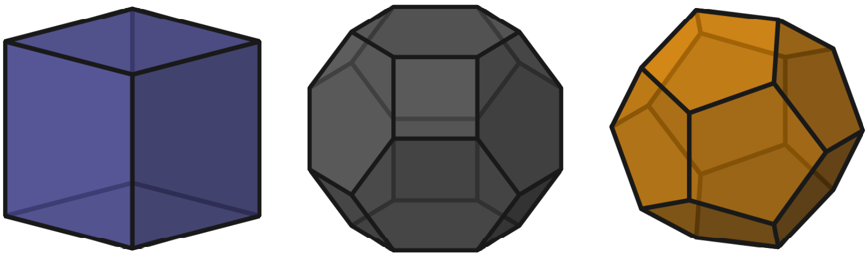


Figure 25: Simulation boxes that can be stacked spacefillingly, i. e. without gaps. From left to right: cube, truncated octahedron, dodecahedron.

Minimum Image Convention When applying periodic boundary conditions, the closest neighbour in a simulation system is not necessarily the one in the same box. Here holds the Minimum Image Convention which states that interactions happening in a simulation are always calculated between the closest, i. e. nearest, image of the molecule.[143, 155] Figure 26 illustrates this in a two dimensional case with periodic boundary conditions. Only the interaction across the boundary with the nearest dark blue circle is evaluated. Thus, it is assured that just one, the closest, image is evaluated to avoid multiple counting of interactions.

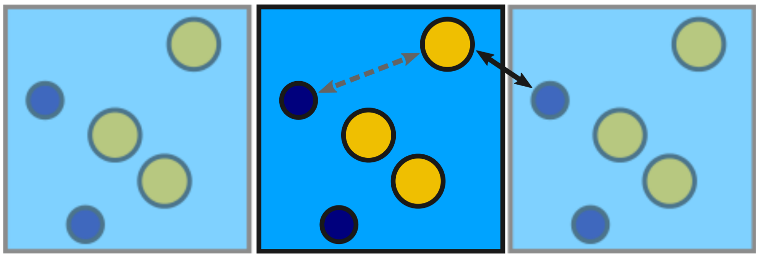


Figure 26: Minimum Image Convention: Only the interaction with the dark continuous arrow, indicating the closest image of the dark blue sphere, is counted.

5.3.2. Temperature Coupling

To accomplish simulating a *canonical* ensemble (nVT), i. e. the number of molecules, the volume, and the temperature is kept constant, the necessity arises to fix the temperature. Thus, a thermostat has to be applied to the system with a reasonable algorithm. Several different algorithms have been evolved and proposed over the years. In the following, two important examples are explained, which are used throughout this work.

***Berendsen* thermostat** The *Berendsen* weak coupling algorithm was proposed by BERENDSEN ET AL. in 1984.[156] The general idea of this thermostat is to couple either parts of or the whole simulation system to a surrounding temperature bath with a fixed temperature T_0. It is realised by calculating velocities according to (15).

$$m_i \frac{d\mathbf{v}_i}{dt} = \mathbf{F}_i + \frac{m_i}{2\tau_{T_B}} \left(\frac{T_0}{T} - 1 \right) \mathbf{v}_i \tag{15}$$

Where m_i and $\mathbf{v}_i$ are the mass and velocity of atom i, τ_{T_B} a damping factor that determines the strength of coupling, and T_0 and T as the heat bath temperature and the actual temperature, respectively. Mind that the same damping factor τ_{T_B} is used for every atom in (15), but it is also possible to define different coupling groups in which the coupling can occur with other damping factors τ_{iT_B}.

This essentially means that the velocities are rescaled as:

$$v = \lambda_T v \tag{16}$$

with

$$\lambda_T = 1 + \frac{\Delta t}{2\tau_{T_B}} \left(\frac{T_0}{T} - 1 \right) \tag{17}$$

Hence, the temperature is adapted by damping the molecular velocities.

***Nosé-Hoover* thermostat** Another variant to keep the temperature in a molecular dynamics simulation constant is the *Nosé-Hoover* algorithm.[157] While the *Berendsen* algorithm works in terms of rescaling the velocities after they were calculated with the standard *Newton* equations of motion the *Nosé-Hoover* algorithm adds a virtual heat bath component as an additional degree of freedom to the equations of motion.[144]

$$m_i \frac{d\mathbf{v}_i}{dt} = \mathbf{F}_i - \frac{m_i p_\xi}{Q} \mathbf{v}_i \tag{18}$$

In this formula p_ξ is a time dependent function to adjust the temperature

$$\frac{dp_\xi}{dt} = (T - T_0) \tag{19}$$

and Q is a constant containing the reference temperature T_0 and the oscillational time $\tau_{T_{NH}}$

$$Q = \frac{\tau_{T_{NH}}^2 T_0}{4\pi^2} \tag{20}$$

The temperature is adjusted in an oscillational fashion, where the oscillational time $\tau_{T_{NH}}$ represents the period which is used to perform one temperature adaption. Thus, this thermostat samples a canonical ensemble and adapts the temperature slower than the strongly damping *Berendsen* algorithm.

5.3.3. Pressure Coupling

Many experimentally acquired data are measured under constant temperature and pressure. To reproduce such results, an isothermal *and* isobaric ensemble is necessary which means that a barostat is needed additionally to the thermostat. Throughout this work two different barostats have been used, the *Berendsen* and the *Parrinello-Rahman* algorithm.

***Berendsen* barostat** In a similar fashion as the *Berendsen* thermostat (cf. Section 5.3.2, page 37) the *Berendsen* barostat adapts the pressure in the simulated system by an exponential damping.[156] While the thermostat rescales velocities, the barostat adjusts the atomic positions with:

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i - \frac{\beta(P_0 - P)}{3\tau_{P_B}} \mathbf{x}_i \tag{21}$$

with $\mathbf{x}_i$ describing the positions, $\mathbf{v}_i$ velocity of the ith particle, P_0 and P being the reference pressure and the actual pressure, respectively.

***Parrinello-Rahman* barostat** Comparably to the fashion in which the *Nosé-Hoover* thermostat adapts the temperature, the *Parrinello-Rahman* thermostat adds the rescaling of the unit cell vectors as an additonal degree of freedom to the equations of motion. Thus, the volume becomes variable in the simulation.[158, 159] Furthermore, the box vectors become independent and an anisotropic scaling is also possible.

In the `GROMACS` tool suite, which was used for this work, this can lead to some problems with deformations of the box. When the system changes from rigid to fluid or isotropic the box tends to abnormally elongate. In such a case the simulation finally stops because other box vectors become shorter than twice the cut-off length for the calculation of *van der Waals* and *Coulomb* interactions.

6. Observables

A lot of information can be drawn out of the so called trajectories, i. e. the coordinates stored with respect to time. These observables can be used to determine whether the system has reached the equilibrium or extract physical informations that can be compared to experimental findings.

6.1. Hydrogen Bonds

According to the IUPAC, the "International Union of Pure and Applied Chemistry",[160] a hydrogen bond is the association between an electronegative atom and a hydrogen atom that is attached to another, rather electronegative atom (cf. Figure 27).

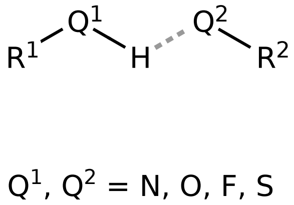


Figure 27: Structural definition of a hydrogen bond.

The electronegativity of the atoms leads to a constant localised distribution of electronic charge. While the hydrogen atom attached to the electronegative atom is partially positively charged (hydrogen bond donor), the hydrogen bond acceptor is a partially negatively charged atom due to its electronegativity. This association is not a bond in the sense of a shared electron pair between the hydrogen atom and the acceptor, but a form of electrostatic attraction. With an energetic contribution of 20–25 kJ mol^{-1} hydrogen bonds can stabilise intra- as well as intermolecular structures and play an important role in supramolecular chemistry and biology.

Beside the general structural definition, two additional aspects are crucial for the formation of hydrogen bonds, the distance between donor and acceptor (hydrogen bond length) and the entrance angle of the hydrogen to the acceptor. In the physical world the length of a hydrogen bond varies according to the temperature, the electronegativity of the participating atoms, and the system pressure. The optimal entrance angle depends strongly on the acceptor atom.

Analysis tools like `g_hbond` of the `GROMACS` suite[153, 161] or the visualisation tool `VMD` (Visual Molecular Dynamics)[162] use the distance between the participating atoms and the angle between the acceptor atom, the donor atom, and the hydrogen atom (cf. Figure 28) to define a hydrogen bond.

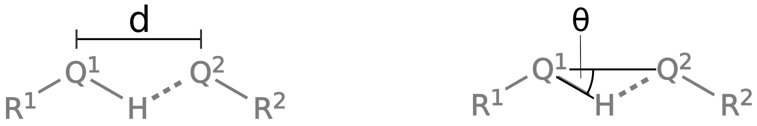


Figure 28: Geometrical description of a hydrogen bond by the distance d between the electronegative centers and the angle θ between the acceptor (Q^2), the donor (Q^1) and the hydrogen.

6.2. Diffusion Coefficient

The macroscopic definition of the diffusion coefficient, is described by *Fick*'s laws. The first law describes the diffusive flux as

$$J = -D\frac{\delta c}{\delta x} \tag{22}$$

where J is the diffusive flux ($\mathrm{mol\,m^{-2}\,s^{-1}}$), i. e. the amount of molecules that move through a certain area in a certain time, c is the concentration ($\mathrm{mol\,m^{-3}}$), x is a spatial component (m), and D is the diffusive coefficient ($\mathrm{m^2\,s^{-1}}$). D is a constant depending on the temperature, and the combination of materials. Figure 29 on page 42 illustrates the relation between a concentration profile c, its spatial derivative, and the resulting diffusive flux. While the sign of the flux curve describes the direction of the flux at a certain concentration, the absolute value shows its strength.

Fick's second law of diffusion

$$\begin{aligned}\frac{\delta \phi}{\delta t} &= -\frac{\delta}{\delta x}J \\ &= D\frac{\delta^2 \phi}{\delta x^2} \qquad (23)\end{aligned}$$

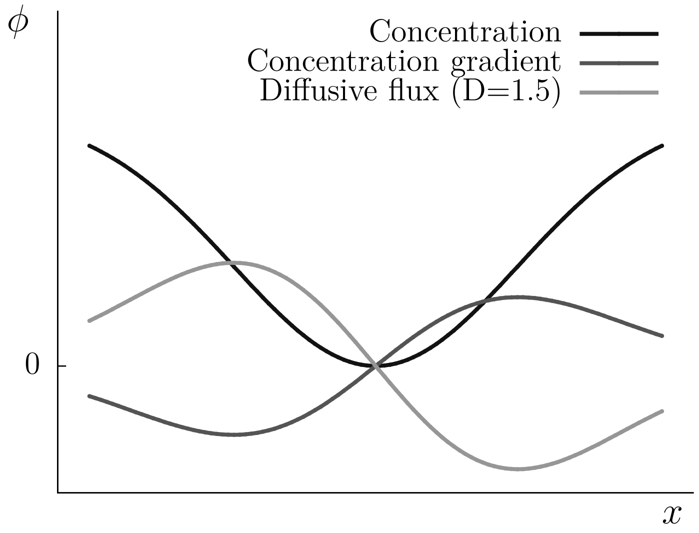


Figure 29: *Fick*'s first law of diffusion. Showing a concentration profile, its gradient, and the resulting diffusive flux with a diffusion coefficient of $D = 1.5$.

states the relation between the change of concentration ϕ over time and the spatial change of diffusive flux J. It shows that under conservation of mass and no chemical reaction the concentration changes according to the flux of the molecules.

If the concentration ϕ is rephrased as a density function dependent on time and space $\rho(x,t)$, a solution for this differential equation can be found as the following *Gaussian* function

$$\begin{aligned}\rho(x,t) &= \frac{1}{\sqrt{4\pi Dt}} \exp^{\frac{-x^2}{4Dt}} \\ &= \frac{1}{\sqrt{2Dt}\sqrt{2\pi}} \exp^{-\frac{1}{2}\left(\frac{x}{\sqrt{2Dt}}\right)^2} \end{aligned} \tag{24}$$

assuming that at time $t = 0$ all particles or concentrated in one point. Since this function is a normal distribution as

$$f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{25}$$

σ and μ can be directly derived as

$$\mu = 0$$
$$\sigma^2 = 2Dt$$

Considering the moments of a normal distribution, which are the expected values $E\left[(x-\mu)^p\right]$ of the normally distributed variable x with the order $p = 1, 2, \ldots$, the first two moments can be found as

$$\begin{aligned} E\left[(x-\mu)^1\right] &= \mu \\ \overline{x} &= 0 \end{aligned} \tag{26}$$

and

$$\begin{aligned} E\left[(x-\mu)^2\right] &= \sigma^2 \\ \overline{x^2} &= 2Dt \end{aligned} \tag{27}$$

That means that there is no preference of the molecule to move back or forth, left or right, up or down, depending on which dimension is considered (first moment) and that there is a linear relation between the mean squared displacement $\overline{x^2}$ and the diffusion coefficient D.

This relation is known as the *Einstein-Smoluchowski* relation.[163, 164] The diffusion coefficient can be determined in a one dimensional case as the slope of a plot of $\overline{x^2}$ over the time

$$D = \overline{x^2}/2t \tag{28}$$

In the two (layer) and three dimensional case (bulk) this equation is modified to

$$D = \overline{x^2}/4t \tag{29}$$

and

$$D = \overline{x^2}/6t \tag{30}$$

since every new dimension adds another degree of freedom in which the molecule can move.

In layered systems the separation of the components of the diffusion coefficient in the direction of the layer and perpendicular to the layer is a possibility to measure the formation and existence of layers. These diffusion coefficients are denoted by $D_\perp$ for diffusion perpendicular to the layer $D_\|$ for diffusion within the layer.

In order to measure these diffusion components appropriately with the `GROMACS` tool `g_msd` the trajectory frames have to be rotated in such a way that the director or the layer normal points in one of the coordinate directions. After that the diffusion in this direction and the diffusion in the other two directions can be measured independently. If the system is rotated in the y-direction then $D_\perp$ and $D_\|$ become

$$D_\perp = D_y \tag{31}$$

$$D_\| = \frac{D_x + D_z}{2} \tag{32}$$

6.3. Order Parameters

An essential aspect of liquid crystals is the definition and determination of the degree of molecular order in the mesophase. Several different order parameters have been defined in literature but amongst them the orientational and positional order parameters are the most important in the field of smectic liquid crystals. In the course of this work a tool was developed to analyse and determine the orientational and positional order parameter in `GROMACS` trajectories. This tool is named `g_order_tensor` and can be found in the appendix (cf. Section B, page V-28).

6.3.1. Orientational Order Parameter

The orientational order parameter is a measure to which degree the molecules in a liquid crystalline material are oriented in a main direction. This main direction is by definition characterised by a unit vector $\mathbf{n}$ that points in the average orientation of all molecules. $\mathbf{n}$ is called director.

Q-Tensor There are several ways to determine the orientational order parameter computationally. One is to form the orientational order tensor $Q_{\alpha\beta}$ for all molecules N in a sample

$$Q_{\alpha\beta} = \frac{1}{N}\sum_{1}^{N}\left(\frac{3}{2}u_{i\alpha}u_{i\beta} - \frac{1}{2}\delta_{\alpha\beta}\right) \qquad \text{with } \alpha, \beta = x, y, z \tag{33}$$

and diagonalise this tensor. $u_{i[\alpha\beta]}$ are the α or β component, respectively of unit vectors indicating the main orientation of the ith molecule. This main direction is in this work determined by the biggest moment of inertia. The diagonalization yields three eigenvalues λ_+, λ_0, and λ_-. The order parameter is derived from the biggest eigenvalue (λ_+). The corresponding eigenvector is at the same time the director of the system. In some cases it can be reasonable to use $-2\lambda_0$ as orientational order parameter since this value undulates around zero for isotropic configurations while λ_+ tends to be slightly above 0.

Second Rank Order Parameter Another approach is the determination of the director by averaging over all orientation vectors of the molecules and the measurement of the angle θ between the director and the molecular orientation axis. By using this value in the second Legendre polynomial $P_2(\cos\theta)$ the second rank order parameter can be calculated as

$$P_2(\cos\theta) = \left\langle \frac{3}{2}\cos^2\theta - \frac{1}{2} \right\rangle \tag{34}$$

where the angle brackets $\langle\rangle$ denote a spatial and temporal average over all molecules.

6.3.2. Positional Order Parameter

The smectic or positional order parameter is based on the idea that on the microscopical level the molecules arrange in layers. Mathematically this can be described by a density wave of the form:

$$\rho(z) = \rho_0 \left[1 + \sum_n \tau_n \cos\left(\frac{2\pi n}{d_{Sm}} z \right) \right] \tag{35}$$

where z denotes the projection of the center of mass of a molecule onto the versor (which is identical with the director in a smectic A phase), ρ_0 is the overall density, d_{Sm} is the smectic layer distance, and τ_n a modulating factor for each contribution of the cosine series. Figure 30 on page 47 shows schematically how the distribution of the centers of mass (COM) can be described by the superposition of e. g. four cosine functions.

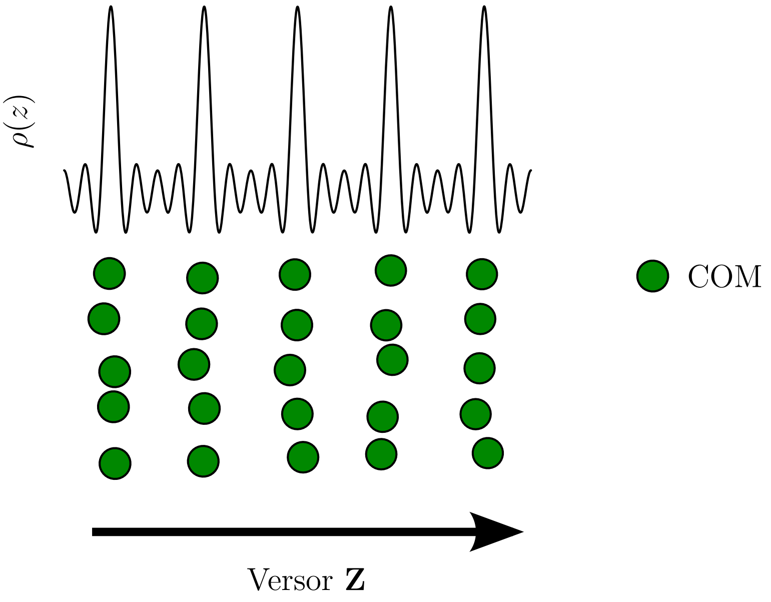


Figure 30: Schematical representation of the density wave $\rho(z)$ of centers of mass (COM) of molecules arranged in layers. The density wave is approximated with the first four terms of (35) on page 46. z is the projection of the position of the COM on the versor **Z**.

Gaetani[127] defined an order parameter based on the work of McMillan[165–168] as a complex quantity:

$$\begin{aligned}\langle\tau_n\rangle &= \left|\frac{1}{N_{mol}}\sum_{j=1}^{N_{mol}}\exp\left(i\frac{2\pi n}{d_{Sm}}z_j\right)\right| \\ &= \left|\frac{1}{N_{mol}}\sum_{j=1}^{N_{mol}}\cos\left(\frac{2\pi n}{d_{Sm}}z_j\right)+i\sin\left(\frac{2\pi n}{d_{Sm}}z_j\right)\right| \\ &= \left|\frac{1}{N_{mol}}\sum_{j=1}^{N_{mol}}\cos\left(\frac{2\pi n}{d_{Sm}}z_j\right)+i\frac{1}{N_{mol}}\sum_{j=1}^{N_{mol}}\sin\left(\frac{2\pi n}{d_{Sm}}z_j\right)\right| \\ &= \sqrt{\left\langle\cos\left(\frac{2\pi n}{d_{Sm}}z\right)\right\rangle^2+\left\langle\sin\left(\frac{2\pi n}{d_{Sm}}z\right)\right\rangle^2} \qquad (36)\end{aligned}$$

with $\langle\tau_n\rangle$ as the nth order parameter, d_{Sm} the layer distance, and z_n the projection of the position of COM of the molecule j onto the layer normal or director (SmA), respectively.

In an MD trajectory the layerspacing d_{Sm} is undefined, hence Gaetani devised an algorithm in O(n) to measure the smectic order parameter and the layer distance at once. The algorithm works by searching the layer distance δ that maximises $\langle\tau\rangle$:

$$\max\left[\langle\tau_n(\delta)\rangle\right] = \max\left[\sqrt{\left\langle\cos\left(\frac{2\pi n}{\delta}z\right)\right\rangle^2 + \left\langle\sin\left(\frac{2\pi n}{\delta}z\right)\right\rangle^2}\right] \tag{37}$$

This algorithm can be used in the same way for bilayer structures and to derive an order parameter for the separate parts of a bilayer:

$$\begin{aligned}\max\left[\left\langle\tau_{nup/down}(\delta)\right\rangle\right] = \\ \max\left[\sqrt{\left\langle\cos\left(\frac{2\pi n}{\delta}z_{up/down}\right)\right\rangle^2 + \left\langle\sin\left(\frac{2\pi n}{\delta}z_{up/down}\right)\right\rangle^2}\right]\end{aligned} \tag{38}$$

where *up* and *down* denote arbitrarily chosen orientations of the molecules with respect to the layer plane.

While Gaetani averages the order parameter over all time frames and all molecules, Palermo introduced a time resolved solution. The order parameters and instantaneous layerspacings are derived per time frame. With this approach fluctuations in the layerspacing do not reduce the instantaneous order parameter.[169]

Furthermore, he found a term that is only dependent on the boundaries chosen and a phase shift of the density wave with respect to the sampling region. The sampling region is a probing cylinder with a fixed radius oriented in the direction of the layer normal. For this he found the analytical solutions for the $\langle\cos\rangle$ and $\langle\sin\rangle$ terms shown before by applying the *rule of the unconsciousness statistician*. The probability function that is used is the density distribution (35) on page 46 with an additional phase shift z_0:

$$\begin{aligned}\left\langle \cos \frac{2\pi n z}{\delta} \right\rangle &= \frac{1}{b-a} \int_a^b \cos\left(\frac{2\pi n z}{\delta}\right) \mathrm{d}z \\ &+ \frac{2}{b-a} \sum_{m=1} \tau_m \int_a^b \cos \frac{2\pi m (z+z_0)}{d} \cdot \ \cos \frac{2\pi n z}{\delta} \, \mathrm{d}z \\ &= \frac{1}{\kappa_n} \sin \kappa_n z \\ &+ \sum_{m=1} \tau_m \left[\frac{1}{A} \sin(Az + k_m z_0) + \frac{1}{B} \sin(Bz + k_m z_0) \right] \Bigg|_a^b \cdot \frac{1}{b-a} \end{aligned} \tag{39}$$

$$\begin{aligned}\left\langle \sin \frac{2\pi n z}{\delta} \right\rangle &= \frac{1}{b-a} \int_a^b \sin\left(\frac{2\pi n z}{\delta}\right) \mathrm{d}z \\ &+ \frac{2}{b-a} \sum_{m=1} \tau_m \int_a^b \cos \frac{2\pi m (z+z_0)}{d} \cdot \ \sin \frac{2\pi n z}{\delta} \, \mathrm{d}z \\ &= -\frac{1}{\kappa_n} \cos \kappa_n z \\ &- \sum_{m=1} \tau_m \left[\frac{1}{A} \cos(Az + k_m z_0) - \frac{1}{B} \cos(Bz + k_m z_0) \right] \Bigg|_a^b \cdot \frac{1}{b-a} \end{aligned} \tag{40}$$

with

$$A = \frac{2\pi(m\delta + nd)}{d\delta} \tag{41}$$

$$B = \frac{2\pi(m\delta - nd)}{d\delta} \tag{42}$$

$$k_m = \frac{2\pi m}{d} \tag{43}$$

$$\kappa_n = \frac{2\pi n}{\delta} \tag{44}$$

Using $b = -a$ in (40) the first term vanishes, while it remains in (39). The latter one can mask the actual maximum and hence, has to be substracted in the algorithm.

This algorithm is used in this work to determine the smectic order parameter, i. e. the distinctness of the layers, and the layerspacing.

6.4. Correlation Functions

Correlation functions offer the possibility to discover relations between observables or properties. Generally, they follow the form

$$N \int f(x_1)g(x_2)dx \tag{45}$$

where N is a normalisation factor, which is dependent on the properties or observables f and g, and x_1 and x_2 are states of the system. They measure the correlation, i. e. the degree of common relation, between property f at state x_1 and property g at state x_2.

Considering the quality of the properties or of the states, several forms of correlation functions can be differentiated.

Auto-Correlation Functions If the functions or properties describe the same aspect, e. g. orientation or mean square deviation, this is called auto-correlation. *Time-correlation* is considered, if the states x_1 and x_2 are time dependent. This is a special form of auto-correlation.

$$N \int f(t_1)f(t_2)dt \tag{46}$$

In equilibrium states holds the condition that properties are invariant to changes of time. Thus, time-correlation functions can be expressed by time differences.

$$N \int f(t_2 - t_1)f(0)dt \tag{47}$$

Which means that a correlation is created between $f(0)$, which is a property f at an arbitrary time, and the same property at a certain time $t_2 - t_1$ after that arbitrary moment.

Time correlation functions can reveal time dependent correlations, like the relaxation time of the system, which is derived from the orientation auto-correlation, or the diffusion, which is the mean square displacement auto-correlation (*Einstein-Smoluchowski* relation, compare Section 6.2 on page 41).

Cross-Correlation Functions If the properties under investigation are different, e. g. distance and orientation, this is called cross-correlation. A special form of cross-correlations are spatial correlation functions. They create a relation between a distance and a property.

This allows to discover structural informations in the system like a dependence between the orientation and the distance of molecules, or the degree of orientational order and the distance.

These type of correlation functions were also built into the developed tool `g_order_tensor` (cf. Section B, page V-28).

III

Results

In the following, the results of several approaches are summarised to address the simulation of carbohydrate liquid crystals. The achievements and obstacles which arise when dealing with polar and partially charged groups will be described. Their contribution to hydrogen bond networks will be shown. The computational reproduction of phase transition temperatures of compound **3** will be discussed. Furthermore, an explanation will be given for the different phase behaviour of compounds **8** and **9**. Additionally, it will be investigated and discussed how racemic mixtures and enantiopure samples of compounds **8** and **9** differ in their phase formation.

7. Ressources and Applications

7.1. Hardware

All calculations have been performed on the CHEOPS and SuGI cluster at the Regional Computing Center Cologne (RRZK). Both systems are High Performance Computing (HPC) resources.

CHEOPS is an low latency InfiniBand coupled HPC cluster with dual socket INCA compute nodes:

- 210 x 2 Nehalem EP quad-core processors (Xeon X5550, 2.66 GHz), 24 GB RAM
- 5 x 2 Nehalem EP quad-core processors (Xeon X5550, 2.66 GHz), 96 GB RAM
- 432 x 2 Westmere hexa-core processors (Xeon X5650, 2.66 GHz), 24 GB RAM
- 170 x 2 Westmere hexa-core processors (Xeon X5650, 2.66 GHz), 48 GB RAM

SuGI is an low latency InfiniBand coupled HPC cluster:

- 32 x 2 Intel quad-core processors (Xeon E5345, 2.33 GHz), 32 GB RAM

The results presented in this work could be obtained with the usage of $7 \cdot 10^6$ CPU hours at the CHEOPS cluster and $7 \cdot 10^5$ CPU hours at the SuGI cluster.

7.2. Software

Throughout this work the `GROMACS` tool suite was used.[153, 170] The MD core application `mdrun` was compiled in version 4.5.5 with MKL support (version 10.3) and Intel® MPI (version 4.0.3) in double precision mode.

To determine the diffusion constant the standard gromacs tool `g_msd` in version 4.5.5 was used. The orientational order and positional order parameters were determined with the tool `g_order_tensor` developed in the course of this work. The spatial correlation functions were calculated with `g_order_tensor` as well.

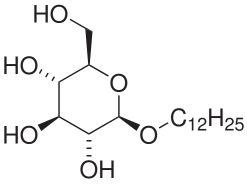
The starting configurations were created with the `packmol` tool in version 1.1.2.023.[171] With this tool it could be ensured that the molecules are distributed isotropically.

The quantum chemical optimisation of the starting structures and the population analysis with the *CHelpG* algorithm[152] were performed with the `Gaussian`[172] application in version 03 revision E01.

Visualisations of the trajectories and the rendering of the molecular representations were performed with `VMD` version 1.9.1 and the `tachyon` renderer version 0.98.9.[162]

8. Bulk investigations of G_1C_{12}

As a model system exhibiting smectic liquid crystalline behaviour compound **3** was chosen (cf. Figure 31).



3

Figure 31: Structure of *n*-dodecyl-β-D-glucopyranoside.

In the following denoted by G_1C_{12}, since the molecule is not actually the described compound, but an approximation, in which some atoms are united (*united atom* description of alkyl moiety). In this case the polar part is a sugar moiety, β-D-glucose, and the apolar part is an dodecyl-alkyl moiety. Due to this difference in polarity a microphase separation is expected to happen. The problem of phase transition temperature reproduction shall be addressed by cooling simulations. This will ensure that the ordered phases are not influenced by expectations, and metastable configurations, which tend to be preserved in MD simulations, are avoided.

8.1. Simulation Setup

The simulation experiment was approached with a moderate system size of 216 molecules in a rectangular box. The starting configurations were created with the `packmol` tool.

All simulations were performed in an nPT ensemble, i. e. the number of molecules, the pressure, and the temperature remain constant. The *Parrinello-Rahman* barostat was used to keep the pressure constant anisotropically at 1 bar with an adjustment frequency of 2 ps and a compressibility of $4.5 \cdot 10^{-5}\,\text{bar}^{-1}$. Two cooling simulation series were performed with the *Nosé-Hoover* thermostat. An adjustment step of every 500 fs was used. The first cooling schedule was applied between 430 K to 400 K with 5 K cooling steps. The system should, according to experiments, exhibit a phase transition from the isotropic liquid to the smectic A mesophase at 410.6 K (cf. Table 1, page 16). The investigation of the results of this simulation lead to a second cooling setup between 490 K and 400 K also with cooling steps of 5 K. The character of the thermostat creates fluctuations which lead to a slightly increased temperature (ca. 2.5 K) with respect to the target temperature. In both calculations the system was simulated at each temperature for 200 ns.

The bonds to hydrogens were constraint with the LINear Constraint Solver (LINCS) algorithm. The neighbour list was updated every 20 calculation steps in a range of 1.4 nm. The short-range electrostatics were cut off at 1.4 nm as well as the *van der Waals* interactions. Those were modelled with a twin range cut-off potential. The long-range electrostatics were modelled with the Particle Mesh Ewald (PME) algorithm with a fourierspacing of 0.3 nm, an order of 3, and an accuracy of 10^{-6}.

The applied integrator was a leap frog algorithm with four random *Gaussian* numbers and an integration time step of 2 fs.

The resulting trajectories were investigated on behalf of their orientational and positional order, and their diffusion coefficient.

8.2. Cooling I

The first cooling simulation was performed around the expected clearing point, i. e. the transition temperature from the isotropic liquid into a smectic A phase, $T_{\mathrm{SmAIso}} = 410.6\,\mathrm{K}$.

8.2.1. Orientational Order

Although the temperature range was chosen to reproduce the isotropic to SmA transition, the simulation did not reveal any onset of orientational order (cf. Figure 32, page 58). The onset of orientational order is a prerequisite for the formation of smectic A layers (cf. Figure 2, page 8).

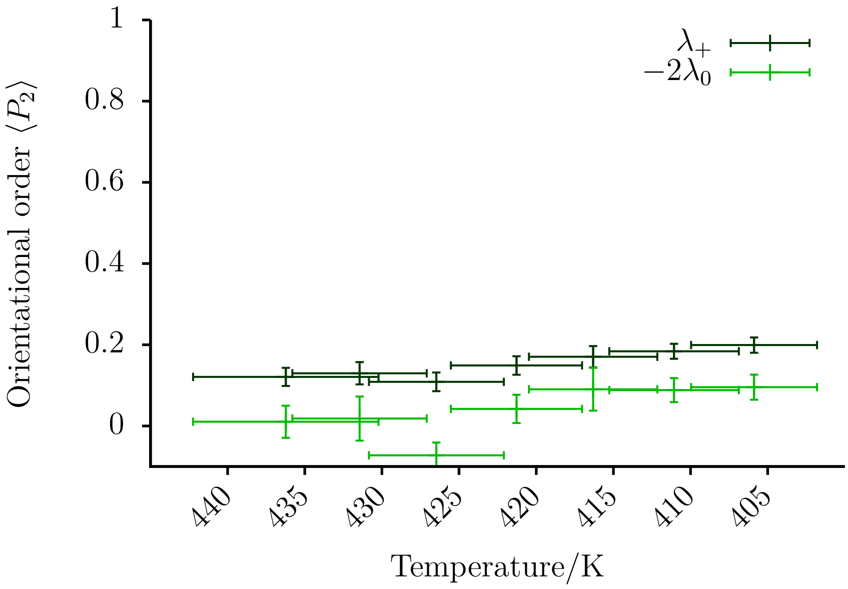


Figure 32: Evolution of the orientational order parameter of G_1C_{12} around the originally expected transition temperature T_{SmAIso}. Here λ_+ as well as $-2\lambda_0$ are plotted for the same trajectory to indicate the difference in their value for isotropic configurations, i. e. $-2\lambda_0$ slightly lower than λ_+.

8.2.2. Self-diffusion

By checking the diffusion coefficient and plotting it in an *Arrhenius* plot, it became obvious that there was a distinct change in the trend (cf. Figure 33, page 59). Since the diffusion coefficient does not change below 420 K, this indicates that there is a transition into a compacted state, in which the mobility is greatly reduced.

DIMITROV introduced the *law of equal MSD.*[173] He states that the mean square displacement (MSD) at the transition from the solid to the liquid phase is the same independent of the nature of the solid or liquid. That the MSD is equal at the glass-liquid and solid-liquid transition has also been stated in a recent review by ANGELL.[174] Considering the transition from a liquid state to a solid phase, the abrupt change in diffusivity indicates the transition to a solid and marks the so called *jamming* temperature. At this temperature all molecules are compacted and caged in by their neighbouring molecules in such a way, that every movement of a molecule does not allow a macroscopic concerted movement. The temperature measured here is not the actual temperature of the transition from the SmA mesophase to crystal but a temperature below at which the system prepares for a glass state formation. This temperature is always below the T_{CrSmA}. Hence, the term solid depicts in the following a supercooled amorphous metastable system. This state is dependent on the cooling rate used in this simulation study and the time scales. A direct comparison to the liquid to crystal transition temperature can not be drawn.

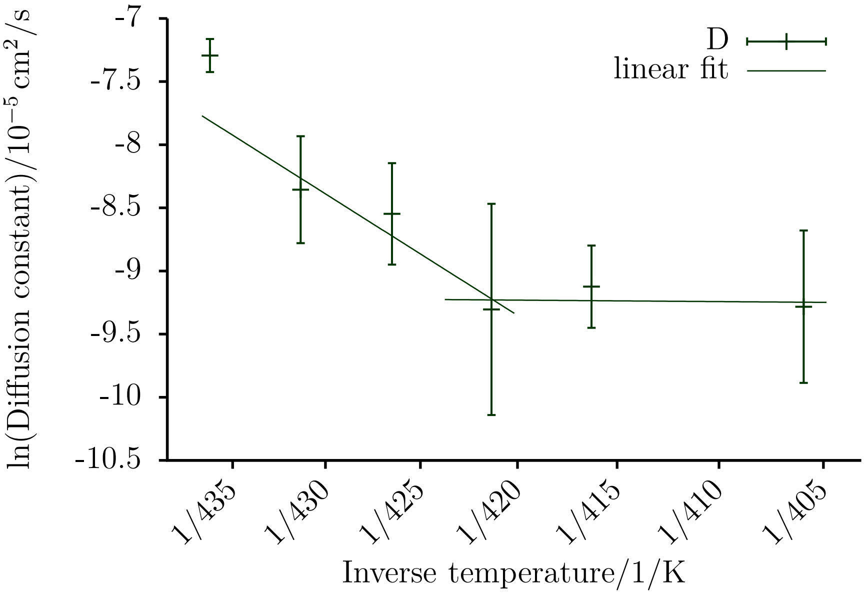


Figure 33: Evolution of the self-diffusion of G_1C_{12} around the experimentally detected transition temperature T_{SmAIso} in an Arrhenius plot.

8.3. Cooling II

Assuming that the force field representation G_1C_{12} could reproduce a SmA phase and observing that a transition into a solid happens at 420 K, it was concluded, that the force field overestimates the transition temperatures. A new experiment was designed to confirm this assumption. Since the experimental transition temperature from the smectic A phase to the solid was measured to be $T_{CrSmA} = 353\,K$, the calculation overestimates the experimental findings by 67 K. Hence, the investigated temperature range was shifted and elongated and a simulation between 490 K to 400 K was performed.

8.3.1. Orientational Order

In this simulation an onset of orientational order could be observed at around 475 K, where the order parameter $\langle P_2 \rangle$ goes up to 0.6 (cf. Figure 34). In the range between 490 K and 475 K orientational preorganisation happens, indicated by a jump of $\langle P_2 \rangle$ from 0.1 to 0.3. This is in perfect agreement with the hypothesis that the force field represents the correct phase sequence but overestimates the temperature by about 70 K.

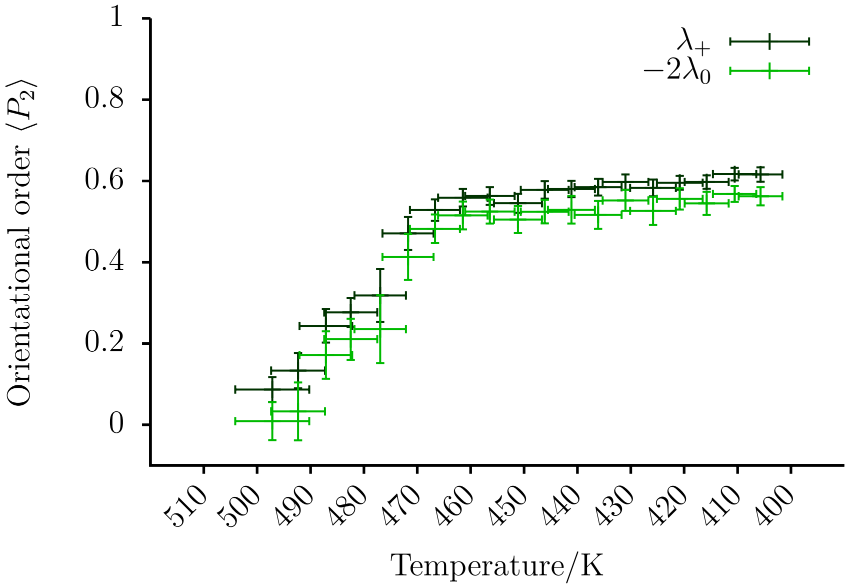


Figure 34: Development of the orientational order parameters λ_+ and $-2\lambda_0$ of 216 molecules of G_1C_{12}.

8.3.2. Positional Order

Though the formation of orientational order is a necessary condition for the formation of a SmA phase, it is not sufficient to confirm its existence. It is obligatory to measure the positional or smectic order parameter to prove the formation of layers.

In the same temperature region, in which the orientational preorder happens, also an onset of the positional order, i. e. the formation of layers, is detectable. The smectic order parameter τ also rises to a value of almost 0.4 at 475 K. The smectic order parameter is a measure for the distinctness of the layers. Since the phase formed is a SmA_d phase, the parts of the double layers can be considered separately as "upwards" and "downwards" oriented molecules. Looking at the layers separately they become even more distinct (cf. Figure 35, page 61). Due to the interdigitation of the layers the order parameter of the double layer is smaller than the one of the separate layers.

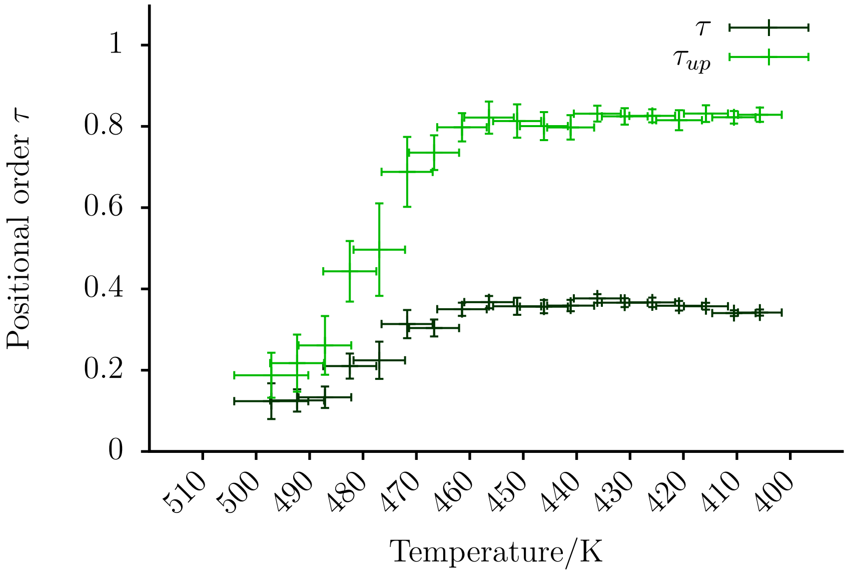


Figure 35: Evolution of the positional order parameter τ of 216 molecules of G_1C_{12}. τ_{up} denotes the positional order of the molecules oriented in the same arbitrarily chosen direction in the bilayer.

8.3.3. Layerspacing

Also the measurement of the layerspacing shows the formation of layers between 485 K and 475 K with a high degree of fluctuations. This ends in a slightly rising, very confined layerspacing of 3.5 nm at temperatures below 475 K. The maximum value is 3.7 nm at 440 K, which reduces again to 3.5 nm on further cooling to the solid state (cf. Figure 36, page 62).

8.3.4. Lateral Self-diffusion

Measuring the self-diffusion of a system can give valuable hints on its phase state. A distinctive property regarding the self-diffusion in smectic phases is

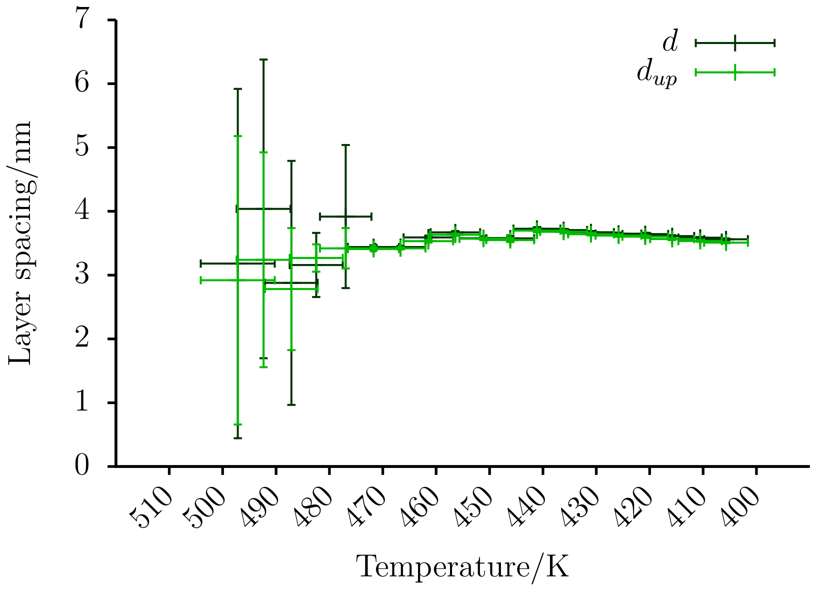


Figure 36: Evolution of the layerspacing of 216 molecules of G_1C_{12}. Above 475 K no layers are present, i. e. the layerspacing is indefinite. d: layerspacing between whole layer; d_{up}: layerspacing between layers of "upwards" oriented molecules.

the lateral diffusion $D_{\|}$ in the layers, which is significantly higher than the diffusion in the direction of the layer normal $D_{\perp}$.

In Figure 37 on page 63 a clear split can be observed. At 470 K the parallel diffusion $D_{\|}$ becomes two orders of magnitude higher than $D_{\perp}$ (the natural logarithm is plotted on the y-axis).

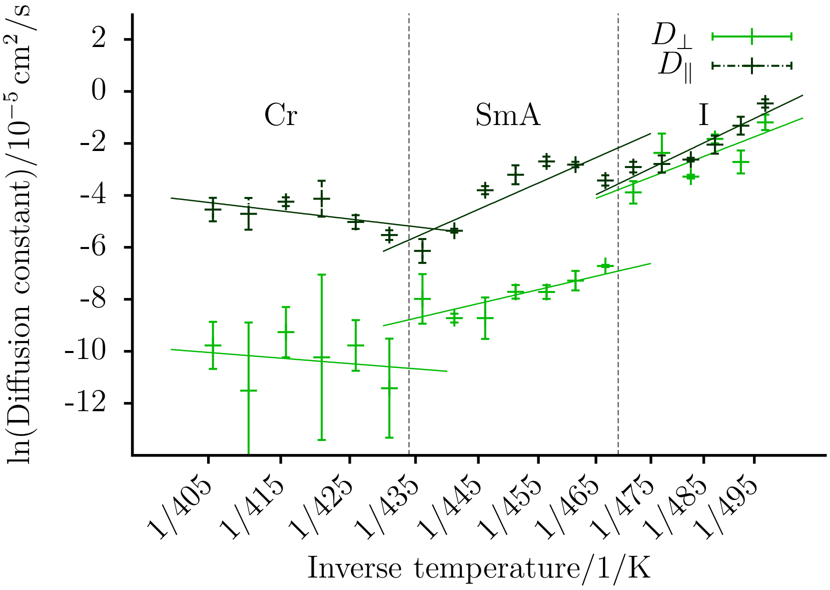
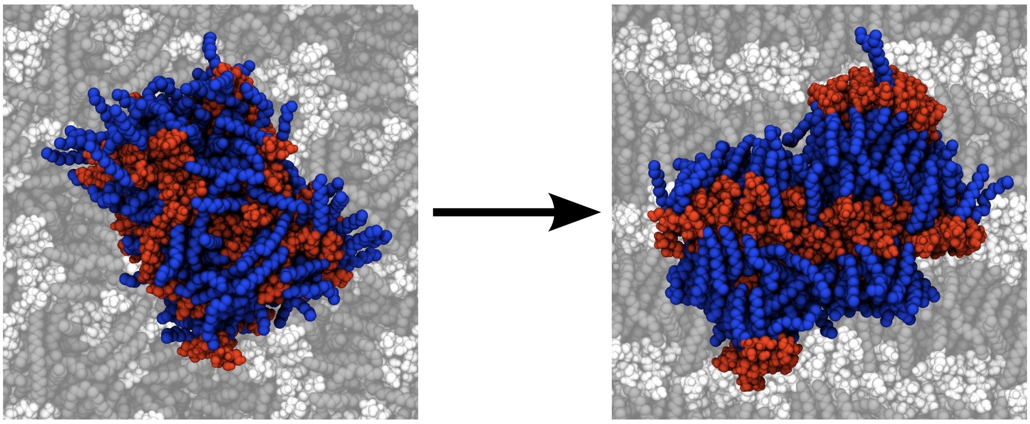


Figure 37: Evolution of the self-diffusion coefficient of 216 molecules of G_1C_{12} divided into perpendicular ($D_\perp$) and parallel ($D_\parallel$) movements with respect to the layers. A drop in $D_\perp$ indicates the transition to a SmA phase.

8.4. Summary

It was possible to simulate the phase sequence of **3** with a combined *all atom/united atom* model. The visual differences between the calculated isotropic state and the smectic A mesophase are depicted in Figure 38. The model overestimates the transition temperatures by a constant value of 70 K. It could be shown that an onset of orientational order appears in parallel with the onset of positional order. Furthermore, a clear differentiation between lateral diffusion inside the layer and perpendicular diffusion, i. e. in the direction of the layer normal could be observed.

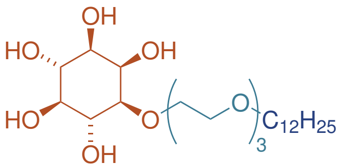


497 K 406 K

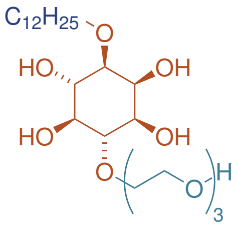
Figure 38: Transition from an isotropic configuration of 216 molecules of G_1C_{12} into a smectic A phase. The formation of layers can be observed. Blue: alkyl chains; red: sugar moieties; gray: periodic images.

9. Cooling of UA-$C_{12}E_3I_1$ and UA-$E_3I_1C_{12}$ in a bulk

Aim of this work was to investigate the structural properties and differences of the two compounds **8** and **9** (cf. Figure 39).



8



9

Figure 39: Structures of the two model compounds 1-*O*-[2′-[2″-[2‴-(dodecyloxy)ethoxy]ethoxy]ethyl]-*myo*-inositol (**8**) and 1-*O*-dodecyl-4-O-[2′-[2″-[2‴-(hydroxy)ethoxy]ethoxy]ethyl]-*myo*-inositol (**9**).[44]

Compounds **8** and **9** are represented by a *united atom* description. Hence, they will be denoted in the follwoing by UA-$C_{12}E_3I_1$ and UA-$E_3I_1C_{12}$, respectively. UA stands for the *united atom* character of the whole compound, C_{12} for the alkyl moiety, E_3 for the triethylene glycol substructure, and I_1 for the *myo*-inositol substructure.

9.1. Simulation Setup

The simulation setup was chosen as described in Section 8.1 on page 56. The starting configurations of the samples were generated with 1024 molecules of one compound in a cubic box. With the `packmol` tool it was ensured that there was an isotropic distribution of the molecules.

The simulations were performed in an nPT ensemble. The pressure was kept anisotropically constant at 1 bar with the *Parrinello-Rahman* barostat with an adjustment frequency of 2 ps and a compressibility of $4.5 \cdot 10^{-5}\,\text{bar}^{-1}$. A *Nosé-Hoover* thermostat was used to keep the temperature around a constant value with an adjustment frequency of every 500 fs. A cooling schedule was applied between 420 K to 400 K with 2.5 K cooling steps. Every temperature was simulated for 144 ns. The system should exhibit a transition from the isotropic liquid to the smectic A mesophase at 426.15 K according to experiments (cf. Table 2, page 19). The thermostat does not keep the temperature constant exactly. Fluctuations were detected which led to a slightly increased temperature (ca. 2.5 K).

The bonds between heavy atoms and hydrogens were constraint with the LINCS algorithm and the neighbour list was updated every 20 calculation steps in a range of 1.4 nm. The short-range electrostatics were cut off at 1.4 nm as well as the *van der Waals* interactions. Those were modelled with a twin range cut-off potential. The long-range electrostatics were modelled with the Particle Mesh Ewald (PME) algorithm with a fourierspacing of 0.3 nm, an order of 3, and an accuracy of 10^{-6}.

The applied integrator was a leap frog algorithm with four random *Gaussian* numbers and an integration time step of 2 fs.

The resulting trajectories were subjected to analyses regarding the evolution of order (short-range and long-range), the diffusion coefficient, and the formation of the hydrogen bond networks.

9.2. Formation of Order

During the course of the simulation it became obvious that the mobility of the molecules is quite reduced due to their tendency to form O—H$\cdots$O hydrogen bond networks. These interactions are possible between the inositol substructures itself and between the inositol groups and the triethylene glycol moieties. Thus, a strong tendency to form clusters or regions with strongly interacting parts (sugars; ethoxy groups) and weakly interacting parts gathered (alkyl) could be observed (cf. Figure 40, page 67 and Figure 41, page 67).

Remarkable was the formation of a diffuse monodomain in the model of the non-liquid crystalline compound (UA-$E_3I_1C_{12}$) while the representation of the liquid crystal (UA-$C_{12}E_3I_1$) was characterised by microdomains.

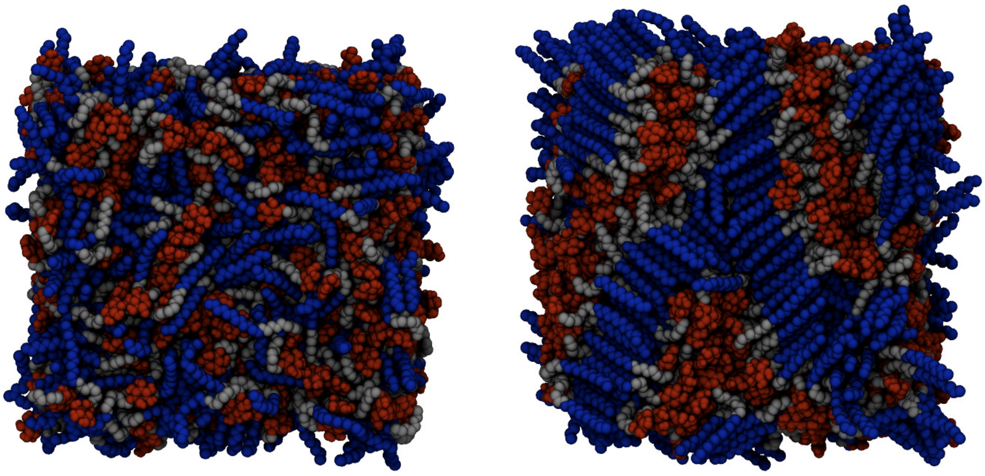


Figure 40: Snapshots of 1024 molecules UA-$C_{12}E_3I_1$ in a rectangular box at the beginning (left) and at 1296 ns (right) of a cooling simulation. Blue: alkyl; red: sugar; gray: triethylene glycol.

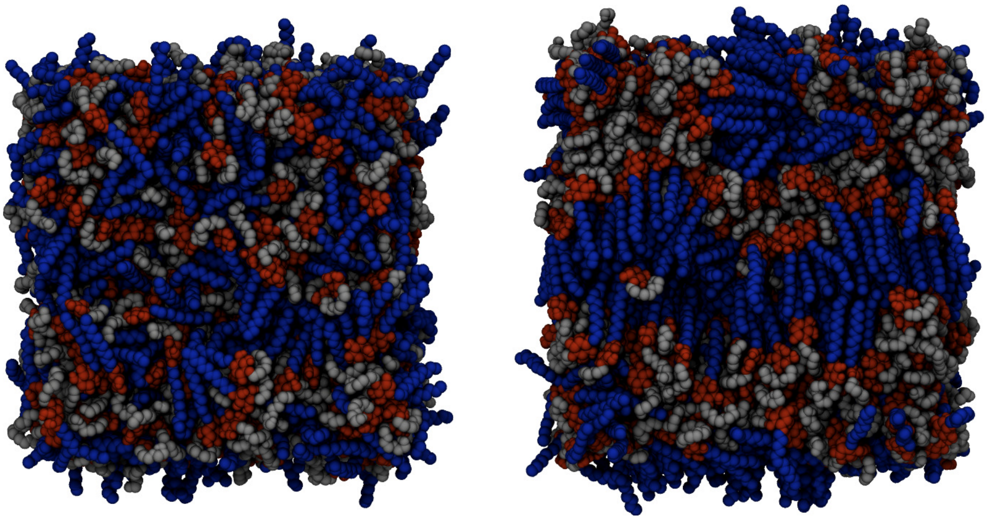


Figure 41: Snapshots of 1024 molecules UA-$E_3I_1C_{12}$ in a rectangular box at the beginning (left) and at 1296 ns (right) of a cooling simulation. Blue: alkyl; red: sugar; gray: triethylene glycol.

9.2.1. Orientational Order

The trajectories were examined according to the orientational order parameter to gain a deeper understanding of the structural differences of compounds **8** and **9**. The orientational order parameter $\langle P_2 \rangle$ was determined from the orientational order matrix (cf. Section 6.3.1, page 45). λ_+ and $-2\lambda_0$, i. e. the largest and middle eigenvalue of the orientational matrix Q, were taken into consideration as orientational order parameter $\langle P_2 \rangle$. As mentioned before, if the system is disordered, λ_0 gives values closer to zero.

Figure 42 plots the evolution of order during the simulation of UA-$C_{12}E_3I_1$. Although the order slightly rises at around 413 K while cooling the system, λ_0 stays at about zero and λ_+ stagnates around 0.2 after further lowering the temperature. Obviously, according to the long range orientational order the system gets stuck in a disordered state.

Visual inspection of the trajectory nonetheless gives the impression that there are microdomains present in the bulk (cf. Figure 43, page 69). These confined structures of alkyl or inositol aggregates can not be examined by the averaging method used for the $\langle P_2 \rangle$. Hence spatial correlation functions are used to determine the short-range structure in the system.

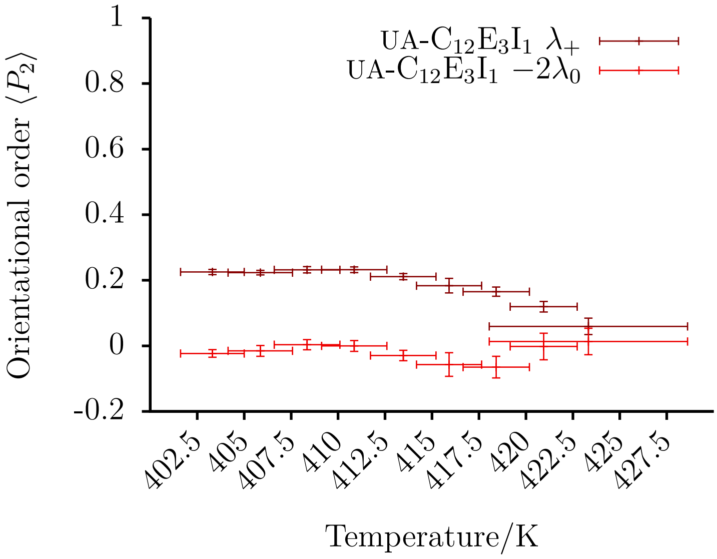


Figure 42: Evolution of the orientational order parameter $\langle P_2 \rangle$ of 1024 molecules of UA-$C_{12}E_3I_1$. λ_0 gives a value closer to zero for isotropic configurations.

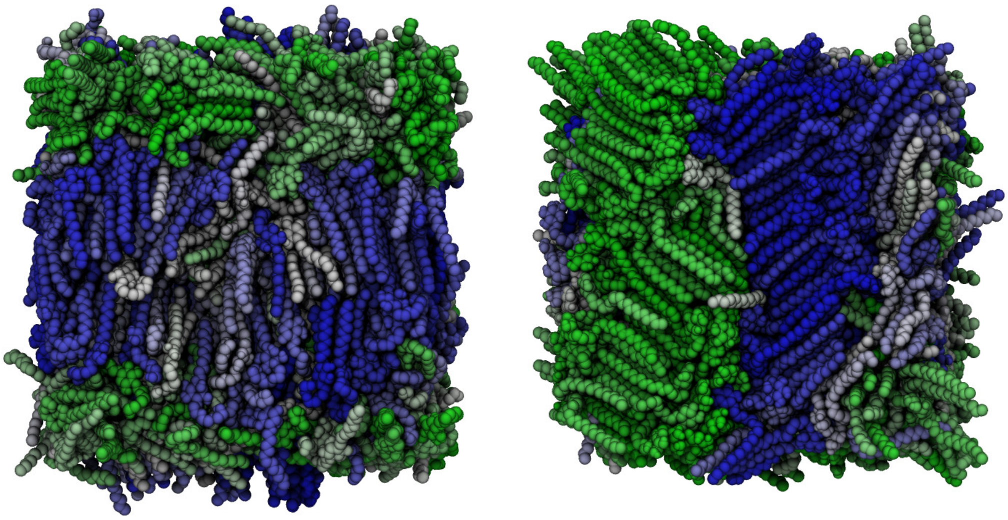


Figure 43: Snapshots of 1024 molecules UA-$E_3I_1C_{12}$ (left) and UA-$C_{12}E_3I_1$ (right) in a rectangular box at 1296 ns of a cooling simulation. The molecules are colored according to their deviation of a fixed axis: blue: $\|$; green: $\perp$; white: 45°. The axis was chosen to represent the orientation in the biggest ordered domain.

Another picture is drawn by the simulation of UA-$E_3I_1C_{12}$.Here a continuous rise of orientational order can be observed and the trends of λ_0 and λ_+ are the same (cf. Figure 44). Even visual inspection reveals the onset of a monodomain (cf. Figure 43, page 69).

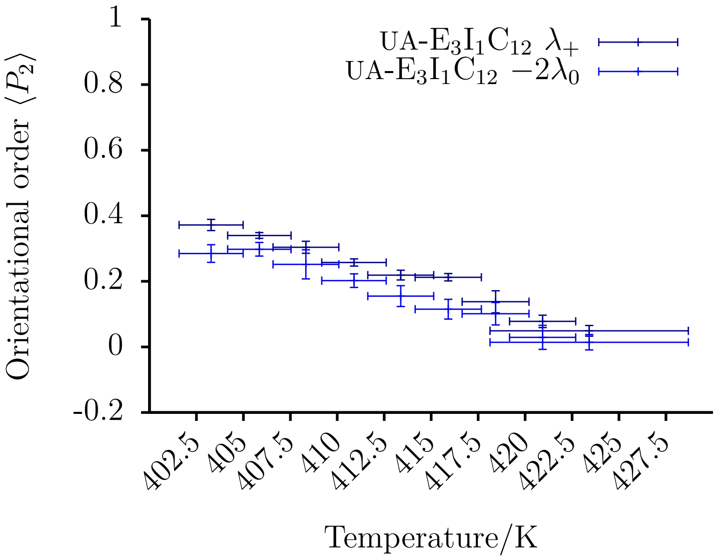


Figure 44: Evolution of the orientational order parameter $\langle P_2 \rangle$ of 1024 molecules of UA-$E_3I_1C_{12}$ in a rectangular box at 1 bar.

Comparing these findings with the actual expectations shown in Figure 45 it becomes clear that the simulated system does not reproduce the real world findings. This can happen due to the hydrogen bonds, which slow down the system. Another option would be that the force field parameters overestimate the transition temperatures which would mean that the system is probably supercooled.

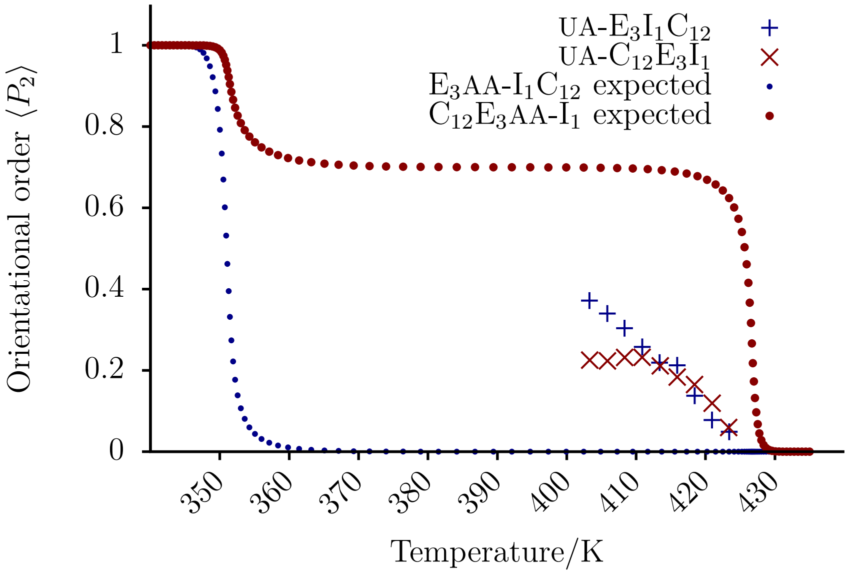


Figure 45: Expected evolution of orientational order (dots) and simulated evolution of order.

9.2.2. Spatial Correlation Function

The spatial orientation correlation functions (cf. Section 6.4, page 51) represent the avarage values of the Legendre polynomials $P_L(\mathbf{u}_i \cdot \mathbf{u}_j)$ in correlation to their distance. Here $\mathbf{u}$ denotes the longitudinal axis of a molecule. The g_0 correlation function is basically the standard radial distribution function. g_1 shows the correlation between the distance and the cosine of the angle β_{ij} between the tested molecules ($P_1(\mathbf{u}_i \cdot \mathbf{u}_j) = \cos(\beta_{ij})$). Finally, g_2 measures the correlation between the orientational order parameter ($P_2(\mathbf{u}_i \cdot \mathbf{u}_j) = \frac{3}{2}\cos^2(\beta_{ij}) - \frac{1}{2}$) and the distance. These functions can be used to derive information about the local or short range structure in the system.

Figure 46 on page 72 shows the spatial orientation correlation functions for the systems with 1024 molecules of UA-$C_{12}E_3I_1$ and UA-$E_3I_1C_{12}$ for every temperature in the cooling schedule. In both systems an organisation can be observed over time, though very different. The measure describes the probability of finding another molecule in a certain distance. While UA-$E_3I_1C_{12}$ evolves a peak at 0.6 nm, which is based on the dense packing of the molecules, there is no additional structure detectable, especially not beyond one molecular length. Due to different conformations this length lies between 2.0 nm and 2.7 nm. UA-$C_{12}E_3I_1$ on the contrary shows a more confined peak at 0.6 nm, another small maximum at 2.1 nm, and a slight minimum at 1.4 nm. This indicates a strong order at 0.6 nm distance, a slightly increased probability of finding another molecule in a distance of 2.1 nm, and a reduced probability to find a molecule at 1.4 nm.

An even bigger difference can be seen in the g_1. In the UA-$E_3I_1C_{12}$ simulation, the molecules that can be found in the direct proximity of a molecule tend to be oriented rather in the same direction. This can be derived from the cosine that is giving in average positive values. Combining this with the form of g_2 it can be seen that the molecules close to each other tend to be aligned alongside to each other (the local orientational order shown by g_2 is around 0.7) with a slight higher tendency to be oriented in the same direction.

For UA-$C_{12}E_3I_1$ there is a clear orientational dependence between molecules in a distance of 0.6 nm or 2.7 nm. While molecules tend to be parallel (the average cosine value is 0.6), in case that they are close to each other, they tend to be antiparallel in a distance of 2.7 nm, since the average cosine becomes negative around that value. This trend is also confirmed by g_2, which has a peak at 0.6 nm, and a shoulder in the curve at 2 nm.

9.3. Self-diffusion

Figure 47 on page 73 plots the diffusion coefficients of UA-$C_{12}E_3I_1$ and UA-$E_3I_1C_{12}$ in an Arrhenius graph. While the graph for UA-$E_3I_1C_{12}$ shows a continuous linear relation between the data points, the graph of UA-$C_{12}E_3I_1$ has a change in slope at 413 K. This change indicates the transition into another

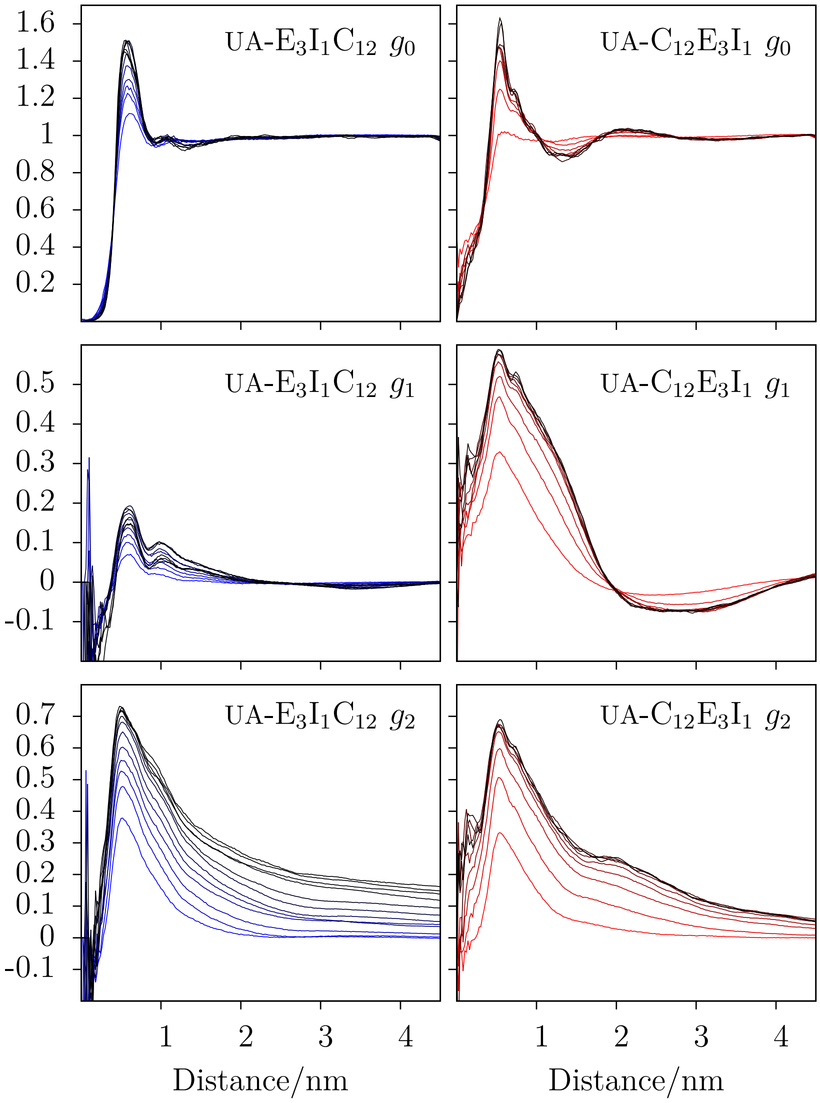


Figure 46: Spatial correlation functions of 1024 molecules of UA-$E_3I_1C_{12}$ (left) and UA-$C_{12}E_3I_1$ (right). The colors from light to dark denote the simulations at 420 K to 400 K in 2.5 K cooling steps. In both simulations a trend of self-organisation is evident.

phase. Based on the observations regarding the local order, a possible explanation could be a transition from the SmA mesophase into a solid. Thus, the force field parameters overestimate the transition temperatures of the compound UA-$C_{12}E_3I_1$. If the same holds for the UA-$E_3I_1C_{12}$ compound the system is in a supercooled state and is forming a locally ordered glass. The direct isotropic to crystalline phase transitions are not easy to simulate since the formation of a crystal from a liquid takes longer than the time scales which can be simulated with Molecular Dynamics. Especially in systems with that high amount of internal degrees of freedom the computational costs simply run out of any reasonable amount. Thus, the formation of an amorphous solid, i. e. glass, is highly probable.

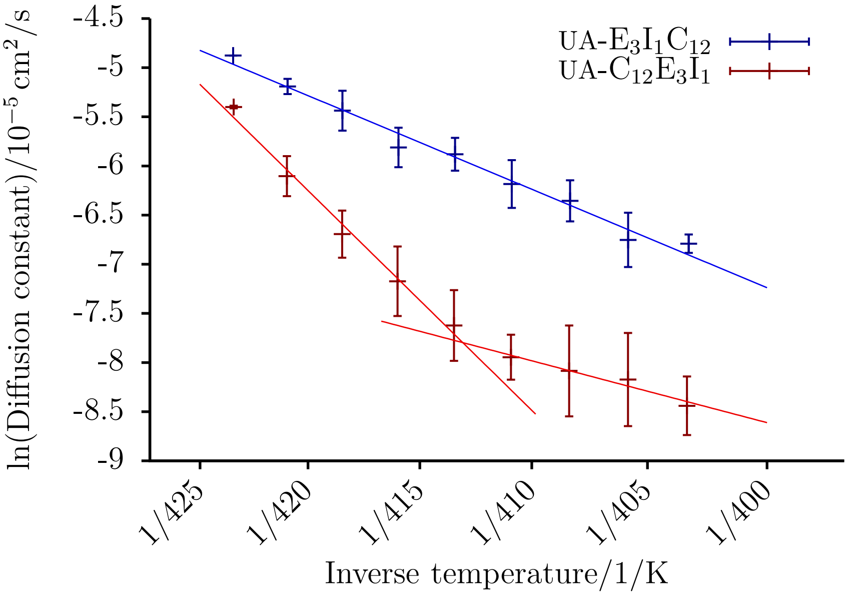


Figure 47: Self-diffusion of 1024 molecules of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$ plotted in an Arrhenius plot. The straight lines show a fit of the data points against a linear function.

9.4. Hydrogen Bonds

A second observation with regard to the diffusion coefficient is that the mobility in UA-$E_3I_1C_{12}$ is one order of magnitude higher than that in the simulation of compound UA-$C_{12}E_3I_1$. Especially intermolcular interactions have an important influence on the mobility. The prominent forces that play a role in this inositol-based compounds are hydrogen bonds. In conjunction with the triethylene glycol moiety and other inositol moieties O—H···O hydrogen bonds can be established.

These hydrogen bonds form between the inositol groups, as homo interactions, and between the cyclitol units and the triethylene glycol chains, as hetero interactions. The simulations show that in general UA-$C_{12}E_3I_1$ tends to have 2.5 times more inositol-based hydrogen bonds than UA-$E_3I_1C_{12}$ (cf. Figure 48). One reason for that finding is surely that UA-$C_{12}E_3I_1$ has one free hydroxyl group more than UA-$E_3I_1C_{12}$. Another point is the terminal position of the inositol in UA-$C_{12}E_3I_1$, which due to this situation is sterically less hindered.

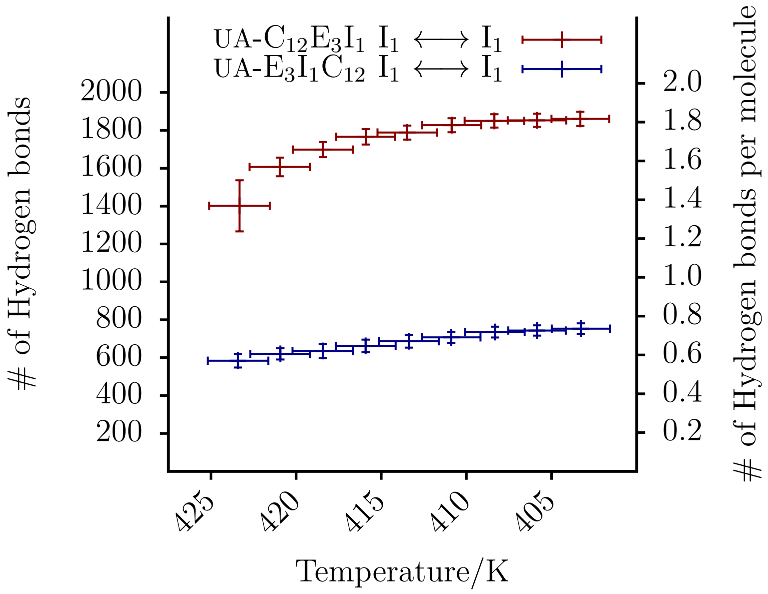


Figure 48: Number of intermolecular hydrogen bonds of 1024 molecules of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$ based on inositol-inositol interactions in the whole system and averaged per molecule. The amount of inositol-inositol O—H···O hydrogen bonds is 2.5 times higher in UA-$C_{12}E_3I_1$. I_1: inositol substructure.

While measuring the hetero interactions it appeared that UA-$E_3I_1C_{12}$ has two to three times more hydrogen bonds between inositols and the ethoxy groups than UA-$C_{12}E_3I_1$ (cf. Figure 49, page 75). Splitting them in inter- and intramolecular contributions (cf. Figure 50, page 76) it can be seen that there are three times more interactions within the molecule in UA-$E_3I_1C_{12}$ than in UA-$C_{12}E_3I_1$. This explains the finding of the strongly reduced mobility in UA-$C_{12}E_3I_1$, since there are way more sugar-based intermolecular forces.

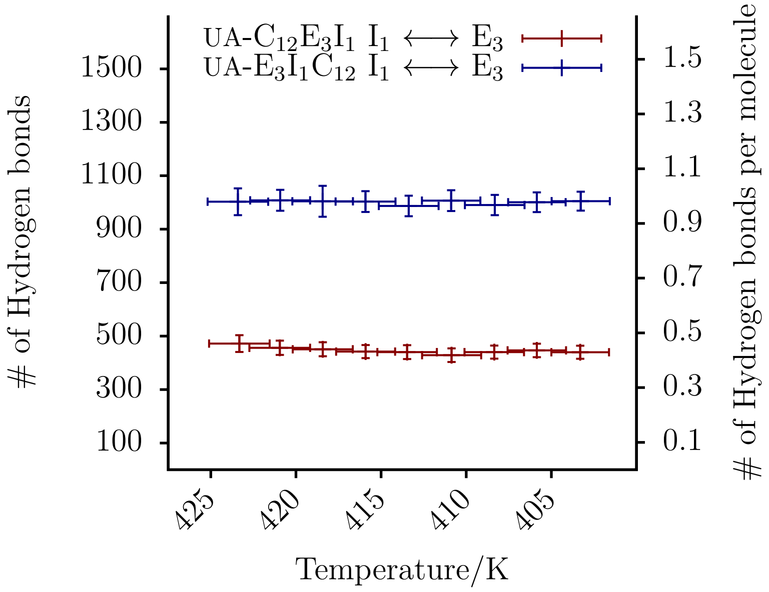


Figure 49: Number of intermolecular and intramolecular hydrogen bonds of 1024 molecules of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$ based on inositol-ethoxy interactions in the whole system and averaged per molecule. The inositol-ethoxy interactions are two times higher in UA-$E_3I_1C_{12}$. I_1: inositol substructure; E_3: triethylene glycol substructure.

9.5. Summary

A cooling simulation was performed with a *united atom* OPLS force field description of the enantiopure compounds **8** and **9**. The diffusion coefficient suggests that for UA-$C_{12}E_3I_1$ a transition into a solid takes place at 413 K.

Compared with experimental findings (cf. Table 2, page 19), this means an overestimation by 60 K. The mobility in UA-$E_3I_1C_{12}$ is one order of magnitude higher than in UA-$C_{12}E_3I_1$. This is due to an intramolecular shielding of the sugar moiety by the triethylene glycol moieties. Thus, the formation of intermolecular hydrogen bonds is reduced in comparison to UA-$C_{12}E_3I_1$. This findings are in agreement to the qualitative assumption of CATANOIU, who stated that

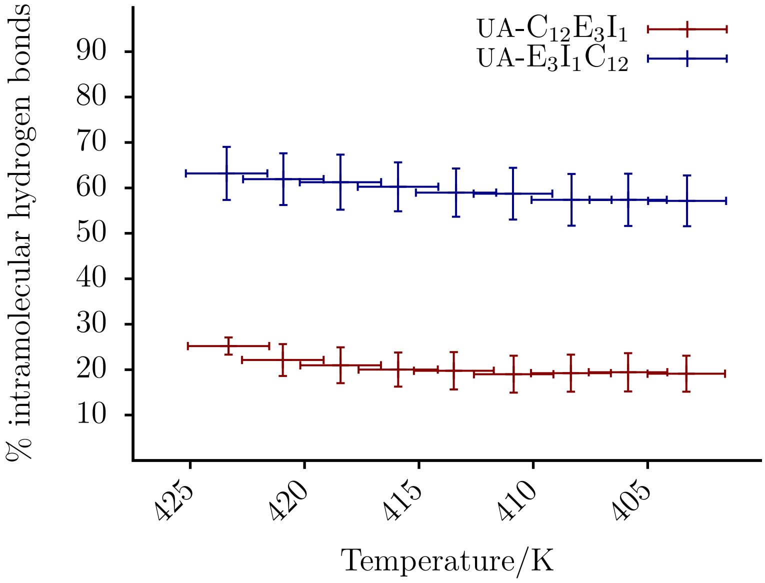


Figure 50: Percentage of intramolecular O—H···O hydrogen bonds of all O—H···O hydrogen bonds in systems of 1024 molecules of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$, respectively.

the flexible terminal triethylene oxide chain disturbs the formation of the hydrogen bond network. This hydrogen bond network built up by the terminal cyclitols in UA-$C_{12}E_3I_1$ is indispensable for the formation of the highly ordered layered structure.[43]

In both computational experiments the formation of locally ordered microdomains was possible.

10. Heating UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$ in a bulk material

The cooling simulations of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$ revealed that the transition temperature T_{SmAIso} for UA-$C_{12}E_3I_1$ and T_{CrIso} for UA-$E_3I_1C_{12}$ could not be easily obtained.

An alternative approach was to simulate, instead of observing the onset of order in a cooling simulation, the extinction of order, i. e. the melting into the liquid crystal or the isotropic liquid, in a heating simulation. Since the crystal structure was not experimentally determined an almost crystalline or highly ordered starting state should be created via simulation techniques. The simulations performed before showed that the experimentally determined bilayer structure could be reproduced. This confirmed that a layered structure is system inherent. Thus, the creation of a bilayered structure could be a reasonably good starting point for a highly ordered starting configuration. The preparation of this kind of starting structure could be performed by creating thin free standing films in a vacuum simulation and stack them.

10.1. Thin Films

The generation of thin films appeared to be a valid approach to generate layered structures in a bulk. The operational procedure was to generate free standing, thin films by organising a sample of compounds between a vacuum space, i. e. the molecular sample has two surfaces. The vacuum worked as a strongly organising field, since the apolar parts of the molecules tend to orient towards the free space (cf. Figure 51, page 78) due to the hydrophobic effect. This orientation had also an impact on the other molecules and led to the formation of a monodomain also in the bulk part of the film. Such films could also be obtained at higher temperatures since the organising influence of the free space balances the tendency to form isotropic phases at higher temperatures. The simulations had to be performed in an nVT ensemble, i. e. the volume was kept constant instead of the pressure, to retain the vacuum space. Otherwise, this space would have been deleted by the barostat. An important aspect which had to be taken into consideration in the fixed box was a proper amount of molecules that would fit into the cross-sectional area of the box. Hence, the minimum surface area of the molecules was determined by dividing the surface area by the molecules that assemble on the surface. Under the assumption that the molecules in bilayered structures in the bulk would consume the same minimum surface area as in the free standing film, the basis area of the box was then adapted according to this value in a new simulation.

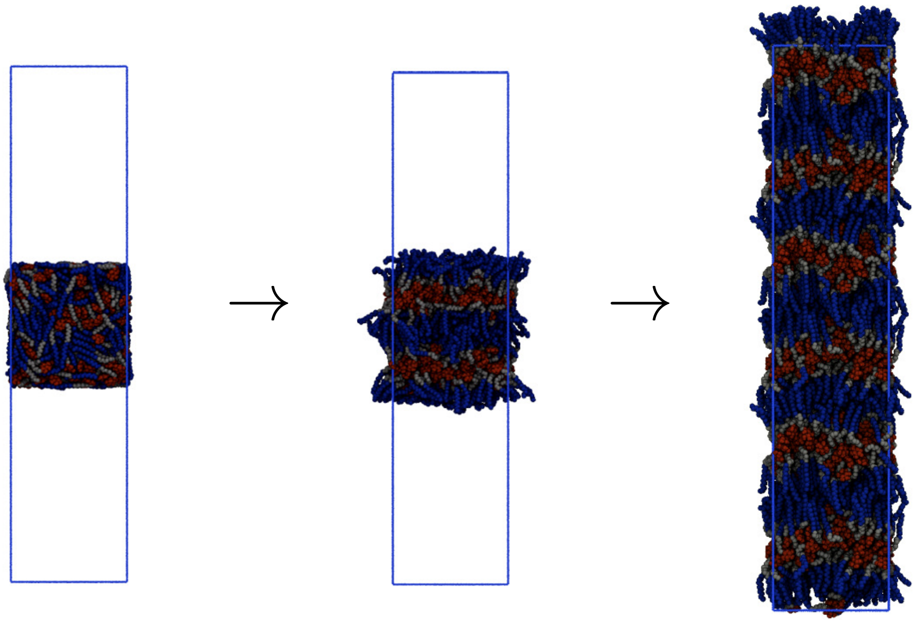


Figure 51: Creating a highly ordered and layered starting configuration by creating thin films and stacking them on top of each other.

With this approach a layered starting configuration could be created for each compound (UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$; Figure 52, page 79). Those configurations are the glass of a frozen SmA mesophase. Since there are no experimentally determined crystal structures available, this is the best starting point for a melting simulation, which can be created with simulation conditions.

These configurations were then subjected to a heating schedule. Each temperature was simulated for 200 ns. The evolution of the orientational and positional order parameters was investigated with respect to time and the diffusion coefficient was measured in the bulk as well as the lateral diffusivity.

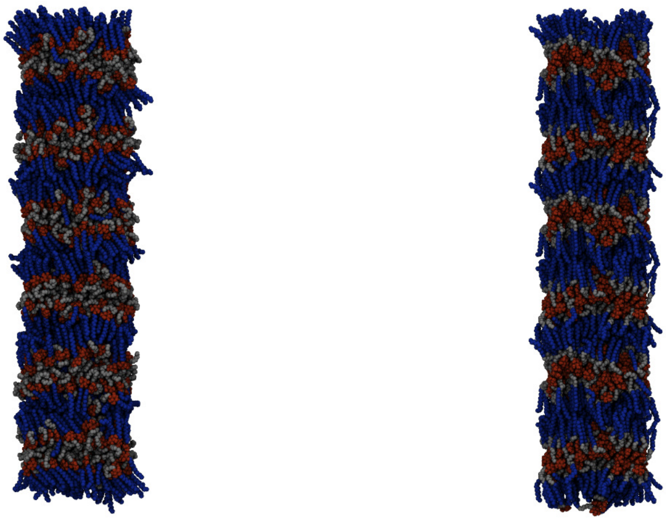


Figure 52: Starting configurations for the melting simulations. 1536 molecules of UA-$E_3I_1C_{12}$ (left) and 1536 molecules of UA-$C_{12}E_3I_1$ (right).

10.2. Simulation Setup

The simulation setup was chosen as the one described in Section 9.1 on page 65. To have a representative amount of layers a thin film system with two bilayers was stacked three times for every starting configuration, resulting in 1536 molecules per system.

The simulations were performed in an nPT ensemble. The *Parrinello-Rahman* barostat was used to keep the pressure constant anisotropically at 1 bar with an adjustment frequency of 2 ps and a compressibility of $4.5 \cdot 10^{-5}$ bar^{-1}. Since the anisotropic pressure scaling in `GROMACS` led to a strong deformation of the box in the UA-$C_{12}E_3I_1$ simulation, the pressure scaling was switched to isotropic at this temperature. This strong deformation took place when the system started to become isotropic. A heating simulation series was performed with the *Nosé-Hoover* thermostat. An adjustment frequency of every 500 fs was used. The heating schedule was applied between 390 K to 500 K with 4 K heating steps. In both simulations each temperature was simulated for 200 ns.

The bonds to hydrogens were constraint with the LINCS algorithm. In a range of 1.4 nm the neighbour list was updated every 20 calculation steps. The short-range electrostatics were cut off at 1.4 nm as well as the *van der Waals* interactions. Those were modelled with a twin range cut-off potential, while long-range electrostatics were modelled with the Particle Mesh Ewald (PME) algorithm with a fourierspacing of 0.3 nm, an order of 3, and an accuracy of 10^{-6}.

The applied integrator was a leap frog algorithm with four random *Gaussian* numbers and an integration time step of 2 fs.

10.3. Orientational Order Parameter

The orientational order parameter for the melting of UA-$E_3I_1C_{12}$ shows a transition into an isotropic liquid at 420 K. Figure 53 on page 81 shows that the transition happens over a temperature range of 10 K indicating that the simulation time of 200 ns per temperature is too short. An elongation of this time would be a reasonable improvement of the simulation conditions.

For UA-$C_{12}E_3I_1$ the transition into a totally disordered configuration happens at 488 K. Though this is not sufficient to ensure that there is a SmA to istropic liquid transition, at least the transition into the isotropic liquid can be stated. In both cases it is overestimated, for UA-$C_{12}E_3I_1$ by 62 K and for UA-$E_3I_1C_{12}$ by 67 K. This is consistent with the trend already shown by the force field description of compound **3** (cf. Section 9.5, page 75).

10.4. Positional Order Parameter

The measurement of the positional order parameter was performed by the method of PRAMPOLINI with the optimisation by PALERMO (cf. Section 6.3.2, page 46).[127, 169] Since, as mentioned in Section 8.3.2 on page 60, the positional order is more confined when considering the "upwards" and "downwards" oriented molecules separately, only the "upwards" order parameter τ_{up} is plotted in Figure 54 on page 82.

One notable finding is that the molecules in the UA-$E_3I_1C_{12}$ case, which does not show a smectic A phase in an experiment, form less confined layers than UA-$C_{12}E_3I_1$. This is indicated by a the lower order parameter of 0.7 in comparison to a starting value of 0.84 for the UA-$C_{12}E_3I_1$ case. The vanishing of the order appears in a consistent manner at the same temperatures as the

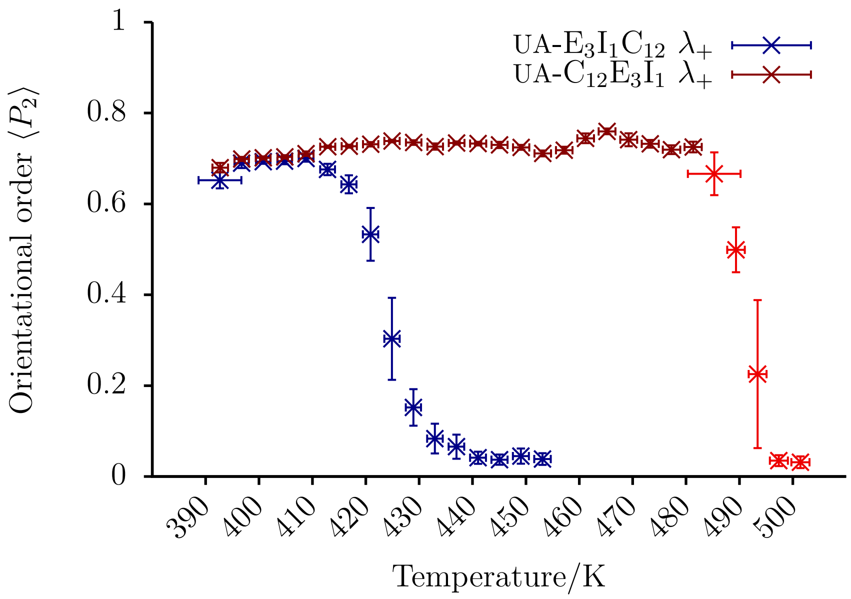


Figure 53: Evolution of the orientational order parameter, derived from the highest eigenvalue λ_+ of the orientational order tensor Q, in a heating regime of 1536 molecules of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$, respectively. The lighter color indicates the switch from anisotropic to isotropic pressure scaling.

orientational order. At about 428 K UA-$C_{12}E_3I_1$ shows the onset of a slight decrease in positional order, which could indicate the formation of a smectic phase. The appearance of a small jump at 482 K marks the transition temperature, at which the formation of smectic order, i. e. layered structures, is favoured.

10.5. Self-diffusion

The analysis of the self-diffusion should further characterise the phases which were observed. The general diffusion in the system of UA-$E_3I_1C_{12}$ faces a change at about 425 K (cf. Figure 55, page 82). This coincides with the change in orientational and positional order and confirms the melting into an isotropic liquid without the formation of a mesophase. Furthermore, the UA-$E_3I_1C_{12}$ system has a higher mobility by more than one order of magnitude, which is in agreement with the observation in the smaller and cooler system (cf. Section 9.3, page 71).

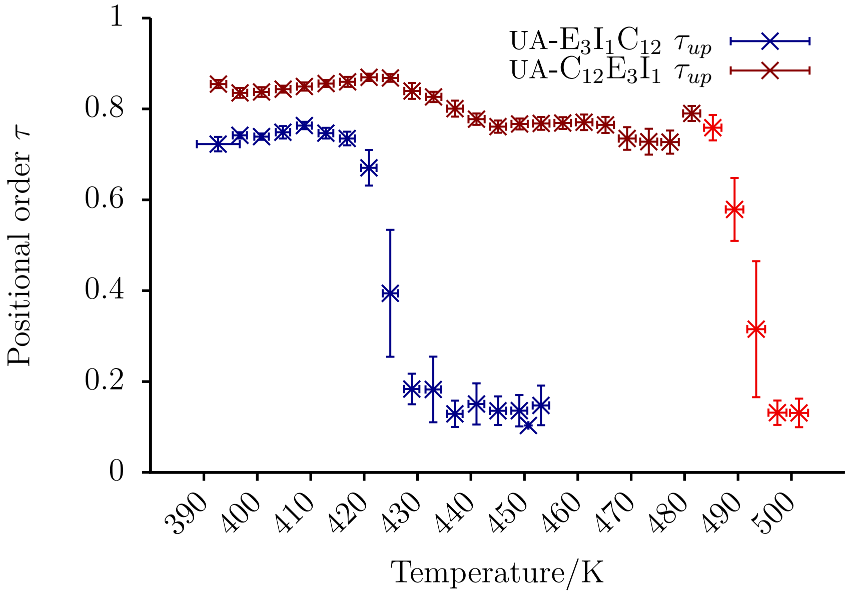


Figure 54: Evolution of the positional order parameter in a heating regime of 1536 molecules of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$, respectively. The lighter color indicates the switch from anisotropic to isotropic pressure scaling.

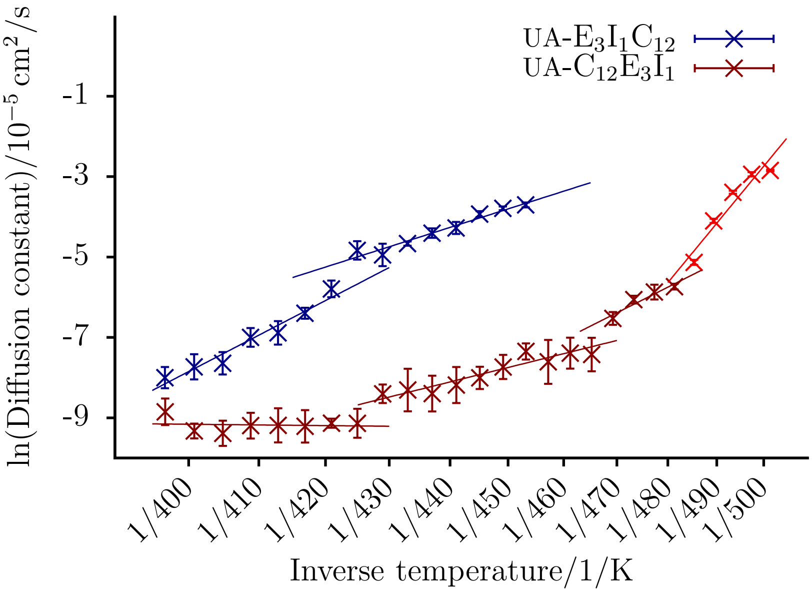


Figure 55: Evolution of the self-diffusion in a heating regime of 1536 molecules of UA-$E_3I_1C_{12}$ and UA-$C_{12}E_3I_1$, respectively. The lighter color indicates the switch from anisotropic to isotropic pressure scaling. The lines indicate linear fits of the data points.

The graph of the diffusion coefficient for the UA-$C_{12}E_3I_1$ simulation shows four different regions. Between 390 K and 428 K the diffusion remains constantly low. From 428 K to 465 K the diffusivity slightly rises continuously. Between 465 K and 483 K the diffusion constant rises more quickly, until it becomes very mobile between 483 K and 500 K. Since in the region of 428 K an onset of mobility can be detected, but both order parameters remain high, this marks the transition from the solid into the SmA phase. When comparing the transition temperature of 428 K with the value of 413 K obtained in Section 9.3 on page 71, a deviation of 15 K can be stated. This indicates a transition temperature hysteresis, i. e. a different behaviour on cooling and on heating. This is not reported for this compound but not uncommon for LC phases.[175, 176]

Though not very prominent, a split of the perpendicular and lateral diffusion components can be detected at the same temperature (cf. Figure 56, page 84). In the temperature range between 428 K and 485 K a trend can be observed that the lateral diffusion in the layer becomes slightly higher than the perpendicular diffusion. This finding supports the tendency of the system to form a smectic phase which destabilises at 485 K. In a small temperature range of 10 K between 470 K to 480 K this characteristic has its clearest expression.

10.6. Summary

It could be shown that UA-$C_{12}E_3I_1$ and UA-$E_3I_1C_{12}$ reproduce the phase behaviour of the compounds **8** and **9**. The temperature is overestimated by about 60 K, which is in good agreement with the trend observed for G_1C_{12} (cf. Section 8.4, page 64). Despite this overestimation, the phase sequence and also the phase differences can be reproduced. The higher mobility in UA-$E_3I_1C_{12}$ in comparison to UA-$C_{12}E_3I_1$ is the reason for the direct melt into a liquid.

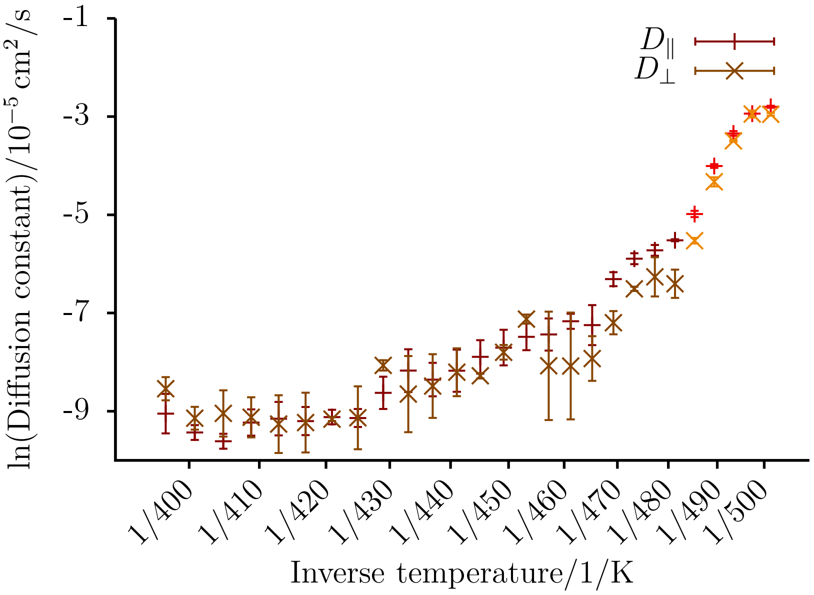


Figure 56: Separation of the diffusion into lateral $D_\parallel$ and perpendicular $D_\perp$ components in a heating regime of 1536 molecules of UA-$C_{12}E_3I_1$. The lighter color indicates the switch from anisotropic to isotropic pressure scaling.

11. Simulations of Racemic and Enantiopure Samples

Compounds **8** and **9** were synthesised and experimentally analysed as racemic mixtures.[43, 44] In order to study the differences between the enantiopure or racemic samples, respectively, two comparative simulations for each compound were started. For these simulation setups the inositol substructure of the molecules were modelled in an *all atom* fashion. The choice of an *all atom* model assures reliably that the configuration at each stereochemical center remains constant. This is a crucial aspect when simulating differences of enantiomeric mixtures and enantiopure samples. The general parameters were taken from the OPLS force field, while the local charges for these compounds were derived with a quantum mechanical population analysis. For that purpose compounds **8** and **9** were optimised by DFT using the B3LYP functional [177, 178] and the 6-31G* basis set[179–184] with the `Gaussian` 03 Suite.[172] Afterwards, a population analysis was performed with the CHelpG[152] algorithm. The enantiopure version of the simulation of compound **8** is denoted by $C_{12}E_3$AA-I_1 and the racemic setup by *rac*-$C_{12}E_3$AA-I_1. The respective calculations of compound **9** are denoted with E_3AA-I_1C_{12} and *rac*-E_3AA-I_1C_{12}, respectively. AA-I_1 stands for *myo*-inositol in an *all atom* description, E_3 for the *united atom* triethylene glycol substructure, and C_{12} for the *united atom* dodecyl moiety.

Four starting configurations with 1024 molecules each, organised in two layers were created with the *thin film* approach described in Section 10.1 on page 77. In the both cases of $C_{12}E_3$AA-I_1 and *rac*-$C_{12}E_3$AA-I_1 they could be easily obtained and the layered organisation appeared spontaneously on contact with vacuum. Furthermore, the layered structure is quite confined and the alkyl chains are oriented neatly alongside each other. In contrast to that, in the case of E_3AA-I_1C_{12} and its racemic mixture *rac*-E_3AA-I_1C_{12} the starting configurations were obtained by using only the ordered parts of the system at the surface, replicating and subsequently compressing them. Hence, in this case the configurations are less organised and the layers are not as confined as in the $C_{12}E_3$AA-I_1 cases (cf. Figure 57, page 86). This adds a higher artificial character to these configurations.

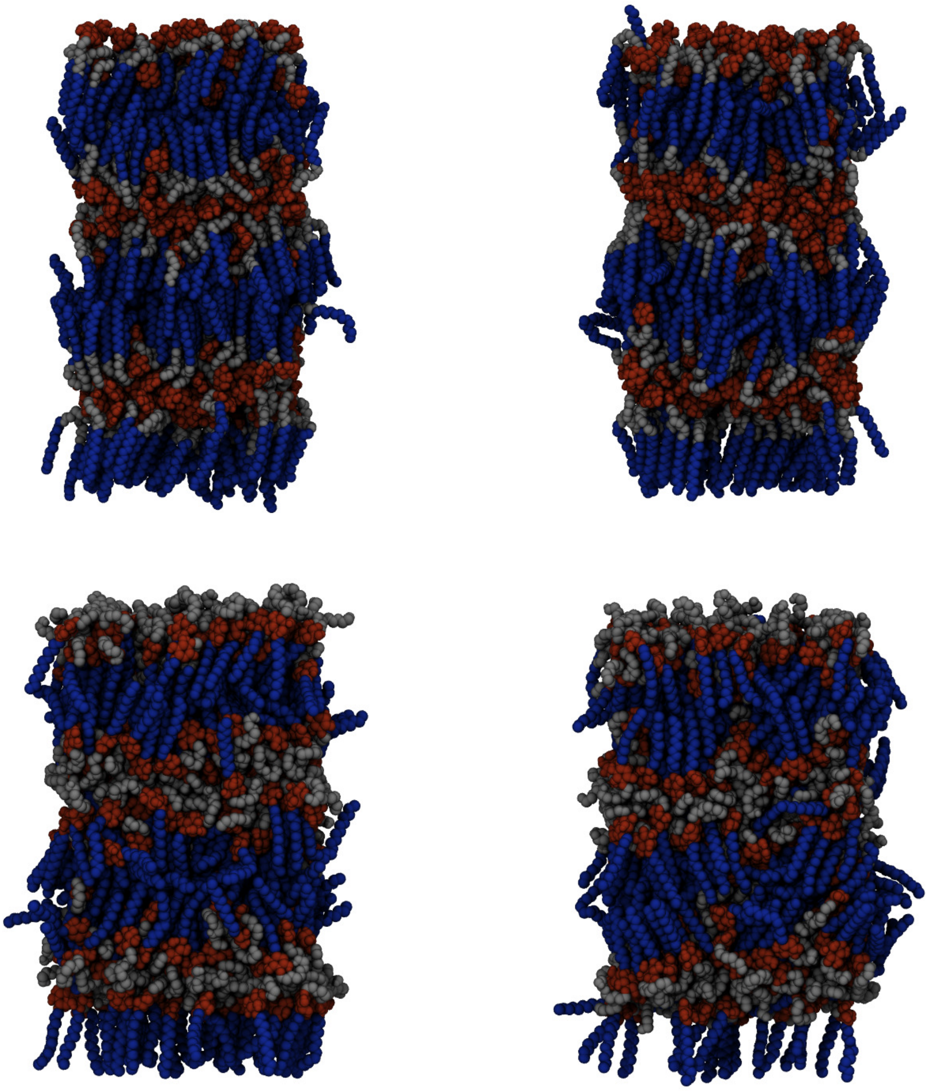


Figure 57: Four starting configurations created for equilibration simulations at different temperatures. Top: $C_{12}E_3$AA-I_1; bottom: E_3AA-I_1C_{12}; left: racemic; right: enantiopure.

All four configurations were equilibrated at different temperatures. In contrast to the previously discussed simulations, in which all the configurations were sequentially heated or sequentially cooled, here the temperature range was sampled by independent simulations.

11.1. Simulation Setup

The simulation setup was chosen as in Section 9.1 on page 65 and the configurations were equilibrated at different temperatures. To localise the transition temperature into the isotropic liquid a *binary search* approach[185] was chosen. For each configuration 400 K or 500 K, respectively, were chosen as starting temperatures. If an isotropic configuration was obtained a new simulation was started at a temperature halfway between the current temperature and the next lower one (*binary search algorithm*).

The simulations were performed in an nPT ensemble. In order to avoid a box deformation which may occur with the anisotropic pressure scaling in `GROMACS`, the isotropic *Parrinello-Rahman* barostat was used to keep the pressure constant at 1 bar. The adjustment frequency was set to 2 ps with a compressibility of $4.5 \cdot 10^{-5}\,\mathrm{bar}^{-1}$.

Each temperature equilibration was performed with the *Nosé-Hoover* thermostat and an adjustment frequency of 500 fs. Depending on the time which the system needed to equilibrate, different simulation times were chosen between 300 ns and 900 ns.

The bonds to hydrogen atoms were constraint with the LINCS algorithm. The higher amount of hydrogen atoms in the system (due to the *all atom* description of the carbon atoms in the inositol) lead to a significant slow-down in performance by factor 2.

In a range of 1.4 nm the neighbour list was updated every 20 calculation steps. The short-range electrostatics were cut off at 1.4 nm as well as the *van der Waals* interactions. Those were modelled with a twin range cut-off potential, while long-range electrostatics were modelled with the Particle Mesh Ewald (PME) algorithm with a fourierspacing of 0.3 nm, an order of 3, and an accuracy of 10^{-6}.

The applied integrator was a leap frog algorithm with four random *Gaussian* numbers and an integration time step of 2 fs.

11.2. $C_{12}E_3$AA-I_1 and *rac*-$C_{12}E_3$AA-I_1

Considering that the intermolecular interactions in racemic mixtures can be different from enantiopure substances, a racemic mixture and an enantiopure sample of $C_{12}E_3$AA-I_1 was studied by MD. The simulations should reveal the differences and the common aspects on a microscopical level. As it is known for other compounds that the enantiopurity can have an influence on the phase polymorphism or the transition temperatures, such differences can also be possible in this case.

11.2.1. Orientational Order

An analysis of the orientational order clearly reveals a transition from the orientationally ordered phase into the isotropic liquid at 426 K. This is indicated by the step in the graph of the orientational order parameter (cf. Figure 58, page 89). The temperature matches exactly the experimentally determined transition from the smectic A phase to the isotropic liquid (cf. Table 2, page 19). Furthermore, no difference in orientational order can be detected between the racemat and the enantiopure sample.

11.2.2. Positional Order

A strong decrease of order also occurs with respect to the positional order (cf. Figure 59, page 90). As shown before the order parameter is more pronounced if the "upwards" and "downwards" layers of the double layer system are analysed separately. Due to this only τ_{up} is considered. This analysis clearly shows that there is an immediate loss of the layered structure, suggesting that a direct transition from a smectic A phase into the isotropic liquid happens. This is also supported by the development of the layerspacing (cf. Figure 60, page 91). It shows a constant value in the temperature range from 380 K to 423 K. At higher temperatures the layers vanish and hence, a layerspacing can

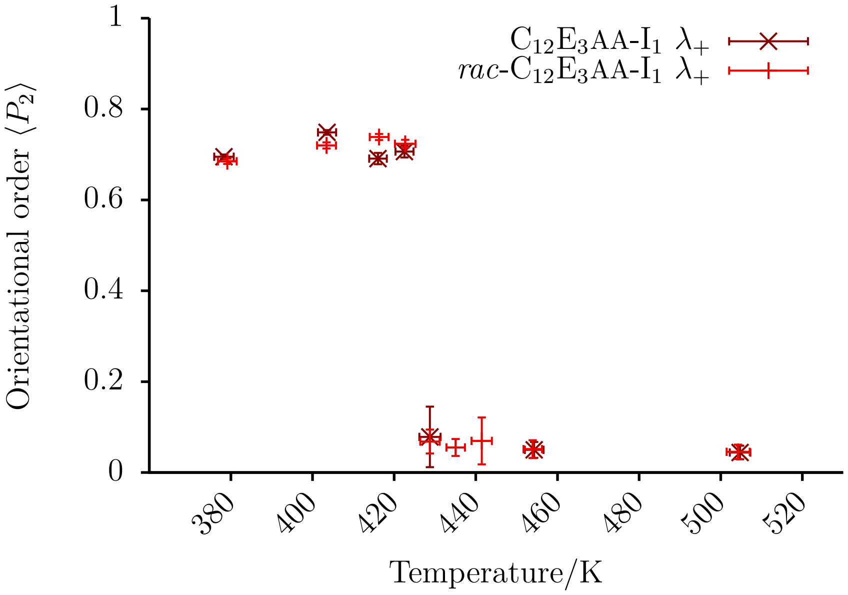


Figure 58: Development of the orientational order of 1024 molecules of enantiopure $C_{12}E_3AA$-I_1 and racemic *rac*-$C_{12}E_3AA$-I_1, respectively, in a layered structure at different temperatures. The degree of order was measured over the last 200 ns after sufficient equilibration.

not be determined, indicated by the high standard deviation. In the case of the enantiopure system the layerspacing is 5.2 nm, while in the racemic mixture this distance is 5.4 nm. This is in good agreement with the experimental findings, too. A remarkable detail is that in the enantiopure case the layers appear to be less neatly aligned. The smectic order parameter is almost 0.1 higher in the racemic mixture. An influence of the starting configuration with respect to this result is unlikely since both configurations start with the same value of the order parameter. Furthermore, the different temperatures and the duration of the simulations balances the starting conditions.

11.2.3. Hydrogen bonds

A reason for the higher degree of positional order of the racemate could be the better alignment of molecules in case that the pairing of enantiomers enable a better intermolecular interaction. An indicator for a better alignment is the amount of O—H$\cdots$O hydrogen bonds in the system. A respective analysis of the trajectory reveals that in the racemic systems 10 % more hydrogen bonds are

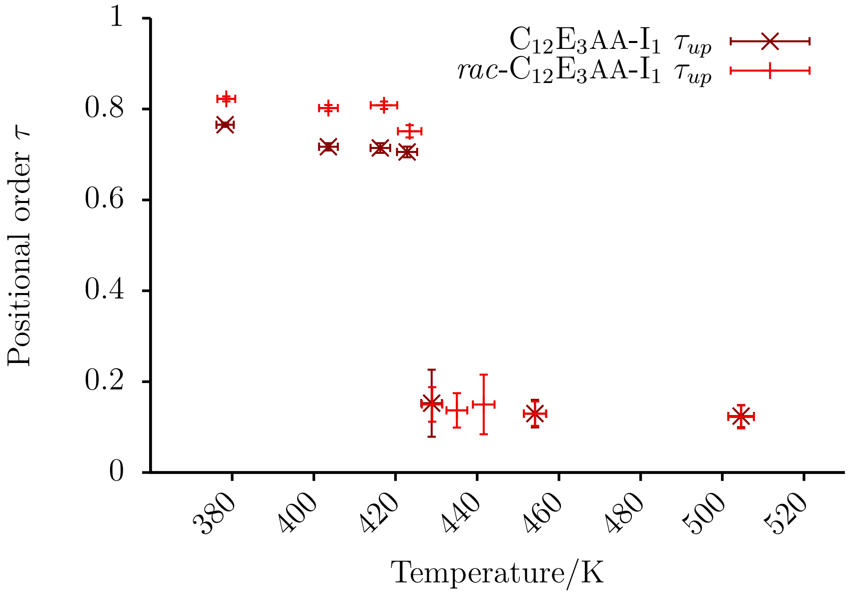


Figure 59: Development of the positional order parameter of 1024 molecules of enantiopure ($C_{12}E_3AA$-I_1) and racemic samples (*rac*-$C_{12}E_3AA$-I_1), respectively. In each case the order was measured over the last 200 ns after positional and orientational order parameter remained constant.

present than in the enantiopure systems (cf. Figure 61, page 92). In both systems the significant drop by 200 hydrogen bonds is notable.

11.2.4. Diffusion

To further characterise the phases the diffusion coefficient was calculated. A noteworthy observation is a significant increase of more than one order of magnitude at the transition into the isotropic liquid (cf. Figure 62, page 93). This is also in agreement with the aforementioned decrease of the number of hydrogen bonds. Compared with the diffusion calculated before for the UA-$C_{12}E_3I_1$ it can be stated that for the investigated temperature range it is found to be higher by more than one order of magnitude (cf. Figure 56, page 84).

Hence, the choice between *all atom* or *united atom* description of the carbohydrate has a big influence on the degree of mobility and thus on the phase formation.

In other simulations of smectic phases the lateral diffusion is a strong indicator for the layered character. A notable difference between the racemic and enan-

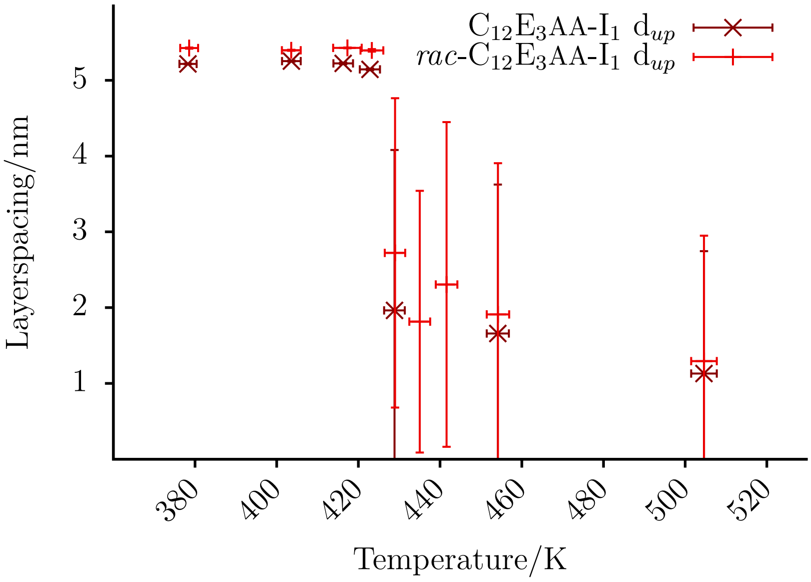


Figure 60: Temperature dependence of the layerspacing of 1024 molecules of enantiopure $C_{12}E_3AA$-I_1 or racemic *rac*-$C_{12}E_3AA$-I_1, respectively, in a layered structure. In each case the layerspacing was measured over the last 200 ns after positional and orientational order parameter remained constant. The layerspacing was measured between the "upwards" oriented layers.

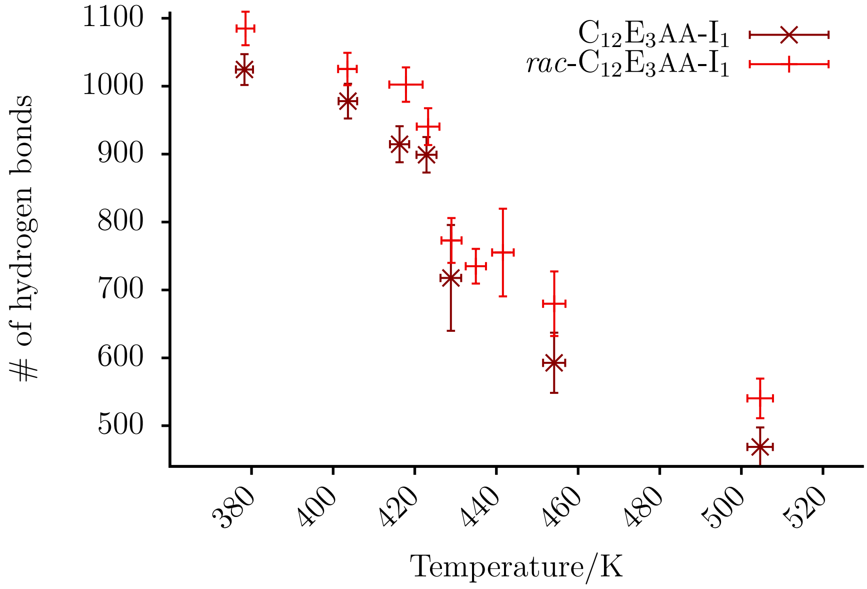


Figure 61: Number of hydrogen bonds of 1024 molecules of enantiopure $C_{12}E_3AA\text{-}I_1$ or racemic *rac*-$C_{12}E_3AA\text{-}I_1$, respectively, in a layered structure at different temperatures. The hydrogen bonds were measured over the last 200 ns after the positional and orientational order parameters remained constant.

tionpure sample UA-$C_{12}E_3I_1$ with respect to the lateral ($D_{\parallel}$) and perpendicular diffusion ($D_{\perp}$) can not be made. A reason for that can be the strong attraction between the sugar moieties. Thus, in average the molecules get stuck at their positions in the layer. A general or concerted lateral movement of the layers would be possible, this is the reason for the soapy character of smectic LCs, but in the simulation no shear stress was applied. Therefore, no eminent lateral diffusion or a distinction between lateral and perpendicular diffusion is observable (cf. Figure 63, page 94).

11.2.5. Summary

The developed combined *all atom/united atom* topology, i. e. force field description, for compound **8** describes perfectly the behaviour around the SmA to isotropic transition. This underlines the importance of fitting the atom charges to a molecule for a subsequent computational chemistry study. The experimental phase transition temperature is accurately reproduced, and the clearing process is confirmed by the positional and orientational order parameter as well as

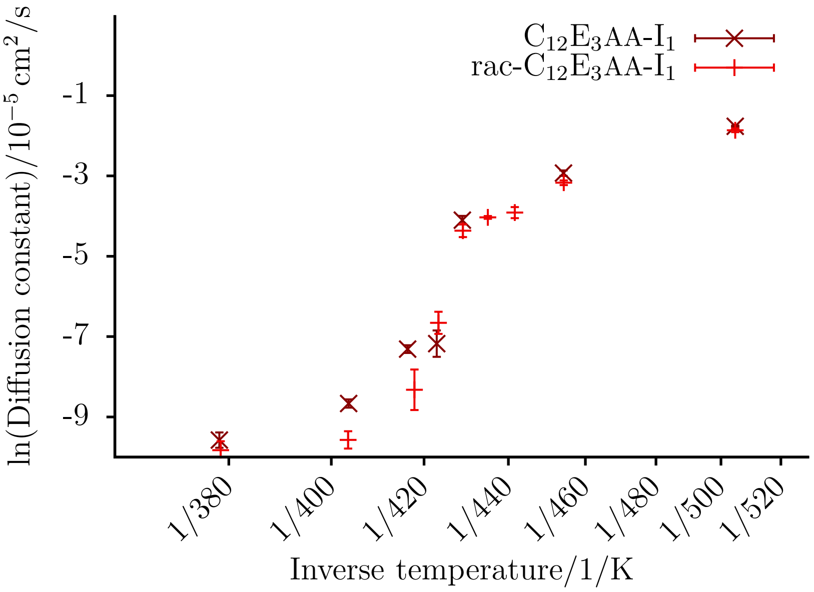


Figure 62: Development of the diffusion coefficient of 1024 molecules of enantiopure $C_{12}E_3AA\text{-}I_1$ and racemic *rac*-$C_{12}E_3AA\text{-}I_1$, respectively, in a layered structure at different temperatures. In each case the diffusion coefficient was measured over the last 200 ns after positional and orientational order parameter remained constant.

the diffusion coefficient. A difference between the racemic and enantiopure samples can be observed in the formation of hydrogen bond networks. This leads to a better organisation of the molecules in the racemic mixture.

11.3. $E_3AA\text{-}I_1C_{12}$ and *rac*-$E_3AA\text{-}I_1C_{12}$

Experimentally, compound **9** shows no LC phase. This behaviour should be modelled by a combined *all atom/united atom* description denoted by $E_3AA\text{-}I_1C_{12}$. The sugar part is represented with all its atoms and the alkyl moiety by *united atoms*. Furthermore, also in this case the influence of racemic and enantiopure samples should be investigated. The starting structure was generated, as mentioned before, with the *thin film* approach. Due to issues in obtaining a monodomain of two layers in the system, the layered structure was created artificially by stacking the ordered parts that organised at the surface of the *thin film* system. Hence, it can not be excluded that a metastable configuration was created.

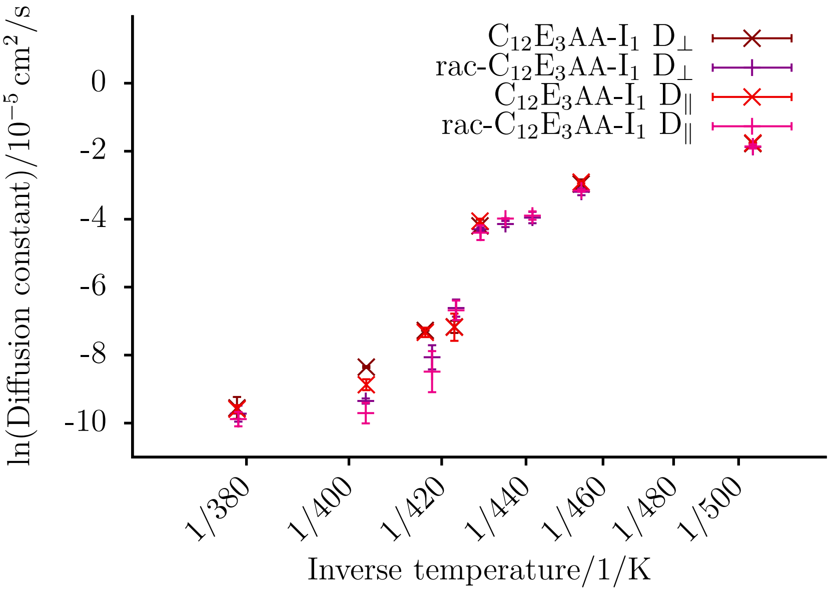


Figure 63: Development of lateral ($D_\parallel$) and perpendicular ($D_\perp$) diffusion of 1024 molecules of enantiopure $C_{12}E_3AA$-I_1 and racemic *rac*-$C_{12}E_3AA$-I_1, respectively, in a layered structure at different temperatures. The diffusion coefficient was measured over the last 200 ns after positional and orientational order parameter remained constant.

For all measurements or calculations, respectively, the last 200 ns of the simulation were taken into account. To ensure that the systems were equilibrated, especially the smectic and the orientational order parameter were taken into consideration. These observables take a lot of time to equilibrate, more than temperature, density, or the diffusion coefficient. The respective time spans for equillibration are dependent on the system and the interactions in the system. With lower temperatures systems take even longer to equilibrate. Figure 64 on page 95 shows an example of how the positional order parameter evolves in a trajectory over time. It becomes apparent that with lower temperatures the simulation takes longer for the order parameter to equilibrate. Especially the 400 K simulation shows that even after 800 ns, the system can still undergo a structural rearrangement.

Thus, it cannot be excluded that the simulations need more time to equilibrate to show reliable results. Hence, the results presented in the following are considered to be preliminary.

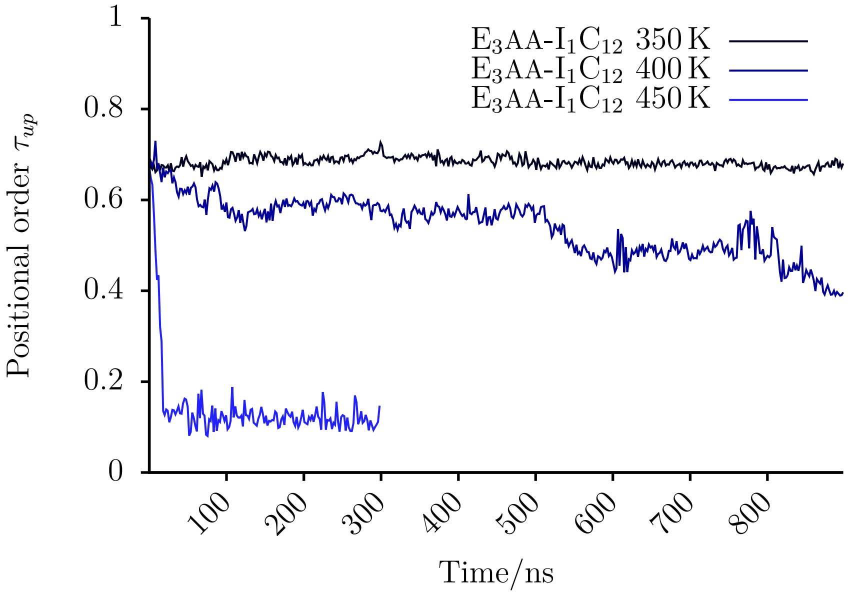


Figure 64: Evolution of the positional order parameter over time of 1024 molecules of enantiopure E_3AA-I_1C_{12} in a layered structure. Depicted are three different temperatures. There are still structural changes after an apparent equilibration of 500 ns at 400 K.

11.3.1. Orientational Order

The analyis of the orientational order parameter revealed a steady decrease of the orientational organisation from 400 K to 420 K. At temperatures above 430 K the system is clearly disordered (cf. Figure 65, page 96). Furthermore, *rac*-E_3AA-I_1C_{12} shows a higher order by 0.1. This finding could be ascribed to the different order parameters in the starting configurations and was thereby not neccessarily system inherent.

11.3.2. Positional Order

In Figure 66 on page 96 the averaged positional order is plotted for the simulated temperatures. Also here a steady decrease in order is notable. This decrease starts already at 370 K, and at 420 K the system has totally lost its layered structure.

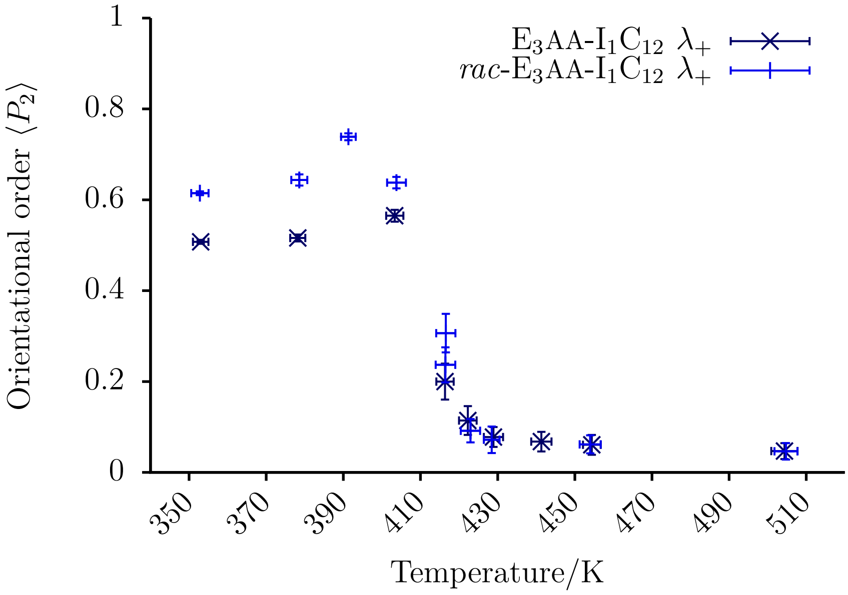


Figure 65: Development of the orientational order of 1024 molecules of enantiopure $E_3AA\text{-}I_1C_{12}$ and racemic *rac*-$E_3AA\text{-}I_1C_{12}$, respectively, in a layered structure at different temperatures. In each case the order was measured over the last 200 ns after sufficient equilibration.

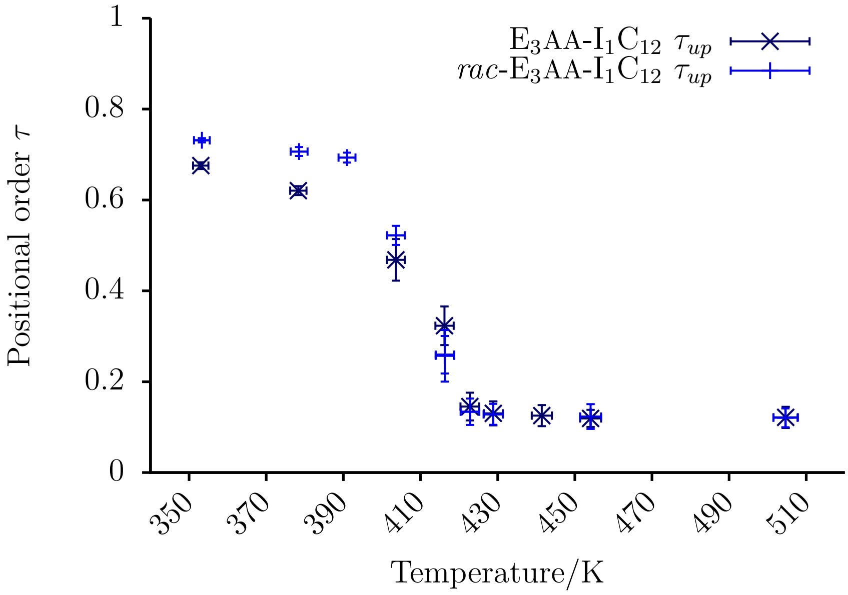


Figure 66: Development of the positional order of 1024 molecules of enantiopure $E_3AA\text{-}I_1C_{12}$ and racemic *rac*-$E_3AA\text{-}I_1C_{12}$, respectively, in a layered structure at different temperatures. In each case the order was measured over the last 200 ns after equilibration.

11.3.3. Diffusion

The evolution of the diffusion coefficient is a valuable measure to determine phase transition temperatures or detect sudden changes in mobility. Such changes can be caused by a structural alignment that hinders the displacement in some directions. The diffusion coefficient is plotted in Figure 67. The run of the curves is slightly sigmoidal and continuous.

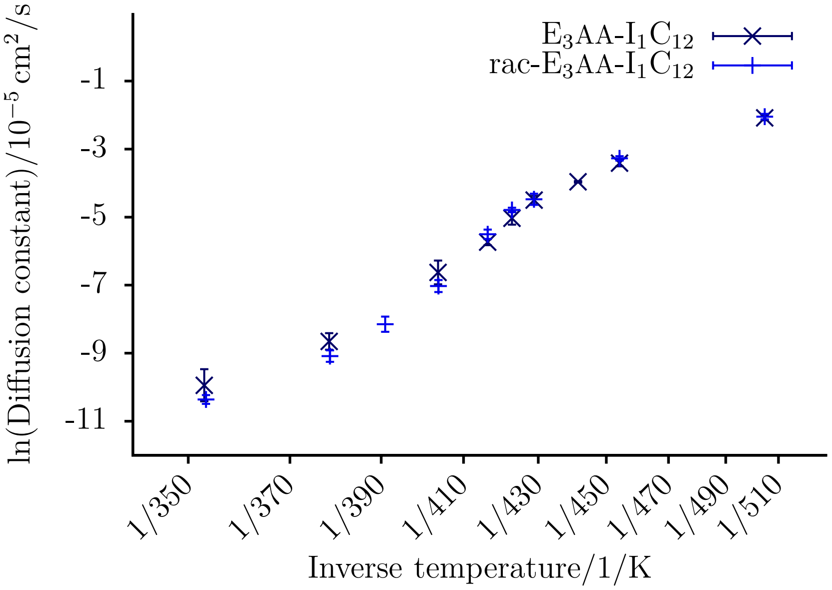


Figure 67: Development of the diffusion coefficient of 1024 molecules of enantiopure E_3AA-I_1C_{12} and racemic *rac*-E_3AA-I_1C_{12}, respectively, in a layered structure at different temperatures. In each case the diffusion coefficient was measured over the last 200 ns after equilibration.

Such a continuity can also be found with regard to the number of O—H···O hydrogen bonds as shown in Figure 68 on page 98. The combination of this mobile and structural property leads to the conclusion that the system appears to have dynamics of an isotropic liquid.

11.3.4. Summary

It becomes apparent that the simulations of E_3AA-I_1C_{12} and *rac*-E_3AA-I_1C_{12} suffer from their metastable starting configurations. Almost no differences are detectable between the enantiopure and racemic samples except for a significant

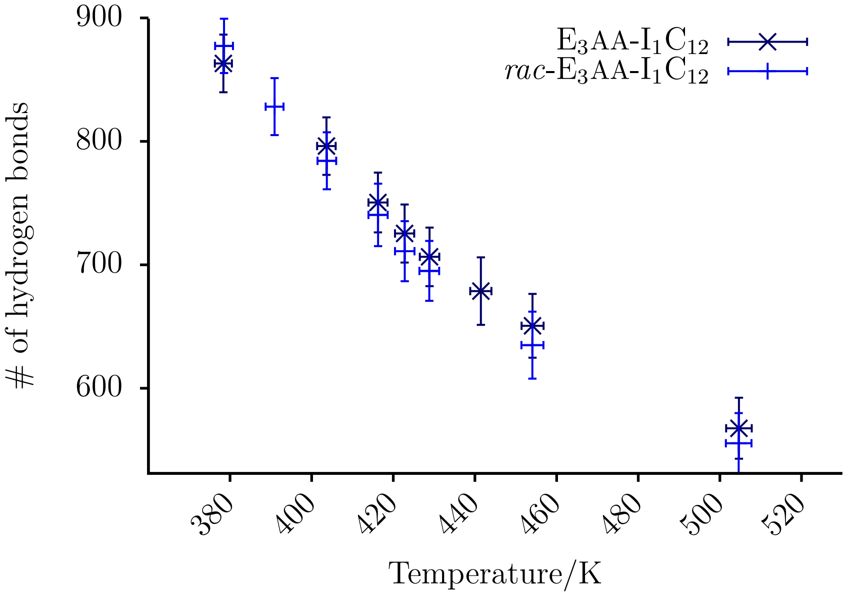


Figure 68: Development of the number of hydrogen bonds of 1024 molecules of enantiopure E_3AA-I_1C_{12} and racemic *rac*-E_3AA-I_1C_{12}, respectively, in a layered structure at different temperatures. The number of hydrogen bonds was measured over the last 200 ns after equilibration.

higher orientational order in *rac*-E_3AA-I_1C_{12}. Although the simulated time span reached almost 1 µs it can not be stated with certainty that the states were fully equilibrated. For real experiments this is a rather short time, but compared with the simulation time step of 2 fs the simulation time spans already nine orders of magnitude.

The diffusion coefficients and the development of the hydrogen bonds indicate that the system resembles a liquid. The supramolecular arrangement is mainly based on the artificial starting configuration. Sufficiently long equillibration times can not be reached with current methods, since the necessary timescale is beyond the scope of classical Molecular Dynamics.

11.4. $C_{12}E_3AA$-I_1 and E_3AA-I_1C_{12}

Comparing the simulation results of $C_{12}E_3AA$-I_1 and E_3AA-I_1C_{12} one observation is most remarkable: While the simulation of $C_{12}E_3AA$-I_1 reproduces perfectly the supramolecular structure change and the experimental transition temperature from SmA to the isotropic liquid T_{SmAIso} of compound **8**, E_3AA-I_1C_{12} does not reproduce the isotropic liquid characteristics of compound **9**.

Though the diffusion dynamics of E_3AA-I_1C_{12} resemble a liquid, the structural reorganisation, i. e. the loss of preferential orientation and positional organisation could not have been reproduced.

11.5. Summary

The influence of enantiopurity of the sample was addressed in this section. With a sequence of independent nPT simulations on starting configurations consisting of four layers the common aspects and differences were investigated. For each of the two compounds **8** and **9** two combined *all atom/united atom* OPLS force field descriptions were generated ($C_{12}E_3$AA-I_1; E_3AA-I_1C_{12}), one for the compound and one for its enantiomer. The enantiopure starting configurations were generated with 1024 molecules of one compound ($C_{12}E_3$AA-I_1 or E_3AA-I_1C_{12}), while the racemates were generated from 512 molecules of the compound and 512 molecules of its enantiomer. The *thin film* approach was used to form layered starting configurations. While in the $C_{12}E_3$AA-I_1 case the formation of two neatly aligned layers was possible, the E_3AA-I_1C_{12} configuration was built up by extracting the ordered parts of the configuration and stack them.

In case of $C_{12}E_3$AA-I_1, the model representing the liquid crystalline compound, it could be shown that the racemic mixture allows a higher organisation of the layered structure. Furthermore, the simulations concerning $C_{12}E_3$AA-I_1 reproduce perfectly the experimental findings. Due to computationally impossible equillibration times and influences of the artificial starting configuration reliable results for E_3AA-I_1C_{12} could not be obtained.

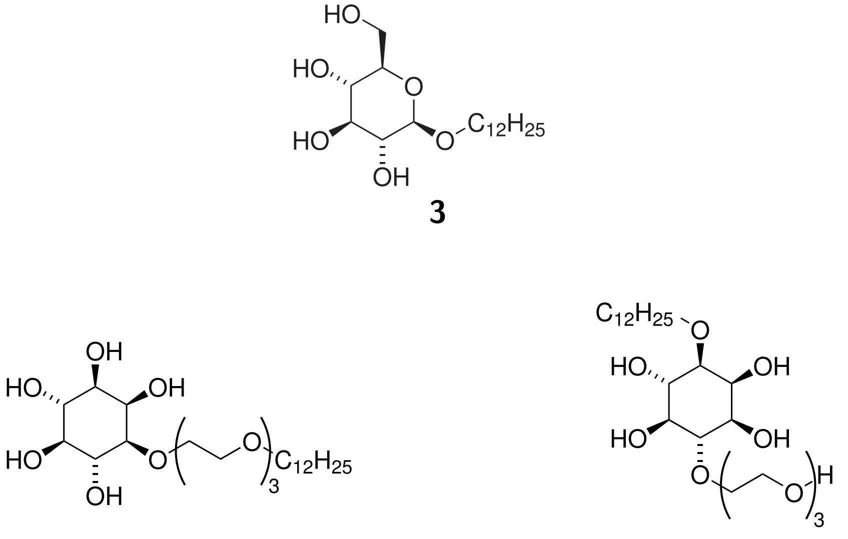
IV

Outlook

This chapter summarises the results that could be obtained in the course of this thesis. Additional methods are shown and suggestions are made for future works on the demanding topic of simulations of sugar- or inositol-based liquid crystal simulation.

12. Summary

In this thesis the influence of two $\mathrm{O{-}H}\cdots\mathrm{O}$ hydrogen bond network forming head groups on the liquid crystal phase formation was investigated. Three model compounds **3**, *rac*-**8**, and *rac*-**9**, which were investigated experimentally in former studies guided these simulations (cf. Figure 69).[42–44]



3

rac-**8**

rac-**9**

Figure 69: The three model compounds investigated in this thesis. *n*-dodecyl-β-D-glucopyranoside (**3**), 1-*O*-[2′-[2″-[2‴-(dodecyloxy)ethoxy]-ethoxy]ethyl]-*myo*-inositol (*rac*-**8**), and 1-*O*-dodecyl-4-O-[2′-[2″-[2‴-(hydroxy)ethoxy]ethoxy]ethyl]-*myo*-inositol (*rac*-**9**).

A *united atom/all atom* OPLS force field description could be derived for the alkyl-carbohydrate **3**. As stipulated in the OPLS force field the default charges were used. In an MD simulation with a negative temperature gradient (cooling) it was possible to reproduce the experimental phase sequence (cf. Table 1, page 16). The developed topology, i. e. the force field description, overestimates the transition temperatures T_{SmAIso} and T_{CrSmA} by 60 K. While the instantaneous onset of orientational and positional order can be observed at the transition from the isotropic liquid to smectic A phase (cf. Figure 70, page 103), the SmA to crystal transition is only notable in the change in mobility (cf. Figure 37, page 63).

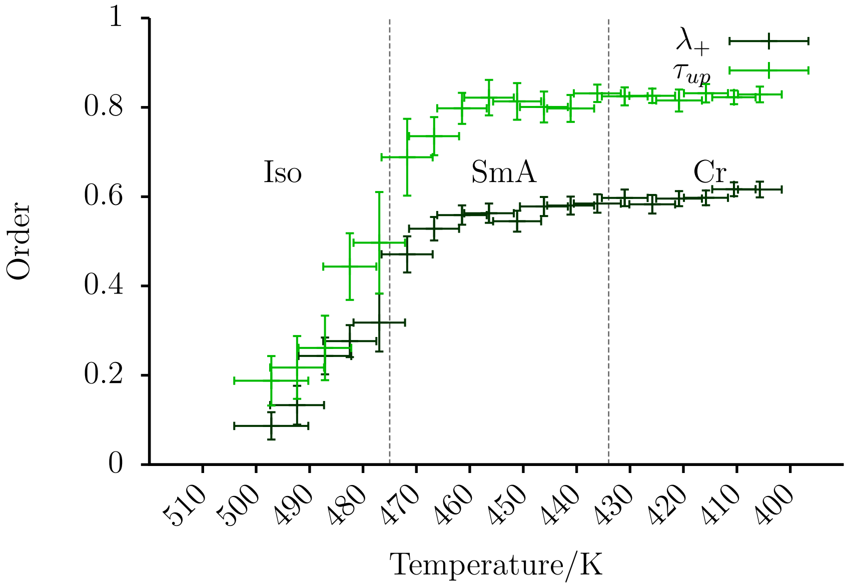


Figure 70: Evolution of the orientational (λ_+) and smectic order parameter (τ_{up}) of the force field description of *n*-dodecyl-β-D-glucopyranoside. τ_{up} denotes the order between the arbitrarily chosen "upwards" oriented molecules. The concerted onset of positional and orientational order marks the transition into the smectic A phase. The vertical dashed lines mark the transition temperatures in the simulation based on the diffusion coefficient, the positional and orientational order parameters.

The model systems of the compounds **8** and **9** were designed as comparative studies to understand the differences between these two compounds despite their similar modular setup. For these systems full *united atom* approaches based on the OPLS force field were used. Also in these cases the OPLS charges were used.

For both systems cooling as well as heating simulations were performed, which showed an overestimation of the transition temperatures. In the case of the liquid crystalline compound **8** the T_{CrSmA} is overestimated by 75 K and the T_{SmAIso} by 54 K. In the case of the non liquid crystal **9** is the overestimation found to be 70 K. While the simulation of the cooling cycle gave insights into the structural details of the molecular arrangement, it indicated that the simulation temperature of the system was too cool for the actual isotropic to smectic A transition. The heating of a mostly ordered configuration of six layers could reveal and confirm this observation. A direct transition from the solid into the isotropic liquid could be reproduced for compound **9**. Additionally, the transition from the solid to the SmA phase and from the SmA phase into the isotropic liquid could be emulated for the compound with the terminal inositol **8** (cf. Figure 71, page 105).

Both simulation experiments reveal that the standard OPLS force field description, independent of the *united atom* or *all atom* description of the carbohydrate, overestimates the transition temperatures. Approaches of other authors used force fields with case by case derived point charges.[22, 24, 186] Thus, this approach was exploited as an improvement over the aforementioned descriptions. Additionally to this change the inositol carbohydrates were described with *all atoms* to ensure the stereochemical configuration of the carbon atoms.

Combined *all atom*/*united atom* topologies of the compounds **8** and **9** were developed. The point charges were derived by quantum chemical population analyses. Layered starting configurations of 1024 molecules of these topologies were equilibrated. In the case of the thermotropic liquid crystal **8** a perfect reproduction of the experimental transition temperature from SmA to the isotropic liquid was possible. This shows the strong influence of point charges and the improvement by deriving them in a case by case manner.

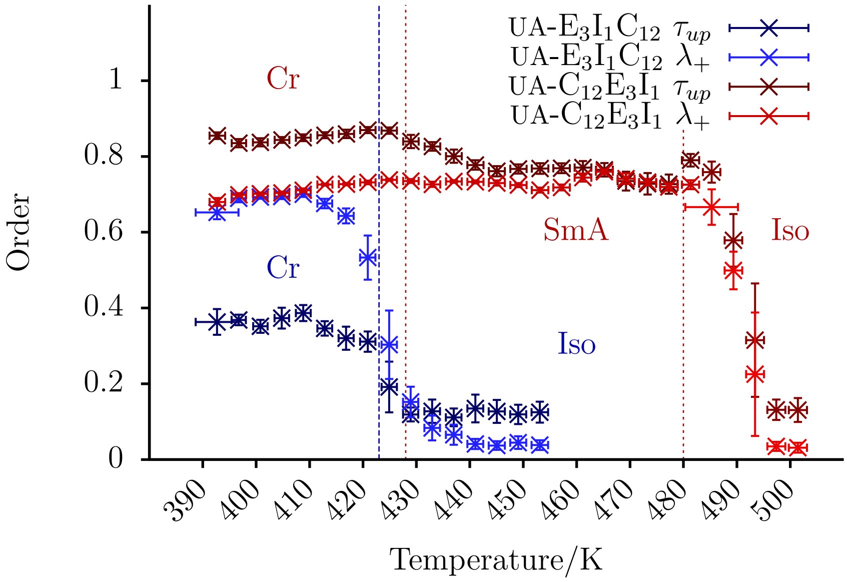


Figure 71: Evolution of the orientational and positional order parameters of the *united atom* force field description of compounds **8** (UA-$C_{12}E_3I_1$) and **9** (UA-$E_3I_1C_{12}$). τ_{up} denotes the translational order between the "upwards" oriented molecules. λ_+ is the overall orientational order parameter. The phases of the different compounds are denoted with differently colored labels. The lighter color in the curves of UA-$C_{12}E_3I_1$ indicates the switch to isotropic pressure scaling. The vertical dashed lines mark the transition temperatures obtained in the simulations based on the orientational order parameter, the positional order parameter, and the diffusion coefficient.

The simulation of the non liquid crystalline **9** did not reproduce the behaviour as perfectly as shown by the liquid crystalline compound. In the investigated temperature range the equilibration of the order observables could not be detected. Especially in the temperature range below 410 K this lead to very long simulation times of over 1 µs. A reason for this could be the artificial starting configuration, which possibly lies in a local minimum in the phasespace. Molecular Dynamics methods can not easily escape these potential wells.[187, 188] Besides, the diffusion coefficient and the hydrogen bond network indicate an isotropic behaviour and a phase transition could not be detected in this case.

Another aspect which was investigated was the influence of enantiopurity of the sample. Though no significant change in the phase sequence could be detected comparing the racemic with the enantiopure setup, a structural difference became evident. In both simulation series the racemic sample showed a higher degree of positional order, which is a sign for a better alignment of the molecules in a layered structure.

13. Future Work

The simulation of the model systems **3** and *rac*-**8** and *rac*-**9**, respectively, revealed a lot of insights into the structural features of these compounds in bulk materials. Nonetheless further approaches could be evaluated to gain an even better understanding of inositol- and carbohydrate-based LCs.

To ensure that the found topologies fully reproduce the experimental findings further data needs to be gathered. For example the layerspacing of compound **8** could be experimentally determined. Furthermore, the density of experimental samples is a valuable information to fine-tune force field parameters. Gathering more experimental data could lead to an improvement of the force field description and hence, to the *in silico* reproduction and prediction of liquid crystal phase properties.

The application of case by case derived point charges could be one approach for a remarkable improvement of the force field. Another possibility could be the generation of new standardised point charges that fit more appropriately the inositol compounds. Since the usage of quantum chemically derived atom charges showed a perfect reproduction of the experiment, this approach could be also applied in case of *n*-dodecyl-β-D-glucopyranoside.

Another obstacle in simulations of hydrogen bond network forming compounds is the neccessary but comparably long simulation time to sample the phase space appropriately. A new approach to take care of this issue could be the usage of multiscale simulations as described by MUKHERJEE and PETER.[119, 120] They introduced a multi-site coarse grained description to approximate the molecules. This accelerates the equilibration of the configurations. These configurations can subsequently be sampled again with a higher degree of accuracy.

A problem in finding appropriate force field descriptions is also the long time spans needed before the phase formation appears out of the isotropic liquid. One approach to address this issue was shown in this work by creating reasonable starting structures and melt these into the isotropic liquid. A reasonable starting structure could also be an actual, experimentally determined crystal. Temperature equilibration series could be run in parallel on those structures and could reveal the quality of the force field.

The future aim would be to derive a description of the liquid crystalline properties of new compounds out of computed model systems.

An interesting aspect, especially for cosmetics and pharmacology, is the simulation of lyotropic liquid crystals. While the focus of the work presented here was on the thermotropic behaviour, studies on the lyotropic aspects could reveal insights into new applications in medicine, pharmacy, biology, or cosmetics.

References

[1] I. A. Shanks, *Contemp. Phys.* **1982**, *23*, 65–91.

[2] B. Bahadur, *Mol. Cryst. Liq. Cryst.* **1984**, *109*, 3–93.

[3] S. E. Friberg, *J. Soc. Cosmet. Chem.* **1990**, *41*, 155–171.

[4] H. K. Bisoyi, S. Kumar, *Chem. Soc. Rev.* **2011**, *40*, 306–319.

[5] J. P. Lagerwall, G. Scalia, *Curr. Appl. Phys.* **2012**, *12*, 1387–1412.

[6] N. Garti, D. Libster, A. Aserin, *Food Funct.* **2012**, *3*, 700–713.

[7] C. L. Stevenson, D. B. Bennett, D. Lechuga-Ballesteros, *J. Pharm. Sci.* **2005**, *94*, 1861–1880.

[8] C. Guo, J. Wang, F. Cao, R. J. Lee, G. Zhai, *Drug Discovery Today* **2010**, *15*, 1032–1040.

[9] D. Chapman, *Ann. N. Y. Acad. Sci.* **1966**, *137*, 745–754.

[10] R. L. Rill, *Proc. Natl. Acad. Sci. U. S. A.* **1986**, *83*, 342–346.

[11] G. T. Stewart, *Liq. Cryst.* **2004**, *31*, 443–471.

[12] C. Tschierske, *Prog. Polym. Sci.* **1996**, *21*, Liquid Crystalline Polymers Part 1, 775–852.

[13] A. S. Sonin, *J. Struct. Chem.* **1991**, *32*, 111–129.

[14] C. M. Care, D. J. Cleaver, *Rep. Prog. Phys.* **2005**, *68*, 2665–2700.

[15] M. R. Wilson, *Chem. Soc. Rev.* **2007**, *36*, 1881–1888.

[16] R. Berardi, L. Muccioli, S. Orlandi, M. Ricci, C. Zannoni, *J. Phys.: Condens. Matter* **2008**, *20*, 463101–463116.

[17] P. A. Lebwohl, G. Lasher, *Phys. Rev. A: At. Mol. Opt. Phys.* **1972**, *6*, 426–429.

[18] J. Talbot, D. Kivelson, M. P. Allen, G. T. Evans, D. Frenkel, *J. Chem. Phys.* **1990**, *92*, 3048–3057.

[19] J. A. C. Veerman, D. Frenkel, *Phys. Rev. A: At. Mol. Opt. Phys.* **1992**, *45*, 5632–5648.

[20] J. G. Gay, B. J. Berne, *J. Chem. Phys.* **1981**, *74*, 3316–3319.

[21] L. Muccioli, C. Zannoni, *Chem. Phys. Lett.* **2006**, *423*, 1–6.

[22] R. Berardi, D. Micheletti, L. Muccioli, M. Ricci, C. Zannoni, *J. Chem. Phys.* **2004**, *121*, 9123–9130.

[23] M. R. Wilson, *J. Chem. Phys.* **1997**, *107*, 8654–8663.

[24] G. Tiberio, L. Muccioli, R. Berardi, C. Zannoni, *ChemPhysChem* **2009**, *10*, 125–136.

[25] M. J. Cook, M. R. Wilson, *Mol. Cryst. Liq. Cryst.* **2001**, *357*, 149–165.

[26] R. Memmer, *J. Chem. Phys.* **2001**, *114*, 8210–8222.

[27] C. Vega, D. Frenkel, *Mol. Phys.* **1989**, *67*, 633–650.

[28] R. Eppenga, D. Frenkel, *Mol. Phys.* **1984**, *52*, 1303–1334.

[29] D. Frenkel, R. Eppenga, *Phys. Rev. Lett.* **1982**, *49*, 1089–1092.

[30] D. Frenkel, B. M. Mulder, *Mol. Phys.* **1985/2002**, *100*, 201–217.

[31] G. Lasher, *Phys. Rev. A: At. Mol. Opt. Phys.* **1972**, *5*, 1350–1354.

[32] J. M. Briggs, T. Matsui, W. L. Jorgensen, *J. Comput. Chem.* **1990**, *11*, 958–971.

[33] C. Chiccoli, P. Pasini, F. Biscarini, C. Zannoni, *Mol. Phys.* **1988**, *65*, 1505–1524.

[34] A. Pizzirusso, R. Berardi, L. Muccioli, M. Ricci, C. Zannoni, *Chem. Sci.* **2012**, *3*, 573–579.

[35] S. J. Picken, W. H. de Jeu, *Liq. Cryst.* **2006**, *33*, 1359–1362.

[36] C. McBride, M. R. Wilson, J. A. K. Howard, *Mol. Phys.* **1998**, *93*, 955–964.

[37] T. T. Chong, R. Hashim, R. A. Bryce, *J. Chem. Phys. B* **2006**, *110*, 4978–4984.

[38] P. Konidala, L. He, B. Niemeyer, *J. Mol. Graphics Modell.* **2006**, *25*, 77–86.

[39] S. J. Marrink, X. Periole, D. P. Tieleman, A. H. de Vries, *Phys. Chem. Chem. Phys.* **2010**, *12*, 2254–2256.

[40] V. Marcon, T. Vehoff, J. Kirkpatrick, C. Jeong, D. Y. Yoon, K. Kremer, D. Andrienko, *J. Chem. Phys.* **2008**, *129*, 094505/1–094505/8.

[41] G. A. Jeffrey, L. M. Wingert, *Liq. Cryst.* **1992**, *12*, 179–202.

[42] G. Catanoiu, V. Gärtner, C. Stubenrauch, D. Blunk, *Langmuir* **2007**, *23*, 12802–12805.

[43] G. Catanoiu, D. Blunk, C. Stubenrauch, *J. Colloid Interface Sci.* **2012**, *371*, 82–88.

[44] G. Catanoiu, PhD thesis, Universität zu Köln, **2009**.

[45] C. M. Paleos, D. Tsiourvas, *Curr. Opin. Colloid Interface Sci.* **2001**, *6*, 257–267.

[46] V. Vill, R. Hashim, *Curr. Opin. Colloid Interface Sci.* **2002**, *7*, 395–409.

[47] H. A. van Doren, E. Smits, J. M. Pestman, J. B. F. N. Engberts, R. M. Kellogg, *Chem. Soc. Rev.* **2000**, *29*, 183–199.

[48] F. Reinitzer, *Monatsh. Chem.* **1888**, *9*, 421–441.

[49] F. Reinitzer, *Ann. Phys. (Berlin Ger.)* **1908**, *332*, 213–224.

[50] G. S. Ginsburg, D. Atkinson, D. M. Small, *Prog. Lipid Res.* **1984**, *23*, 135–167.

[51] O. Lehmann, *Ann. Phys. (Berlin Ger.)* **1890**, *277*, 525–537.

[52] R. A. Shanks, D. Staszczyk, *Int. J. Polym. Sci.* **2012**, *2012*, 767581–1–767581–13.

[53] G. Friedel, *Ann. Phys. (Paris Fr.)* **1922**, *18*, 273–474.

[54] L. Onsager, *Ann. N. Y. Acad. Sci.* **1949**, *51*, 627–659.

[55] M. P. Taylor, J. Herzfeld, *J. Phys.: Condens. Matter* **1993**, *5*, 2651–2678.

[56] M. Barón, *Pure Appl. Chem.* **2001**, *73*.

[57] C. Tschierske, G. Pelzl, S. Diele, *Angew. Chem.* **2004**, *116*, 6340–6368.

[58] R. S. Porter, J. F. Johnson, *J. Appl. Phys.* **1963**, *34*, 51–54.

[59] M. Kleman, J. Friedel, *Rev. Mod. Phys.* **2008**, *80*, 61–115.

[60] I. Dierking, *Textures of Liquid Crystals*, WILEY-VCH Verlag, **2003**.

[61] H. Sackmann, D. Demus, *Mol. Cryst. Liq. Cryst.* **1966**, *2*, 81–102.

[62] A. Saupe, *Mol. Cryst. Liq. Cryst.* **1969**, *7*, 59–74.

[63] I. G. Chistyakov, L. S. Schabischev, R. I. Jarenov, L. A. Gusakova, *Mol. Cryst. Liq. Cryst.* **1969**, *7*, 279–284.

[64] T. R. Taylor, J. L. Fergason, S. L. Arora, *Phys. Rev. Lett.* **1970**, *24*, 359–362.

[65] A. de Vries, *J. Chem. Phys.* **1979**, *71*, 25–31.

[66] A. D. Vries, A. Ekachai, N. Spielberg, *Mol. Cryst. Liq. Cryst.* **1979**, *49*, 143–152.

[67] H. Sackmann, D. Demus, *Z. Phys. Chem. (Leipzig)* **1968**, *238*, 215.

[68] S. Diele, P. Brand, H. Sackmann, *Mol. Cryst. Liq. Cryst.* **1972**, *16*, 105–116.

[69] E. Sackmann, S. Meiboom, L. C. Snyder, *J. Am. Chem. Soc.* **1967**, *89*, 5981–5982.

[70] S. Meiboom, M. Sammon, *Phys. Rev. Lett.* **1980**, *44*, 882–885.

[71] D. L. Johnson, J. H. Flack, P. P. Crooker, *Phys. Rev. Lett.* **1980**, *45*, 641–644.

[72] J. W. Goodby, P. S. Robinson, B.-K. Teo, P. E. Cladi, *Mol. Cryst. Liq. Cryst.* **1980**, *56*, 303–309.

[73] C. Destrade, Nguyen Huu Tinh, J. Malthête, A. Levelut, *J. Phys. (Paris)* **1983**, *44*, 597–602.

[74] A. M. Levelut, F. Hardouin, H. Gasparoux, C. Destrade, Nguyen Huu Tinh, *J. Phys. (Paris)* **1981**, *42*, 147–152.

[75] M. Cotrait, P. Marsau, M. Pesquer, V. Volpilhac, *J. Phys. (Paris)* **1982**, *43*, 355–359.

[76] Y. Bouligand, *J. Phys. (Paris)* **1980**, *41*, 1307–1315.

[77] C. Destrade, P. Foucher, H. Gasparoux, N. H. Tinh, A. M. Levelut, J. Malthete, *Mol. Cryst. Liq. Cryst.* **1984**, *106*, 121–146.

[78] C. Tschierske, *Curr. Opin. Colloid Interface Sci.* **2002**, *7*, 355–370.

[79] T. P. Silverstein, *J. Chem. Educ.* **1998**, *75*, 116–118.

[80] M. C. Holmes, *Curr. Opin. Colloid Interface Sci.* **1998**, *3*, 485–492.

[81] H. Prade, R. Miethchen, V. Vill, *J. Prakt. Chem./Chem.-Ztg.* **1995**, *337*, 427–440.

[82] J. W. Goodby, V. Görtz, S. J. Cowling, G. Mackenzie, P. Martin, D. Plusquellec, T. Benvegnu, P. Boullanger, D. Lafont, Y. Queneau, S. Chambert, J. Fitremann, *Chem. Soc. Rev.* **2007**, *36*, 1971–2032.

[83] J. W. Goodby, *Mol. Cryst. Liq. Cryst.* **1984**, *110*, 205–219.

[84] N. Noiraud, L. Maurousset, R. Lemoine, *Plant Physiol. Bioch.* **2001**, *39*, 717–728.

[85] J. Dvořáková, *Folia Microbiol. (Dordrecht Neth.)* **1998**, *43*, 323–338.

[86] G. E. Stutzmann, *Neuroscientist* **2005**, *11*, 110–115.

[87] G. R. Monteith, B. D. Roufogalis, *Cell Calcium* **1995**, *18*, 459–470.

[88] A. J. Verkhratsky, O. H. Petersen, *Cell Calcium* **1998**, *24*, 333–343.

[89] C. D'Santos, J. Clarke, N. Divecha, *Biochim. Biophys. Acta Mol. Cell Biol. Lipids* **1998**, *1436*, 201–232.

[90] A. M. Martelli, M. Y. Follo, C. Evangelisti, F. Falà, R. Fiume, A. M. Billi, L. Cocco, *J. Cell. Biochem.* **2005**, *96*, 285–292.

[91] P. A. Janmey, *Chem. Biol. (Oxford U. K.)* **1995**, *2*, 61–65.

[92] S. K. Danoff, C. A. Ross, *Prog. Neuro-Psychopharmacol. Biol. Psychiatry* **1994**, *18*, 1–16.

[93] I. Vucenik, A. M. Shamsuddin, *Nutr. Cancer* **2006**, *55*, 109 –125.

[94] G. D. Prestwich, *Chem. Biol. (Oxford U. K.)* **2004**, *11*, 619–637.

[95] W. Maier, A. Saupe, *Z. Naturforsch. A: Phys. Sci.* **1959**, *14a*, 882–889.

[96] W. Maier, A. Saupe, *Z. Naturforsch. A: Phys. Sci.* **1960**, *15a*, 287–292.

[97] A. Saupe, *Angew. Chem. Int. Ed. Engl.* **1968**, *7*, 97–112; A. Saupe, *Angew. Chem.* **1968**, *80*, 99–115.

[98] G. R. Luckhurst, C. Zannoni, *Nature* **1977**, *267*, 412–414.

[99] M. P. Allen, D. Frenkel, J. Talbot, *Comput. Phys. Rep.* **1989**, *9*, 301–353.

[100] M. P. Allen, D. Frenkel, *Phys. Rev. Lett.* **1987**, *58*, 1748–1750.

[101] D. Frenkel, B. M. Mulder, J. P. McTague, *Phys. Rev. Lett.* **1984**, *52*, 287–290.

[102] A. Stroobants, H. N. W. Lekkerkerker, D. Frenkel, *Phys. Rev. Lett.* **1986**, *57*, 1452–1455.

[103] A. Stroobants, H. N. W. Lekkerkerker, D. Frenkel, *Phys. Rev. A: At. Mol. Opt. Phys.* **1987**, *36*, 2929–2945.

[104] D. Frenkel, *Mol. Phys.* **1987**, *60*, 1–20.

[105] D. Frenkel, *Liq. Cryst.* **1989**, *5*, 929–940.

[106] B. J. Berne, P. Pechukas, *J. Chem. Phys.* **1972**, *56*, 4213–4216.

[107] D. J. Adams, G. R. Luckhurst, R. W. Phippen, *Mol. Phys.* **1987**, *61*, 1575–1580.

[108] G. R. Luckhurst, *Liq. Cryst. Today* **1993**, *3*, 3–5.

[109] G. R. Luckhurst, R. A. Stephens, R. W. Phippen, *Liq. Cryst.* **1990**, *8*, 451–464.

[110] M. A. Bates, G. R. Luckhurst, *Liq. Cryst.* **1998**, *24*, 229–241.

[111] R. Memmer, *Liq. Cryst.* **2000**, *27*, 533–546.

[112] R. Memmer, H.-G. Kuball, A. Schönhofer, *Liq. Cryst.* **1993**, *15*, 345–360.

[113] J. S. Lintuvuori, M. R. Wilson, *J. Chem. Phys.* **2008**, *128*, 044906.

[114] Z. E. Hughes, M. R. Wilson, L. M. Stimson, *Soft Matter* **2005**, *1*, 436–443.

[115] A. Dewar, P. J. Camp, *Phys. Rev. E* **2004**, *70*, 011704.

[116] S. J. Johnston, R. J. Low, M. P. Neal, *Phys. Rev. E* **2002**, *65*, 051706.

[117] M. A. Horsch, Z. Zhang, S. C. Glotzer, *Phys. Rev. Lett.* **2005**, *95*, 056105.

[118] S.-J. Marrink, A. E. Mark, *Biophys. J.* **2004**, *87*, 3894–3900.

[119] C. Peter, L. D. Site, K. Kremer, *Soft Matter* **2008**, *4*, 859–869.

[120] B. Mukherjee, L. D. Site, K. Kremer, C. Peter, *J. Phys. Chem. B* **2012**, *116*, 8474–8484.

[121] M. Cestari, S. Diez-Berart, D. A. Dunmur, A. Ferrarini, M. R. de la Fuente, D. J. B. Jackson, D. O. Lopez, G. R. Luckhurst, M. A. Perez-Jubindo, R. M. Richardson, J. Salud, B. A. Timimi, H. Zimmermann, *Phys. Rev. E* **2011**, *84*, 031704.

[122] M. Cestari, A. Bosco, A. Ferrarini, *J. Chem. Phys.* **2009**, *131*, 054104.

[123] M. Cestari, A. Ferrarini, *Soft Matter* **2009**, *5*, 3879–3887.

[124] A. Ferrarini, G. J. Moro, P. L. Nordio, G. R. Luckhurst, *Mol. Phys.* **1992**, *77*, 1–15.

[125] F. Chami, M. R. Wilson, V. S. Oganesyan, *Soft Matter* **2012**, *8*, 6823–6833.

[126] W. Ran, W. Zeng-Hui, W. Yong-Gang, W. Z. Zhi-Gang, W. X. Li, *Chin. Phys. B* **2009**, *18*, 4380–4385.

[127] L. D. Gaetani, G. Prampolini, *Soft Matter* **2009**, *5*, 3517–3526.

[128] L. D. Gaetani, G. Prampolini, A. Tani, *J. Chem. Phys.* **2008**, *128*, 194501.

[129] L. D. Gaetani, G. Prampolini, A. Tani, *J. Phys. Chem. B* **2007**, *111*, 7473–7477.

[130] I. Cacelli, L. D. Gaetani, G. Prampolini, A. lessandro Tani, *J. Phys. Chem. B* **2007**, *111*, 2130–2137.

[131] A. J. McDonald, S. Hanna, *J. Chem. Phys.* **2006**, *124*, 164906.

[132] L. D. Gaetani, G. Prampolini, A. Tani, *J. Phys. Chem. B* **2006**, *110*, 2847–2854.

[133] Y. Lansac, M. A. Glaser, N. A. Clark, *Phys. Rev. E* **2001**, *64*, 051703–051703.

[134] M. Tsige, M. P. Mahajan, C. Rosenblatt, P. L. Taylor, *Phys. Rev. E* **1999**, *60*, 638–644.

[135] S. S. Patnaik, J. A. Lupo, R. Pachter, *Comput. Theor. Polym. Sci.* **1998**, *8*, 39–47.

[136] A. Pizzirusso, M. Savini, L. Muccioli, C. Zannoni, *J. Mater. Chem.* **2011**, *21*, 125–133.

[137] F. Yan, D. J. Earl, *J. Chem. Phys.* **2012**, *136*, 124506.

[138] X.-G. Xue, L. Zhao, Z.-Y. Lu, M.-H. Li, Z.-S. Li, *Phys. Chem. Chem. Phys.* **2011**, *13*, 11951–11957.

[139] F. Yan, D. J. Earl, *Soft Matter* **2011**, *7*, 10266–10273.

[140] J. Pelaez, M. R. Wilson, *Phys. Rev. Lett.* **2006**, *97*, 267801.

[141] G. Cinacchi, *J. Phys. Chem. B* **2005**, *109*, 8125–8131.

[142] P. L. Cristinziano, F. Lelj, *J. Chem. Phys.* **2007**, *127*, 134506.

[143] M. P. Allen, D. J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, USA, **1989**.

[144] D. Frenkel, B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, Academic Press, Second Edition, **2002**.

[145] A. Leach, *Molecular Modelling: Principles and Applications (2nd Edition)*, Prentice Hall, 2nd ed., **2001**.

[146] S. J. Weiner, P. A. Kollman, D. A. Case, C. Singh, C. Ghio, G. Alagona, S. Profeta, P. Weiner, *J. Am. Chem. Soc.* **1984**, *106*, 765–784.

[147] S. J. Weiner, P. A. Kollman, D. T. Nguyen, D. A. Case, *J. Comput. Chem.* **1986**, *7*, 230–252.

[148] W. L. Jorgensen, D. S. Maxwell, J. Tirado-Rives, *J. Am. Chem. Soc.* **1996**, *118*, 11225–11236.

[149] W. L. Jorgensen, J. Tirado-Rives, *J. Am. Chem. Soc.* **1988**, *110*, 1657–1666.

[150] J. Tirado-Rives, W. L. Jorgensen, *J. Am. Chem. Soc.* **1990**, *112*, 2773–2781.

[151] G. Kaminski, W. L. Jorgensen, *J. Phys. Chem.* **1996**, *100*, 18010–18013.

[152] C. M. Breneman, K. B. Wiberg, *J. Comput. Chem.* **1990**, *11*, 361–373.

[153] B. Hess, C. Kutzner, D. van der Spoel, E. Lindahl, *J. Chem. Theory Comput.* **2008**, *4*, 435–447.

[154] N. D. Winter, G. C. Schatz, *J. Phys. Chem. B* **2010**, *114*, 5053–5060.

[155] M. Hloucha, U. K. Deiters, *Mol. Simul.* **1998**, *20*, 239–244.

[156] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, J. R. Haak, *J. Chem. Phys.* **1984**, *81*, 3684–3690.

[157] W. G. Hoover, *Phys. Rev. A: At. Mol. Opt. Phys.* **1985**, *31*, 1695–1697.

[158] M. Parrinello, A. Rahman, *J. Appl. Phys.* **1981**, *52*, 7182–7190.

[159] S. Nosé, M. L. Klein, *Mol. Phys.* **1983**, *50*, 1055–1076.

[160] A. D. Mcnaught, A. Wilkinson, *IUPAC. Compendium of Chemical Terminology, 2nd ed. (the "Gold Book").* WileyBlackwell; 2nd Revised edition edition, **1997**.

[161] H. J. C. Berendsen, D. van der Spoel, R. van Drunen, *Comput. Phys. Commun.* **1995**, *91*, 43–56.

[162] W. Humphrey, A. Dalke, K. Schulten, *J. Mol. Graphics* **1996**, *14*, 33–38.

[163] M. von Smoluchowski, *Ann. Phys. (Berlin Ger.)* **1906**, *326*, 756–780.

[164] A. Einstein, *Ann. Phys. (Berlin Ger.)* **1905**, *322*, 549–560.

[165] W. L. McMillan, *Phys. Rev. A: At. Mol. Opt. Phys.* **1973**, *8*, 328–331.

[166] W. L. McMillan, *Phys. Rev. A: At. Mol. Opt. Phys.* **1973**, *7*, 1673–1678.

[167] W. L. McMillan, *Phys. Rev. A: At. Mol. Opt. Phys.* **1972**, *6*, 936–947.

[168] W. L. McMillan, *Phys. Rev. A: At. Mol. Opt. Phys.* **1971**, *4*, 1238–1246.

[169] M. F. Palermo, A. Pizzirusso, L. Muccioli, C. Zannoni, "An Atomistic Description of the Nematic and Smectic Phases of 8CB", Private communication: O(n) algorithm to determine smectic order parameter, **2012**.

[170] E. Lindahl, B. Hess, D. van der Spoel, *J. Mol. Modeling* **2001**, *7*, 306–317.

[171] L. Martínez, R. Andrade, E. G. Birgin, J. M. Martínez, *J. Comput. Chem.* **2009**, *30*, 2157–2164.

[172] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, J. A. Montgomery, Jr., T. Vreven, K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, C. Gonzalez, J. A. Pople, Gaussian 03, Revision C.02, Gaussian, Inc., Wallingford, CT, 2004.

[173] V. I. Dimitrov, *J. Non-Cryst. Solids* **2006**, *352*, 216–231.

[174] C. A. Angell, K. L. Ngai, G. B. McKenna, P. F. McMillan, S. W. Martin, *J. Appl. Phys.* **2000**, *88*, 3113–3157.

[175] R. N. A. H. Lewis, D. Zweytick, G. Pabst, K. Lohner, R. N. McElhaney, *Biophys. J.* **2007**, *92*, 3166–3177.

[176] A. Nesrullajev, Çiğdem Yörür-Göreci, B. Bilgin-Eran, English, *Int. J. Thermophys.* **2012**, *33*, 58–68.

[177] A. D. Becke, *J. Chem. Phys.* **1993**, *98*, 5648–5652.

[178] C. Lee, W. Yang, R. G. Parr, *Phys. Rev. B: Condens. Matter Mater. Phys.* **1988**, *37*, 785–789.

[179] R. Ditchfield, W. J. Hehre, J. A. Pople, *J. Chem. Phys.* **1971**, *54*, 724–728.

[180] W. J. Hehre, R. Ditchfield, J. A. Pople, *J. Chem. Phys.* **1972**, *56*, 2257–2261.

[181] P. C. Hariharan, J. A. Pople, English, *Theor. Chem. Acc.* **1973**, *28*, 213–222.

[182] P. C. Hariharan, J. A. Pople, *Mol. Phys.* **1974**, *27*, 209–214.

[183] M. M. Francl, W. J. Pietro, W. J. Hehre, J. S. Binkley, M. S. Gordon, D. J. DeFrees, J. A. Pople, *J. Chem. Phys.* **1982**, *77*, 3654–3665.

[184] M. S. Gordon, *Chem. Phys. Lett.* **1980**, *76*, 163–168.

[185] C. H. Davis, *Am. Doc.* **1969**, *20*, 167–167.

[186] R. Berardi, G. Cainelli, P. Galletti, D. Giacomini, A. Gualandi, L. Muccioli, C. Zannoni, *J. Am. Chem. Soc.* **2005**, *127*, PMID: 16045358, 10699–10706.

[187] B. Bouvier, H. Grubmüller, *Biophys. J.* **2007**, *93*, 770–786.

[188] O. F. Lange, L. V. Schäfer, H. Grubmüller, *Journal of Computational Chemistry* **2006**, *27*, 1693–1702.

[189] H. Senderowitz, C. Parish, W. C. Still, *J. Am. Chem. Soc.* **1996**, *118*, 2078–2086.

[190] W. Damm, A. Frontera, J. Tirado-Rives, W. L. Jorgensen, *J. Comput. Chem.* **1997**, *18*, 1955–1970.

V

Appendix

This appendix contains the topologies and force field descriptions that were used in the simulations. Moreover, the C source code developed during this work to measure the positional and orientational order parameters in `GROMACS` trajectories. Several scripts are collected to easily extract observables and create graphs intantly from trajectories.

A. Topologies

A.1. UA-$C_{12}E_3I_1$

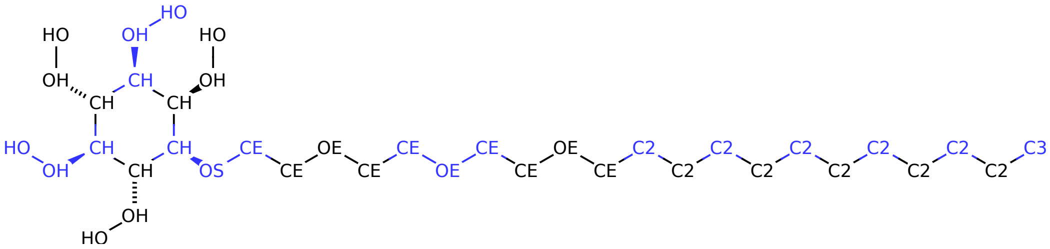


Figure 72: Assignment of atomtypes to *united atom* version of compound **8** (UA-$C_{12}E_3I_1$). Every color change between adjacent atoms indicates the beginning of a new charge group.

Table 3: Atomtypes and non-bonded parameters of UA-$C_{12}E_3I_1$.

Atomtype	Mass/u	Charge	σ/nm	ϵ/kJ mol^{-1}
C3	15.0350	0.000	0.391	0.669 888[149]
C2	14.0270	0.000	0.3905	0.494 042 4[149]
CE	14.0270	0.250	0.380	0.494 042 4[32]
OE	15.9994	−0.500	0.300	0.711 756[32]
OS	15.9994	−0.500	0.290	0.586 152[32]
CH	13.0110	0.265	0.385	0.334 944[149]
OH	15.9994	−0.683	0.3120	0.711 756[148]
HO	1.0080	0.418	0.0	0.0[148]

Table 4: Bond definitions of UA-$C_{12}E_3I_1$.

Atom_1	Atom_2	b_0/nm	k_b/kJ mol^{-1} nm^{-2}
C3	C2	0.1526	217 713.6[146]
C2	C2	0.1526	217 713.6[146]
OE	CE	0.1425	267 955.2[146]
CE	CE	0.1526	217 713.6[146]
CE	C2	0.1526	217 713.6[146]
CE	OS	0.1425	293 076.0[189]
HO	OH	0.0960	463 060.08[147]
CH	CH	0.1526	217 713.6[146]
CH	OH	0.1425	323 220.96[146]
CH	OS	0.1425	267 955.2[146]

Table 5: Angle definitions of UA-$C_{12}E_3I_1$.

Atom_1	Atom_2	Atom_3	ϕ_0/°	k_ϕ/kJ mol^{-1} rad^{-2}
C3	C2	C2	112.4	527.5368[146]
C2	C2	C2	112.4	527.5368[146]
C2	C2	CE	112.4	527.5368[146]
C2	CE	OE	109.5	669.888[146]
CE	CE	OS	109.5	669.888[146]
CE	CE	OE	109.5	669.888[146]
CE	OE	CE	111.8	837.36[146]
CH	CH	CH	111.5	527.5368[146]
CH	OH	HO	108.5	460.548[146]
CH	CH	OH	109.5	669.888[146]
CH	CH	OS	109.5	669.888[146]
CH	OS	CE	111.8	837.36[146]

Table 6: Dihedral definitions of UA-$C_{12}E_3I_1$.

Atom_1	Atom_2	Atom_3	Atom_4	$\frac{\phi_1}{\circ}$	$\frac{V_1}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_2}{\circ}$	$\frac{V_2}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_3}{\circ}$	$\frac{V_3}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_4}{\circ}$	$\frac{V_4}{\mathrm{kJ\,mol^{-1}}}$
C3	C2	C2	C2	—	—	—	—	0.0	8.3736	—	—[146]
C2	C2	C2	C2	—	—	—	—	0.0	8.3736	—	—[146]
C2	C2	C2	CE	—	—	—	—	0.0	8.3736	—	—[146]
C2	C2	CE	OS	—	—	—	—	0.0	8.3736	—	—[146]
C2	C2	CE	OE	—	—	—	—	0.0	8.3736	—	—[146]
CE	CE	OE	CE	—	—	0.0	0.418 68	0.0	6.070 86	—	—[147]
C2	CE	OE	CE	—	—	0.0	0.418 68	0.0	6.070 86	—	—[147]
OE	CE	CE	OE	—	—	0.0	2.0934	0.0	8.3736	—	—[147]
OE	CE	CE	OS	—	—	0.0	2.0934	0.0	8.3736	—	—[147]
CH	CH	OH	HO	—	—	—	—	0.0	2.0934	—	—[147]
CH	OS	CE	CE	—	—	0.0	0.418 68	0.0	3.035 43	—	—[147]
CH	OS	CE	C2	—	—	0.0	0.418 68	0.0	3.035 43	—	—[147]
CE	OS	CH	CH	—	—	0.0	0.418 68	0.0	3.035 43	—	—[147]
CH	CH	OH	HO	—	—	—	—	0.0	2.0934	—	—[147]
CH	CH	CH	OH	—	—	—	—	0.0	8.3736	—	—[147]
CH	CH	CH	CH	—	—	—	—	0.0	8.3736	—	—[147]
OH	CH	CH	OH	—	—	0.0	2.0934	0.0	2.0934	—	—[147]

(continued on next page)

Table 6: Dihedral definitions of UA-$C_{12}E_3I_1$.

Atom_1	Atom_2	Atom_3	Atom_4	$\frac{\phi_1}{\circ}$	$\frac{V_1}{\text{kJ mol}^{-1}}$	$\frac{\phi_2}{\circ}$	$\frac{V_2}{\text{kJ mol}^{-1}}$	$\frac{\phi_3}{\circ}$	$\frac{V_3}{\text{kJ mol}^{-1}}$	$\frac{\phi_4}{\circ}$	$\frac{V_4}{\text{kJ mol}^{-1}}$
OH	CH	CH	OS	—	—	0.0	2.0934	0.0	2.0934	—	—[147]
CH	CH	CH	OS	—	—	—	—	180.0	58.6152	—	—[147]
CH	CH	CH	OH	—	—	—	—	180.0	58.6152	—	—[147]

A.2. $C_{12}E_3AA$-I_1

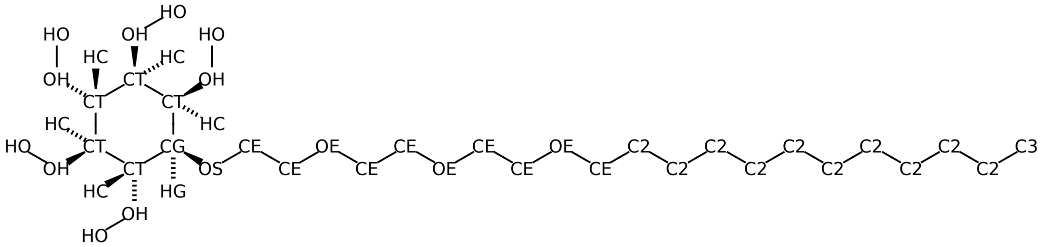


Figure 73: Assignment of atomtypes to the combined *united atom* (alkyl moiety; ethoxy moiety) and *all atom* (inositol moiety) version of compound **8** ($C_{12}E_3AA$-I_1).

Table 7: Atomtypes and non-bonded parameters of $C_{12}E_3AA$-I_1.

Atomtype	Mass/u	σ/nm	ϵ/kJ mol^{-1}
C3	15.0350	0.391	0.669 888[149]
C2	14.0270	0.3905	0.494 042 4[149]
CE	14.0270	0.380	0.494 042 4[32]
OE	15.9994	0.300	0.711 756[32]
OS	15.9994	0.290	0.586 152[32]
CT	12.0110	0.3500	0.276 328 8[148]
CG	12.0110	0.3500	0.276 328 8[148]
HC	1.0080	0.2500	0.125 604 0[148]
HG	1.0080	0.2500	0.125 604 0[148]
OH	15.9994	0.3120	0.711 756[148]
HO	1.0080	0.0	0.0[148]

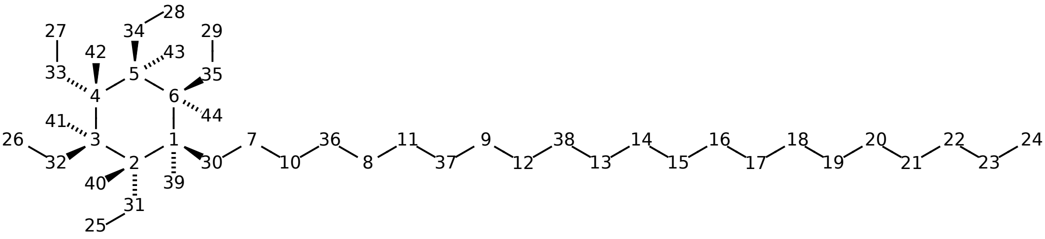


Figure 74: Assignment of atom numbers to the combined *united atom* (alkyl moiety; ethoxy moiety) and *all atom* (inositol moiety) version of compound **8** ($C_{12}E_3AA$-I_1).

Table 8: Atom point charges derived with `Gaussian` for $\mathrm{C_{12}E_3AA\text{-}I_1}$. Structure optimised with DFT//B3LYP/6-31G*. Population analysis on heavy atoms with *CHelpG* and DFT//B3LYP/6-31G*. `Gaussian` root sections: Optimisation: #opt b3lyp/6-31G*
Population: #b3lyp/6-31G* pop=CHelpG IOP(6/20=1000)

#Atom	Charge	#Atom	Charge
1	0.261 534	23	0.005 882
2	−0.036 462	24	−0.007 565
3	0.245 227	25	0.395 666
4	0.194 618	26	0.423 204
5	0.260 890	27	0.438 320
6	0.161 070	28	0.431 703
7	0.355 841	29	0.404 980
8	0.338 213	30	−0.376 384
9	0.272 014	31	−0.627 046
10	0.059 913	32	−0.628 567
11	0.203 706	33	−0.663 186
12	0.274 051	34	−0.666 457
13	0.163 558	35	−0.644 615
14	0.101 190	36	−0.471 864
15	−0.038 884	37	−0.545 065
16	−0.006 490	38	−0.493 785
17	0.016 440	39	−0.022 353
18	0.004 798	40	0.160 320
19	−0.005 728	41	−0.003 330
20	0.000 087	42	0.017 601
21	−0.004 113	43	0.007 760
22	0.010 163	44	0.033 145

Table 9: Bond definitions of $C_{12}E_3$AA-I_1.

Atom_1	Atom_2	b_0/nm	k_b/kJ mol^{-1} nm^{-2}
C3	C2	0.1526	217 713.6[146]
C2	C2	0.1526	217 713.6[146]
OE	CE	0.1425	267 955.2[146]
CE	CE	0.1526	217 713.6[146]
CE	C2	0.1526	217 713.6[146]
CE	OS	0.1425	293 076.0[189]
CG	OS	0.1425	293 076.0[189]
HO	OH	0.0960	463 060.08[147]
CT	CT	0.1529	224 412.48[148]
CT	CG	0.1529	224 412.48[148]
CT	HC	0.1090	284 702.4[148]
CG	HG	0.1090	284 702.4[148]
CT	OH	0.1410	267 955.2[147]
HO	OH	0.0960	463 060.08[147]

Table 10: Angle definitions of $C_{12}E_3$AA-I_1.

Atom_1	Atom_2	Atom_3	$\phi_0/°$	k_ϕ/kJ mol^{-1} rad^{-2}
C3	C2	C2	112.4	527.5368[146]
C2	C2	C2	112.4	527.5368[146]
C2	C2	CE	112.4	527.5368[146]
C2	CE	OE	109.5	669.888[146]
CE	CE	OS	109.5	669.888[146]
CE	CE	OE	109.5	669.888[146]
CE	OE	CE	111.8	837.36[146]
CE	OS	CG	111.8	837.36[146]
CT	CT	OH	109.5	418.680[147]
CG	CT	OH	109.5	418.680[147]
CT	CG	OS	109.5	418.680[147]
HG	CG	OS	109.5	293.076[147]
HC	CT	OH	109.5	293.076[147]
CT	OH	HO	108.5	460.548[147]
CT	CT	HC	110.7	314.01[148]
CG	CT	HC	110.7	314.01[148]
CT	CG	HG	110.7	314.01[148]
CT	CT	CT	112.7	488.599 56[148]
CT	CG	CT	112.7	488.599 56[148]
CT	CT	CG	112.7	488.599 56[148]
HC	CT	HC	107.8	276.3288[148]

Table 11: Dihedral definitions of $C_{12}E_3AA$-I_1.

Atom$_1$	Atom$_2$	Atom$_3$	Atom$_4$	$\frac{\phi_1}{°}$	$\frac{V_1}{\text{kJ mol}^{-1}}$	$\frac{\phi_2}{°}$	$\frac{V_2}{\text{kJ mol}^{-1}}$	$\frac{\phi_3}{°}$	$\frac{V_3}{\text{kJ mol}^{-1}}$	$\frac{\phi_4}{°}$	$\frac{V_4}{\text{kJ mol}^{-1}}$
C3	C2	C2	C2	—	—	—	—	0.0	8.3736	—	—[146]
C2	C2	C2	C2	—	—	—	—	0.0	8.3736	—	—[146]
C2	C2	C2	CE	—	—	—	—	0.0	8.3736	—	—[146]
C2	C2	CE	OE	—	—	—	—	0.0	8.3736	—	—[146]
CE	CE	OS	CG	—	—	0.0	0.418 68	0.0	6.070 86	—	—[147]
CE	CE	OE	CE	—	—	0.0	0.418 68	0.0	6.070 86	—	—[147]
C2	CE	OE	CE	—	—	0.0	0.418 68	0.0	6.070 86	—	—[147]
OE	CE	CE	OE	—	—	0.0	2.0934	0.0	8.3736	—	—[147]
OE	CE	CE	OS	—	—	0.0	2.0934	0.0	8.3736	—	—[147]
CE	OS	CG	HG	—	—	—	—	0.0	4.814 82	—	—[147]
CE	OS	CG	CT	—	—	180.0	0.837 36	0.0	1.603 544 4	—	—[147]
HC	CT	CG	OS	—	—	—	—	0.0	5.442 84	—	—[147]
OS	CG	CT	CT	—	—	—	—	0.0	5.442 84	—	—[147]
OS	CG	CT	OH	—	—	0.0	2.0934	0.0	0.602 899 2	—	—[147]
OH	CT	CT	OH	—	—	0.0	2.0934	0.0	0.602 899 2	—	—[147]
CT	CT	OH	HO	0.0	−1.490 500 8	0.0	−0.728 503 2	0.0	2.059 905 6	—	—[148]
CG	CT	OH	HO	0.0	−1.490 500 8	0.0	−0.728 503 2	0.0	2.059 905 6	—	—[148]

(continued on next page)

Table 11: Dihedral definitions of $C_{12}E_3AA$-I_1.

Atom_1	Atom_2	Atom_3	Atom_4	$\frac{\phi_1}{\circ}$	$\frac{V_1}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_2}{\circ}$	$\frac{V_2}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_3}{\circ}$	$\frac{V_3}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_4}{\circ}$	$\frac{V_4}{\mathrm{kJ\,mol^{-1}}}$
CG	CT	CT	OH	0.0	7.163 614 8	0.0	−2.0934	0.0	2.775 848 4	—	—[148]
CT	CG	CT	OH	0.0	7.163 614 8	0.0	−2.0934	0.0	2.775 848 4	—	—[148]
CT	CT	CT	OH	0.0	7.163 614 8	0.0	−2.0934	0.0	2.775 848 4	—	—[148]
CT	CT	CT	HC	—	—	—	—	0.0	1.532 368 8	—	—[148]
CG	CT	CT	HC	—	—	—	—	0.0	1.532 368 8	—	—[148]
CT	CG	CT	HC	—	—	—	—	0.0	1.532 368 8	—	—[148]
CT	CT	CG	HG	—	—	—	—	0.0	1.532 368 8	—	—[148]
HC	CT	CT	OH	—	—	—	—	0.0	1.959 422 4	—	—[148]
HG	CG	CT	OH	—	—	—	—	0.0	1.959 422 4	—	—[148]
HC	CT	CT	HC	—	—	—	—	0.0	1.331 402 4	—	—[148]
HC	CT	CG	HG	—	—	—	—	0.0	1.331 402 4	—	—[148]
HC	CT	OH	HO	—	—	—	—	0.0	1.884 06	—	—[148]
CT	CT	CT	CT	0.0	7.285 032	0.0	−0.657 327 6	0.0	1.168 117 2	—	—[148]
CT	CT	CG	CT	0.0	7.285 032	0.0	−0.657 327 6	0.0	1.168 117 2	—	—[148]
CT	CT	CT	CG	0.0	7.285 032	0.0	−0.657 327 6	0.0	1.168 117 2	—	—[148]

A.3. UA-$E_3I_1C_{12}$

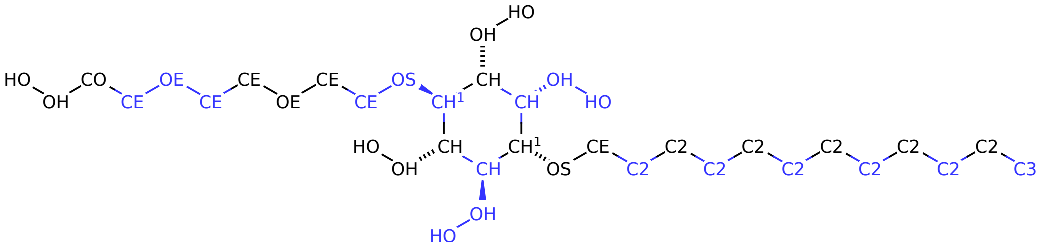


Figure 75: Assignment of atomtypes to *united atom* version of compound **9** (UA-$E_3I_1C_{12}$). The atomtypes labelled with ^{1} had a deviating charge of 0.25 instead of the default 0.265 to ensure the charge groups were neutral. Every color change between adjacent atoms indicates the beginning of a new charge group.

Table 12: Atomtypes and non-bonded parameters of UA-$E_3I_1C_{12}$.

Atomtype	Mass/u	Charge	σ/nm	ϵ/kJ mol^{-1}
C3	15.0350	0.000	0.391	0.669 888[149]
C2	14.0270	0.000	0.3905	0.494 042 4[149]
CE	14.0270	0.250	0.380	0.494 042 4[32]
OE	15.9994	−0.500	0.300	0.711 756[32]
CO	14.0270	0.265	0.3905	0.494 042 4[149]
OS	15.9994	−0.500	0.290	0.586 152[32]
CH	13.0110	0.265	0.385	0.334 944[149]
OH	15.9994	−0.683	0.3120	0.711 756[148]
HO	1.0080	0.418	0.0	0.0[148]

Table 13: Bond definitions of UA-$E_3I_1C_{12}$.

Atom_1	Atom_2	b_0/nm	k_b/kJ mol^{-1} nm^{-2}
C3	C2	0.1526	217 713.6[146]
C2	C2	0.1526	217 713.6[146]
OE	CE	0.1425	267 955.2[146]
CE	CE	0.1526	217 713.6[146]
CE	C2	0.1526	217 713.6[146]
CO	CE	0.1526	217 713.6[146]
CO	OH	0.1425	323 220.96[146]
CE	OS	0.1425	293 076.0[189]
HO	OH	0.0960	463 060.08[147]
CH	CH	0.1526	217 713.6[146]
CH	OH	0.1425	323 220.96[146]
CH	OS	0.1425	267 955.2[146]

Table 14: Angle definitions of UA-$E_3I_1C_{12}$.

Atom_1	Atom_2	Atom_3	ϕ_0/°	k_ϕ/kJ mol^{-1} rad^{-2}
C3	C2	C2	112.4	527.5368[146]
C2	C2	C2	112.4	527.5368[146]
C2	C2	CE	112.4	527.5368[146]
C2	CE	OS	109.5	669.888[146]
CE	CE	OS	109.5	669.888[146]
CE	CE	OE	109.5	669.888[146]
CO	CE	OE	109.5	669.888[146]
CE	OE	CE	111.8	837.36[146]
CE	CO	OH	109.5	669.888[146]
CO	OH	HO	108.5	460.548[146]
CH	CH	CH	111.5	527.5368[146]
CH	OH	HO	108.5	460.548[146]
CH	CH	OH	109.5	669.888[146]
CH	CH	OS	109.5	669.888[146]
CH	OS	CE	111.8	837.36[146]

Table 15: Dihedral definitions of UA-$E_3I_1C_{12}$.

Atom_1	Atom_2	Atom_3	Atom_4	$\frac{\phi_1}{\circ}$	$\frac{V_1}{\text{kJ mol}^{-1}}$	$\frac{\phi_2}{\circ}$	$\frac{V_2}{\text{kJ mol}^{-1}}$	$\frac{\phi_3}{\circ}$	$\frac{V_3}{\text{kJ mol}^{-1}}$	$\frac{\phi_4}{\circ}$	$\frac{V_4}{\text{kJ mol}^{-1}}$
C3	C2	C2	C2	—	—	—	—	0.0	8.3736	—	—[146]
C2	C2	C2	C2	—	—	—	—	0.0	8.3736	—	—[146]
C2	C2	C2	CE	—	—	—	—	0.0	8.3736	—	—[146]
C2	C2	CE	OS	—	—	—	—	0.0	8.3736	—	—[146]
CE	CE	OE	CE	—	—	0.0	0.418 68	0.0	6.070 86	—	—[147]
C0	CE	OE	CE	—	—	0.0	0.418 68	0.0	6.070 86	—	—[147]
OE	CE	CE	OE	—	—	0.0	2.0934	0.0	8.3736	—	—[147]
OE	CE	CE	OS	—	—	0.0	2.0934	0.0	8.3736	—	—[147]
OE	CE	CO	OH	—	—	0.0	2.0934	0.0	8.3736	—	—[147]
CE	CO	OH	HO	—	—	—	—	0.0	2.0934	—	—[147]
CH	CH	OH	HO	—	—	—	—	0.0	2.0934	—	—[147]
CH	OS	CE	CE	—	—	0.0	0.418 68	0.0	3.035 43	—	—[147]
CH	OS	CE	C2	—	—	0.0	0.418 68	0.0	3.035 43	—	—[147]
CE	OS	CH	CH	—	—	0.0	0.418 68	0.0	3.035 43	—	—[147]
CH	CH	OH	HO	—	—	—	—	0.0	2.0934	—	—[147]
CH	CH	CH	OH	—	—	—	—	0.0	8.3736	—	—[147]
CH	CH	CH	OS	—	—	—	—	0.0	8.3736	—	—[147]

(continued on next page)

Table 15: Dihedral definitions of UA-$E_3I_1C_{12}$.

Atom_1	Atom_2	Atom_3	Atom_4	$\frac{\phi_1}{\circ}$	$\frac{V_1}{\text{kJ mol}^{-1}}$	$\frac{\phi_2}{\circ}$	$\frac{V_2}{\text{kJ mol}^{-1}}$	$\frac{\phi_3}{\circ}$	$\frac{V_3}{\text{kJ mol}^{-1}}$	$\frac{\phi_4}{\circ}$	$\frac{V_4}{\text{kJ mol}^{-1}}$
CH	CH	CH	CH	—	—	—	—	0.0	8.3736	—	—[147]
OH	CH	CH	OH	—	—	0.0	2.0934	0.0	2.0934	—	—[147]
OH	CH	CH	OS	—	—	0.0	2.0934	0.0	2.0934	—	—[147]
CH	CH	CH	OS	—	—	—	—	180.0	58.6152	—	—[147]
CH	CH	CH	OH	—	—	—	—	180.0	58.6152	—	—[147]

A.4. E_3AA-I_1C_{12}

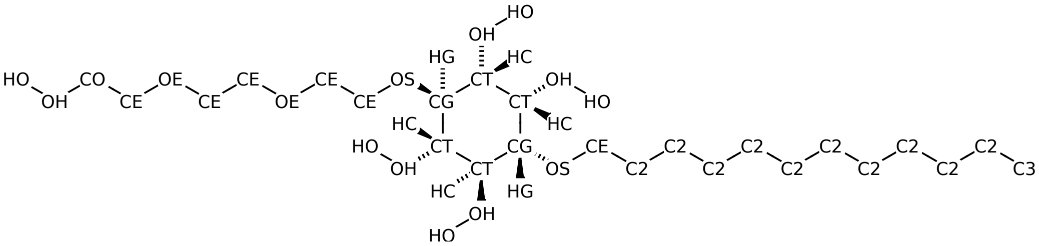


Figure 76: Assignment of atomtypes to combined *united atom* (alkyl moiety; ethoxy moiety) and *all atom* (inositol moiety) version of compound **9** (E_3AA-I_1C_{12}). Every color change between adjacent atoms indicates the beginning of a new charge group.

Table 16: Atomtypes and non-bonded parameters of E_3AA-I_1C_{12}.

Atomtype	Mass/u	σ/nm	ϵ/kJ mol^{-1}
C3	15.0350	0.391	0.669 888[149]
C2	14.0270	0.3905	0.494 042 4[149]
CE	14.0270	0.380	0.494 042 4[32]
CO	14.0270	0.3905	0.494 042 4[149]
OE	15.9994	0.300	0.711 756[32]
OS	15.9994	0.290	0.586 152[32]
CT	12.0110	0.3500	0.276 328 8[148]
CG	12.0110	0.3500	0.276 328 8[148]
HC	1.0080	0.2500	0.125 604 0[148]
HG	1.0080	0.2500	0.125 604 0[148]
OH	15.9994	0.3120	0.711 756[148]
HO	1.0080	0.0	0.0[148]

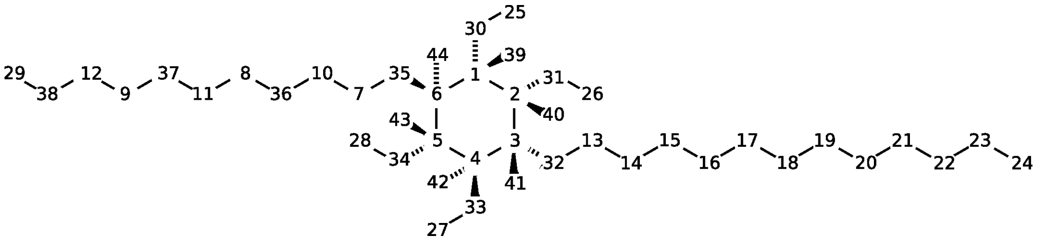


Figure 77: Assignment of atom numbers to the combined *united atom* (alkyl moiety; ethoxy moiety) and *all atom* (inositol moiety) version of compound **9** ($E_3AA\text{-}I_1C_{12}$).

Table 17: Atom point charges derived with `Gaussian` for $E_3AA\text{-}I_1C_{12}$. Structure optimised with DFT//B3LYP/6-31G*. Population analysis on heavy atoms with *CHelpG* and DFT//B3LYP/6-31G*. `Gaussian` root sections: Optimisation: #opt b3lyp/6-31G*
Population: #b3lyp/6-31G* pop=CHelpG IOP(6/20=1000)

#Atom	Charge	#Atom	Charge
1	0.261 534	23	0.005 882
2	−0.036 462	24	−0.007 565
3	0.245 227	25	0.395 666
4	0.194 618	26	0.423 204
5	0.260 890	27	0.438 320
6	0.161 070	28	0.431 703
7	0.355 841	29	0.404 980
8	0.338 213	30	−0.376 384
9	0.272 014	31	−0.627 046
10	0.059 913	32	−0.628 567
11	0.203 706	33	−0.663 186
12	0.274 051	34	−0.666 457
13	0.163 558	35	−0.644 615
14	0.101 190	36	−0.471 864
15	−0.038 884	37	−0.545 065
16	−0.006 490	38	−0.493 785
17	0.016 440	39	−0.022 353
18	0.004 798	40	0.160 320
19	−0.005 728	41	−0.003 330
20	0.000 087	42	0.017 601
21	−0.004 113	43	0.007 760
22	0.010 163	44	0.033 145

Table 18: Bond definitions of $\mathrm{E_3AA\text{-}I_1C_{12}}$.

Atom_1	Atom_2	b_0/nm	$k_b/\mathrm{kJ\,mol^{-1}\,nm^{-2}}$
C3	C2	0.1526	217 713.6[146]
C2	C2	0.1526	217 713.6[146]
OE	CE	0.1425	267 955.2[146]
CE	CE	0.1526	217 713.6[146]
CE	C2	0.1526	217 713.6[146]
CO	CE	0.1526	217 713.6[146]
CO	OH	0.1425	323 220.96[146]
CE	OS	0.1425	293 076.0[189]
CG	OS	0.1425	293 076.0[189]
CT	CT	0.1529	224 412.48[148]
CT	CG	0.1529	224 412.48[148]
CT	HC	0.1090	284 702.4[148]
CG	HG	0.1090	284 702.4[148]
CT	OH	0.1410	267 955.2[147]
HO	OH	0.0960	463 060.08[147]

Table 19: Angle definitions of E_3AA-I_1C_{12}.

Atom_1	Atom_2	Atom_3	$\phi_0/°$	$k_\phi/\text{kJ mol}^{-1}\,\text{rad}^{-2}$
C3	C2	C2	112.4	527.5368[146]
C2	C2	C2	112.4	527.5368[146]
C2	C2	CE	112.4	527.5368[146]
C2	CE	OE	109.5	669.888[146]
CE	CE	OS	109.5	669.888[146]
CE	CE	OE	109.5	669.888[146]
CO	CE	OE	109.5	669.888[146]
CE	CO	OH	109.5	669.888[146]
CO	OH	HO	108.5	460.548[146]
CE	OE	CE	111.8	837.36[146]
CE	OS	CG	111.8	837.36[146]
CT	CT	OH	109.5	418.680[147]
CG	CT	OH	109.5	418.680[147]
CT	CG	OS	109.5	418.680[147]
HG	CG	OS	109.5	293.076[147]
HC	CT	OH	109.5	293.076[147]
CT	OH	HO	108.5	460.548[147]
CT	CT	HC	110.7	314.01[148]
CG	CT	HC	110.7	314.01[148]
CT	CG	HG	110.7	314.01[148]
CT	CT	CT	112.7	488.599 56[148]
CT	CG	CT	112.7	488.599 56[148]
CT	CT	CG	112.7	488.599 56[148]
HC	CT	HC	107.8	276.3288[148]

Table 20: Dihedral definitions of E_3AA-I_1C_{12}.

Atom_1	Atom_2	Atom_3	Atom_4	$\frac{\phi_1}{°}$	$\frac{V_1}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_2}{°}$	$\frac{V_2}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_3}{°}$	$\frac{V_3}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_4}{°}$	$\frac{V_4}{\mathrm{kJ\,mol^{-1}}}$
C3	C2	C2	C2	—	—	—	—	0.0	8.3736	—	—[146]
C2	C2	C2	C2	—	—	—	—	0.0	8.3736	—	—[146]
C2	C2	C2	CE	—	—	—	—	0.0	8.3736	—	—[146]
C2	C2	CE	OS	—	—	—	—	0.0	8.3736	—	—[146]
C2	CE	OS	CG	—	—	0.0	0.418 68	0.0	6.070 86	—	—[147]
CE	CE	OS	CG	—	—	0.0	0.418 68	0.0	6.070 86	—	—[147]
CE	CE	OE	CE	—	—	0.0	0.418 68	0.0	6.070 86	—	—[147]
CO	CE	OE	CE	—	—	0.0	0.418 68	0.0	6.070 86	—	—[147]
OE	CE	CE	OE	—	—	0.0	2.0934	0.0	8.3736	—	—[147]
OE	CE	CE	OS	—	—	0.0	2.0934	0.0	8.3736	—	—[147]
OE	CE	CO	OH	—	—	0.0	2.0934	0.0	8.3736	—	—[147]
CE	CO	OH	HO	—	—	—	—	0.0	2.0934	—	—[147]
CE	OS	CG	HG	—	—	—	—	0.0	4.814 82	—	—[147]
CE	OS	CG	CT	—	—	180.0	0.837 36	0.0	1.603 544 4	—	—[147]
HC	CT	CG	OS	—	—	—	—	0.0	5.442 84	—	—[147]
OS	CG	CT	CT	—	—	—	—	0.0	5.442 84	—	—[147]
OS	CG	CT	OH	—	—	0.0	2.0934	0.0	0.602 899 2	—	—[147]

(continued on next page)

Table 20: Dihedral definitions of $\mathrm{E_3AA\text{-}I_1C_{12}}$.

Atom$_1$	Atom$_2$	Atom$_3$	Atom$_4$	$\frac{\phi_1}{\circ}$	$\frac{V_1}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_2}{\circ}$	$\frac{V_2}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_3}{\circ}$	$\frac{V_3}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_4}{\circ}$	$\frac{V_4}{\mathrm{kJ\,mol^{-1}}}$
OH	CT	CT	OH	—	—	0.0	2.0934	0.0	0.602 899 2	—	—[147]
CT	CT	OH	HO	0.0	−1.490 500 8	0.0	−0.728 503 2	0.0	2.059 905 6	—	—[148]
CG	CT	OH	HO	0.0	−1.490 500 8	0.0	−0.728 503 2	0.0	2.059 905 6	—	—[148]
CG	CT	CT	OH	0.0	7.163 614 8	0.0	−2.0934	0.0	2.775 848 4	—	—[148]
CT	CG	CT	OH	0.0	7.163 614 8	0.0	−2.0934	0.0	2.775 848 4	—	—[148]
CT	CT	CT	OH	0.0	7.163 614 8	0.0	−2.0934	0.0	2.775 848 4	—	—[148]
CT	CT	CT	HC	—	—	—	—	0.0	1.532 368 8	—	—[148]
CG	CT	CT	HC	—	—	—	—	0.0	1.532 368 8	—	—[148]
CT	CG	CT	HC	—	—	—	—	0.0	1.532 368 8	—	—[148]
CT	CT	CG	HG	—	—	—	—	0.0	1.532 368 8	—	—[148]
HC	CT	CT	OH	—	—	—	—	0.0	1.959 422 4	—	—[148]
HG	CG	CT	OH	—	—	—	—	0.0	1.959 422 4	—	—[148]
HC	CT	CT	HC	—	—	—	—	0.0	1.331 402 4	—	—[148]
HC	CT	CG	HG	—	—	—	—	0.0	1.331 402 4	—	—[148]
HC	CT	OH	HO	—	—	—	—	0.0	1.884 06	—	—[148]
CT	CT	CT	CT	0.0	7.285 032	0.0	−0.657 327 6	0.0	1.168 117 2	—	—[148]
CT	CT	CG	CT	0.0	7.285 032	0.0	−0.657 327 6	0.0	1.168 117 2	—	—[148]

(continued on next page)

Table 20: Dihedral definitions of $\mathrm{E_3AA\text{-}I_1C_{12}}$.

Atom$_1$	Atom$_2$	Atom$_3$	Atom$_4$	$\frac{\phi_1}{\circ}$	$\frac{V_1}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_2}{\circ}$	$\frac{V_2}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_3}{\circ}$	$\frac{V_3}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_4}{\circ}$	$\frac{V_4}{\mathrm{kJ\,mol^{-1}}}$
CT	CT	CT	CG	0.0	7.285 032	0.0	−0.657 327 6	0.0	1.168 117 2	—	—[148]
CG	CT	CT	CG	0.0	7.285 032	0.0	−0.657 327 6	0.0	1.168 117 2	—	—[148]

A.5. G_1C_{12}

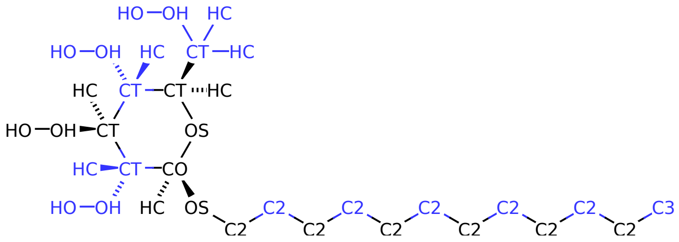


Figure 78: Assignment of atomtypes to combined *united atom* (alkyl moiety) and *all atom* (sugar moiety) version of compound **3** (G_1C_{12}). Every color change between adjacent atoms indicates the beginning of a new charge group.

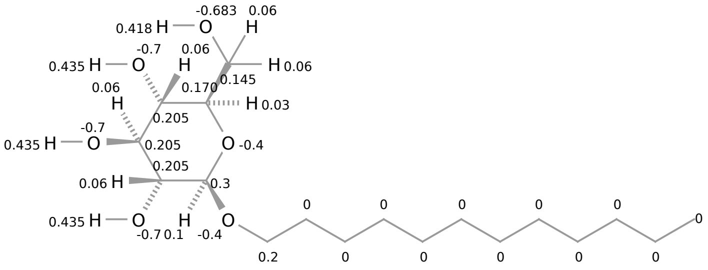


Figure 79: Assignment of charges to combined *united atom* (alkyl moiety) and *all atom* (sugar moiety) version of compound **3** (G_1C_{12}).

Table 21: Atomtypes and non-bonded parameters of G_1C_{12}.

Atomtype	Mass/u	Charge	σ/nm	ϵ/kJ mol^{-1}
C3	15.0350	0.000	0.391	0.669 888[149]
C2	14.0270	0.000	0.3905	0.494 042 4[149]
HC	1.0080	0.060	0.2500	0.125 604 0[190]
CT	12.0110	0.205	0.3500	0.276 328 8[190]
CO	12.0110	0.205	0.3500	0.276 328 8[190]
OH	15.9994	−0.683	0.3120	0.711 756[190]
HO	1.0080	0.418	0.0	0.0[190]
OS	15.9994	−0.450	0.290	0.586 152[190]
OD	15.9994	−0.700	0.307	0.711 756[190]

Table 22: Bond definitions of G_1C_{12}.

Atom_1	Atom_2	b_0/nm	k_b/kJ mol^{-1} nm^{-2}
C3	C2	0.1526	217 713.6[146]
C2	C2	0.1526	217 713.6[146]
CT	CT	0.1529	224 412.48[190]
CT	CO	0.1529	224 412.48[190]
CT	HC	0.1090	284 702.4[190]
CO	HC	0.1090	284 702.4[190]
CT	OS	0.1410	267 955.2[190]
CT	OH	0.1410	267 955.2[190]
CO	OS	0.1380	267 955.2[190]
C2	OS	0.1380	267 955.2[190]
OH	HO	0.0945	463 060.08[190]

Table 23: Angle definitions of G_1C_{12}.

Atom_1	Atom_2	Atom_3	$\phi_0/°$	$k_\phi/\text{kJ mol}^{-1}\text{ rad}^{-2}$
C3	C2	C2	112.4	527.5368[146]
C2	C2	C2	112.4	527.5368[146]
CT	CT	CT	112.7	488.599 56[190]
CT	CT	CO	112.7	488.599 56[190]
CT	CT	HC	110.7	314.010 00[190]
CT	CO	HC	110.7	314.010 00[190]
CO	CT	HC	110.7	314.010 00[190]
CT	CT	OH	109.5	418.680 00[190]
CO	CT	OH	109.5	418.680 00[190]
CT	CT	OS	109.5	418.680 00[190]
CT	CO	OS	109.5	418.680 00[190]
CT	CO	OH	109.5	418.680 00[190]
CT	OS	CO	109.5	502.416 00[190]
CT	OS	CT	109.5	502.416 00[190]
CT	OH	HO	108.5	460.548 00[190]
CO	OH	HO	108.5	460.548 00[190]
HC	CT	HC	107.8	276.328 80[190]
HC	CT	OS	109.5	293.076 00[190]
HC	CT	OH	109.5	293.076 00[190]
HC	CO	OS	109.5	293.076 00[190]
HC	CO	OH	109.5	293.076 00[190]
OS	CO	OS	111.55	775.395 36[190]
OS	CO	OH	111.55	775.395 36[190]
CO	OS	C2	109.5	502.416 00[190]
C2	C2	OS	109.5	418.680 00[190]

Table 24: Dihedral definitions of $\mathrm{G_1C_{12}}$.

Atom_1	Atom_2	Atom_3	Atom_4	$\frac{\phi_1}{\circ}$	$\frac{V_1}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_2}{\circ}$	$\frac{V_2}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_3}{\circ}$	$\frac{V_3}{\mathrm{kJ\,mol^{-1}}}$	$\frac{\phi_4}{\circ}$	$\frac{V_4}{\mathrm{kJ\,mol^{-1}}}$
C3	C2	C2	C2	—	—	—	—	0.0	8.3736	—	— [146]
C2	C2	C2	C2	—	—	—	—	0.0	8.3736	—	— [146]
HC	CT	CT	HC	—	—	—	—	0.0	1.331 402 4	—	— [190]
HC	CO	CT	HC	—	—	—	—	0.0	1.331 402 4	—	— [190]
CT	CT	CT	HC	—	—	—	—	0.0	1.532 368 8	—	— [190]
CO	CT	CT	HC	—	—	—	—	0.0	1.532 368 8	—	— [190]
CT	CT	CO	HC	—	—	—	—	0.0	1.532 368 8	—	— [190]
CT	CT	CT	CT	0.0	7.285 032	0.0	−0.657 327 6	0.0	1.168 117 2	—	— [190]
CT	CT	CT	CO	0.0	7.285 032	0.0	−0.657 327 6	0.0	1.168 117 2	—	— [190]
HC	CT	CT	OH	—	—	—	—	0.0	1.959 422 4	—	— [190]
HC	CO	CT	OH	—	—	—	—	0.0	1.959 422 4	—	— [190]
HC	CT	CT	OS	—	—	—	—	0.0	1.959 422 4	—	— [190]
HC	CT	CO	OS	—	—	—	—	0.0	1.959 422 4	—	— [190]
HC	CT	OH	HO	—	—	—	—	0.0	1.884 06	—	— [190]
HC	CT	OS	CO	—	—	—	—	0.0	3.181 968	—	— [190]
HC	CT	OS	CT	—	—	—	—	0.0	3.181 968	—	— [190]
CO	OS	CT	CT	0.0	2.721 42	0.0	−1.0467	0.0	2.805 156	—	— [190]

(continued on next page)

Table 24: Dihedral definitions of G_1C_{12}.

Atom_1	Atom_2	Atom_3	Atom_4	$\frac{\phi_1}{^\circ}$	$\frac{V_1}{\text{kJ mol}^{-1}}$	$\frac{\phi_2}{^\circ}$	$\frac{V_2}{\text{kJ mol}^{-1}}$	$\frac{\phi_3}{^\circ}$	$\frac{V_3}{\text{kJ mol}^{-1}}$	$\frac{\phi_4}{^\circ}$	$\frac{V_4}{\text{kJ mol}^{-1}}$
CT	OS	CO	CT	0.0	2.721 42	0.0	−1.0467	0.0	2.805 156	—	— [190]
CT	OS	CO	OS	0.0	−1.570 05	0.0	−5.685 674 4	0.0	0.016 747 2	—	— [190]
CO	CT	CT	OH	0.0	−5.593 564 8	—	—	—	—	—	— [190]
CT	CT	CT	OH	0.0	−5.593 564 8	—	—	—	—	—	— [190]
CT	CT	CT	OS	0.0	−5.593 564 8	—	—	—	—	—	— [190]
CT	CT	CO	OS	0.0	−5.593 564 8	—	—	—	—	—	— [190]
CO	CT	OH	HO	0.0	11.195 503 2	0.0	−12.070 544 4	0.0	4.295 656 8	—	— [190]
CT	CT	OH	HO	0.0	11.195 503 2	0.0	−12.070 544 4	0.0	4.295 656 8	—	— [190]
OH	CT	CT	OS	0.0	18.082 789 2	—	—	—	—	—	— [190]
OH	CT	CO	OS	0.0	18.082 789 2	—	—	—	—	—	— [190]
OH	CT	CT	OH	0.0	37.957 528 8	—	—	—	—	—	— [190]
CO	OS	C2	C2	0.0	2.721 42	0.0	−1.0467	0.0	2.805 156	—	— [190]
C2	OS	CO	CT	0.0	2.721 42	0.0	−1.0467	0.0	2.805 156	—	— [190]
C2	OS	CO	OS	0.0	−1.570 05	0.0	−5.685 674 4	0.0	0.016 747 2	—	— [190]
C2	C2	C2	OS	0.0	−5.593 564 8	—	—	—	—	—	— [190]
HC	CO	OS	C2	—	—	—	—	0.0	3.181 968	—	— [190]

B. Analysis Tool

B.1. `g_order_tensor.c`

```
/*
 *
 *                 This source code is part of
 *
 *                  G   R   O   M   A   C   S
 *
 *           GROningen MAchine for Chemical Simulations
 *
 * Written by David van der Spoel, Erik Lindahl, Berk Hess, and others.
 * Copyright (c) 1991-2000, University of Groningen, The Netherlands.
 * Copyright (c) 2001-2009, The GROMACS development team,
 * check out http://www.gromacs.org for more information.

 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License
 * as published by the Free Software Foundation; either version 2
 * of the License, or (at your option) any later version.
 *
 * If you want to redistribute modifications, please consider that
 * scientific software is very special. Version control is crucial -
 * bugs must be traceable. We will be happy to consider code for
 * inclusion in the official distribution, but derived work must not
 * be called official GROMACS. Details are found in the README & COPYING
 * files - if they are missing, get the official version at www.gromacs.org.
 *
 * To help us fund GROMACS development, we humbly ask that you cite
 * the papers on the package - you can find them in the top README file.
 *
 * For more info, check our website at http://www.gromacs.org
 */

/*! \brief
 * The main function.
 *
 * In Gromacs, most analysis programs are implemented such that the \p main
 * function is only a wrapper for a \p gmx_something function that does all
 * the work, and that convention is also followed here. */
int
main(int argc, char *argv[])
{
    gmx_order_tensor(argc, argv);
    return 0;
}
```

B.2. `gmx_order_tensor.c`

```
/*
 *
 *                 This source code is part of
 *
 *                  G   R   O   M   A   C   S
 *
 *           GROningen MAchine for Chemical Simulations
 *
 * Written by David van der Spoel, Erik Lindahl, Berk Hess, and others.
 * Copyright (c) 1991-2000, University of Groningen, The Netherlands.
 * Copyright (c) 2001-2009, The GROMACS development team,
 * check out http://www.gromacs.org for more information.

 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License
 * as published by the Free Software Foundation; either version 2
 * of the License, or (at your option) any later version.
 *
 * If you want to redistribute modifications, please consider that
 * scientific software is very special. Version control is crucial -
 * bugs must be traceable. We will be happy to consider code for
 * inclusion in the official distribution, but derived work must not
 * be called official GROMACS. Details are found in the README & COPYING
 * files - if they are missing, get the official version at www.gromacs.org.
 *
 * To help us fund GROMACS development, we humbly ask that you cite
 * the papers on the package - you can find them in the top README file.
 *
 * For more info, check our website at http://www.gromacs.org
 */

/*! \brief
 * The main function.
 *
 * In Gromacs, most analysis programs are implemented such that the \p main
 * function is only a wrapper for a \p gmx_something function that does all
 * the work, and that convention is also followed here. */
int
main(int argc, char *argv[])
{
    gmx_order_tensor(argc, argv);
    return 0;
}


/*
 *
 *                This source code is part of
 *
 *                 G   R   O   M   A   C   S
 *
 *          GROningen MAchine for Chemical Simulations
 *
 * Written by David van der Spoel, Erik Lindahl, Berk Hess, and others.
 * Copyright (c) 1991-2000, University of Groningen, The Netherlands.
 * Copyright (c) 2001-2009, The GROMACS development team,
 * check out http://www.gromacs.org for more information.
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU General Public License
 * as published by the Free Software Foundation; either version 2
 * of the License, or (at your option) any later version.
 *
 * If you want to redistribute modifications, please consider that
 * scientific software is very special. Version control is crucial -
 * bugs must be traceable. We will be happy to consider code for
 * inclusion in the official distribution, but derived work must not
 * be called official GROMACS. Details are found in the README & COPYING
 * files - if they are missing, get the official version at www.gromacs.org.
 *
 * To help us fund GROMACS development, we humbly ask that you cite
 * the papers on the package - you can find them in the top README file.
 *
 * For more info, check our website at http://www.gromacs.org
 */
#include <copyrite.h>
#include <filenm.h>
#include <macros.h>
#include <pbc.h>
#include <smalloc.h>
#include <statutil.h>
#include <vec.h>
#include <xvgr.h>
#include <princ.h>
#include <centerofmass.h>
#include "eigensolver.h"
#include "selelem.h"
#include "confio.h"
```

```
#include <trajana.h>

/*! \brief
 * Calculating the degree of order for a group of
 * molecules by diagonalizing the second order tensor
 * of the molecules. For this the axes of inertia are
 * determined per molecule in group. Every molecular
 * main axis of inertia is taken to build the second
 * order tensor
 * Q_{\alpha\beta}=
 * \frac{1}{2N}\sum_{i=0}^N \left \langle
 * 3\cdot l_{i\alpha}\cdot l_{i\beta} -
 * \delta_{\alpha\beta}\right \rangle
 * where Q is the second order tensor and l is the main
 * orientation of a molecule determind by its moment
 * of inertia.
 */
typedef struct
{
  gmx_bool        b_total;
  FILE            *fp_order;
  real            *order_per_frame;

  FILE            *fp_lambda0_order;
  real            *lambda0_order_per_frame;

  FILE            *fp_p2_order;
  real            *P2_order_per_frame;

  FILE            *fp_p4_order;
  real            *p4_order_per_frame;
  rvec            **axis_of_inertia_per_frame;
  rvec            **cogs_per_frame;
  rvec            **orientation_per_frame;
  int             frames_analyzed;
  int             *no_of_axes;
  real            *volume_per_frame;
  matrix          *box_per_frame;

  FILE            *fp_gL_pair_distribution;
  real            *g0_pair_distribution;
  real            *g1_pair_distribution;
  real            *g2_pair_distribution;
  int             samples;
  real            max_radius;

  FILE            *fp_average_rotation;
  real            *average_rotation1;
  real            *average_rotation2;

  real            *min_box_length;

  FILE            *fp_smectic_palermo;
  real            **smectic_palermo_cosine;
  real            **smectic_palermo_sine;
  real            **smectic_palermo_cosine_up;
  real            **smectic_palermo_sine_up;
  real            **smectic_palermo_cosine_down;
  real            **smectic_palermo_sine_down;

  FILE            *fp_smectic;
  int             sm_bins;

  int             number_of_replicas;
  rvec*           replica;
  int             repl_per_dim;

  unsigned int    *sm_hist;
  unsigned int    *sm_hist_x;
  unsigned int    *sm_hist_y;
  unsigned int    *sm_up_hist;
  unsigned int    *sm_down_hist;
  unsigned int     *sm_up_down_hist;
  unsigned int     *radial_distribution_hist;

  FILE             *fp_p2_histogram;
  int              *p2_histogram;
  real             t_p2_hist;
  int              p2_histogram_number_bins;

  int              *number_of_downs;
  int              *number_of_ups;
  int              *number_of_up_downs;

  FILE             *fp_cos_histogram;
  int              *cos_histogram;
  real             t_cos_hist;
  int              cos_histogram_number_bins;

  int            orientation_1;
  int            orientation_2;

  real             start_time;
  real             time_step;
  gmx_bool         start_time_captured;
  gmx_bool         time_step_captured;
  real             end_time;
  output_env_t     *oenv;

  const char       *fn_p2_order_pdb;
  real             t_p2_order_pdb;
  const char       *fn_up_down_pdb;
  real             t_up_down_pdb;

  int              no_of_entities;
  rvec             versor;
} t_analysisdata;

short int inint_(real rval)
{
  if(rval < 0.0)
    return (rval - 0.5);
  else
    return (rval + 0.5);
}


static void
sdebug (char* text) {
  fprintf(stderr, "%s", text);
}

static void
fdebug (real value) {
  fprintf(stderr, "%f", value);
}

static void
idebug (int value) {
  fprintf(stderr, "%d", value);
}

/*! \brief
 * Function to calculate the Kronecker delta
 * for two integers
 *                   / 1      i=j
 * Kronecker_delta( i, j) =
 *                              \ 0      i!=j
 */
  static int
kronecker_delta(int i, int j)
{
  if ( i == j)
  {
    return 1;
  }
  else
  {
    return 0;
  }
}

static real
min_value(const real* values, const int dim) {
  real min=GMX_REAL_MAX;
  int i;
  for (i=0; i<dim; ++i)
    min = values[i] < min ? values[i] : min;
  return min;
}

static real
max_value(const real* values, const int dim) {
  real max=-GMX_REAL_MAX;
  int i;
  for (i=0; i<dim; ++i) {
    max = values[i] > max ? values[i] : max;
  }
  return max;
}

static real
max_value_and_index(const real* values, const int dim, int *index) {
  real max=-GMX_REAL_MAX;
  *index = 0;
  int i;
  for (i=0; i<dim; ++i) {
    if ( values[i] > max ) {
      max = values[i];
      *index = i;
    }
  }
  return max;
}

static int
i_max_value(const int* values, const int dim) {
  int max=-INT_MAX;
  int i;
  for (i=0; i<dim; ++i)
    max = values[i] > max ? values[i] : max;
  return max;
}

static unsigned int
ui_max_value(const unsigned int* values, const int dim) {
  int max=-INT_MAX;
  int i;
  for (i=0; i<dim; ++i)
    max = values[i] > max ? values[i] : max;
  return max;
}


/*! \brief
 * function to create an second rank order tensor
 * for every molecule axis in axes_of_inertia
 * Returns 0 if successful else -1.
 */
  static int
create_order_tensor(int no_of_axes, int first_axis, int start_frame, int no_of_frames,
    int groups, rvec **axes_of_inertia, matrix order_tensor)
{
  if ( 0 != no_of_axes )
  {
    int row = 0;
    int column = 0;
    int axis = first_axis;
    int frame = 0;

    for ( row = 0; row < DIM; ++row)
    {
      for ( column = 0; column < DIM; ++column)
      {
        order_tensor[row][column] = 0;
        for ( axis = first_axis; axis < first_axis + no_of_axes; ++axis )
        {
          for ( frame = start_frame; frame < start_frame + no_of_frames * groups;
              frame += groups)
          {
            order_tensor[row][column] += 3 * axes_of_inertia[frame][axis][row] *
              axes_of_inertia[frame][axis][column] -
              kronecker_delta(row, column);
          }
        }
        order_tensor[row][column] /= no_of_frames * 2 * no_of_axes;
      }
    }
  }
  else
  {
    return -1;
  }

  return 0;
};

  static int
get_order_from_tensor(matrix order_tensor, real* order, rvec* director, int type)
{

  // with type the eigenvalue that should be taken
  // as order parameter from tensor is specified
  // if type == 0 take the biggest eigenvalue
  // if type == 1 take the second biggest eigenvalue * -2
  real *eigenvalues, *eigenvectors;
  snew(eigenvalues, DIM);
  snew(eigenvectors, DIM*DIM);

  matrix copy;
  int i, j;
  for ( i = 0 ; i < DIM ; ++i )
    for ( j = 0; j < DIM ; ++j )
      copy[i][j]=order_tensor[i][j];

  eigensolver((real*)copy, DIM, 0, DIM-1, eigenvalues, eigenvectors);

  int pos = XX;
  if ( type == 0 ) {
    if ( eigenvalues[XX] > eigenvalues[YY] ) {
      if ( eigenvalues[XX] > eigenvalues[ZZ] ) {
        pos = XX;
      } else {
        pos = ZZ;
      }
    } else {
      if ( eigenvalues[YY] > eigenvalues[ZZ] ) {
        pos = YY;
      } else {
        pos = ZZ;
      }
    }
  } else if ( type == 1 ) {
    if ( eigenvalues[XX] > eigenvalues[YY] ) {
      if ( eigenvalues[XX] < eigenvalues[ZZ] ) {
        pos = XX;
      } else if ( eigenvalues[ZZ] > eigenvalues[YY] ) {
        pos = ZZ;
        }
      } else if ( eigenvalues[XX] < eigenvalues[ZZ] ) {
        if ( eigenvalues[YY] < eigenvalues[ZZ] ) {
          pos = YY;
        } else {
          pos = ZZ;
        }
      }
    } else if ( type == 2 ) {
      if ( eigenvalues[XX] > eigenvalues[YY] ) {
        if ( eigenvalues[YY] > eigenvalues[ZZ] ) {
          pos = ZZ;
        } else {
          pos = YY;
        }
      } {
        if ( eigenvalues[XX] < eigenvalues[ZZ] ) {
          pos = XX;
        } else {
          pos = ZZ;
        }
      }
    }

    (*director)[XX] = eigenvectors[DIM*pos+XX];
    (*director)[YY] = eigenvectors[DIM*pos+YY];
    (*director)[ZZ] = eigenvectors[DIM*pos+ZZ];
    if ( type == 0 ) {
      *order = eigenvalues[pos];
    } else if ( type == 1 ) {
      *order = -2 * eigenvalues[pos];
    }

    sfree(eigenvectors);
    sfree(eigenvalues);

    return 0;
}

/*! \brief
 * Function to print the content of a matrix
 * to standard error
 */
  static int
print_matrix(matrix mat)
{
  fprintf(stderr, "\n");
  fprintf(stderr, "%15.10f  %15.10f %15.10f\n", mat[XX][XX], mat[XX][YY], mat[XX][ZZ]);
  fprintf(stderr, "%15.10f  %15.10f %15.10f\n", mat[YY][XX], mat[YY][YY], mat[YY][ZZ]);
  fprintf(stderr, "%15.10f  %15.10f %15.10f\n", mat[ZZ][XX], mat[ZZ][YY], mat[ZZ][ZZ]);
  return 0;
}

/*! \brief
 * Function to print the content of a vector
 * to standard error
 */
  static int
print_vector(rvec vec, int dim)
{
  fprintf(stderr, "\n");
  int i;
  for ( i = 0 ; i < dim ; ++i )
  {
    fprintf(stderr, "%15.10f\n", vec[i]);
  }
  return 0;
}

  char *
replace_str ( const char *string, const char *substr, const char *replacement )
{
  char *tok = NULL;
  char *newstr = NULL;
  char *oldstr = NULL;
  char *head = NULL;

  /* if either substr or replacement is NULL, duplicate string a let caller handle it */
  if ( substr == NULL || replacement == NULL ) return strdup (string);
  newstr = strdup (string);
  head = newstr;
  while ( (tok = strstr ( head, substr ))) {
    oldstr = newstr;
    newstr = malloc ( strlen ( oldstr ) - strlen ( substr ) + strlen ( replacement ) + 1 );
    /*failed to alloc mem, free old string and return NULL */
    if ( newstr == NULL ) {
      free (oldstr);
      return NULL;
    }
    memcpy( newstr, oldstr, tok - oldstr );
    memcpy( newstr + (tok - oldstr), replacement, strlen( replacement ) );
    memcpy( newstr + (tok - oldstr) + strlen( replacement ), tok + strlen( substr ),
        strlen( oldstr ) - strlen( substr ) - ( tok - oldstr ) );
    memset( newstr + strlen( oldstr ) - strlen( substr ) + strlen( replacement ) , 0, 1 );
    /* move back head right after the last replacement */
    head = newstr + (tok - oldstr) + strlen( replacement );
    free (oldstr);
  }
  return newstr;
}

static void
resize_arrays (t_analysisdata *d, int no_of_selections) {
  int nr = no_of_selections;
  srenew(d->axis_of_inertia_per_frame, nr * d->frames_analyzed);
  srenew(d->cogs_per_frame, nr * d->frames_analyzed);
  srenew(d->orientation_per_frame, nr * d->frames_analyzed);
  srenew(d->order_per_frame, nr * d->frames_analyzed);
  srenew(d->volume_per_frame, d->frames_analyzed);
  srenew(d->box_per_frame, d->frames_analyzed);
  srenew(d->smectic_palermo_sine, d->frames_analyzed);
  srenew(d->smectic_palermo_cosine, d->frames_analyzed);
  srenew(d->smectic_palermo_sine_up, d->frames_analyzed);
  srenew(d->smectic_palermo_cosine_up, d->frames_analyzed);
  srenew(d->smectic_palermo_sine_down, d->frames_analyzed);
  srenew(d->smectic_palermo_cosine_down, d->frames_analyzed);
  srenew(d->average_rotation1, nr * d->frames_analyzed);
  srenew(d->average_rotation2, nr * d->frames_analyzed);
  srenew(d->number_of_downs, d->frames_analyzed);
  srenew(d->number_of_ups, d->frames_analyzed);
  srenew(d->number_of_up_downs, d->frames_analyzed);
  srenew(d->min_box_length, d->frames_analyzed);
  srenew(d->P2_order_per_frame, nr * d->frames_analyzed);
  srenew(d->p4_order_per_frame, nr * d->frames_analyzed);
  srenew(d->lambda0_order_per_frame, nr * d->frames_analyzed);
}

static void
write_frame_time_to_filepointers(t_analysisdata *d, real time) {
  if (d->fp_order)
    fprintf(d->fp_order, "%10.3f", time*output_env_get_time_factor(*(d->oenv)));
  if (d->fp_p2_order)
    fprintf(d->fp_p2_order, "%10.3f", time*output_env_get_time_factor(*(d->oenv)));
  if (d->fp_p4_order)
    fprintf(d->fp_p4_order, "%10.3f", time*output_env_get_time_factor(*(d->oenv)));
  if (d->fp_lambda0_order)
    fprintf(d->fp_lambda0_order, "%10.3f", time*output_env_get_time_factor(*(d->oenv)));
}

static void
write_newline_to_filepointers(t_analysisdata *d) {
  if (d->fp_order)
    fprintf(d->fp_order, "\n");
  if (d->fp_lambda0_order)
```

```
    fprintf(d->fp_lambda0_order, "\n");
  if (d->fp_p2_order)
    fprintf(d->fp_p2_order, "\n");
  if (d->fp_p4_order)
    fprintf(d->fp_p4_order, "\n");
}

static void
create_replica_vectors(int repl_per_dim, matrix box, rvec** replicas, int *index) {
  rvec xdir, ydir, zdir;
  *index=0;
  int j, k, l;
  for ( j= 0 ; j < repl_per_dim ; ++j ) {
    for ( k= 0 ; k < repl_per_dim ; ++k ) {
      for ( l= 0 ; l < repl_per_dim ; ++l ) {
        svmul(j-repl_per_dim/2. + 0.5, box[0], xdir);
        svmul(k-repl_per_dim/2. + 0.5, box[1], ydir);
        svmul(l-repl_per_dim/2. + 0.5, box[2], zdir);
        rvec_add((*replicas)[*index], xdir, (*replicas)[*index]);
        rvec_add((*replicas)[*index], ydir, (*replicas)[*index]);
        rvec_add((*replicas)[*index], zdir, (*replicas)[*index]);
        ++(*index);
      }
    }
  }
  if (*index == 0)
    *index = 1;
}

// Pair correlation functions with Legendre polynomials
/*
 * in other words, in a spherical (or cylindrical if you are focused just in one
 * direction), one computes the histogram of the distances r1-r2 like you do for g(r).
 * in a separate histogram with the same bins you instead sum the values of,
 * e.g. P1 = (u1.u2) or P2= 3/2-1/2 (u1.u2)^2.
 * at the end of the calculation you divide the value of each bin of the second histogram
 * for the corresponding one in the first histogram, so in practice you compute
 * sum [ P(u1.u2) / sum molecules ] for each interval of r1-r2.
 */

  static void
calculate_gL_pair_distribution(
    matrix *box,
    int frames,
    int molecules,
    int no_of_selections,
    int selection,
    rvec** axis_of_orientation,
    rvec** axis_of_inertia,
    rvec** cogs,
    real* volumes,
    int samples,
    real max_radius,
    real* g0,
    real* g1,
    real* g2)
{
  real bucket_width=max_radius/samples;
  real average_volume = 0;
  int i;
  for ( i = 0; i < samples; ++i ) {
    g0[i] = 0;
    g1[i] = 0;
    g2[i] = 0;
  }
  rvec cog1 = {0, 0, 0};
  rvec cog2 = {0, 0, 0};
  rvec dist_vec = {0, 0, 0};
  real dist;
  int frame;
  t_pbc *l_pbc;
  snew(l_pbc, 1);
  for (frame = 0; frame < frames; ++frame) {
    set_pbc(l_pbc, -1, box[frame]);

    average_volume += volumes[frame];
    fprintf(stderr, "\rCalculating autocorrelation: %7.3f %%", 100.0*(frame+1)/(frames));
    int molecule1;
    for ( molecule1 = 0; molecule1 < molecules-1; ++molecule1 ) {
      int molecule2;
      for ( molecule2 = molecule1; molecule2 < molecules; ++molecule2 ) {
        if ( molecule2 != molecule1 ) {
          pbc_dx(l_pbc, cogs[frame][molecule1], cogs[frame][molecule2], dist_vec);
          real dist = norm(dist_vec);
          if ( dist < max_radius ) {
            int bucket = dist * samples / max_radius + 0.5;
            real cos_alpha =
              iprod( axis_of_orientation[no_of_selections * frame + selection][molecule1],
                  axis_of_orientation[no_of_selections * frame + selection][molecule2] );
            real order_p2 = 1.5*
              sqr(iprod(axis_of_inertia[no_of_selections*frame+selection][molecule1],
                    axis_of_inertia[no_of_selections*frame+selection][molecule2] )) - 0.5;
            ++g0[bucket];
            g1[bucket] += cos_alpha;
            g2[bucket] += order_p2;
          }
        }
      }
    }
  }
  average_volume /= frames;
  for ( i = 0; i < samples; ++i ) {
    if ( g0[i] != 0 ) {
      g1[i] /= g0[i];
      g2[i] /= g0[i];
    }
    real radius = 1.0 * i/samples * max_radius + bucket_width/2;
    real rim_volume = 4.18879033*((radius)*(radius)*(radius)-(radius-bucket_width/2)*
        (radius-bucket_width/2)*(radius-bucket_width/2));
    g0[i] /= rim_volume * frames * molecules * (molecules -1 ) / average_volume;
  }
}

static void
calculate_average_rotation(int frames, int molecules, int no_of_selections, int selection,
    rvec** axis_of_inertia, real* average_rotation1, real* average_rotation2 ) {
  int dt;
  for (dt = 0; dt < frames; ++dt) {
    average_rotation1[dt] = 0;
    average_rotation2[dt] = 0;
    int combinations = 0;
    int frame;
    for (frame = 0; frame < (frames-dt); ++frame) {
      int molecule;
      for ( molecule = 0 ; molecule < molecules; ++molecule ) {
        real cos = iprod(axis_of_inertia[no_of_selections * frame + selection][molecule],
            axis_of_inertia[no_of_selections * (frame+dt) + selection][molecule]);
        average_rotation1[dt] += cos;
        average_rotation2[dt] += 1.5*cos-0.5;
        ++combinations;
      }
    }
    average_rotation1[dt] /= combinations;
    average_rotation2[dt] /= combinations;
  }
}

static gmx_bool
is_in_cylinder(const rvec position, const rvec director, real radius, real height) {
  real r_max2 = sqr(height)+sqr(radius);
  real norm_square_position = norm2(position);
  /*
     if (r2 < r_max2 ) then ! Checking if molecule is inside the cylinder
```

```
   r      = sqrt(r2)
   cosr   = abs(z/r)
   sinr   = sqrt(1._rk - cosr**2)
   x      = r*sinr

   if (abs(z) < z_max .and. x < x_max) then
 */
if ( norm_square_position < r_max2 ) {
  real projection = fabs(iprod(position, director));
  real norm_position = sqrt(norm_square_position);
  real cosr  = fabs(projection/norm_position);
  real sinr  = sqrt(1. - sqr(cosr));
  real x     = norm_position*sinr;

  if (fabs(projection) < height && x < radius) {
    /*
     *  if ( projection <= height) {
     rvec radius1_vec, radius2_vec;
     cprod(position, director, radius1_vec);
     cprod(director, radius1_vec, radius2_vec);
     unitv(radius1_vec, radius1_vec);
     unitv(radius2_vec, radius2_vec);
     real radius1 = iprod(position, radius1_vec);
     real radius2 = iprod(position, radius2_vec);
     if ( sqr(radius1) + sqr(radius2) <= sqr(radius) ) */
    return TRUE;
  }
  }
  return FALSE;
}
/*! \brief
 * Function that does the analysis for a single frame.
 *
 * It is called once for each frame.
 */
static int analyze_frame(t_topology *top, t_trxframe *fr, t_pbc *pbc,
    int nr, gmx_ana_selection_t *sel[], void *data)
{
  t_analysisdata *d = (t_analysisdata *)data;
  if ( d->start_time_captured && ! d->time_step_captured )
  {
    d->time_step = fr->time - d->start_time;
    d->time_step_captured = TRUE;
  }
  if ( ! d->start_time_captured )
  {
    d->start_time = fr->time;
    d->start_time_captured = TRUE;
  }

  ++d->frames_analyzed;
  int frame_in_array = d->frames_analyzed - 1;

  resize_arrays(d, 1);
  write_frame_time_to_filepointers(d, fr->time);

  d->volume_per_frame[frame_in_array] = det(fr->box);
  copy_mat(fr->box, d->box_per_frame[frame_in_array]);

  rvec *backup_x;
  snew (backup_x, fr->natoms);
  int atom = 0;
  rvec center = { -fr->box[XX][XX]/2, -fr->box[YY][YY]/2, -fr->box[ZZ][ZZ]/2 };
  for ( atom = 0 ; atom < fr->natoms; ++atom )
    rvec_add(fr->x[atom], center, backup_x[atom]);

  int no_of_entities=top->mols.nr;
  if ( d->no_of_entities != -1 )
    no_of_entities=d->no_of_entities;
  if ( 0 != no_of_entities )
  {
    int no_of_atoms_in_one_selected_entity = sel[0]->g->isize / no_of_entities;
int no_of_atoms_in_one_selected_entity_orientation_point2;

d->no_of_axes[0] = no_of_entities;
rvec *axis_per_molecule = NULL;
snew(axis_per_molecule, no_of_entities);
rvec *cog_per_molecule = NULL;
snew(cog_per_molecule, no_of_entities);
rvec *orientation_per_molecule = NULL;
snew(orientation_per_molecule, no_of_entities);

int first_atom_in_molecule = 0;
atom_id* index;
snew(index, no_of_atoms_in_one_selected_entity);
atom_id* index_orientation_point2;
for ( first_atom_in_molecule = 0; first_atom_in_molecule < sel[0]->g->isize;
    first_atom_in_molecule += no_of_atoms_in_one_selected_entity) {
  int atom_index = 0;
  for (atom_index=0; atom_index<no_of_atoms_in_one_selected_entity; ++atom_index)
  {
    index[atom_index] = sel[0]->g->index[first_atom_in_molecule + atom_index];
  }

  matrix axes;
  rvec inertia, xcm;
  sub_xcm(fr->x, no_of_atoms_in_one_selected_entity, index, top->atoms.atom, xcm,
      FALSE);
  principal_comp(no_of_atoms_in_one_selected_entity, index, top->atoms.atom, fr->x,
      axes, inertia);
  int pos, pos2, pos3;
  if ( inertia[XX] < inertia[YY] )
  {
    if ( inertia[XX] < inertia[ZZ] )
    {
      pos = XX;
      if ( inertia [YY] < inertia[ZZ] )
      {
        pos2 = YY;
        pos3 = ZZ;
      } else {
        pos3 = YY;
        pos2 = ZZ;
      }
    } else {
      pos = ZZ;
      if ( inertia [YY] < inertia[XX] ) {
        pos2 = YY;
        pos3 = XX;
      } else {
        pos3 = YY;
        pos2 = XX;
      }
    }
  } else {
    if ( inertia[YY] < inertia[ZZ] ) {
      pos = YY;
      if ( inertia [ZZ] < inertia[XX] ) {
        pos2 = ZZ;
        pos3 = XX;
      } else {
        pos3 = ZZ;
        pos2 = XX;
      }
    } else {
      pos = ZZ;
      if ( inertia [YY] < inertia[XX] ) {
        pos2 = YY;
        pos3 = XX;
      } else {
        pos3 = YY;
        pos2 = XX;
      }
    }
```

```
        }

        // take the main axis of inertia and store it.
        copy_rvec(axes[pos],
            axis_per_molecule[first_atom_in_molecule/no_of_atoms_in_one_selected_entity]);

        // calculate the center of geometry for the current selection entity
        rvec cog;
        gmx_calc_cog_pbc(top, backup_x, pbc,
            no_of_atoms_in_one_selected_entity, index, cog);
        copy_rvec(cog,
            cog_per_molecule[first_atom_in_molecule/no_of_atoms_in_one_selected_entity]);

        // calculate an orientation axis based on the first
        // and the last atom in the selection or on the atom indices specified by the user
        rvec entity_orientation = {0, 0, 0};
        real dist = -1;
        real old_dist = -1;
        if ( d->orientation_1 == -1 )
          d->orientation_1 = 0;
        if ( d->orientation_2 == -1 )
          d->orientation_2 = no_of_atoms_in_one_selected_entity -1;
        pbc_dx(pbc, backup_x[index[d->orientation_1]], backup_x[index[d->orientation_2]],
            entity_orientation);
        copy_rvec(entity_orientation,
            orientation_per_molecule[first_atom_in_molecule/
            no_of_atoms_in_one_selected_entity]);
      }
      free(index);

      // copy arrays to the main arrays
      d->axis_of_inertia_per_frame[frame_in_array] = axis_per_molecule;
      d->cogs_per_frame[frame_in_array] = cog_per_molecule;
      d->orientation_per_frame[frame_in_array] = orientation_per_molecule;
    }
    else
    {
      fprintf( stderr, "The selection with name %s and selection ", sel[0]->name);
      fprintf( stderr, "string %s does not contain molecules.\r", sel[0]->selstr);
    }

    matrix order_tensor;
    rvec director;
    rvec second_director;
    rvec third_director;
    real order = -1, lambda0_order = -1;
    if ( 0 == create_order_tensor(no_of_entities, 0, (d->frames_analyzed - 1), 1, 1,
          d->axis_of_inertia_per_frame, order_tensor) )
    {
      get_order_from_tensor(order_tensor, &order, &director, 0);
      d->order_per_frame[(d->frames_analyzed-1)] = order;
      get_order_from_tensor(order_tensor, &lambda0_order, &second_director, 1);
      if (d->fp_lambda0_order) {
        d->lambda0_order_per_frame[(d->frames_analyzed-1)] = lambda0_order;
      }
      real dummy;
      get_order_from_tensor(order_tensor, &dummy, &third_director, 2);
    }

    if (d->fp_order) {
      fprintf(d->fp_order, " %.3f", order);
    }
    if (d->fp_lambda0_order) {
      fprintf(d->fp_lambda0_order, " %.3f", lambda0_order);
    }

    t_pdbinfo *p2_pdbinfo = NULL;
    if ( ( d->fn_up_down_pdb && ((int)fr->time == (int)d->t_up_down_pdb) ) ||
         ( d->fn_p2_order_pdb && ((int)fr->time == (int)d->t_p2_order_pdb) ) ) {
      if (top->atoms.pdbinfo == NULL)
        snew(top->atoms.pdbinfo, top->atoms.nr);
    }
```

```
if (  d->fn_p2_order_pdb && ((int)fr->time == (int)d->t_p2_order_pdb)  ) {
  p2_pdbinfo = top->atoms.pdbinfo;
}

t_pdbinfo *up_down_pdbinfo = NULL;
if ( ( d->fn_up_down_pdb && ((int)fr->time == (int)d->t_up_down_pdb) ) ) {
  if (  d->fn_p2_order_pdb && ((int)fr->time == (int)d->t_p2_order_pdb)  ) {
    snew(up_down_pdbinfo, top->atoms.nr);
  } else {
    up_down_pdbinfo = top->atoms.pdbinfo;
  }
}

static int replica = 0;
if ( ( d->fp_smectic_palermo || d->fp_smectic || d->fp_gL_pair_distribution) &&
    (replica == 0) ) {
  snew(d->replica , d->repl_per_dim*d->repl_per_dim*d->repl_per_dim);
  create_replica_vectors(d->repl_per_dim , fr->box, &(d->replica),
      &(d->number_of_replicas));
  replica = 1;
}

if (
    d->fn_p2_order_pdb
    || d->fn_up_down_pdb
    || d->fp_p2_order
    || d->fp_p4_order
    || d->fp_p2_histogram
    || d->fp_cos_histogram
    || d->fp_smectic_palermo
    || d->fp_smectic
    || d->fp_average_rotation
    || d->fp_gL_pair_distribution
   ) {
  real p2_order = 0.0;
  real p4_order = 0.0;
  d->number_of_downs[frame_in_array] = 0;
  d->number_of_ups[frame_in_array] = 0;
  d->number_of_up_downs[frame_in_array] = 0;

  int entity=0;

  real *smectic_palermo_cosine_per_frame , *smectic_palermo_sine_per_frame;
  snew(smectic_palermo_cosine_per_frame , d->samples);
  snew(smectic_palermo_sine_per_frame , d->samples);
  real *smectic_palermo_cosine_up_per_frame, *smectic_palermo_sine_up_per_frame;
  snew(smectic_palermo_cosine_up_per_frame, d->samples);
  snew(smectic_palermo_sine_up_per_frame, d->samples);
  real *smectic_palermo_cosine_down_per_frame, *smectic_palermo_sine_down_per_frame;
  snew(smectic_palermo_cosine_down_per_frame, d->samples);
  snew(smectic_palermo_sine_down_per_frame , d->samples);
  int sample;
  for ( sample = 0 ; sample < d->samples; ++sample) {
    smectic_palermo_cosine_per_frame[sample] = 0;
    smectic_palermo_sine_per_frame[sample] = 0;
    smectic_palermo_cosine_up_per_frame[sample] = 0;
    smectic_palermo_sine_up_per_frame[sample] = 0;
    smectic_palermo_cosine_down_per_frame[sample] = 0;
    smectic_palermo_sine_down_per_frame[sample] = 0;
  }
  int counted_molecules = 0;
  int counted_molecules_up = 0;
  int counted_molecules_down = 0;

  real max_radius = d->max_radius;
  if (max_radius <= 0 )
    max_radius = -sqrt(2) * max_value(center , 3);
  real max_height = 80;
  if ( d->fp_smectic || d->fp_smectic_palermo || d->fn_up_down_pdb) {
    matrix multiple_box;
    msmul(fr->box, d->repl_per_dim , multiple_box);
```

```
set_pbc(pbc, -1, multiple_box);
for ( entity = 0 ; entity < no_of_entities ; ++entity ) {

  int repl1;
  for (repl1=0; repl1<d->number_of_replicas; ++repl1) {
    rvec cog1;
    if ( d->fp_smectic_palermo ) {
      rvec_add (d->cogs_per_frame[frame_in_array][entity],
          d->replica[repl1], cog1);
    } else {
      if (entity == 0) {
        srenew(d->cogs_per_frame[frame_in_array],
            d->number_of_replicas*no_of_entities);
      } else {
        copy_rvec(d->cogs_per_frame[frame_in_array][repl1*no_of_entities+entity],
            cog1) ;
      }
    }

    rvec axis_first;
    copy_rvec(d->orientation_per_frame[frame_in_array][entity], axis_first) ;
    real orient_first = 0;
    if ( norm(d->versor) == 0) {
      orient_first = iprod(director, axis_first);
    } else {
      orient_first = iprod(d->versor, axis_first);
    }

    if ( d->fp_smectic_palermo ) {
      if ( is_in_cylinder(cog1, director, max_radius*0.5,
            max_radius*d->repl_per_dim*0.5) == TRUE ) {
        for ( sample = 0; sample < d->samples; ++sample) {
          real radius = (1.0 + sample)/d->samples*d->max_radius;
          real inv_radius = d->samples / ( d->max_radius * ( 1.0 + sample ) );
          real projection = iprod(director, cog1);
          real proj_pi_2 = 2 * M_PI * projection * inv_radius;
          smectic_palermo_cosine_per_frame[sample] += cos(proj_pi_2);
          smectic_palermo_sine_per_frame[sample] += sin(proj_pi_2);
          if (orient_first <= 0 ) {
            smectic_palermo_cosine_up_per_frame[sample] += cos(proj_pi_2);
            smectic_palermo_sine_up_per_frame[sample] += sin(proj_pi_2);
            if ( sample == 0 )
              ++counted_molecules_up;
          } else {
            smectic_palermo_cosine_down_per_frame[sample] += cos(proj_pi_2);
            smectic_palermo_sine_down_per_frame[sample] += sin(proj_pi_2);
            if ( sample == 0 )
              ++counted_molecules_down;
          }
        }
        ++counted_molecules;
      }
    }

    if (orient_first <= 0 && repl1 == 0 ) {
      if ( d->fn_up_down_pdb && ((int)fr->time == (int)d->t_up_down_pdb) ) {
        int j;
        for( j=top->mols.index[entity]; j < top->mols.index[entity+1]; ++j ) {
          up_down_pdbinfo[j].bfac = 99;
        }
      }
      ++d->number_of_downs[frame_in_array];
    }

    if (orient_first > 0 && repl1 == 0 ) {
      if ( d->fn_up_down_pdb && ((int)fr->time == (int)d->t_up_down_pdb) ) {
        int j;
        for( j=top->mols.index[entity]; j < top->mols.index[entity+1]; ++j ) {
          up_down_pdbinfo[j].bfac = 0;
        }
      }
      ++d->number_of_ups[frame_in_array];
```

```
}

if ( d->fp_smectic ) {
  if ( entity < no_of_entities-1 ) {
    int second_entity;
    for ( second_entity = entity + 1; second_entity < no_of_entities;
        ++second_entity ) {
      int repl2;
      for (repl2=0; repl2<d->number_of_replicas; ++repl2) {
        rvec cog2;
        if (entity == 0 && repl2 != 0) {
          rvec_add (d->cogs_per_frame[frame_in_array][second_entity],
              d->replica[repl2], cog2);
          copy_rvec(cog2, d->cogs_per_frame[frame_in_array]
              [repl2*no_of_entities+second_entity]);
        } else {
          copy_rvec(d->cogs_per_frame[frame_in_array]
              [repl2*no_of_entities+second_entity], cog2);
        }

        rvec axis_second;
        copy_rvec(d->orientation_per_frame[frame_in_array][second_entity],
            axis_second) ;
        real orient_second = 0;
        if ( norm(d->versor) == 0)
          orient_second = iprod(axis_second, director);
        else
          orient_second = iprod(axis_second, d->versor);

        rvec diff;
        pbc_dx(pbc, cog1, cog2, diff);

        double r = norm(diff);
        if (r <= max_radius*d->repl_per_dim) {
          int histindex = ((r * d->sm_bins) / (max_radius*d->repl_per_dim));
          ++d->radial_distribution_hist[histindex];
        }

        real r_z = fabs(iprod(diff, director));
        real r_x = sqrt(norm2(diff) - sqr(r_z));

        int histindex = 0;
        if ( r_z < max_radius*d->repl_per_dim*0.5) {
          if ( r_x <= max_radius) {
            histindex = ((r_z * d->sm_bins) /
                (max_radius*d->repl_per_dim*0.5));
            ++d->sm_hist[histindex];
            if (orient_first*orient_second > 0 ) {
              if (orient_first < 0 ) {
                ++d->sm_down_hist[histindex];
              } else {
                ++d->sm_up_hist[histindex];
              }
            } else {
              ++d->sm_up_down_hist[histindex];
            }
          }
        }

        real y = fabs(iprod(diff, second_director));
        real x = fabs(iprod(diff, third_director));

        if ( y < max_radius*d->repl_per_dim && x*x+r_z*r_z < sqr(max_radius) ) {
          histindex = ((y * d->sm_bins) / (max_radius*d->repl_per_dim));
          ++d->sm_hist_y[histindex];
        }
        if ( x < max_radius*d->repl_per_dim && y*y+r_z*r_z < sqr(max_radius) ) {
          histindex = ((x * d->sm_bins) / (max_radius*d->repl_per_dim));
          ++d->sm_hist_x[histindex];
        }
      }
    }
```

```
          }

        }
      }
    }

    if ( d->fp_p2_order || d->fp_p4_order
        || d->fp_p2_histogram || d->fp_cos_histogram ) {
      real cos_beta = iprod(d->axis_of_inertia_per_frame[frame_in_array][entity],
          director);
      real sqr_cos_beta = sqr(cos_beta);
      if ( d->fp_p2_order || d->fp_p2_histogram) {
        p2_order += 1.5 * sqr_cos_beta - 0.5;
        if ( d->t_p2_hist == -1 || ((int)fr->time == (int)d->t_p2_hist ) ) {
          ++(d->p2_histogram[(int)(((sqr_cos_beta+0.5)/1.5)*
                d->p2_histogram_number_bins)]);
        }
      }
      if ( d->fp_cos_histogram && ( d->t_cos_hist == -1 ||
            ((int)fr->time == (int)d->t_cos_hist ) ) ) {
        ++(d->cos_histogram[(int)(((cos_beta+1.)/2)*d->cos_histogram_number_bins)]);
      }
      if ( d->fp_p4_order )
        p4_order += (35 * sqr(sqr_cos_beta) -
            30 * sqr_cos_beta + 3) / 8.;
    }
    if ( d->fn_p2_order_pdb && ((int)fr->time == (int)d->t_p2_order_pdb) ) {
      int j;
      for( j=top->mols.index[entity]; j < top->mols.index[entity+1]; ++j ) {
        p2_pdbinfo[j].bfac = sqr(iprod(
              d->axis_of_inertia_per_frame[frame_in_array][entity], director)) * 100;
      }
    }
  }
  if (d->fp_smectic_palermo ) {
    for ( sample = 0 ; sample < d->samples; ++sample) {
      if (counted_molecules > 0) {
        smectic_palermo_cosine_per_frame[sample] /= counted_molecules;
        smectic_palermo_sine_per_frame[sample] /= counted_molecules;
        smectic_palermo_cosine_up_per_frame[sample] /= counted_molecules_up;
        smectic_palermo_sine_up_per_frame[sample] /= counted_molecules_up;
        smectic_palermo_cosine_down_per_frame[sample] /= counted_molecules_down;
        smectic_palermo_sine_down_per_frame[sample] /= counted_molecules_down;
        real radius = (1.+sample)/d->samples*d->max_radius;
        real correction = radius/(M_PI*max_radius*d->repl_per_dim)*
          sin(M_PI*max_radius*d->repl_per_dim/radius);
        smectic_palermo_cosine_per_frame[sample] -= correction;
        smectic_palermo_cosine_up_per_frame[sample] -= correction;
        smectic_palermo_cosine_down_per_frame[sample] -= correction;
      }
    }
    d->smectic_palermo_cosine[frame_in_array] = smectic_palermo_cosine_per_frame;
    d->smectic_palermo_sine[frame_in_array] = smectic_palermo_sine_per_frame;
    d->smectic_palermo_cosine_up[frame_in_array] = smectic_palermo_cosine_up_per_frame;
    d->smectic_palermo_sine_up[frame_in_array] = smectic_palermo_sine_up_per_frame;
    d->smectic_palermo_cosine_down[frame_in_array] =
      smectic_palermo_cosine_down_per_frame;
    d->smectic_palermo_sine_down[frame_in_array] = smectic_palermo_sine_down_per_frame;
  }

  if (d->fp_p2_order) {
    p2_order /= no_of_entities;
    d->P2_order_per_frame[frame_in_array] = p2_order;
    fprintf(d->fp_p2_order, " %.3f", p2_order);
  }
  if (d->fp_p4_order) {
    p4_order /= no_of_entities;
    d->p4_order_per_frame[frame_in_array] = p4_order;
    fprintf(d->fp_p4_order, " %.3f", p4_order);
  }

  if ( d->fn_p2_order_pdb && ((int)fr->time == (int)d->t_p2_order_pdb) )
```

```
      write_sto_conf(d->fn_p2_order_pdb, "molecular order", &top->atoms, backup_x, NULL,
          fr->ePBC, fr->box);

    if ( d->fn_up_down_pdb && ((int)fr->time == (int)d->t_up_down_pdb) ) {
      if ( d->fn_p2_order_pdb && ((int)fr->time == (int)d->t_p2_order_pdb) ) {
        top->atoms.pdbinfo = up_down_pdbinfo;
      }
      write_sto_conf(d->fn_up_down_pdb, "up down", &top->atoms, backup_x, NULL, fr->ePBC,
          fr->box);
    }

  }
  sfree(backup_x);

  write_newline_to_filepointers(d);
  /* We need to return 0 to tell that everything went OK */
  d->end_time = fr->time;
  return 0;
}

static void
  init_arrays (t_analysisdata *d)
  {
    snew(d->average_rotation1, 1);
    snew(d->average_rotation2, 1);
    snew(d->axis_of_inertia_per_frame, 1);
    snew(d->cogs_per_frame, 1);
    snew(d->orientation_per_frame, 1);
    snew(d->sm_hist, d->sm_bins);
    snew(d->sm_hist_x, d->sm_bins);
    snew(d->sm_hist_y, d->sm_bins);
    snew(d->smectic_palermo_cosine, 1);
    snew(d->smectic_palermo_sine, 1);
    snew(d->smectic_palermo_cosine_up, 1);
    snew(d->smectic_palermo_sine_up, 1);
    snew(d->smectic_palermo_cosine_down, 1);
    snew(d->smectic_palermo_sine_down, 1);
    snew(d->sm_up_down_hist, d->sm_bins);
    snew(d->radial_distribution_hist, d->sm_bins);
    if ( d->fp_p2_histogram )
      snew(d->p2_histogram, d->p2_histogram_number_bins);
    if ( d->fp_cos_histogram )
      snew(d->cos_histogram, d->cos_histogram_number_bins);
    snew(d->sm_up_hist, d->sm_bins);
    snew(d->sm_down_hist, d->sm_bins);
    snew(d->no_of_axes, 1);
    snew(d->volume_per_frame, 1);
    snew(d->box_per_frame, 1);
    snew(d->g0_pair_distribution, d->samples);
    snew(d->g1_pair_distribution, d->samples);
    snew(d->g2_pair_distribution, d->samples);
    snew(d->min_box_length, 1);
    snew(d->number_of_downs, 1);
    snew(d->number_of_ups, 1);
    snew(d->number_of_up_downs, 1);
    snew(d->order_per_frame, 1);
    snew(d->P2_order_per_frame, 1);
    snew(d->p4_order_per_frame, 1);
    snew(d->lambda0_order_per_frame, 1);
  }

static void
  init_file_pointers(t_analysisdata *d, t_filenm fnm[], int NFILE, output_env_t *oenv) {
    d->fn_p2_order_pdb = opt2fn_null("-p2_pdb", NFILE, fnm);
    d->fn_up_down_pdb = opt2fn_null("-up_down_pdb", NFILE, fnm);
    /* We also open the output file */
    d->fp_order              =    NULL;
    d->fp_order = xvgropen(opt2fn("-o", NFILE, fnm),
        "Biggest Eigenvalue from Order Tensor",
        output_env_get_xvgr_tlabel(*oenv), "Degree of order", *oenv);
    d->fp_lambda0_order      =    NULL;
    if (opt2bSet("-lamb0", NFILE, fnm)) {
      d->fp_lambda0_order = xvgropen(opt2fn("-lamb0", NFILE, fnm),
          "Middle Eigenvalue times -2 from Order Tensor",
          output_env_get_xvgr_tlabel(*oenv), "Degree of order", *oenv);
    }
    d->fp_p2_order              =   NULL;
    if (opt2bSet("-p2", NFILE, fnm)) {
      d->fp_p2_order = xvgropen(opt2fn("-p2", NFILE, fnm),
          "P2 Order Parameter with Director",
          output_env_get_xvgr_tlabel(*oenv), "Degree of order", *oenv);
    }
    d->fp_p4_order              =   NULL;
    if (opt2bSet("-p4", NFILE, fnm)) {
      d->fp_p4_order = xvgropen(opt2fn("-p4", NFILE, fnm),
          "P4 Order Parameter with Director",
          output_env_get_xvgr_tlabel(*oenv), "Degree of order", *oenv);
    }
    d->fp_p2_histogram          =   NULL;
    if (opt2bSet("-p2_hist", NFILE, fnm)) {
      d->fp_p2_histogram = xvgropen_type(opt2fn("-p2_hist", NFILE, fnm),
          "Distribution of P2 Order Parameter",
          "Distribution", "# of values", exvggtNR, *oenv);
      xvgr_line_props(d->fp_p2_histogram, 0, elNone, ecWhite, *oenv);
    }
    d->fp_cos_histogram         =   NULL;
    if (opt2bSet("-cos_hist", NFILE, fnm)) {
      d->fp_cos_histogram = xvgropen_type(opt2fn("-cos_hist", NFILE, fnm),
          "Distribution of the Cosine of Tilt Angle",
          "Distribution", "# of values", exvggtNR, *oenv);
      xvgr_line_props(d->fp_cos_histogram, 0, elNone, ecWhite, *oenv);
    }
    d->fp_smectic_palermo               =   NULL;
    if (opt2bSet("-smectic_palermo", NFILE, fnm)) {
      d->fp_smectic_palermo = fopen(opt2fn("-smectic_palermo", NFILE, fnm), "w");
    }
    d->fp_smectic               =   NULL;
    if (opt2bSet("-smectic", NFILE, fnm)) {
      d->fp_smectic = xvgropen(opt2fn("-smectic", NFILE, fnm), "Smectic Order Parameter",
          "r / nm", "Density", *oenv);
    }
    d->fp_average_rotation      =   NULL;
    if (opt2bSet("-rotation", NFILE, fnm)) {
      d->fp_average_rotation = xvgropen(opt2fn("-rotation", NFILE, fnm),
          "Average Rotation",
          output_env_get_xvgr_tlabel(*oenv), "<z\\s0\\N*z\\scurr\\N>", *oenv);
    }
    d->fp_gL_pair_distribution          =   NULL;
    if (opt2bSet("-gL",NFILE,fnm)) {
      d->fp_gL_pair_distribution = xvgropen(opt2fn("-gL", NFILE, fnm),
          "gL autocorrelation function",
          "radius/nm", "<dirac_delta(r-r\\sij\\N)*(axis\\si\\N*axis\\sj\\N>", *oenv);
    }
  }

static void
  init_input_variables(gmx_ana_traj_t *trj, t_analysisdata *d, int argsize, t_pargs pa[],
      int argc, char *argv[], int descsize, const char* desc[], t_filenm fnm[], int NFILE,
      output_env_t *oenv)
  {
    d->b_total                      = FALSE;
    d->t_p2_order_pdb               = -1;
    d->t_p2_hist                    = -1;
    d->p2_histogram_number_bins     = 20;
    d->t_up_down_pdb                = -1;
    d->t_cos_hist                   = -1;
    d->max_radius                   = -1;
    d->samples                      = 1000;
    d->cos_histogram_number_bins    = 20;
    d->sm_bins                      = 400;
    d->repl_per_dim                 = 3;
    d->no_of_entities               = -1;
    d->versor[0]                    = 0;
    d->versor[1]                    = 0;
```

```
      d->versor[2]                    = 0;
      d->oenv                         = oenv;
      d->start_time_captured          = FALSE;
      d->time_step_captured           = FALSE;
      d->frames_analyzed              = 0;
      d->number_of_replicas           = 1;
      d->replica                      = NULL;
      d->orientation_1                = -1;
      d->orientation_2                = -1;

      parse_trjana_args(trj, &argc, argv, PCA_CAN_VIEW | PCA_TIME_UNIT, NFILE, fnm,
          argsize, pa, descsize, desc, 0, NULL, oenv);
    }

  /*! \brief
   * Function that implements the a tool to determine the degree of order in
   * a trajectory per frame or for the whole trajectory, respectively.
   *
   * Following the style of Gromacs analysis tools, this function is called
   * \p gmx_order_tensor.
   */
  int
    gmx_order_tensor(int argc, char *argv[])
    {
      const char *desc[] = {
        "This tool calculates the second rank order parameter ",
        "by deriving the second rank order tensor from the ",
        "inertia axes of each molecule in a selection group and ",
        "subsequently diagonalizing this tensor. An order parameter ",
        "can then be derived by taking \n",
        "\n",
        "1. the largest eigenvalue ([TT]-o[tt]), \n",
        "\n",
        "2. the second largest eigenvalue (lambda_0) and multiply \n",
        "   that with -2 ([TT]-lamb0[tt]), \n",
        "\n",
        "3. the angle (beta) between the instantanous director\n ",
        "   and the inertia axis of each molecule and using the formula: \n",
        "   P2 = < (3 * cos(beta)^2 - 1)/2 > \n",
        "   The brackets mean a temporal average. ([TT]-p2[tt]) \n",
        "\n",
        "4. the angle (beta) between the instantanous director \n",
        "   and the inertia axis of each molecule and using the formula: \n",
        "   P4 = < (35 * cos(beta)^4 - 30 * cos(beta)^2 + 3) / 8 > \n",
        "   The brackets mean a temporal average. ([TT]-p4[tt]) \n",
        "\n",
        "The deviation of one molecule from the instantanous director",
        "can then be stored for a selected frame ([TT]-tpdb[tt]) in the",
        "B factor of a pdb file ([TT]-pdb[tt]).\n",
        "\n",
        "A histogram describing the distribution of the deviation of the molecules",
        "from the instantanous director can be written ([TT]-hist[tt]) either for ",
        "a certain frame ([TT]-thist[tt]) or for the whole trajectory.",
        "The number of bins to sort the deviations can also be specified ([TT]-bins[tt]).\n",
      };

      /* Output files */
      t_filenm fnm[] = {
        {efXVG, "-o",               "tensor_p2",              ffWRITE},
        {efXVG, "-lamb0",           "lambda0",                ffOPTWR},
        {efXVG, "-p2",              "p2",                     ffOPTWR},
        {efXVG, "-p4",              "p4",                     ffOPTWR},
        {efXVG, "-p2_hist",         "hist_p2",                ffOPTWR},
        {efXVG, "-cos_hist",        "hist_cos",               ffOPTWR},
        {efXVG, "-smectic",         "smectic",                ffOPTWR},
        {efXVG, "-smectic_palermo", "smectic_palermo",          ffOPTWR},
        {efXVG, "-rotation",        "average_rotation",       ffOPTWR},
        {efXVG, "-gL",              "gL_pair_distribution",   ffOPTWR},
        {efPDB, "-p2_pdb",          "p2_dev",                 ffOPTWR},
        {efPDB, "-up_down_pdb",     "up_down",                ffOPTWR}
      };
#define NFILE asize(fnm)
```

```
gmx_ana_traj_t      *trj;
output_env_t        oenv;
t_analysisdata      d;
int                 ngrps;
gmx_ana_selection_t **sel;

CopyRight(stderr, argv[0]);
/* Here, we can use flags to specify requirements for the selections and/or
 * other features of the library. */

gmx_ana_traj_create(&trj, ANA_REQUIRE_TOP);
gmx_ana_set_nrefgrps(trj, 0);
gmx_ana_set_nanagrps(trj, -1);
/* If required, other functions can also be used to configure the library
 * before calling parse_trjana_args(). */

t_pargs             pa[] = {
  {   "-total",   FALSE, etBOOL, {&(d.b_total)},
    "determine the total order in trajectory." },
  {   "-samples",   FALSE, etINT, {&(d.samples)},
    "The number of bins used for [TT]-p2_hist[tt]." },
  {   "-radius",   FALSE, etREAL, {&(d.max_radius)},
    "The length of the radial distribution autocorrelation functions.\
      The cut-off distance for the probing cylinder in [TT]-smectic[tt].\
      [TT]-radius[tt] will be radius and height/2 of the cylinder.\
      If using -1 the program will take min(box_length)/sqrt(2) as \
      radius and frame_volume/radius^2/2 as height." },
  {   "-tp2_pdb",   FALSE, etTIME, {&(d.t_p2_order_pdb)},
    "The frame to use for option [TT]-p2_pdb[tt] (%t)" },
  {   "-tp2_hist",   FALSE, etTIME, {&(d.t_p2_hist)},
    "The frame to use for option [TT]-p2_hist[tt] (%t). \
      If -1 then the histogram is built over the whole trajectory." },
  {   "-p2_bins",   FALSE, etINT, {&(d.p2_histogram_number_bins)},
    "The number of bins used for [TT]-p2_hist[tt]." },
  {   "-tcos_hist",   FALSE, etTIME, {&(d.t_cos_hist)},
    "The frame to use for option [TT]-cos_hist[tt] (%t). \
      If -1 then the histogram is built over the whole trajectory."      },
  {   "-cos_bins",   FALSE, etINT, {&(d.cos_histogram_number_bins)},
    "The number of bins used for [TT]-cos_hist[tt]." },
  {   "-smbins",   FALSE, etINT, {&(d.sm_bins)},
    "The number of bins used for building the density distribution \
      for [TT]-smectic[tt]." },
  {   "-replcount",   FALSE, etINT, {&(d.repl_per_dim)},
    "The number of replicas in every dimension used for [TT]-smectic[tt]." },
  {   "-tup_down_pdb",   FALSE, etTIME, {&(d.t_up_down_pdb)},
    "The frame to use for option [TT]-up_down_pdb[tt] (%t)" },
  {   "-no_of_molecules",   FALSE, etINT, {&(d.no_of_entities)},
    "If there are different compounds in the trajectory this option specifies \
      the number of molecules that are regarded in the selection." },
  {   "-orient-1",   FALSE, etINT, {&(d.orientation_1)},
    "First atom in molecule to define the orientation." },
  {   "-orient-2",   FALSE, etINT, {&(d.orientation_2)},
    "Second atom in molecule to define the orientation." },
  {   "-versor",   FALSE, etRVEC, {&(d.versor)},   "Layer normal/Versor" }
};
init_input_variables(trj, &d, asize(pa), pa, argc, argv, asize(desc),
    desc, fnm, NFILE, &oenv);

gmx_ana_get_nanagrps(trj, &ngrps);
gmx_ana_get_anagrps(trj, &sel);

init_file_pointers(&d, fnm, NFILE, &oenv);
init_arrays(&d);

xvgr_selections(d.fp_order, trj);
if ( d.fp_lambda0_order )
  xvgr_selections(d.fp_lambda0_order, trj);
if ( d.fp_p2_order )
  xvgr_selections(d.fp_p2_order, trj);
if ( d.fp_p4_order )
  xvgr_selections(d.fp_p4_order, trj);
```

```c
if ( d.fp_p2_histogram )
  xvgr_selections(d.fp_p2_histogram, trj);
if ( d.fp_cos_histogram )
  xvgr_selections(d.fp_cos_histogram, trj);

/* Now, we do the actual analysis */
gmx_ana_do(trj, 0, &analyze_frame, &d);

int nframes=0;
gmx_ana_get_nframes(trj, &nframes);

char ** legend_names;
snew(legend_names, 1);

char *buf;
legend_names[0] = replace_str(sel[0]->name, "\"", "\\\"");
if (d.b_total)
{
  matrix order_tensor;
  rvec director;
  real order = -1;
  real std_dev = 0;
  if ( 0 == create_order_tensor(d.no_of_axes[0], 0, 0, nframes, 1,
        d.axis_of_inertia_per_frame, order_tensor) )
  {
    get_order_from_tensor(order_tensor, &order, &director, 0);
  }
  fprintf(stdout, "Director:\t%10.10f\t%10.10f\t%10.10f\n",
      director[0], director[1], director[2]);

  if ( d.frames_analyzed > 1 )
  {
    real deviation_sum = 0;
    real sum = 0;
    int j;
    for ( j = 0 ; j < d.frames_analyzed ; ++j )
    {
      sum = d.order_per_frame[j] - order;
      deviation_sum += sqr(sum);
    }
    std_dev = sqrt( 1. / ( d.frames_analyzed - 1. ) * deviation_sum ) ;
  }
  snew (buf, 1024);
  sprintf(buf, "%s: %.2f +/- %.2f", replace_str(sel[0]->name, "\"", "\\\""),
      order, std_dev);

  legend_names[0] = buf;
  fprintf(stdout, "Total P2 tensor order for group '%s':\t%10.10f\t+/-\t%10.10f\n",
      sel[0]->name, order, std_dev);
  if ( d.fp_p2_order ) {
    real p2_order = 0;
    real p2_deviation = 0;
    int j;
    for ( j = 0 ; j < d.frames_analyzed ; ++j )
      p2_order += d.P2_order_per_frame[j];
    p2_order /= d.frames_analyzed;
    for ( j = 0 ; j < d.frames_analyzed ; ++j )
      p2_deviation += sqr(d.P2_order_per_frame[j] - p2_order);
    p2_deviation /= d.frames_analyzed;
    fprintf(stdout, "Total P2 order for group '%s':\t\t%10.10f\t+/-\t%10.10f\n",
        sel[0]->name, p2_order, p2_deviation);
  }
  if ( d.fp_p4_order ) {
    real p4_order = 0;
    real p4_deviation = 0;
    int j;
    for ( j = 0 ; j < d.frames_analyzed ; ++j )
      p4_order += d.p4_order_per_frame[j];
    p4_order /= d.frames_analyzed;
    for ( j = 0 ; j < d.frames_analyzed ; ++j )
      p4_deviation += sqr(d.p4_order_per_frame[j] - p4_order);
    p4_deviation /= d.frames_analyzed;
```

```
      fprintf(stdout, "Total P4 order for group '%s':\t\t%10.10f\t+/-\t%10.10f\n",
          sel[0]->name, p4_order, p4_deviation);
    }
    if ( d.fp_lambda0_order ) {
      real lambda0_order = 0;
      real lambda0_deviation = 0;
      int j;
      for ( j = 0 ; j < d.frames_analyzed ; ++j )
        lambda0_order += d.lambda0_order_per_frame[j];
      lambda0_order /= d.frames_analyzed;
      for ( j = 0 ; j < d.frames_analyzed ; ++j )
        lambda0_deviation += sqr(d.lambda0_order_per_frame[j] - lambda0_order);
      lambda0_deviation /= d.frames_analyzed;
      fprintf(stdout, "Total lambda_0 order for group '%s':\t%10.10f\t+/-\t%10.10f\n",
          sel[0]->name, lambda0_order, lambda0_deviation);
    }
  }

  if ( d.fp_gL_pair_distribution ) {
    calculate_gL_pair_distribution(d.box_per_frame, d.frames_analyzed, d.no_of_axes[0],
        1, 0, d.orientation_per_frame, d.axis_of_inertia_per_frame, d.cogs_per_frame,
        d.volume_per_frame, d.samples, d.max_radius, d.g0_pair_distribution,
        d.g1_pair_distribution, d.g2_pair_distribution);
    int j;
    for ( j=0; j < d.samples; ++j) {
      fprintf(d.fp_gL_pair_distribution, "%f\t%f\t%f\t%f\n",
          1.0*j/d.samples*d.max_radius, d.g0_pair_distribution[j],
          d.g1_pair_distribution[j], d.g2_pair_distribution[j]);
    }
  }

  if ( d.fp_average_rotation ) {
    calculate_average_rotation(d.frames_analyzed, d.no_of_axes[0], 1, 0,
        d.axis_of_inertia_per_frame, d.average_rotation1, d.average_rotation2);
    int j;
    for ( j=0; j < d.frames_analyzed; ++j) {
      fprintf(d.fp_average_rotation, "%f\t%f\t%f\n",
          j*d.time_step*output_env_get_time_factor(*(d.oenv)),
          d.average_rotation1[j], d.average_rotation2[j]);
    }
  }

  if ( d.fp_order )
  {
    xvgr_legend(d.fp_order, 1, (const char**)legend_names, oenv);
    xvgr_world(d.fp_order, d.start_time*output_env_get_time_factor(*(d.oenv)), -0.5,
        d.end_time*output_env_get_time_factor(*(d.oenv)), 1.0, oenv);
  }
  if ( d.fp_p2_order )
  {
    xvgr_legend(d.fp_p2_order, 1, (const char**)legend_names, oenv);
    xvgr_world(d.fp_p2_order, d.start_time*output_env_get_time_factor(*(d.oenv)), -0.5,
        d.end_time*output_env_get_time_factor(*(d.oenv)), 1.0, oenv);
  }
  if ( d.fp_p4_order )
  {
    xvgr_legend(d.fp_p4_order, 1, (const char**)legend_names, oenv);
    xvgr_world(d.fp_p4_order, d.start_time*output_env_get_time_factor(*(d.oenv)), -0.5,
        d.end_time*output_env_get_time_factor(*(d.oenv)), 1.0, oenv);
  }
  if ( d.fp_lambda0_order )
  {
    xvgr_legend(d.fp_lambda0_order, 1, (const char**)legend_names, oenv);
    xvgr_world(d.fp_lambda0_order, d.start_time*output_env_get_time_factor(*(d.oenv)),
        -0.5, d.end_time*output_env_get_time_factor(*(d.oenv)), 1.0, oenv);
  }
  if ( d.fp_p2_histogram )
  {
    int i;
    for ( i=0 ; i < d.p2_histogram_number_bins ; ++i) {
      fprintf(d.fp_p2_histogram, "%.3f\t%d\n",
          ((real)i)/d.p2_histogram_number_bins*1.5-0.5, d.p2_histogram[i]);
  }
  xvgr_legend(d.fp_p2_histogram, 1, (const char**)legend_names, oenv);
  xvgr_world(d.fp_p2_histogram, -0.5, 0, 1.0, 10, oenv);
}
if ( d.fp_cos_histogram )
{
  int i;
  int total = 0;
  for ( i=0 ; i < d.cos_histogram_number_bins ; ++i)
    total += d.cos_histogram[i];
  for ( i=0 ; i < d.cos_histogram_number_bins ; ++i) {
    fprintf(d.fp_cos_histogram , "%.3f\t%d\t%f\n",
        ((real)i)/d.cos_histogram_number_bins*2-1.0, d.cos_histogram[i],
        1.*d.cos_histogram[i]/total);
  }
  xvgr_legend(d.fp_cos_histogram , 1, (const char**)legend_names, oenv);
  xvgr_world(d.fp_cos_histogram , -0.5, 0, 1.0, 10, oenv);
}

if ( d.fp_smectic_palermo ) {
  int frame;
  real average_smectic_order_parameter = 0;
  real average_layer_distance = 0;
  for ( frame = 0 ; frame < d.frames_analyzed ; ++frame ) {
    real index = 0;
    real index_up = 0;
    real index_down = 0;
    real max_value = -GMX_REAL_MAX;
    real max_value_up = -GMX_REAL_MAX;
    real max_value_down = -GMX_REAL_MAX;
    int sample;
    fprintf(d.fp_smectic_palermo ,
        "#Time\tlayer_spacing\tcosine\tsine\tsum\tcosine_up\tsine_up\tsum_up\t\
        cosine_down\tsine_down\tsum_down\n");
    for ( sample = 0; sample < d.samples; ++sample ) {
      real sum = sqrt( sqr(d.smectic_palermo_cosine[frame][sample])+
          sqr(d.smectic_palermo_sine[frame][sample]) );
      real sum_up = sqrt( sqr(d.smectic_palermo_cosine_up[frame][sample])+
          sqr(d.smectic_palermo_sine_up[frame][sample]) );
      real sum_down = sqrt( sqr(d.smectic_palermo_cosine_down[frame][sample])+
          sqr(d.smectic_palermo_sine_down[frame][sample]) );
      fprintf(d.fp_smectic_palermo , "%.8f\t%.8f\t", (d.start_time+frame*(d.end_time-
            d.start_time)/d.frames_analyzed)*output_env_get_time_factor(*(d.oenv)),
          (1.+sample)/d.samples*d.max_radius);
      fprintf(d.fp_smectic_palermo , "%.8f\t%.8f\t%.8f\t",
          d.smectic_palermo_cosine[frame][sample],
          d.smectic_palermo_sine[frame][sample], sum );
      fprintf(d.fp_smectic_palermo , "%.8f\t%.8f\t%.8f\t",
          d.smectic_palermo_cosine_up[frame][sample],
          d.smectic_palermo_sine_up[frame][sample], sum_up );
      fprintf(d.fp_smectic_palermo , "%.8f\t%.8f\t%.8f\n",
          d.smectic_palermo_cosine_down[frame][sample],
          d.smectic_palermo_sine_down[frame][sample], sum_down );
      if ( sum > max_value ) {
        index = sample;
        max_value = sum;
      }
      if ( sum_up > max_value_up ) {
        index_up = sample;
        max_value_up = sum_up;
      }
      if ( sum_down > max_value_down ) {
        index_down = sample;
        max_value_down = sum_down;
      }
    }
    average_layer_distance += (1.+index)/d.samples*d.max_radius;
    average_smectic_order_parameter += max_value;
    fprintf(d.fp_smectic_palermo , "\n#Max time\tlayer_spacing\torder\t\
        layer_spacing_up\torder_up\tlayer_spacing_down\torder_down\n");
    fprintf(d.fp_smectic_palermo , "%.8f\t",
        (d.start_time+frame*(d.end_time-d.start_time)/
           d.frames_analyzed)*output_env_get_time_factor(*(d.oenv)));
      fprintf(d.fp_smectic_palermo, "%.8f\t%.8f\t",
          (1.+index)/d.samples*d.max_radius, max_value);
      fprintf(d.fp_smectic_palermo, "%.8f\t%.8f\t",
          (1.+index_up)/d.samples*d.max_radius, max_value_up);
      fprintf(d.fp_smectic_palermo, "%.8f\t%.8f\n\n\n",
          (1.+index_down)/d.samples*d.max_radius, max_value_down);
    }
    average_layer_distance /= d.frames_analyzed;
    average_smectic_order_parameter /= d.frames_analyzed;
    fprintf(stdout, "Average smectic order parameter:\t%4.4f\nAverage layer distance:\t\
        \t\t%4.4f nm\n", average_smectic_order_parameter, average_layer_distance);
  }

  if ( d.fp_smectic )
  {
    real average_volume = 0;
    real average_radius = 0;
    real average_no_of_ups = 0;
    real average_no_of_downs = 0;
    real average_no_of_up_downs = 0;

    int i;
    for ( i=0; i<d.frames_analyzed; ++i ) {
      average_volume  += d.volume_per_frame[i];
      average_radius  += d.min_box_length[i];
      average_no_of_ups += d.number_of_ups[i];
      average_no_of_downs += d.number_of_downs[i];
      average_no_of_up_downs += d.number_of_up_downs[i];
    }
    average_volume /= d.frames_analyzed;
    average_radius /= d.frames_analyzed;
    average_no_of_ups /= d.frames_analyzed;
    average_no_of_downs /= d.frames_analyzed;
    average_no_of_up_downs /= d.frames_analyzed;
    double average_height = d.max_radius;
    if ( d.max_radius == -1 ) {
      average_height = average_volume / sqr(average_radius) / M_PI;
    } else {
      average_height =  d.max_radius*d.repl_per_dim*0.5;
      average_radius =  d.max_radius;
    }


    double inverse_couple_density = average_volume /
      (d.no_of_axes[0])/(d.number_of_replicas*d.no_of_axes[0]-1);
    double inverse_couple_density_up = average_volume /
      ((average_no_of_ups)*(d.number_of_replicas*average_no_of_ups -1));
    double inverse_couple_density_down = average_volume /
      ((average_no_of_downs)*(d.number_of_replicas*average_no_of_downs-1));
    double inverse_couple_density_up_down = average_volume /
      ((average_no_of_ups)*(d.number_of_replicas*average_no_of_downs));
    double cylinder_volume = M_PI * sqr(average_radius) * average_height;
    double norm_factor = inverse_couple_density * d.sm_bins /
      ( d.frames_analyzed * cylinder_volume );
    double norm_factor_up = 0.5 * inverse_couple_density_up * d.sm_bins /
      ( d.frames_analyzed * cylinder_volume );
    double norm_factor_down = 0.5 * inverse_couple_density_down * d.sm_bins /
      ( d.frames_analyzed * cylinder_volume );
    double norm_factor_up_down = 0.5 * inverse_couple_density_up_down * d.sm_bins /
      ( d.frames_analyzed * cylinder_volume );

    for ( i=0 ; i < d.sm_bins ; ++i) {
      double norm_factor_radial_distribution = 4./3.*M_PI*(
          pow((real)1.0*i/d.sm_bins*average_height+0.5*average_height/d.sm_bins, 3)-
          pow((real)1.0*i/d.sm_bins*average_height -0.5*average_height/d.sm_bins, 3));
      fprintf(d.fp_smectic, "%.3f\t%.9f\t%.9f\t%.9f\t%.9f\t%.9f\t%.9f\t%.9f\n",
          ((real)i/d.sm_bins*average_height),
          (d.sm_hist[i])*norm_factor,
          (d.sm_up_hist[i])*norm_factor_up,
          (d.sm_down_hist[i])*norm_factor_down,
          (d.sm_up_down_hist[i])*norm_factor_up_down,
             d.radial_distribution_hist[i]/norm_factor_radial_distribution*
             inverse_couple_density/d.frames_analyzed,
             (d.sm_hist_x[i])*norm_factor,
             (d.sm_hist_y[i])*norm_factor);
        }
        xvgr_world(d.fp_smectic, -1.0, 0, 1.0, 10, oenv);
      }

      // cleanup
      int frame = 0;
      for ( frame = 0; frame < nframes; ++frame )
      {
        sfree(d.axis_of_inertia_per_frame[frame]);
      }
      sfree(d.axis_of_inertia_per_frame);

      if (d.b_total)
      {
        sfree(legend_names[0]);
      }

      /* For the template, we close the output file if one was opened */
      if (d.fp_order)
        xvgrclose(d.fp_order);
      if (d.fp_p2_order)
        xvgrclose(d.fp_p2_order);
      if (d.fp_p4_order)
        xvgrclose(d.fp_p4_order);
      if (d.fp_lambda0_order)
        xvgrclose(d.fp_lambda0_order);
      if (d.fp_p2_histogram)
        xvgrclose(d.fp_p2_histogram);
      if (d.fp_cos_histogram)
        xvgrclose(d.fp_cos_histogram);
      if (d.fp_gL_pair_distribution)
        xvgrclose(d.fp_gL_pair_distribution);
      if (d.fp_average_rotation)
        xvgrclose(d.fp_average_rotation);
      if (d.fp_smectic_palermo)
        fclose(d.fp_smectic_palermo);

      thanx(stderr);
      return 0;
    }
```

B.3. `getPositionalAndOrientationalOrder.sh`

Shell script to automatically extract the orientational and positional order parameter with respect to the temperature from a trajectory containing more than one temperature sampling.

```
#!/bin/bash
export LANG=CC
set -e
set -u

if [ $# -lt 4 ] ; then
        echo -e "Error:\tToo few arguments!"
        echo -e "Usage:\t$0 file.xtc file.edr file.tpr timePerTemperatureInNS"
        exit 1
fi

XTC=$1
EDR=$2
TPR=$3
time_per_temp=$4
```

```
time_per_traj=`gmxcheck -e $EDR 2>&1 1>/dev/null |tr
 '\n'| \
              awk '/Last energy frame/ { print int($7)}'`
echo "Measured $time_per_traj ps in trajectory." >&2
temp_in_traj=$(($time_per_traj/$time_per_temp/1000+1))
echo "Calculated $temp_in_traj temperatures in trajectory." >&2

for radius in 7.00 ; do
  for i in `seq $temp_in_traj` ; do
    echo "name \".*\"" | g_order_tensor -quiet -f $XTC -s $TPR -b $(((i-1)*$time_per_temp))\
      -e $((i*$time_per_temp)) -orient-1 0 -orient-2 34 \
      -radius $radius -replcount 3 -smectic_palermo smectic-$i-$radius.xvg -dt 2 -tu ns \
      -xvg no -lamb0 tensor_p2_0-$i.xvg -o tensor_p2_1-$i.xvg
    rm -f \#*
  done
done

for i in `seq $temp_in_traj` ; do
  echo "Temp" | g_energy -quiet -xvg no -f $EDR -b $(((i-1)*$time_per_temp*1000)) \
    -e $((i*$time_per_temp*1000)) -o energy-$i.xvg
done

echo -e "#Temperature\tError\tspacing\tError\torder\tError\tError\tspacing_up\tError\t\
       order_up\tError\tError\tspacing_down\tError\torder_down\tError " > \
       positional-order-param.xvg
for radius in 7.00 ; do
  for i in `seq $temp_in_traj`  ; do
    smectic=smectic-$i-$radius.xvg
    smectic_max=smectic-max-$i-$radius.xvg
    awk '/#Max time/ {getline; print $1, $2, $3, $4, $5, $6, $7}' $smectic > $smectic_max
    energy=energy-$i.xvg
    echo -n "$radius "
    echo 'split_long_rows(false); a=load("'$smectic_max'"); b=load("'$energy'"); \
      [ mean(b(:,2)) std(b(:,2)) mean(a(:,2)) std(a(:,2)) mean(a(:,3)) \
      std(a(:,3)) mean(a(:,4)) std(a(:,4)) mean(a(:,5)) std(a(:,5)) mean(a(:,6)) \
      std(a(:,6)) mean(a(:,7)) std(a(:,7)) ]' |octave | tail -n2|head -n1 ;
  done
done >> positional-order-param.xvg

echo -e "#Temperature\tError\tspacing\tError\torder\tError\tError\tspacing_up\tError\t\
       order_up\tError\tError\tspacing_down\tError\torder_down\tError " > \
       positional-order-param-last.xvg
for radius in 7.00 ; do
  for i in `seq $temp_in_traj`  ; do
    smectic=smectic-$i-$radius.xvg
    smectic_max=smectic-max-$i-$radius.xvg
    awk '/#Max time/ {getline; print $1, $2, $3, $4, $5, $6, $7}' $smectic > $smectic_max
    energy=energy-$i.xvg
    echo -n "$radius "
    echo 'split_long_rows(false); a=load("'$smectic_max'"); b=load("'$energy'"); \
      [ mean(b(:,2)) std(b(:,2)) mean(a(end-100:end,2)) std(a(end-100:end,2)) \
      mean(a(end-100:end,3)) std(a(end-100:end,3)) mean(a(end-100:end,4)) \
      std(a(end-100:end,4)) mean(a(end-100:end,5)) std(a(end-100:end,5)) \
      mean(a(end-100:end,6)) std(a(end-100:end,6)) mean(a(end-100:end,7)) \
      std(a(end-100:end,7)) ]' |octave | tail -n2|head -n1 ;
  done
done >> positional-order-param-last.xvg

echo -e "#Temperature\tError\tlambda_+\tError\tlambda_0\tError" > \
       orientational-order-param.xvg
for i in `seq $temp_in_traj`  ; do
  tensor0=tensor_p2_0-$i.xvg
  tensor1=tensor_p2_1-$i.xvg
  energy=energy-$i.xvg
  echo 'a=load("'$tensor1'"); b=load("'$energy'"); c=load("'$tensor0'"); [mean(b(:,2)) \
    std(b(:,2)) mean(a(:,2)) std(a(:,2)) mean(c(:,2)) std(c(:,2))]' \
    |octave | tail -n2|head -n1 ;
done >> orientational-order-param.xvg
echo -e "#Temperature\tError\tlambda_+\tError\tlambda_0\tError" > \
       orientational-order-param-last.xvg
for i in `seq $temp_in_traj`  ; do
  tensor0=tensor_p2_0-$i.xvg
  tensor1=tensor_p2_1-$i.xvg
  energy=energy-$i.xvg
  echo 'a=load("'$tensor1'"); b=load("'$energy'"); c=load("'$tensor0'"); \
    [mean(b(end-100:end,2)) std(b(end-100:end,2)) mean(a(end-100:end,2)) \
    std(a(end-100:end,2)) mean(c(end-100:end,2)) std(c(end-100:end,2))]' \
    |octave | tail -n2|head -n1 ;
done >> orientational-order-param-last.xvg

echo 'set term x11 enhanced; plot "./positional-order-param.xvg" using 2:10:3:11 w xye \
       t "{/Symbol t}_{up}"; pause mouse key'| gnuplot
echo 'set term x11 enhanced; plot "./positional-order-param.xvg" using 2:8:3:9 w xye \
       t "d_{up}"; pause mouse key'| gnuplot
echo 'set term x11 enhanced; plot "./orientational-order-param.xvg" using 1:3:2:4 w xye \
       t "{/Symbol l}_+", "" using 1:5:2:6 w xye t "{/Symbol l}_0"; pause mouse key'| gnuplot
```

B.4. getHBondsIntra-Inter-everyTimeStep.sh

Shell script to automatically extract the intra- and intermolecular hydrogen bonds with respect to the temperature from a trajectory containing more than one temperature sampling.

```
#!/bin/bash
set -e
set -u

if [ $# -ne 4 ] ; then
        echo "Usage: $0 file.xtc file.edr file.tpr time_per_temp_in_ns"
        exit 1
fi

XTC=$1
EDR=$2
TPR=$3
time_per_temp=$4

time_per_traj=`gmxcheck -e $EDR 2>&1 1>/dev/null |tr
 '\n'| \
              awk '/Last energy frame/ { print int($7)}'`
echo "Measured $time_per_traj ps in trajectory." >&2
temp_in_traj=$(($time_per_traj/$time_per_temp/1000+1))
echo "Calculated $temp_in_traj temperatures in trajectory." >&2
time_step_ps=`gmxcheck -f $XTC 2>&1 1>/dev/null | tr
 '\n' | awk '/^Coords/ {print $3}'`
echo "Measured a storing time step for the coordinates of $time_step_ps ps." >&2
frames_per_ns=$((1000/time_step_ps))
echo "Calculated the number of stored frames per ns as $frames_per_ns." >&2

INDEX=hbond-relevant-parts.ndx
echo -e 'del 0\ndel 0\ndel 0\na o1 o2 o3 o4 o5 o6 ho1 ho2 ho3 ho4 ho5 ho6\n\
       a hoc3 oc1 oc2 oc3\nname 0 sugar\nname 1 ethoxy\nq' \
       | make_ndx -f $TPR -o $INDEX

relation_file=hbonds-intra-inter-relation.xvg
echo > $relation_file
for i in `seq $temp_in_traj` ; do
  hbindex=hbonds-$i.ndx
  start=$((($i-1)*$time_per_temp))
  end=$(($i*$time_per_temp))
  sugar_inter_file=sugar-inter-hbond-$i.xvg
  sugar_intra_file=sugar-intra-hbond-$i.xvg
  sugar_ethoxy_inter_file=sugar-ethoxy-inter-hbond-$i.xvg
  sugar_ethoxy_intra_file=sugar-ethoxy-intra-hbond-$i.xvg
  echo > $sugar_inter_file
  echo > $sugar_intra_file
  echo > $sugar_ethoxy_inter_file
  echo > $sugar_ethoxy_intra_file
  for j in `seq $((time_per_temp*frames_per_ns))` ; do
    frame=$(((j-1)*time_step_ps+start*1000))
    echo 0 0 | g_hbond -f $XTC -s $TPR -n $INDEX -xvg none -hbn $hbindex -tu ps \
      -b $frame -e $frame
    rm \#*
    sed -ni '/hbonds_/,$p' $hbindex
    awk '{printf "%d %d %d %s\n", ($1-$1%38)/38==($3-$3%38)/38, ($1-$1%38)/38, \
      ($3-$3%38)/38, $0}' $hbindex | awk '/^0/{print}' > temp_inter-hbond-$i
    awk '{printf "%d %d %d %s\n", ($1-$1%38)/38==($3-$3%38)/38, ($1-$1%38)/38, \
      ($3-$3%38)/38, $0}' $hbindex | awk '/^1/{print}' > temp_intra-hbond-$i
    intra=`wc -l < temp_intra-hbond-$i`
    inter=`wc -l < temp_inter-hbond-$i`
    echo $frame $intra >> $sugar_intra_file
    echo $frame $inter >> $sugar_inter_file
    echo 0 1 | g_hbond -f $XTC -s $TPR -n $INDEX -xvg none -hbn $hbindex -tu ps \
      -b $frame -e $frame
    rm \#*
    sed -ni '/hbonds_/,$p' $hbindex
    awk '{printf "%d %d %d %s\n", ($1-$1%38)/38==($3-$3%38)/38, ($1-$1%38)/38, \
      ($3-$3%38)/38, $0}' $hbindex | awk '/^0/{print}' > temp_inter-hbond-$i
    awk '{printf "%d %d %d %s\n", ($1-$1%38)/38==($3-$3%38)/38, ($1-$1%38)/38, \
      ($3-$3%38)/38, $0}' $hbindex | awk '/^1/{print}' > temp_intra-hbond-$i
    intra=`wc -l < temp_intra-hbond-$i`
    inter=`wc -l < temp_inter-hbond-$i`
    echo $frame $intra >> $sugar_ethoxy_intra_file
    echo $frame $inter >> $sugar_ethoxy_inter_file
  done
  temp=temperature-$i.xvg
  echo 'Temp' | g_energy -f $EDR -xvg none -b $(($start*1000)) -e $(($end*1000)) -o $temp
  temperature=`echo 'a=load("'$temp'"); [mean(a(:,2)) std(a(:,2)) ]' \
             |octave | tail -n2|head -n1 `;
  intra_results=`echo 'a=load("'$sugar_intra_file'"); [mean(a(:,2)) std(a(:,2)) ]' \
                |octave | tail -n2|head -n1 `;
  inter_results=`echo 'a=load("'$sugar_inter_file'"); [mean(a(:,2)) std(a(:,2)) ]' \
                |octave | tail -n2|head -n1 `;
  echo -n $temperature $intra_results $inter_results >> $relation_file
  intra_results=`echo 'a=load("'$sugar_ethoxy_intra_file'"); [mean(a(:,2)) std(a(:,2)) ]' \
                |octave | tail -n2|head -n1 `;
  inter_results=`echo 'a=load("'$sugar_ethoxy_inter_file'"); [mean(a(:,2)) std(a(:,2)) ]' \
                |octave | tail -n2|head -n1 `;
  echo " " $intra_results $inter_results >> $relation_file
done
```

Parts of this work have been published in two posters:

S. Breuers, D. Blunk "Liquid Crystals in the Grid", *38th Topical Meeting on Liquid Crystals*, **March 10-12, 2010**, Mainz

S. Breuers, D. Blunk "Predicting Molecular Self Organisation in Sugar Based Liquid Crystals Using Grid Computing Facilities", *26th Molecular Modelling Workshop 2012*, **March 12-14, 2012**, Erlangen

Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie – abgesehen von unten angegebenen Teilpublikationen – noch nicht veröffentlicht worden ist, sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen der Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von

Priv.-Doz. Dirk Blunk

betreut worden.

Sebastian Breuers

Lebenslauf

Persönliche Daten

Name:	Sebastian Breuers
Geburt	14.04.1982, Erkelenz

Berufspraxis

seit 12/2009	**Wissenschaftlicher Mitarbeiter** Promotionsstudium, AG Priv.-Doz. Dr. D. Blunk, AG Prof. Dr. H.-G. Schmalz, Institut für Organische Chemie, Universität zu Köln Projektmitarbeiter bei „Molecular Simulation Grid“
09/2011–11/2011	**Projektaufenthalt** bei Prof. Claudio Zannoni, Dipartimento di Chimica Fisica ed Inorganica, Università di Bologna
11/2007–11/2009	**Wissenschaftlicher Mitarbeiter** Regionales Rechenzentrum der Universität zu Köln (RRZK) Projektmitarbeiter bei „Sustainable Grid Infrastructure“
2005–2007	**Studentisches Hilfskraft** RRZK
2004	**Studentisches Hilfskraft** Universitäts- und Stadtbibliothek Köln
2003/2004	**Studentisches Hilfskraft** AG Prof. Dr. M. Melkonian, Institut für Botanik der Universität

Ausbildung

10/2002–10/2007	**Studium der Biologie** Universität zu Köln, Akademischer Grad: Diplom Biologe „Drei-dimensionale Visualisierung genbasierter Metadaten mit Hilfe von Methoden der Formalen Begriffsanalyse.“
08/2001–05/2002	**Zivildienst** Hermann-Josef-Stiftung Ambulantes Pflegezentrum, Ambulanter Pflegedienst, Erkelenz
1992–2001	Cusanus Gymnasium, Erkelenz, Abschluss: Abitur
1991–1992	Gemeinschaftsgrundschule, Gerderath
1988–1991	Luise-Hensel-Schule, Erkelenz

Köln, 28. Mai 2013