



universität
wien

DISSERTATION

Titel der Dissertation

Structural Decomposition and Structural Relaxation of Solvation Shells of Hydrated Molecular Ionic Liquids and Protein Solutions

angestrebter akademischer Grad
Doktor der Naturwissenschaften (Dr. rer. nat.)

Verfasser:	Mag. Gregor Neumayr
Matrikelnummer:	9804013
Dissertationsgebiet (lt. Studienblatt) :	441 Genetik - Mikrobiologie (Stzw)
Betreuer:	Univ.-Prof. Dr. Othmar Steinhauser

Wien, im Mai 2010

Contents

1	General Introduction	11
1.1	Methods	12
1.1.1	Decomposition of Space	12
1.1.2	Atomistic Simulation	13
1.2	Systems	15
1.2.1	Molecular Ionic Liquids	15
1.2.2	Proteins	17
1.3	Application to systems	17
1.4	Author’s declaration	18
2	On the collective network of ionic liquid/water mixtures. III. Structural analysis of ionic liquids on the basis of Voronoi decomposition	23
2.1	Introduction	23
2.2	Theory and Methods	25
2.2.1	Voronoi decomposition	25
2.2.2	Combined g -coefficient/Voronoi analysis	29
2.2.3	Computational setup	34
2.3	Results and Discussion	35
2.3.1	The hydrophobic cationic reference site	35
2.3.2	The hydrophilic cationic reference site	38
2.3.3	The anion–water network	44
2.3.4	Coordination numbers	46
2.4	Conclusion	50
3	Relaxation of Voronoi shells in hydrated molecular ionic liquids	57

Contents

3.1	Introduction	57
3.2	Theory	59
3.2.1	Molecular Dynamics of Voronoi Shells	59
3.2.2	Probabilistic Approach	63
3.3	Methods	67
3.3.1	Implementation of Tessellation	67
3.3.2	Details of Simulation	68
3.3.3	Construction of Transition Matrix \mathbf{W}	70
3.4	Results and Discussion	71
3.4.1	Motional Parameters of Voronoi Dynamics	71
3.4.2	Probabilistic Approach	76
3.5	Conclusion	84
4	Global and Local Voronoi Analysis of Solvation Shells of Proteins	91
4.1	Introduction	91
4.2	Theory	93
4.2.1	Tessellation and Voronoi Shells	93
4.2.2	Radially Resolved and Shell Grained Spatial Distribution Functions	95
4.2.3	Time Series and Time Correlation Functions	97
4.3	Methods	98
4.3.1	Simulation and System Description	98
4.3.2	Implementation of Delaunay Tessellation	99
4.3.3	Data Analysis and Organisation	99
4.4	Results and Discussion	101
4.4.1	Global Protein Analysis	103
4.4.2	Local Residue Analysis	113
4.5	Conclusion	122
5	Summary	129
6	Outlook	133
7	GEPETTO - Implementation	135

7.1	General Program Flow	137
7.2	Instruction Space (IS)	139
7.2.1	Flow Control	139
7.2.2	Program Input	139
7.2.3	Selection	142
7.2.4	Optimization	147
7.2.5	Program Output	154
7.3	Calculation Space (CS)	156
7.3.1	Periodic Boundaries	158
7.3.2	Single Particle Observables - $O(N)$	159
7.3.3	Neighbourhood Calculation - $O(N^2)$	160
7.3.4	Pair Properties $O(N^2)$	164
7.3.5	Collective Properties $O(1)$	170
8	GEPETTO - User Guide	173
8.1	Single Particle Properties	174
8.1.1	Example I: Mean Square Displacement	174
8.1.2	Example II: Atom Coordinates and Unfolding	175
8.1.3	Example III: Dipole Autocorrelation	175
8.1.4	Example IV: Velocity Autocorrelation Functions	176
8.1.5	Example V: Angular Velocity	176
8.2	Pair Correlation Functions	177
8.2.1	Example VI: Simple g-Functions	177
8.2.2	Example VII: Voronoi Decomposition of g-Functions	178
8.3	Nuclear Density Maps	178
8.3.1	Example VIII: Nuclear Density Map	179
8.3.2	Example IX: Multiple Nuclear Density Maps	180
8.3.3	Example X: Nuclear Density Map and Ensemble Average	181
8.4	Coordination and Neighbour Interaction	182
8.4.1	Example XI: Coordination Number	183
8.4.2	Example XII: Species Specific Coordination Number	183
8.4.3	Example XIII: Residue Resolved Coordination Number	184

Contents

8.4.4	Example XIV: Mean Residence Time	185
8.4.5	Example XV: Contact Matrix	185
8.4.6	Example XVI: Markovian Transition Matrix	186
8.4.7	Example XVII: Promiscuity	187
8.5	Voronoi Surfaces and Volumes	187
8.5.1	Example XVIII: Voronoi Volume Time Series of Proteins	187
8.5.2	Example XIX: Voronoi Volume Distribution of Solvation Shell	188
8.5.3	Example XX: Residue Specific VISA	188
8.6	Collective Properties	189
8.6.1	Example XXI: MD, MJ, J Time Series	189
8.6.2	Example XXII: $\langle \text{MD}(0)\text{MD}(t) \rangle$ Autocorrelation LTC	190
8.6.3	Example XXIII: $\langle \text{MD}(0)\text{J}(t) \rangle$ Crosscorrelation	190
8.6.4	Example XXIV: MD Cage Correlation	191
8.6.5	Example XXV: Molecular Dipoles Projected to the Total Dipole Moment . . .	192

Danksagung

Eine Arbeitsatmosphäre, wie ich sie am noch jungen *Institut für Computergestützte Biologische Chemie* erlebt habe, ist nicht selbstverständlich und den Bemühungen jedes einzelnen Mitgliedes zu verdanken.

Als erstes erwähnen möchte ich meinen Betreuer Othmar Steinhauser der in einer persönlichen, offenen und vorausschauenden Art das Institut leitet und meine Arbeit betreute. Neben seinen zahlreichen politischen und betrieblichen Verpflichtungen fand er fast immer die Zeit für *Privatissima*, bei denen er mit viel Geduld die kompliziertesten Zusammenhänge aufbereitete und verständlich machte. Diese Arbeit profitiert in großem Ausmaß von seinen persönlichen Bemühungen die sich durch den gesamten Entstehungsprozess hindurchziehen. Christian Schröder sei gedankt für die kompetente Unterstützung und die produktive Zusammenarbeit aber auch für konstruktive Kritik. Ich möchte Thomas Taylor und Michael Haberler herzlich für das Teilen und Entwickeln einer Vision und deren gemeinsame Umsetzung danken. Wertvolle Hinweise und Tipps erhalten und in zahlreichen, in den meisten Fällen fruchtbaren, Diskussionen gelernt habe ich von Stefan Bruckner, Stefan Boresch, Gerhard König und Sonja Maurer.

Ganz besonderer Dank gebührt meiner Ehefrau Angelika Schöneegger, die mich in dieser nicht immer einfachen Zeit meines Lebens unterstützt und begleitet hat.

Zusammenfassung

Die vorliegende Arbeit liefert neue methodische Beiträge zur Untersuchung der Struktur und Dynamik von Biomolekülen in Lösung mittels Voronoi-Analyse von Computersimulationen. Dabei werden sowohl kollektive wie auch Einteilchen-Eigenschaften der Solvathüllen und des Bulk-Mediums betrachtet.

Als Modellproteine dienen Ubiquitin (PDB-code: 1UBQ), Calbindin (1CLB) und eine Phospholipase (2PLD) deren Solvation in Wasser einen wesentlichen Bestandteil dieser Arbeit darstellt. Darüber hinaus werden Vorstudien zu Molekularen Ionischen Flüssigkeiten (MIL) angestellt die in den letzten Jahren unter anderem als umweltverträgliche polare Lösungsmittel in den Vordergrund getreten sind. Trifluoroazetat-, Tetrafluoroborat- und Trifluoromethylsulfonat- Salze von alkyliertem Imidazolium werden einerseits in Reinform, andererseits in Mischung mit Wasser untersucht.

Neu an dieser Arbeit ist zunächst die Atom-aufgelöste Tesselierung, die für Systeme mit 30000 Atomen mit periodischen Randbedingungen über hundertausende Zeitschritte sehr rechenintensiv, und daher nur durch die effiziente Implementierung geeigneter Algorithmen zu bewerkstelligen ist. Auf dieser Grundlage werden weitestgehend parameterfreie Ansätze zur lokalen und globalen Strukturanalyse entwickelt die einerseits mit konventionellen Methoden wie etwa Radialen Verteilungsfunktionen und Orientierungskorrelationsfunktionen verglichen werden, andererseits zusätzliche Möglichkeiten der Interpretation bieten. Position und Orientierung von benachbarten Molekülen kann direkt anhand von graphentheoretischen Interaktionen beschrieben und interpretiert werden. Ein Markov-Modell für die Dynamik innerhalb und zwischen einzelnen Solvathüllen wird entwickelt und auf MIL Systeme angewendet.

Abstract

The present work provides new methodical contributions to investigation of structural and dynamic behaviour of solvated biomolecules using Voronoi analysis of computer simulations. Thereby, collective as well as single particle properties of solvation shells and the bulk medium are considered. The three proteins ubiquitin (PDB-code: 1UBQ), calbindin (1CLB) and phospholipase (2PLD) serve as model systems. The study of their solvation in water is an integral part of this work. Moreover, preliminary studies of Molecular Ionic Liquids (MIL) are being made, that have come to the fore in recent years as environmentally compliant polar solvents. Alkylated imidazolium salts of Trifluoroacetate, Tetrafluoroborate and Trifluoromethylsulfonate are analysed in the pure form as well as mixed with water. For one thing, new in this work is the atom-resolved tessellation, that is computationally demanding for systems with about 30000 atoms and periodic boundary conditions over 100-thousands of time steps and hence is to be managed only by the efficient implementation of suitable algorithms. Widely parameter free approaches to local and global structure analysis are developed on this basis and compared to conventional methods like radial distribution functions and orientation correlation functions. Furthermore, they provide additional possibilities for interpretation. Position and orientation of neighbouring molecules can be described and interpreted directly by graph theoretical interactions. A Markov model for dynamics within and between solvation shells is being developed and applied to MIL systems.

1 General Introduction

In living cells, large biomolecules like proteins or DNA are surrounded by cytosol. This intracellular fluid consists of a complex mixture of chemical compounds dissolved in water. The surface of a protein can be regarded as a dynamic, heterogeneously charged interface that has an impact on the positioning, orientation and dynamics of polar solvent molecules. Actually, charged dipolar solutes structure their environment in a characteristic way. Thus, the solvent contains and transfers structural information about the solute, necessary for successful interaction with approaching ligands. As the solute's impact on the solvent decreases with increasing distance, a proper partitioning of space into solvation layers is of crucial importance for the understanding of biomolecular behaviour on a physical or chemical level.

The central thesis of this work is that parameter-free (and thus system-independent) methods can be developed and optimised to uniquely describe structural packing, orientation and relaxation in solutions of charged and anisotropic molecules of medium and large size. Methods based on g-functions traditionally used to describe structural aspects and relaxation of small solvents are dependent on the selection of multiple ambiguous distance based parameters. In contrast, Voronoi based analyses being developed and optimised in this work enable a unique description that is free of parameters. Hydrated molecular ionic liquids serve as model systems of medium size solutes while three hydrated proteins have been chosen as representatives for the group of large biomolecules. As no experimentally inferred structures of hydrated proteins or ionic liquids exist, this work relies on detailed atomistic molecular dynamic simulations. The quantities calculated from these simulations are being compared to experimental data.

1.1 Methods

1.1.1 Decomposition of Space

In a mathematic context, the definition of solvation layers or shells is a geometrical problem of spatial decomposition. This decomposition is traditionally done on the basis of radial distances. The necessary parameters for this radial shell definition are usually obtained from radial distribution functions $g(r)$ (RDF). In statistical mechanics, the RDF $g(r) = g^{000}(r)$ and its higher momenta, the orientation correlation functions (OCF) (e.g. the dielectric screening function $g^{110}(r)$) are in widespread use to describe the static average structure of solvation shells and are therefore also being used in this work. They describe packing (RDF) and orientation (OCF) of solvent molecules or search points with respect to a solute molecule or origin point. Furthermore, they are of fundamental importance in thermodynamics because macroscopic thermodynamic quantities like pressure, energy or compressibility can be expressed in terms of the RDF. However, as both biomolecules and MILs can be quite anisotropic, a radial description of solvation is either a rather crude approximation on a global molecule scale, or needs to be done using multiple empirical parameters on a local, residue or atom scale.

Voronoi graphs represent a parameter-free and unique decomposition of space. They have been introduced by Voronoi^[1] as early as 1908 and applied in many scientific fields since then. One seemingly moot - but often cited - example is the problem of finding the nearest post office in a set of post offices. On a two dimensional map, each post office is the reference point to construct a Voronoi polygon around it, representing an imaginary catchment area. This area contains all points closer to the reference point than to any other post office. Every linear segment of the border of each polygonal area bisects the line connecting two neighbouring post offices and is orthogonal to it. Therefore, the Voronoi polygons have an irregular honeycomb structure. As Voronoi tessellation is space-filling, each point on the map lies within a Voronoi polygon belonging to one single post office - the nearest to that point. The connection between post offices and biomolecular solvation might not be immediately evident. Anyway, the example shows the broad applicability of Voronoi diagrams. On an abstract level, the problem of finding the nearest post office reduces to the graph theoretical problem of finding the nearest neighbour in a set of points or vertices. The membership in the first solvation layer corresponds to that same graph theoretical problem. Once the Voronoi graph has been constructed, the nearest neighbours can be identified immediately because their polyhedra share a common face in the Voronoi diagram. Voronoi tessellation provides for an elegant method to classify neighbour relations for highly anisotropic large molecules but it is computationally intensive. Thus, the decisive step to conquer this problem is

to find and implement a suitable algorithm. Fortunately, the dual of a Voronoi graph can be calculated in a more efficient way. This dual graph was presented first by Delaunay,^[2] a student of Voronoi. Dual graphs can be converted into each other without any further information. Both, the Delaunay and the Voronoi diagram of a 1-butyl-3-methyl-imidazolium cation are shown in Fig.1.1. In this work, the basic data for the Voronoi analysis come from atomistic molecular dynamics computer simulations (see Sec.1.1.2). These simulations are conducted using periodic boundary conditions. Therefore, the Delaunay tessellation algorithm has to fulfill two constraints: It must be able to construct three dimensional polyhedra and incorporate periodic boundary conditions. An appropriate algorithm has been suggested by Thompson^[3] in 2002. It has been adapted and optimised by the author of this thesis and implemented in the proprietary trajectory analysis tool GEPETTO which is described in chapter 7.

A second mathematical problem comes into play when asking for the members of a more distant solvation shell. It is the graph theoretical shortest path problem: The length of the shortest path between two vertices in a graph is the minimum number of edges that connect the two vertices. Applying a Delaunay based shell definition, the members of the shell with index d are those molecules with a shortest path of length d in the corresponding Delaunay graph.

1.1.2 Atomistic Simulation

The underlying data for this study originate from atomistic molecular dynamics simulations. Molecular dynamics computer simulation is a method to describe motion of a chemical system in terms of a set of atomic positions and velocities as discrete-time table functions.

The most accurate description of motion at the atomic or subatomic level can only be achieved within quantum mechanics. However, despite the huge progress in computer performance having been made over the past decades, a quantum mechanical simulation of biomolecular solvation processes is still out of reach. Therefore, reasonable simplifications have to be made for our purposes: The Ehrenfest equation^[4]

$$\frac{d\langle\vec{p}_i\rangle}{dt} = \langle\vec{F}_i\rangle \quad (1.1)$$

represents a link between quantum mechanics and classical mechanics. In this formulation, $\langle\vec{p}_i\rangle$ is the expected value of the impulse of particle i and $\langle\vec{F}_i\rangle$ is the expected value of the effective force acting on particle i . The Ehrenfest theorem proves that, under certain conditions, the classical equations of motion are valid for the expected values of quantum mechanics. In quantum mechanics, positions

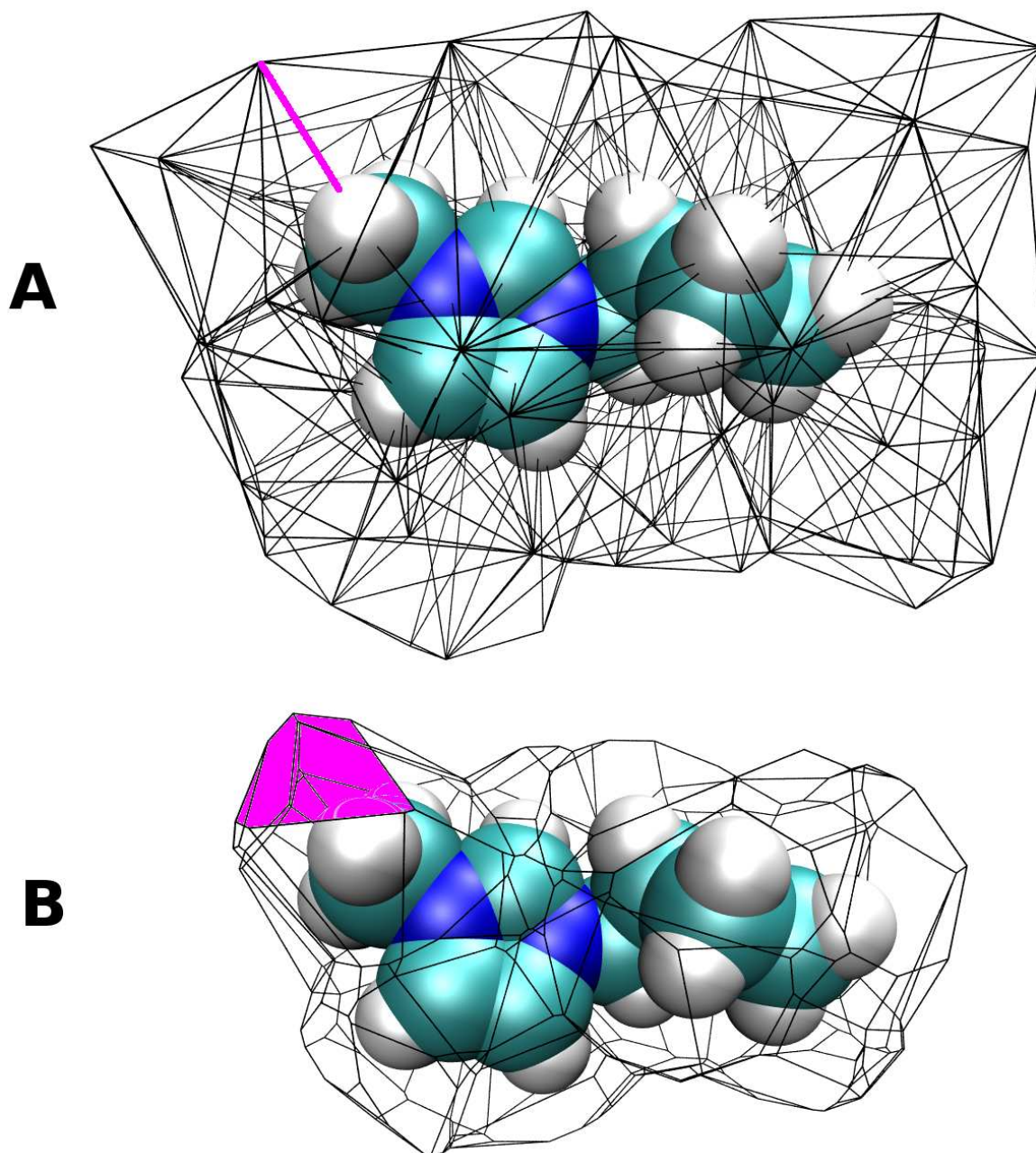


Figure 1.1: Delaunay (A) and Voronoi (B) diagram of a 1-butyl-3-methyl-imidazolium cation in aqueous solution. Each edge in the Delaunay diagram (A) connects two neighbouring atoms. The edge drawn in magenta connects one methyl hydrogen with a water oxygen atom. For the sake of clarity surrounding water molecules are not shown. Based on the Delaunay graph, the Voronoi diagram (B) is constructed by erecting a plane orthogonal to and bisecting each Delaunay edge. This is indicated by the magenta-coloured face. It is the corresponding face to the edge in part A. Both representations can be used to describe a variety of features. Thereby, Delaunay diagrams lend themselves to the classification of neighbourhood, shell membership and contact type. Voronoi diagrams can be used to calculate geometric features like molecular volume or interface surface area.

of atomic nuclei, containing almost the total mass of the atom, show distributions having relatively small deviations as compared to the light-weight electrons. This means, nuclear positions can well be described by instantaneous positions. Thereby, the representation of the location of atom i is reduced to only one point \vec{r}_i in three dimensional space. This simplification makes it possible to apply the equation of motion of classical point mechanics which reads

$$\frac{d^2 \vec{r}_i(t)}{dt^2} = \frac{\vec{F}_i(t)}{m_i}. \quad (1.2)$$

when expressed in cartesian coordinates \vec{r}_i . It describes the acceleration as the quotient of the effective force \vec{F}_i acting on atom i and the atomic mass m_i . The CHARMM force field has been used in all simulations.^[5]

Replacing the second differential quotient by the finite central difference of second order, the Verlet algorithm^[6]

$$\vec{r}_i(t + \Delta t) = 2\vec{r}_i(t) - \vec{r}_i(t - \Delta t) + \frac{\Delta t^2}{m} \vec{F}_i(t) \quad (1.3)$$

can be derived. The time step Δt has to be chosen in accordance with atomic masses. This algorithm can be used to solve the integration problem and yields a three dimensional “molecular movie” called trajectory. Figure 1.2 illustrates the functional principle of the Verlet algorithm. The output is a set of atomic coordinates for each time step, called frame. Molecular dynamics simulations can be used to simulate significantly bigger systems than quantum mechanics. However, the computer resources still limit the systems to macroscopically irrelevant sizes of 10^4 to 10^5 atoms. Simulating 30000 atoms arranged in a cube (the primary cell) leads to significant edge or size effects. Such a model does not reflect the properties of a macroscopic chemical system. To overcome this, periodic boundary conditions are usually included in molecular dynamics simulations. Thereby, the primary cell is copied periodically in all three spatial directions, introducing periodic images for each atom. This way, a particle can leave the primary cell and reenter it at the opposite side with the same velocity.

1.2 Systems

1.2.1 Molecular Ionic Liquids

Molecular Ionic Liquids (MIL), due to certain properties like a low vapor pressure, high thermal and electrochemical stability, are promising charged, polar solvents and a valuable contribution to green

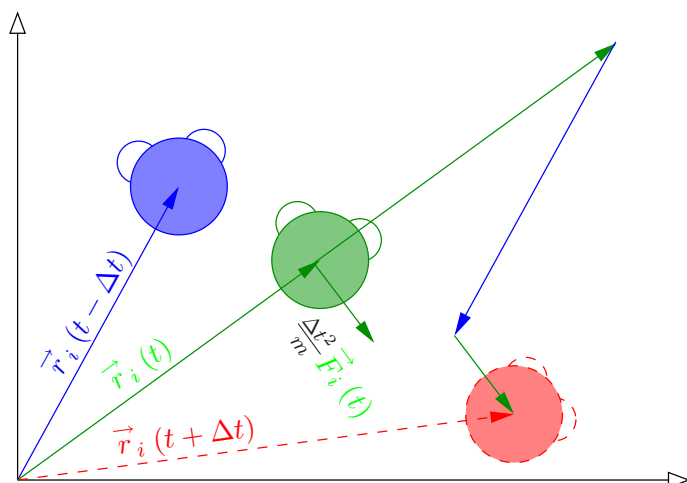


Figure 1.2: Verlet algorithm. This image shows the trajectory of one schematic water molecule in two dimensions. The position vector \vec{r} of the oxygen atom i is shown for three time steps. The Verlet algorithm uses the current frame at time t and the previous one ($t - \Delta t$) in order to calculate the next frame at time $t + \Delta t$. In addition, the current effective force $\vec{F}_i(t)$ needs to be calculated which is by far the most expensive step.

chemistry. A wide range of applications in fields as different as biotechnology, nuclear industry, solar energy, food and bioproducts or waste recycling have come up in the recent years.^[7]

One special area of application is the usage as solvents for enzymes. Lipase and lyase activity has been detected in pure MILs and shown to be 50% higher compared to activity measured in organic solvents. MIL water mixtures provide an opportunity to expand the application range. Among others, peroxidase activity could be detected in such mixtures. These properties turn MILs into interesting alternatives for organic solvents, not least in biotechnology or pharmaceutical industry. Several review articles exist on this topic^{[8]–[14]}

The cations used in this study are based on an imidazole ring substituted by a methyl residue and either a butyl (1-butyl-3-methyl-imidazolium, BMIM) or an ethyl chain (1-ethyl-3-methyl-imidazolium, EMIM). The anions used were tetrafluoroborate (TFB), lacking a permanent electric dipole, and trifluoromethylsulfonate (TRIF), having a pronounced dipole moment along the C-S bond. The choice of BMIM, EMIM, TFB and TRIF has been made due to their conventional usage as MIL model systems. The parameters used for simulation were taken from the references [15]–[19]. All systems were mixed with water at different concentrations.

1.2.2 Proteins

Three systems of hydrated proteins have been simulated, varying in their secondary structure and the total charge:

Ubiquitin is highly conserved and a typical model protein. It can be found ubiquitously in all eukaryotic cells and an analogue might even exist in certain prokaryotic cells. It is involved in regulation of other proteins' function, half-life and localization within the cell. Thereby, post-translational modification of proteins is effected by different ways of ubiquitin binding. The structural features include an α -helical region and a β -sheet and a zero total charge. The coordinates of the start configuration for simulation were taken from the PDB^{[20],[21]} (PDB-code: 1UBQ).

The bovine calcium binding protein apo-calbindin D_{9K} ^[22] (1CLB) with a total charge of -7e was used as a second model protein. It is present in mammalian intestinal epithelial cells and mediates the transport of calcium across these cells. Thereby, the amount of calcium crossing the cell is increased without raising the free concentration. Its secondary structure is dominated by α -helices.

The third model system contains the C-terminal SH2 domain of phospholipase $C-\gamma 1$ ^{[23],[24]} (2PLD) solvated in water. It plays a role in intracellular transduction of tyrosine kinase activators. 2PLD forms two α -helices and two β -sheets and has a total charge of +3e.

1.3 Application to systems

Some of our work has already been published in the *Journal of Chemical Physics*. Three articles, two of which are already published, are included in this compilation in chronological order, thus, documenting the scientific progress:

As a first transitional step towards a Voronoi description of solvation, the traditional g-functions are decomposed into Voronoi shells. This can be seen as an extension of radial distribution functions and orientation correlation functions in the light of Voronoi diagrams. This first Voronoi study was based on molecular dynamics simulations of three different mixtures of 1-butyl-3-methyl-imidazolium tetrafluoroborate with water and can be found in Chapter 2. It focuses mainly on structural properties.

The second step from structural analysis to the study of dynamic aspects was conducted for eight hydrated MIL systems including the ionic liquids 1-butyl-3-methyl-imidazolium tetrafluoroborate and 1-ethyl-3-methyl-imidazolium trifluoromethylsulfonate using a Voronoi based residence function $n(t)$. Autocorrelation functions of $n(t)$ were fitted to Kohlrausch-Williams-Watts functions (KWW). The parameters from these KWW fits were used to obtain mean residence times. Furthermore, we under-

took the effort to develop a Markovian master equation to describe the dynamic processes occurring in solvation shells and results. This probabilistic approach was compared to traditional autocorrelation functions. These dynamic aspects of solvation are described in the article in Chapter 3.

The third step is the transition between medium size ionic liquid molecules and large biomolecules, thereby, considering both, structural and dynamic aspects. The three proteins ubiquitin, calbindin and phospholipase, solvated in water, were used as model systems. We developed the new concept of “shell graining” considering “one shell as a bin”. Again, results are being compared to experimental results. Details are given in Chapter 4.

The actual calculation of Voronoi analysis, traditional g-functions, residence functions and correlation functions as well as all quantities resulting from these analyses was performed completely using self written code. This way, the software package GEPETTO came into existence. An overview over the implementation and a User Guide are given in the two final Chapters 7 and 8.

1.4 Author’s declaration

The *Institute of Computational Biological Chemistry* provides a great breeding ground for ideas and finding solutions for problems that arise. This work, like any work of comparable complexity, profits from united efforts and team work to some extent. In order to meet formal conditions inflicted by the official doctoral examination procedure of the University of Vienna, however, the following can be said:

Chapter 2:

C. Schröder, G. Neumayr and O. Steinhauser J. Chem. Phys. 130, 194503 (2009)

The basic idea of combined g-coefficient/Voronoi analysis stems from GN. The study was designed by GN and OS. GN accounts for the implementation and optimisation of algorithms, most figures, all analysis work, all data of immediate importance to the study. CS provided the underlying MD simulation.

Chapter 3:

G. Neumayr, C. Schröder and O. Steinhauser J. Chem. Phys. 131, 174509 (2009)

GN had the idea for the probabilistic approach. GN and OS designed the study and developed the theoretical framework. Implementation, adaptation and optimization of the basic algorithm was done by GN. All analyses were performed by GN. Data and figures were produced by GN. CS provided the MD simulation.

Chapter 4:

G. Neumayr, T. Rudas and O. Steinhauser submitted to J. Chem. Phys. (April 2010)

The idea of “shell graining” stems from OS. The study was designed by OS and GN. The theoretical framework was developed by OS and GN. GN accounts for all figures, data and analysis as well as implementation and optimization of algorithms. TR provided the MD simulation.

Bibliography

- [1] G. F. Voronoi, *J. Reine Angew. Math.* **134**, 198 (1908).
- [2] B. N. Delaunay, *Bulletin of Academy of Sciences of the USSR* **7** **6**, 793 (1934).
- [3] K. E. Thompson, *Int. J. Numer. Meth. Engng.* **55**, 1345 (2002).
- [4] T. Fliessbach, *Quantenmechanik* (1995).
- [5] B. R. Brooks, C. L. Brooks III, A. D. Mackerell Jr., L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caffisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus, *J. Comput. Chem.* **30**, 1545 (2009).
- [6] L. Verlet *Phys. Rev.* **159**, 98 (1967).
- [7] N. V. Plechova, and K. R. Seddon, *Chem. Soc. Rev.* **37**, 123 (2008).
- [8] P. Wasserscheid, and T. Welton, *VCH-Wiley* (2003).
- [9] T. Welton, *Chem. Rev.* **99**, 2071-2083 (1999).
- [10] F. Endres, and S. Z. El Abedin, *Phys. Chem. Chem. Phys.* **8**, 2101-2116 (2006).
- [11] J. Dupont, and P. A. Z. Suarez, *Phys. Chem. Chem. Phys.* **8**, 2441 (2006).
- [12] P. Wasserscheid, and W. Keim, *Angew. Chem.* **39**, 3772 (2000).
- [13] C. Chiappe, and D. Pieraccini, *J. Phys. Org. Chem.* **18**, 257 (2005).
- [14] Stewart A. Forsyth, Jennifer M. Pringle, and Douglas R. MacFarlane, *Aust. J. Chem.* **57**, 113 (2004).

Bibliography

- [15] J. N. Canongia Lopes, J. Deschamps, and A. A. H. Padua, *J. Phys. Chem. B* **108**, 2038 (2004).
- [16] J. N. Canongia Lopes, J. Deschamps, and A. A. H. Padua, *J. Phys. Chem. B* **108**, 11250 (2004).
- [17] J. N. Canongia Lopes and A. A. H. Padua, *J. Phys. Chem. B* **108**, 16893 (2004).
- [18] J. de Andrade, E. S. Böes, and H. Stassen, *J. Phys. Chem. B* **106**, 13344 (2002).
- [19] W. L. Jorgensen, *J. Am. Chem. Soc.* **103**, 335 (1981).
- [20] S. Vijay-Kumar, C. E. Bugg, and W. J. Cook, **194**, 531 (1987).
- [21] B. P. Monia, D. J. Ecker, and S. T. Crooke, *Biotechnology* **8**, 209 (1990).
- [22] N. J. Skelton, J. Kördel, and W. J. Chazin, **249**, 441 (1995).
- [23] Q. Ji, A. Chattopadhyay, M. Vecchi, and G. Carpenter, *Mol. and Cell. Biol.* **19**, 4961 (1999).
- [24] S. M. Pascal, A. U. Singer, G. Gish, T. Yamazaki, S. E. Shoelson, T. Pawson, L. E. Kay, and J. D. Forman-Kay, *Cell* **77**, 461 (1994).

2 On the collective network of ionic liquid/water mixtures. III. Structural analysis of ionic liquids on the basis of Voronoi decomposition

*C. Schröder, G. Neumayr and O. Steinhauser J. Chem. Phys. **130**, 194503 (2009)*

Three different mixtures of 1-butyl-3-methyl-imidazolium tetrafluoroborate with water have been studied by means of molecular dynamics simulations. Based on the classical Lopes-Padua force field trajectories of approximately 60 ns were computed. This is the third part of a series concerning the collective network of 1-butyl-3-methyl-imidazolium tetrafluoroborate/water mixtures. The first part [J. Chem. Phys. 127 (2007), 234503] dealt with the orientational structure and static dielectric constants. The second part [J. Chem. Phys. 129 (2008), 184501] was focused on the decomposition of the dielectric spectrum of these mixtures. In this work the focus lies on the characterisation of the neighbourhood of ionic liquids by means of the Voronoi decomposition. The Voronoi algorithm is a rational tool to uniquely decompose the space around a reference molecule without using any empirical parameters. Thus, neighbourhood relations, direct and indirect ones, can be extracted and were used in combination with g -coefficients. These coefficients represent the generalization of the traditional radial distribution function in order to include the mutual positioning and orientation of anisotropic molecules. Furthermore, the Voronoi method provides, as a byproduct, the mutual coordination numbers of molecular species.

2.1 Introduction

In ionic systems a cation or an anion cannot be considered independently. While in the gas phase the thinking in terms of ion pairs is appropriate, things become much more complex in the condensed phase. The concept of an ion pair has to be replaced by an ion surrounded by shell of counterions

and/or neutral solvent molecules. Although the central ion interacts with multiple counterions, the overall charge neutrality is fulfilled. This traditional picture of “charge clouds” is inherent to the well-known Debye-Hückel theory^[1] which has already introduced such basic concepts like ion activity and the radius of the charge cloud. In the former case, the Coulomb interaction between the reference ion and its neighbouring shell of counterions reduces the activity which can be calculated as a function of concentration, temperature and dielectric constant of the solvent. The very same parameters also enter the formula giving the radius of the ionic cloud. However, this concept of a radius can be hardly transferred to the situation of molecular ionic liquids with their inherent shape and charge anisotropy. This requires the development of new ideas for classifying neighbourhood relations. Commonly in molecular dynamics simulations of ionic liquids, the investigation of the local structure is based on structure factors, radial distribution functions and three dimensional plots.^{[2]–[11]} Augmenting these analysis tools by a specified distance criterion, neighbourhood relations may be deduced. But, this inevitably necessitates the introduction of certain threshold below which particles are considered as neighbours. Consequently, these thresholds comprise a set of parameters which strongly influences the final results. This is already a problem in simple liquids becoming rather complex for molecular species with their anisotropy in shape and charge. For example, the distance threshold to be used depends on the reference atom to which neighbourhood is defined. This parameter-based approach becomes even more difficult when dealing with mixtures of different molecular species. Consequently, a parameter-free method of spatial decomposition is highly desirable. Fortunately, computational geometry offers such a method: Voronoi polyhedra decompose space into cells, each of which consists of points closer to one particular reference point than to any others. This method has been reinvented, given different names, generalised, studied, and applied many times over in many different fields.^{[12],[13]} A first attempt to use this rational, parameter-free approach in the field can be found in Ref. [14]. There, the ionic liquid cations were represented by a two sphere model whereas the anions are approximated by a sphere. This coarse-grained model was able to reproduce the dependence of density and transport properties on the charge delocalization. But the poor resolution of this coarse-grained model prohibits a detailed analysis of the neighbourhood which is the aim of this work. Therefore, our Voronoi decomposition is performed at the full atomic resolution.

This work is part of a series of publications concerning the collective networks of 1-butyl-3-methylimidazolium tetrafluoroborate $\text{BMIM}^+\text{BF}_4^-$ with water: The first part concentrated on the structural analysis of the mutual networks of cations, anions and water.^[15] These networks were analysed by atom-atom radial distribution functions as well as so-called g -coefficients, which reveal the mutual

orientation and position of the molecules. In addition, the collectivity of the mixture was interpreted in terms of the Kirkwood G_k -factor and the static dielectric constant. The second part extended the analysis of the dielectric constant and compared the computational results of the frequency-dependent dielectric spectrum with experimental data.^[16] Both references observed an increased ordering of water with increasing content of ionic liquids in the mixture. This fact is probably due to the direct neighbourhood to the ionic liquids. Therefore, this work focuses on a detailed analysis of this neighbourhood, which can be unambiguously defined by a Voronoi decomposition. In other words, the Voronoi polyhedra of the neighbour molecule shares a common face with the reference polyhedron. Consequently, this method is particularly apt to cope with highly anisotropic reference sites, e.g. long tails of imidazolium based ionic liquids.

2.2 Theory and Methods

2.2.1 Voronoi decomposition

The Voronoi decomposition is simply defined by a non-degenerate set of points $P_{1...N}$ and allocates all space amongst this set. Each point P_i [in our case atoms] is surrounded by an irregular polyhedron containing all space closer to its associated reference point than to any other point P_j of the given set. The sum of these non-overlapping polyhedra is space-filling and the faces of each of these polyhedra are constructed by planes perpendicular to the vectors between the associated reference point and its neighbour points. For example, in Fig. 2.1a the Voronoi cell of the point P_6 is the gray shaded area. Please note, that all sketches explaining the algorithm are drawn in two dimensions only for the sake of simplicity.

However, this Voronoi decomposition into polyhedra corresponds geometrically to a Delaunay tessellation which has been used extensively in the design of efficient algorithms for interpolation, contouring or mesh generation.^[17] In principle, a Delaunay tessellation is a unique partitioning of the set $P_{1...N}$ into “simplices”,^[13] e.g. $P_iP_jP_kP_l$. In three dimensional space such a simplex is an irregular tetrahedron and its four vertices [P_i , P_j , P_k and P_l] are a subset of $P_{1...N}$. These four vertices of a tetrahedron lie on the surface of a circumscribed sphere which does not contain any further [vertex] point P_i . This definition will be called “Delaunay criterion” subsequently. In two dimensions the circumscribed sphere and the tetrahedron reduces to a circumscribed circle and triangle, respectively. In Fig. 2.1b the circumscribed circle is marked bold for the triangle $P_3P_4P_6$. The center of this circle [or sphere in 3D] coincides with the vertex of the Voronoi cell. The edges [dashed lines] of the Delaunay

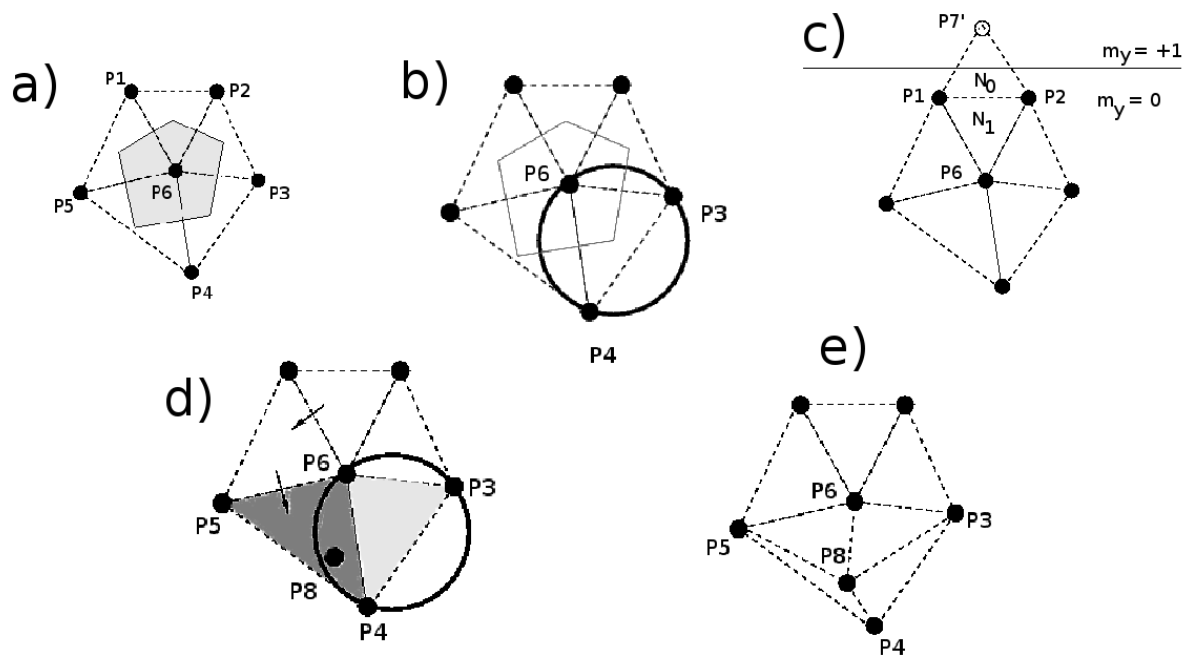


Figure 2.1: Two dimensional sketches illustrating the Voronoi algorithm: a) The dualism of Voronoi decomposition and Delaunay tessellation is shown. The gray shaded area is the Voronoi cell, the triangles with the dashed edges depict the Delaunay simplices. b) The Delaunay criterion [black circle] ensures that no further point lies within the circumscribed circle of the Delaunay triangle. c) The Delaunay triangle N_0 contain an image atoms $P7'$. This fact is saved by periodicity indices m . d) The new insertion point P_8 is located at the simplex $P_4P_5P_6$ which is called base (dark gray shaded area). The light gray shaded area shows a simplex which Delaunay criterion was violated by P_8 . e) “Star-shaped” region around P_8 . After deleting the old Delaunay distances, new distances are constructed in such a way that the Delaunay criterion is fulfilled for all simplex in the box.

simplex [triangle in 2D or tetrahedra in 3D] are the vectors connecting two points P_i and P_j being cut by the orthogonal Voronoi faces [solid line] midway.

In literature a plethora of tessellation algorithms can be found,^{[18]–[21]} but in computer simulations of bulk media periodicity is commonly used to emulate infinite systems. Consequently, Delaunay algorithms^{[17], [20]} taking explicitly into account periodicity are of special importance. Thereby, periodicity is an essential part of the algorithm which excludes algorithms working on the primary cell augmented by its explicit images. Among the periodic Delaunay algorithms “insertion algorithms” have proven particularly successful and were adopted in various ways.^{[17], [18], [22]} They start from a given tessellation of a subset of points and insert a new arbitrarily selected point into the tessellation until all points of the set are tessellated. Then, the finite set of unique simplices is called primary tessellation and can be completely described by specifying the four vertexes of each simplex. Consequently, each simplex is represented by a row in our data structure containing the indices of the vertex points [P_i , P_j , P_k and P_l]:

P_i	m_i^x	m_i^y	m_i^z	P_j	m_j^x	m_j^y	m_j^z	P_k	m_k^x	m_k^y	m_k^z	P_l	m_l^x	m_l^y	m_l^z	N_1	N_2	N_3	N_4
-------	---------	---------	---------	-------	---------	---------	---------	-------	---------	---------	---------	-------	---------	---------	---------	-------	-------	-------	-------

Furthermore, this row also contains the row number of all four adjacent simplices [N_1 , N_2 , N_3 and N_4] which share a face with the simplex $P_i P_j P_k P_l$. In Fig. 2.1c, the reference simplex is N_0 build up by P_1 , P_2 and P'_7 . The first neighbour simplex N_1 [$P_1 P_2 P_6$] shares the face $P_1 P_2$ with N_0 . The storage of this neighbour information facilitates the overall computation. Since we are dealing with periodic tessellations, each vertex point information in that row is augmented by three indices [m_1^x , m_1^y and m_1^z] indicating the periodic displacement of that vertex. For example, in our two dimensional sketch in Fig. 2.1c m_1^y and m_2^y equal zero because both points are located in the primary simulation box. P'_7 is the image of P_7 . Consequently, m_7^y is +1 in this case.

Now, we turn to the description of our insertion algorithm. The initial tessellation is constructed from a randomly chosen point and seven of its images forming a cube. This cube is subdivided into six tetrahedra serving as initial simplices. The insertion of a new point is divided into several steps: First, one looks for the simplex containing this new point or one of its images. This simplex is called “base”. In our example in Fig. 2.1d the point P_8 is to be inserted. In this case the “base” [dark gray shaded area] is $P_4 P_5 P_6$. Unfortunately, the base location cannot be inferred directly from the current tessellation data nor the coordinate array. Therefore, a search starting from one tetrahedron must be performed efficiently to manage large tessellations.^{[17], [22]} In each tetrahedron along the path [arrows

in Fig. 2.1d], one determines which of its four faces can be crossed to move closer to the insertion point and chooses one of these possible pathways randomly. Thereby, the search path may cross the periodic boundaries. If the located base is not in the primary cell it is periodically shifted there assuring that at least one vertex [P_1 , P_2 , P_3 or P_4] is in the primary cell. As a result, the primary tessellation contains no gaps and all tetrahedra are connected.

Second, all neighbouring simplices failing the Delaunay criterion are detected. For example, in Fig. 2.1d the insertion point P_8 violates the criterion for the simplex $P_3P_4P_6$. The collection of these neighbouring simplices plus the base is called “core” [gray shaded areas] and all points on its surface the “boundary surface”. In some cases the circumsphere of a neighbouring simplex contains only an image of the insertion point. These tetrahedra are shifted in such a way that their circumsphere incloses the primary location of the insertion point. Afterwards, a visibility check is performed to check that all faces on the surface of the core are directly visible to the insertion point ensuring a “star-shaped core region” [c. f. Fig. 2.1e] and a non-negligible volume of each new tetrahedron.

Third, all simplices constituting the core are deleted and a new set of simplices is constructed by connecting the inserted point to all points on the boundary surface. Afterwards, a further new point is inserted and the whole procedure is continued iteratively. A more detailed description of each algorithm step is given in Ref. [17].

After completing the final tessellation of a trajectory frame, it can be used in a multitude of ways: A neighbourhood list can be easily created by looking the rows in our data structure. In this sense a “neighbour” is a point which shares at least one tetrahedron with the reference point. The ensemble of all these neighbours constitutes the first coordination shell. The neighbours of the neighbours form the

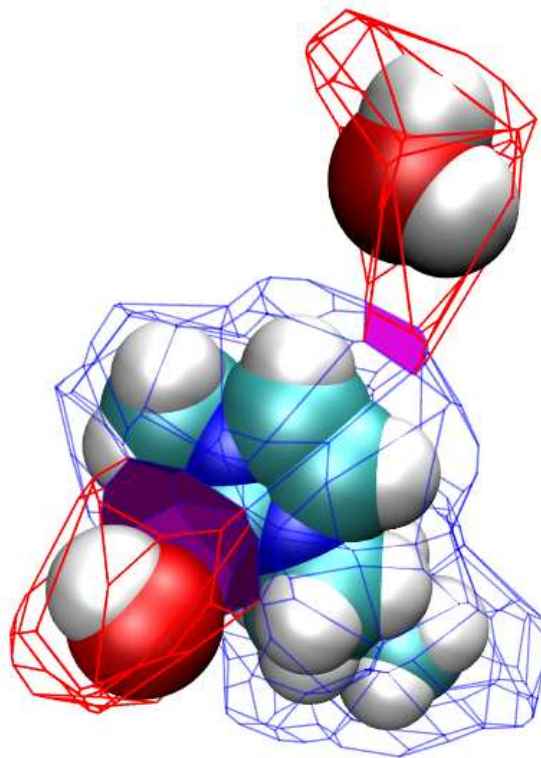


Figure 2.2: Typical Voronoi polyhedron of BMIM^+ and two neighbouring water molecules. The tessellation was performed on an atomic resolution. The shaded areas show the interaction area of BMIM^+ with each water molecule.

second shell and so on. Alternatively, a shell may be defined as a set of particles which share a minimal Delaunay distance to the reference site. The term Delaunay distance is meant in a graph-theoretical sense. In other words, our decomposition provides a parameter-free definition of neighbourhood which can be seen as a “direct interaction” rather than smallest distance criterion. This fact is also visualised by Fig. 2.2. There, a BMIM⁺ and two neighbouring water molecules and their Voronoi polyhedra are depicted. The water molecule on the left bottom side has a small distance to the ring of the cation. Consequently, the interaction surface is bigger than the corresponding surface of the second water molecule at a longer distance. Nevertheless, both water molecules are direct neighbours of the cation.

Another interesting feature of a Voronoi decomposition is the evaluation of volumes and surfaces of the Voronoi polyhedra. These properties can be calculated for each tessellation point exploiting the data structure described above. Each point shares exactly one area of its Voronoi polyhedron with one neighbour point. The corresponding centers of circumspheres used to check the Delaunay criterion in that case are the vertexes of this area. By rotating through the data structure each tetrahedron which shares this Voronoi surface area is found. Thereby, it might be necessary to check the boundary conditions again. In this manner, the Voronoi area can be determined by summing up the areas of the triangles constituting it. The volume of the Voronoi polyhedron is simply the sum of the selected tetrahedra. This rotation is repeated for each Voronoi surface belonging to the tessellation point under investigation. The volume of a molecule can be computed by the sum of the volumes of the tessellation points belonging to that molecule. In case of the surface, one has to keep in mind that there are inner and outer surface areas. The surface area is only build up by the latter ones.

Our self-written program package GEPETTO constructs the sequence of Voronoi shells surrounding each reference site. The members of each shell as well as its volume and its surface are computed. Furthermore, it combines the traditional approach using common structural analyze tool, e.g. common radial distribution functions (RDF) $g^{000}(r)$ and orientational correlation functions [$g^{110}(r)$, $g^{101}(r)$ and $g^{011}(r)$], with the method of Voronoi decomposition.

2.2.2 Combined g -coefficient/Voronoi analysis

The above mentioned g -coefficients are the distance dependent part of the orientational probability function $g(r_{ij}, \Omega_i, \Omega_j, \Omega_{ij})$ of molecular pairs. The orientation of a pair of molecules i and j is each described by a set of three Euler angles, denoted by Ω_i and Ω_j , respectively. The distance vector $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ joining molecular centers is expressed in terms of polar coordinates, i.e. , the intermolecular distance $r_{ij} = |\mathbf{r}_{ij}|$ and two polar angles, denoted here as Ω_{ij} . If one is only interested in the spatial

distribution of molecular center irrespective of molecular orientations the general probability function reduces to $g(r_{ij}, \Omega_{ij})$.^{[23]–[25]} Please note, these $g(r_{ij}, \Omega_{ij})$ always refer to a specific molecule-fixed coordinate system. For example, in Ref. [23]–[25] the z-axis of the coordinate system coincides with the dipole vector of the reference water molecule. Quite generally, the coordinates of all neighbours of a reference molecule have to be projected to the molecule-fixed local coordinate system. This procedure is also inherent to all 3D histogram plot, e.g. Fig. 2.9 in our present work. On the contrary, the g -coefficient method is independent of the choice of the coordinate system and involves the complete set of rotational angles Ω_i , Ω_j and Ω_{ij} .

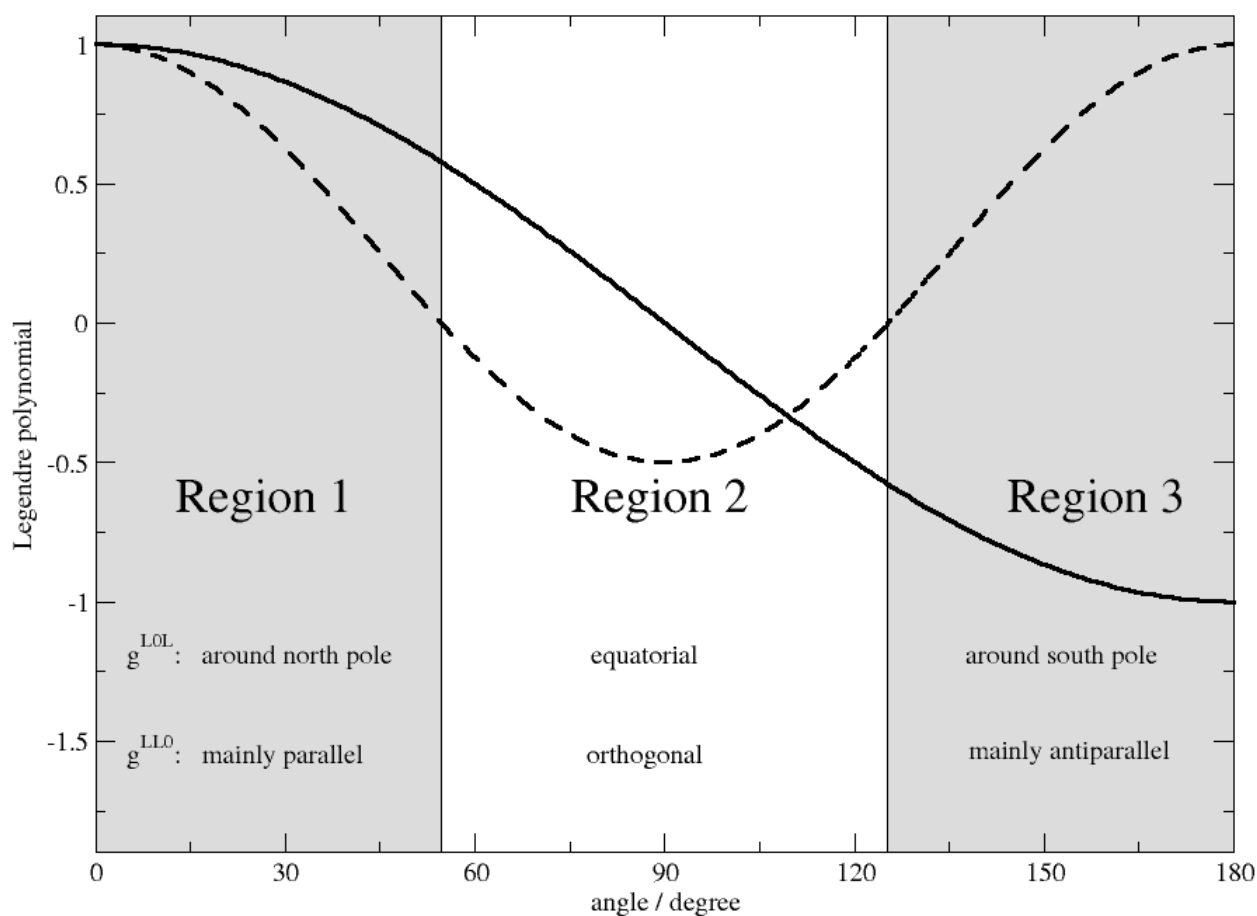


Figure 2.3: First (solid line) and second (dotted line) Legendre polynomials of the cosine of the polar angle. The sign pattern of these functions reveal the position in $g^{LOL}(r)$ and $g^{0LL}(r)$ or the orientation in $g^{LLO}(r)$.

A detailed description of the expansion of the full angle dependent pair correlation function $g(r_{ij}, \Omega_i, \Omega_j, \Omega_{ij})$ into a set of rotationally invariant basis functions can be found in Ref. [26]. The expansion coefficients

g -coefficient	$\Phi^{L_i, L_j, L_{ij}}$	
$g^{000}(r)$	1	radial distribution function
$g^{110}(r)$	$\cos(\mu_i, \mu_j)$	mutual orientation
$g^{101}(r)$	$\cos(\mu_i, \mathbf{r}_{ij})$	position of j with respect to i
$g^{011}(r)$	$\cos(\mu_j, \mathbf{r}_{ij})$	position of i with respect to j
$g^{220}(r)$	$\frac{3}{2} \cos(\mu_i, \mu_j) - \frac{1}{2}$	mutual orientation
$g^{202}(r)$	$\frac{3}{2} \cos(\mu_i, \mathbf{r}_{ij}) - \frac{1}{2}$	position of j with respect to i
$g^{022}(r)$	$\frac{3}{2} \cos(\mu_j, \mathbf{r}_{ij}) - \frac{1}{2}$	position of i with respect to j

Table 2.1: List of the distance dependent $g^{L_i, L_j, L_{ij}}(r)$ -coefficients with $L \in \{0, 1, 2\}$. μ_i and μ_j are the molecular dipole moments of molecule i and j being separated by the vector \mathbf{r}_{ij} .^[51] Higher orders of the angular momentum (L_i , L_j and L_{ij}) replace the cosine function by their respective Legendre polynomial. The latter functions allow a finer resolution of the angular space which can be seen in Fig. 2.3.

$g^{L_i, L_j, L_{ij}}(r)$ depend on the mutual center-of-mass distance of the pair of molecules considered. For computational convenience, the angular basis functions are not expressed in the original Euler angles Ω_i , Ω_j and Ω_{ij} but are evaluated as Legendre polynomials of the cosines tabulated in Table 2.1.^[27] The upper indices (L_i , L_j and L_{ij}) specify the order of the angular momentum of the respective polar angles of the reference molecule i , its (direct or indirect) neighbour j and the distance vector \mathbf{r}_{ij} . Principally, $g^{L_i, L_j, L_{ij}}(r)$ looks like

$$g^{L_i, L_j, L_{ij}}(r) = \frac{1}{\rho \, 4\pi r^2 dr} \sum_j \Phi^{L_i, L_j, L_{ij}} \cdot \delta(r - |\mathbf{r}_{ij}|). \quad (2.1)$$

$4\pi r^2 dr$ is the volume of a *spherical* shell of thickness dr and ρ is a global particle density. In practice, $g^{L_i, L_j, L_{ij}}(r)$ is computed as a histogram accumulating entries of the angular function $\Phi^{L_i, L_j, L_{ij}}$ into a bin from r to $r + dr$. The bin selection is the computational analogue of the mathematical δ -function.

In case of $L_i = L_j = L_{ij} = 0$, $\Phi^{L_i, L_j, L_{ij}}$ equals 1 and the corresponding $g^{000}(r)$ is identical to the well-known center-of-mass RDF. The extension to higher $g^{L_i, L_j, L_{ij}}(r)$ -coefficients is straight forward: The entry of 1 is replaced by the value of the respective Legendre polynomial of the angular function $\Phi^{L_i, L_j, L_{ij}}$.^{[28], [29]} These angular functions reflect the mutual position [$g^{L_i, 0, L_{ij}}(r)$ and $g^{0, L_j, L_{ij}}(r)$] and orientation [$g^{L_i, L_j, 0}(r)$] of a pair of molecules and may be seen as an extension of the traditional RDF, which is sufficient for simple liquids lacking molecular orientation. Additionally, in case of water neighbours $g^{0, L_j, L_{ij}}(r)$ reveals the hydrogen bond donor/acceptor role of the reference site i .^[30] An

intuitive interpretation of Eq. 2.1 takes the value of a $g^{L_i,L_j,L_{ij}}(r)$ -coefficient as the statistical average of the respective angular function at a fixed intermolecular distance r . This averaged value and in particular its sign may be attributed to the most populated angular regions. For the two types of angular functions, $g^{L_i,0,L_{ij}}(r)$ and $g^{L_i,L_j,0}(r)$, these angular regions are depicted in Fig.2.3. We have attached appropriate labels to regions in that figure in order to facilitate the subsequent discussion in literal terms. For the special case of $L = 1$ and $L = 2$, the pattern in sign discriminates three distinct regions: The two regions with a positive sign in the second Legendre polynomial, region 1 and 3, can be distinguished by the sign of the first Legendre polynomial. Region 2 is characterised by a negative sign of the second Legendre polynomial and a change of sign of the first Legendre polynomial.

While molecular anisotropy is reflected by the angular functions used to compute the $g^{L_i,L_j,L_{ij}}(r)$ -coefficients the concept of radial bins still sticks to the picture of *spherical* shells. In our IL system, this description may be sufficient for the anion BF_4^- and water but is certainly too limited for the cation BMIM^+ . As a first remedy BMIM^+ may be splitted into three parts, head (methyl-group, \mathcal{H}), ring (\mathcal{R}) and tail(butyl-group, \mathcal{T}), for the analysis. This decomposition dates from Ref. [31], but there \mathcal{H} and \mathcal{R} were merged together. This still poses the problem of excluded volumes, i.e. spatial areas are occupied by other parts of the reference BMIM^+ and not by its neighbours. This is again a drawback of the concept of spherical shells. In order to overcome this limitation we decided to combine the method of Voronoi decomposition with the g -coefficient analysis retaining the concept of head-ring-tail.

In the combined analysis the histogram of a particular $g^{L_i,L_j,L_{ij}}(r)$ -coefficient is constructed as described above, but the set of neighbours j is restricted to a specific Voronoi shell. Thus, a $g^{L_i,L_j,L_{ij}}(r)$ -coefficient is decomposed into a sum over all these Voronoi shell specific contributions. The advance of this combined method can be demonstrated in Fig.2.4. The solid line represents the overall $g^{000}(r)$. Starting with the radial distribution of BMIM^+ -tails around BMIM^+ -tails, the first peak of $g^{000}(r)$ in Fig.2.4a coincides with the peak of the first Voronoi shell. At first sight this assignment seems to be obvious but is not so straight forward for the second and third Voronoi shell since there is no further clear peak in the overall $g^{000}(r)$. Even more, the tail-tail distribution is a fortunate example for clear-cut assignments of the first peak. As visible in Fig.2.4b the ring-ring distribution shows a clear peak at 8.5Å but this does not correspond to the first Voronoi shell. In other words, this peak is not constituted by the direct neighbours of the cationic ring. In fact, the direct neighbours are hidden in the foot of the peak at 6.5Å indicated by the dotted line (first Voronoi shell). It is the second Voronoi shell containing only the neighbours of the direct neighbours which creates the first peak

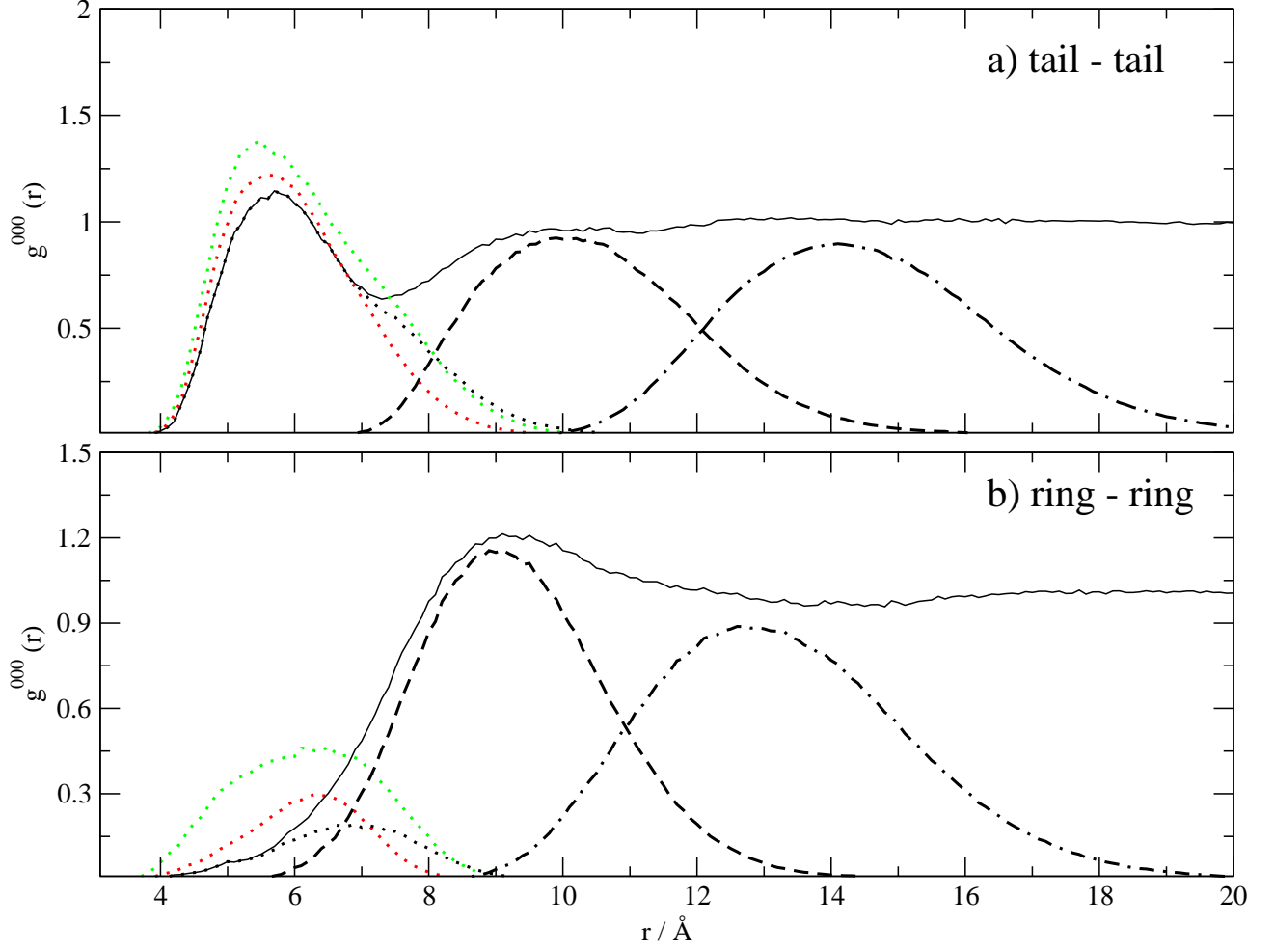


Figure 2.4: Decomposition of the $g^{000}(r)$ -coefficient into its contributions from the first (dotted), second (dashed) and third (dash-dotted line) Voronoi shells at $x_{H_2O} = 0.967$: a) \mathcal{T} - \mathcal{T} distribution, b) \mathcal{R} - \mathcal{R} distribution. Here, the contributions of the first Voronoi shell at a mole fraction of $x_{H_2O} = 0.912$ (green) and $x_{H_2O} = 0.768$ (red) are displayed additionally.

of the overall $g^{000}(r)$. The peak of the third Voronoi shell finally coincides with the first minimum of the overall $g^{000}(r)$ in contradiction to intuition. Another important point to mention is the very large extension of the third shell up to 20Å. This means that in our simulation box of 41.8Å only the first three Voronoi shells can be analysed. We consider this as the minimum correlation length in order to permit a transition of structural oscillations to an uniform bulk property. Nevertheless, we are aware that the third shell is at the edge of this transition. In fact, there are numerous examples for pure ionic liquids where structural oscillations in radial distribution functions extend up to this threshold.^{[7],[32]–[35]} Smaller box sizes reduce the number of solvent layers around the cation to less than three. In systems with periodic boundary conditions this would favor spurious self-interactions between cations and its images.

2.2.3 Computational setup

Since this work is the third part of a series of paper, the detailed description of the molecular dynamics simulation setup is given in Ref. [15] and we will only briefly denoted here. Three different mixtures of BMIM⁺BF₄[−] and TIP3P-water were simulated: The force field of the cation, anion and water were taken from Ref. [36], [37], Ref. [38] and Ref. [39], respectively. Coulomb interactions were calculated by the Particle-Mesh Ewald method,^{[40],[41]} using a 10 Å cutoff and a κ of 0.41 Å^{−1} for the real-space part interactions. All bond lengths were kept fixed by the SHAKE algorithm,^[42] whereas bond angles and dihedrals were left flexible. Trajectories were generated with the molecular dynamics program package CHARMM^[43] under constant volume with a boxlength of 41.8 Å and an average temperature of $T = 300$ K and an average pressure close to 1 atm . The trajectory was propagated for 62 ns with a time increment of $\Delta t = 2$ fs. Each trajectory frame was tessellated at an atomic level. A coarser graining representing the whole molecule by its center-of-mass was considered too crude since essential features of anisotropy are lost in this way. We want to emphasise that each tessellation was checked in a twofold way: First, the Delaunay criterion was carefully revisited in order to secure that pairs of particles recognised as direct neighbours are not separated by overseen interstitial molecules. Second, the sum of all molecular volumes had to equal the total volume of the simulation box.

Table 2.2 summarises the compositions of the mixtures studied. Additionally, average molecular volumina in these liquid mixtures are provided. In combination with the particle numbers, the occupancy of each species (cation, anion and water) of the total simulation box can be computed. As the mole fraction x_{H_2O} only counts particle numbers irrespective of the molecular volumes, we consider this occupancy a more intuitive measure of the composition. This is most evident in case of

$x_{H_2O}=0.768$ where the cations cover one half of the total volume leaving a quarter to water which in terms of mole fractions contributes three quarters. Seen in this way the mixtures studied are far from diluted systems as the might look at first sight.

The simultaneous application of the Voronoi algorithm and the g -coefficients combines the criterion of neighbourhood with the the concept of distance spread. Thereby, we found that in a 41.8 Å simulation box only three Voronoi shells can be completely resolved. This is clear evidence that smaller systems are restricted to direct and indirect neighbours only thus missing any features of a bulk property.

2.3 Results and Discussion

In principle, the Voronoi decomposition of g -coefficients may be applied to any pair of species or moieties, e.g. anion (\mathcal{A}), water (\mathcal{W}), head (\mathcal{H}), ring (\mathcal{R}) and tail(\mathcal{T}) ending up with 15 possible pairs for each g -coefficient. As our box size permits the construction of three complete Voronoi shells, each g -coefficient can be further decomposed into its Voronoi shell contributions. As we use $g^{000}(r)$, $g^{L_i,0,L_{ij}}(r)$, $g^{0,L_j,L_{ij}}(r)$ and $g^{L_i,L_j,0}(r)$ to describe the mutual position and orientation of these pairs, the complete data set covering all this information is already extremely large and is even more enlarged when considering all three compositions of the IL/water mixtures. Nevertheless, we have analysed the whole data field in order to get a consistent picture. For the presentation in this paper, however, we have reduced the information to the essential features.

2.3.1 The hydrophobic cationic reference site

Turning back to the \mathcal{T} - \mathcal{T} configuration displayed in Fig. 2.4a the height of the peak of the first Voronoi-shell increases from 1.14 over 1.24 to 1.38 at the mole fractions $x_{H_2O}=0.967$, 0.912 and 0.768, respectively. An even more pronounced enhancement of the local density can be achieved by lengthening the tail to an octyl chain.^[31] This automatically increases the molecular volume of the cation. As we have learnt from Table 2.2 this enhanced cationic volume leads to very high occupancies. In other words, the total cationic volume overwhelms all other species. Therefore, the enhanced hydrophobicity is merely a matter of occupancy. In the cited systems, the local density is increased to a maximum value of roughly 3. However, the position of the peak remains unchanged. In our systems, the peak of the second Voronoi shell located around 10Å is almost unaffected by the variation of the water content. The increased local density of the first Voronoi shell is in accordance

		x_{H_2O}		
		0.768	0.912	0.967
particle	$N_{BMIM^+}, N_{BF_4^-}$	166	111	55
numbers	N_{H_2O}	548	1147	1592
occupancy	$N_{BMIM^+} \cdot \langle V_{BMIM^+} \rangle / V$	50.2%	33.0%	18.2%
	$N_{BF_4^-} \cdot \langle V_{BF_4^-} \rangle / V$	23.9%	15.0%	7.3%
	$N_{H_2O} \cdot \langle V_{H_2O} \rangle / V$	25.9%	52.0%	74.5%
molecular volumina	$\langle V_{BMIM^+} \rangle / \text{\AA}^3$	270	270	285
	$\langle V_{BF_4^-} \rangle / \text{\AA}^3$	63	59	59
	$\langle V_{H_2O} \rangle / \text{\AA}^3$	33	32	33
reference site	neighbour	coordination number		
cation	cation	9.4	5.5	2.5
	anion	5.6	3.8	2.1
	water	14.0	28.3	39.0
anion	cation	5.6	3.8	2.1
	anion	0.9	0.6	0.3
	water	5.3	10.6	14.9
water	cation	4.2	2.7	1.3
	anion	1.6	1.0	0.5
	water	4.3	8.4	12.2

Table 2.2: Composition and coordination of the simulated $BMIM^+BF_4^-$ /water mixtures.

with the increased number of cations in the solution. However, the enhancement of the mass density does not result in smaller distances between the tails of the cations. Nevertheless, approximately 60% of the cation–cation coordination number in Table 2.2 involves tail contributions. As a result, the \mathcal{T} - \mathcal{T} configuration is the preferred configuration between two adjacent cations and support the thesis of “nonpolar” domains.^[34] The $g^{L_i,0,L_{ij}}(r)$ with $L = 1, 2$ in Fig.2.5 show a preferred position in the proximity of the terminal CH_3 -group. The change in sign of $g^{202}(r)$ indicates a splitting into two

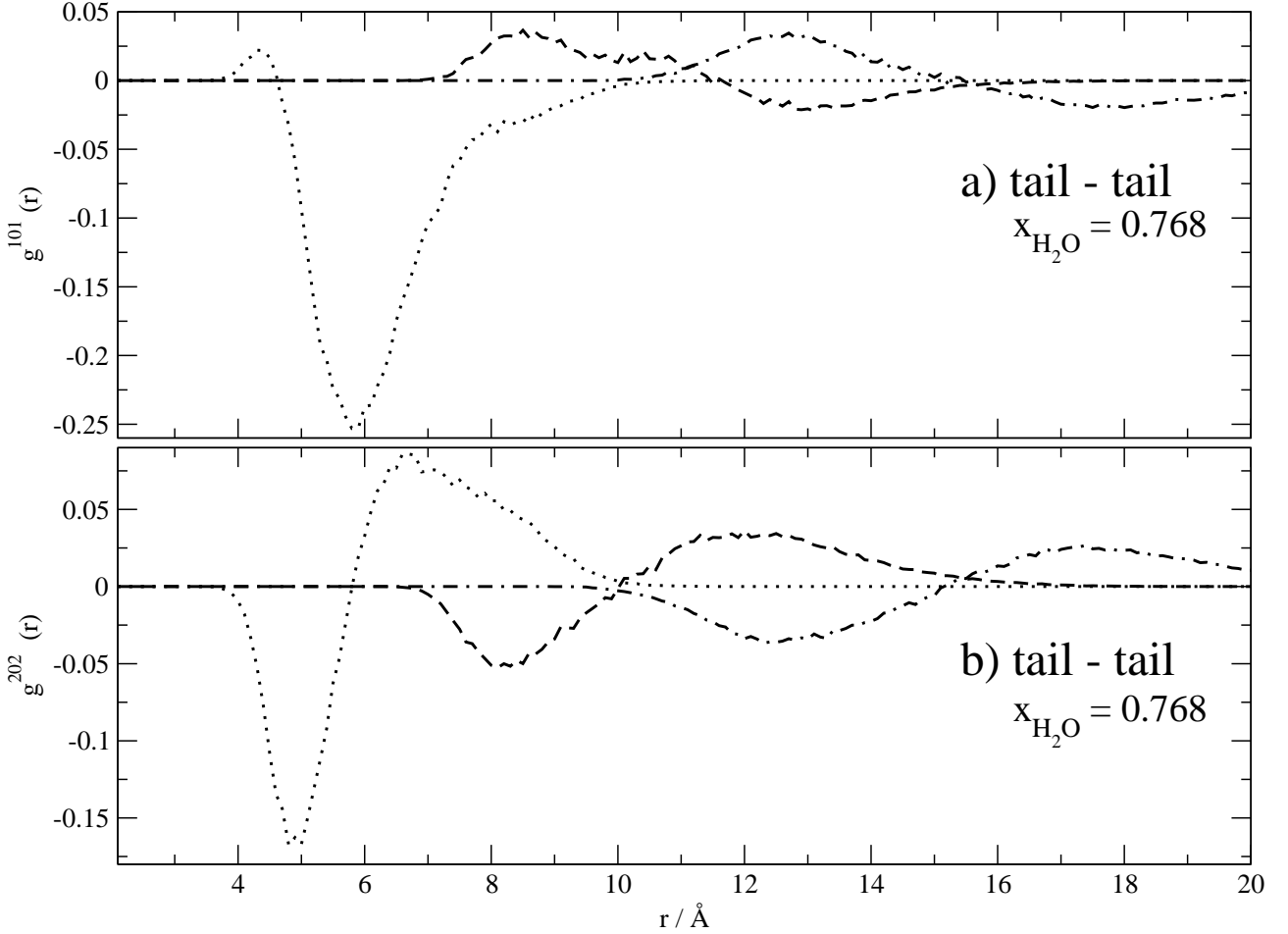


Figure 2.5: First (a) and second (b) $g_{00}^{L0L}(r)$ of \mathcal{T} - \mathcal{T} (tail-tail) angular correlation of $\text{BMIM}^+\text{BF}_4^-$ in water at a mole fraction $x_{\text{H}_2\text{O}} = 0.768$. The dotted, dashed and dash-dotted line correspond to the first, second and third Voronoi shell, respectively.

subgroups: Tails closer to the center-of-mass of the reference tail cover the right part of region 2 in Fig.2.3, i.e. they are aligned sideways with respect to the reference tail. The more distant tails above 6\AA are in region 3 corresponding to position behind the terminal CH_3 -group. The fluctuating sign of $g^{110}(r)$ and the clear sign pattern of $g^{220}(r)$ indicate an overall orthogonal orientation of tails pointing

away from the reference tail. The second and third Voronoi shell show no preferred orientation.

The first peak of the RDF $g^{000}(r)$ of \mathcal{T} - \mathcal{A} (tail-anion) pair is entirely determined by the first Voronoi shell with a peak height of approximately 2.0. In all these RDFs the variation with water content is marginal. This value is lower compared to that of \mathcal{H} - \mathcal{A} (head-anion) and \mathcal{R} - \mathcal{A} (ring-anion). A similar trend is found for \mathcal{T} - \mathcal{W} , \mathcal{R} - \mathcal{W} and \mathcal{H} - \mathcal{W} in accordance with the expected hydrophobicity of the tails. From the combined analysis of g^{101} and g^{202} a clear preference for the region 1 is observed. In other words, anion and water are actually in the proximity of the ring which is in accordance with simulated 1-octyl-3-methylimidazolium nitrate water mixtures.^[31]

2.3.2 The hydrophilic cationic reference site

The local density of water in the first Voronoi shell of the imidazolium ring is a function of the mole fraction x_{H_2O} . It exceeds the global density ρ by a mere 20% in case of $x_{H_2O}=0.967$ but increases to 80% above ρ in case of $x_{H_2O}=0.768$. One reason for this behaviour might be the decreased number of water molecules in the simulation box. Consequently, the probability to find an other water molecule in the vicinity of the reference water decreases as these sites are now occupied by the ions. However, this stoichiometric effect is not sufficient to explain the excess of 80%. Rather, the enhanced viscosity of the mixture sharpens the structure. In molecular terms one would say the electric fields exerted by the additional ions orient the water dipoles. For \mathcal{R} - \mathcal{W} this effect is not restricted to the raise of the first peak but also shows up in additional oscillations of $g^{000}(r)$ in Fig.2.6a for the lowest mole fraction of $x_{H_2O}=0.768$ extending up to 14Å. The decomposition into Voronoi shells reveals the origin of the small peak at 5.7Å: It comes from the superposition of the first and second Voronoi shell and does not correspond to a single discrete region or shell. This occurrence of virtual shells can only be detected by Voronoi decomposition technique. At this distance the molecules are either direct neighbours of the ring (first Voronoi shell) or indirect ones (second Voronoi shell) mediated by first neighbours. The rather different character of these Voronoi shell can be demonstrated by the hydrogen bond donor/acceptor function $g^{011}(r)$ in Fig.2.6b. Up to a distance of 5Å the imidazolium ring acts as a hydrogen bond donor for the direct water neighbours. In other words the water dipoles point away from the imidazolium ring. On the one hand, this is due to the higher affinity of the water oxygen to the hydrogen of the imidazolium ring. On the other hand, the water hydrogens are partially repelled by the hydrogens of the butyl chain attached to the ring. Beyond the distance of 5Å the donor behaviour is reversed visible in negative values of $g^{011}(r)$. This remarkable change from a donor to an acceptor role of the ring can be seen in all three Voronoi shells in Fig.2.6b. As the Voronoi

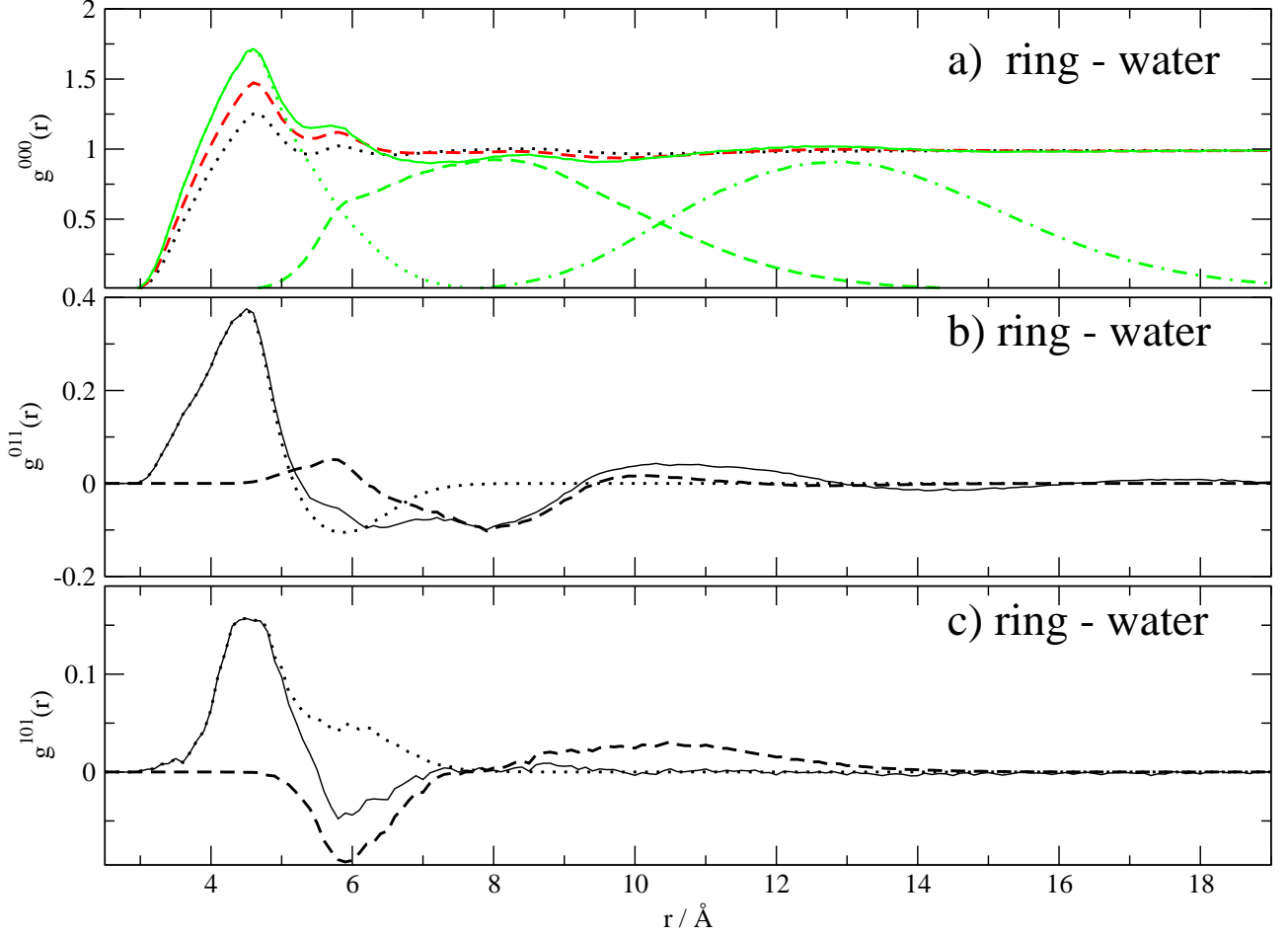


Figure 2.6: Several g -coefficients describe the characteristics of the \mathcal{R} - \mathcal{W} interaction: a) The local density g^{000} is displayed for $x_{H_2O} = 0.967$ (black), 0.912 (red) and 0.768 (green). In case of $x_{H_2O} = 0.768$ the first (green dotted), the second (green dashed) and the third (green dash-dotted) Voronoi shell is shown. b) g^{011} reveals the donor/acceptor role of the imidazolium hydrogens for the first (dotted), second (dashed) and third (dash-dotted) Voronoi shells at $x_{H_2O} = 0.768$. c) g^{101} indicates the position of the water molecules with respect to the reference imidazolium ring for the first (dotted), second (dashed) and third (dash-dotted) Voronoi shells at $x_{H_2O} = 0.768$.

shells are shifted relative to each other on a distance scale the donor region of one shell overlaps with the acceptor region of the preceding shell. While $g^{011}(r)$ describes the orientation of the water dipoles relative to the distance vector from the center of the ring, their position relative to the ring is characterised by $g^{101}(r)$. Fig.2.6c shows a clear preference of \mathcal{W} in the first Voronoi shell to the region in the proximity of the H_2 hydrogen of the imidazolium ring. Positioning near H_4 and H_5 is less favorable in the first Voronoi shell. However, water molecules at approximately 5.9\AA in the second Voronoi shell are located here.

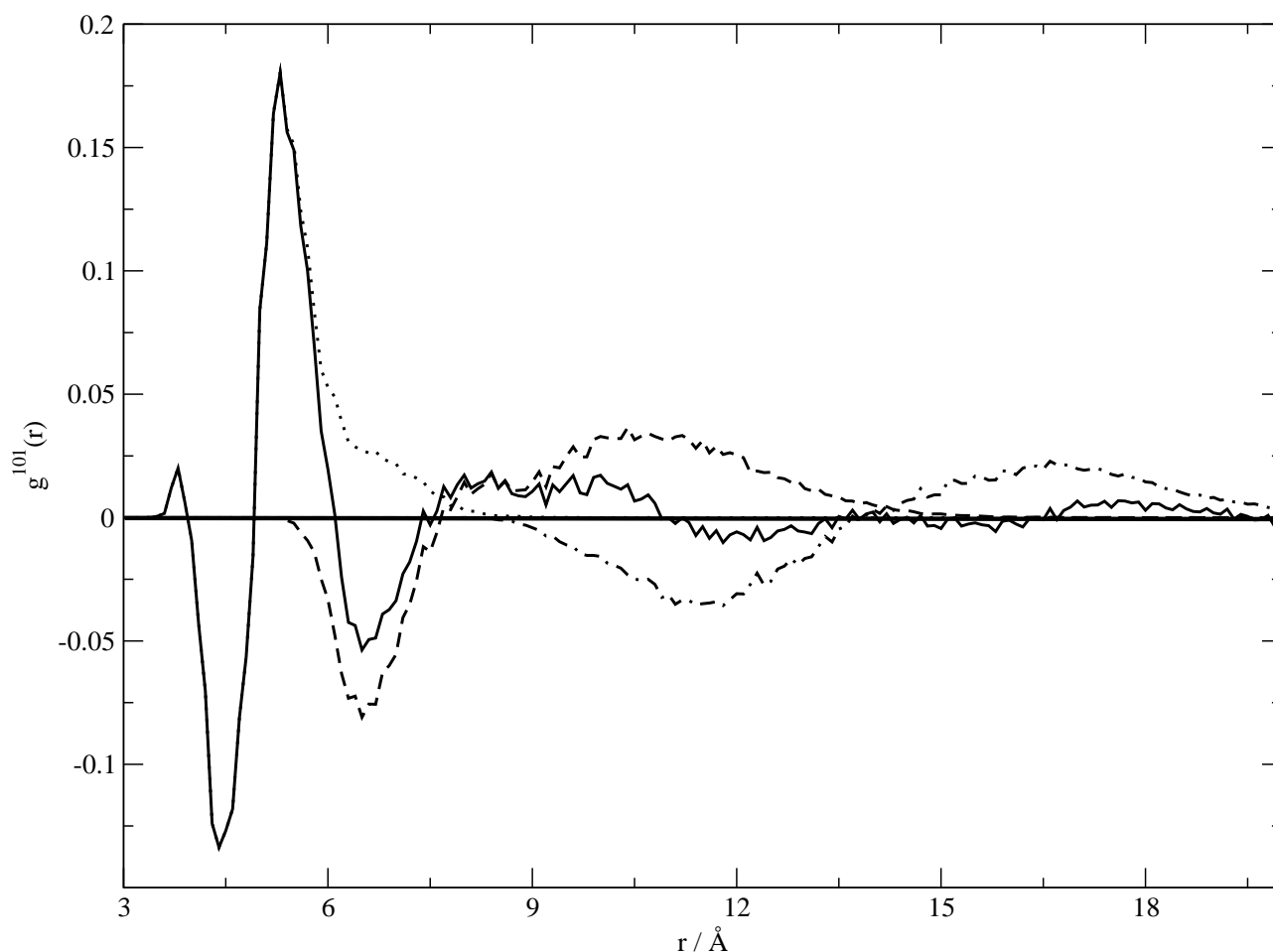


Figure 2.7: g^{101} -coefficient of $\mathcal{R}\text{-}\mathcal{A}$ at a mole fraction of $x_{\text{H}_2\text{O}} = 0.768$. The solid, dotted, dashed and dash-dotted line represent the overall, first, second and third Voronoi shell, respectively.

In contrast, the anions, which are direct neighbours of the ring, populate all regions near the imidazolium hydrogens as shown in Fig.2.7. Anions closer than 4.9\AA to the ring center are found near H_4 and H_5 complementary to nearest water molecules. Beyond that distance anion and water

compete for the H_2 region irrespective of the Voronoi shell to which they belong. Fig.2.7 shows a characteristic sign pattern of $g^{101}(r)$ that occurs in all three Voronoi shells: A negative region at lower distances is followed by a positive one. This alternation in sign corresponds to a population of the H_4 and H_5 region first and the H_2 region afterwards. Since the first peak in the RDF $g^{000}(r)$ of $\mathcal{R}\text{-}\mathcal{A}$ is made up by the first Voronoi shell, other authors find similar locations for the first shell in case of pure $\text{BMIM}^+\text{PF}_6^-$.^{[3],[7],[33],[44]} But the second Voronoi shell which contain all neighbours of the direct neighbours of the BMIM^+ has its maximum at approximately 8\AA which coincides with the first minimum of the overall $g^{000}(r)$. Consequently, this shell cannot be detected by a distance based threshold as used by the cited literature above. In fact, the second maximum in $g^{000}(r)$ originates from a superposition of the second and third Voronoi shell. Since the anions and water molecules occupy the regions near the imidazolium hydrogens, the neighbouring cations are forced to reside above and below the ring. This can also be deduced from the $\mathcal{R}\text{-}\mathcal{T}$ $g^{101}(r)$ (data not shown).

The $\mathcal{H}\text{-}\mathcal{A}$ RDF in Fig.2.8a displays a significant aggregation of anions at 4.2\AA which increases with decreasing water content. This effect seems to be more pronounced if the tail of the cation is elongated to an octyl chain.^[31] The corresponding $g^{202}(r)$ is negative at this distance which we attributed to region 2 in Fig.2.3. This means that the anions prefer an equatorial position in accordance with the findings of the $\mathcal{R}\text{-}\mathcal{A}$ pair correlations. It seems that the affinity of the imidazolium hydrogens for the anions is stronger compared to the methyl hydrogens. Consequently, anions in the vicinity of the methyl-group \mathcal{H} are always near the imidazolium hydrogens, too. This result in the equatorial position of BF_4^- with respect to \mathcal{H} . This feature seems common for all ionic liquids since it was found in numerous molecular dynamics simulations in literature.^{[3],[5],[33],[44]} Furthermore, it is supported by density functional theory calculations on a 6-31++G(d,p) basis set.^[45] In principle, all the observations concerning $\mathcal{H}\text{-}\mathcal{A}$ are also valid for $\mathcal{H}\text{-}\mathcal{W}$ in our mixtures but the local density, positioning and orientation is slightly reduced.

As a first summary we present a schematic 3D vector diagram of cationic neighbourhood in Fig.2.9. The tail \mathcal{T} of that reference BMIM^+ is located in the lower left corner, the head in the upper left. The viewpoint is towards H_4 and H_5 of the imidazolium ring. The blue arrows stand for the dipole moments of the adjacent cations and represent their average orientation. The positioning of the cations is given by the 3D bin from which the arrow emanates. Only bins with population of more than 2.5 times the average density are shown. In case of the anions the threshold was doubled. One has to bear in mind that the arrows do not represent individual cations but display the favored bins. The length and direction of the arrow was calculated by averaging all dipole vectors of that bin. If there

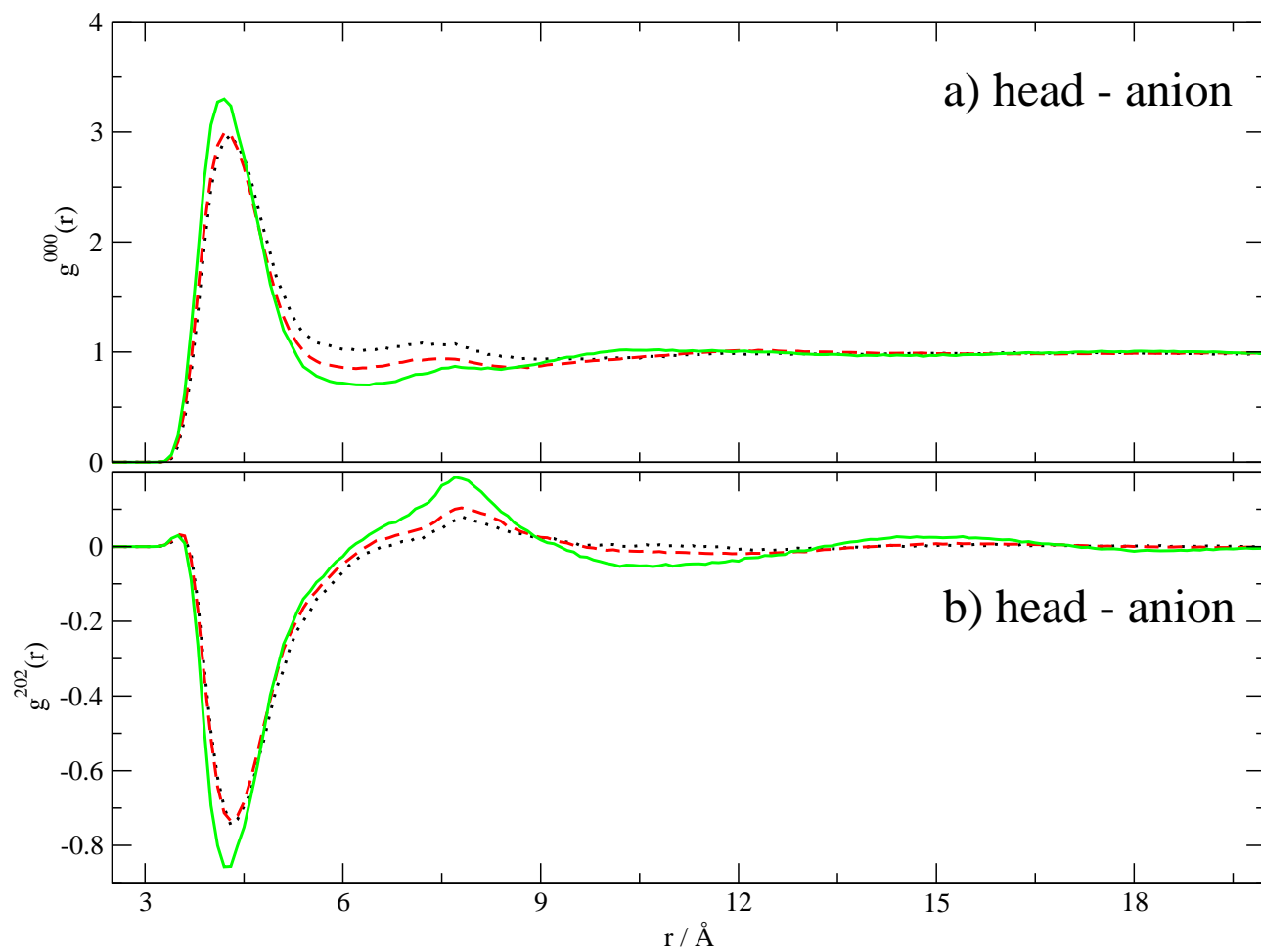


Figure 2.8: Local density (a) and position (b) of the anions with respect to the reference methyl-group (\mathcal{H}) of the cations.

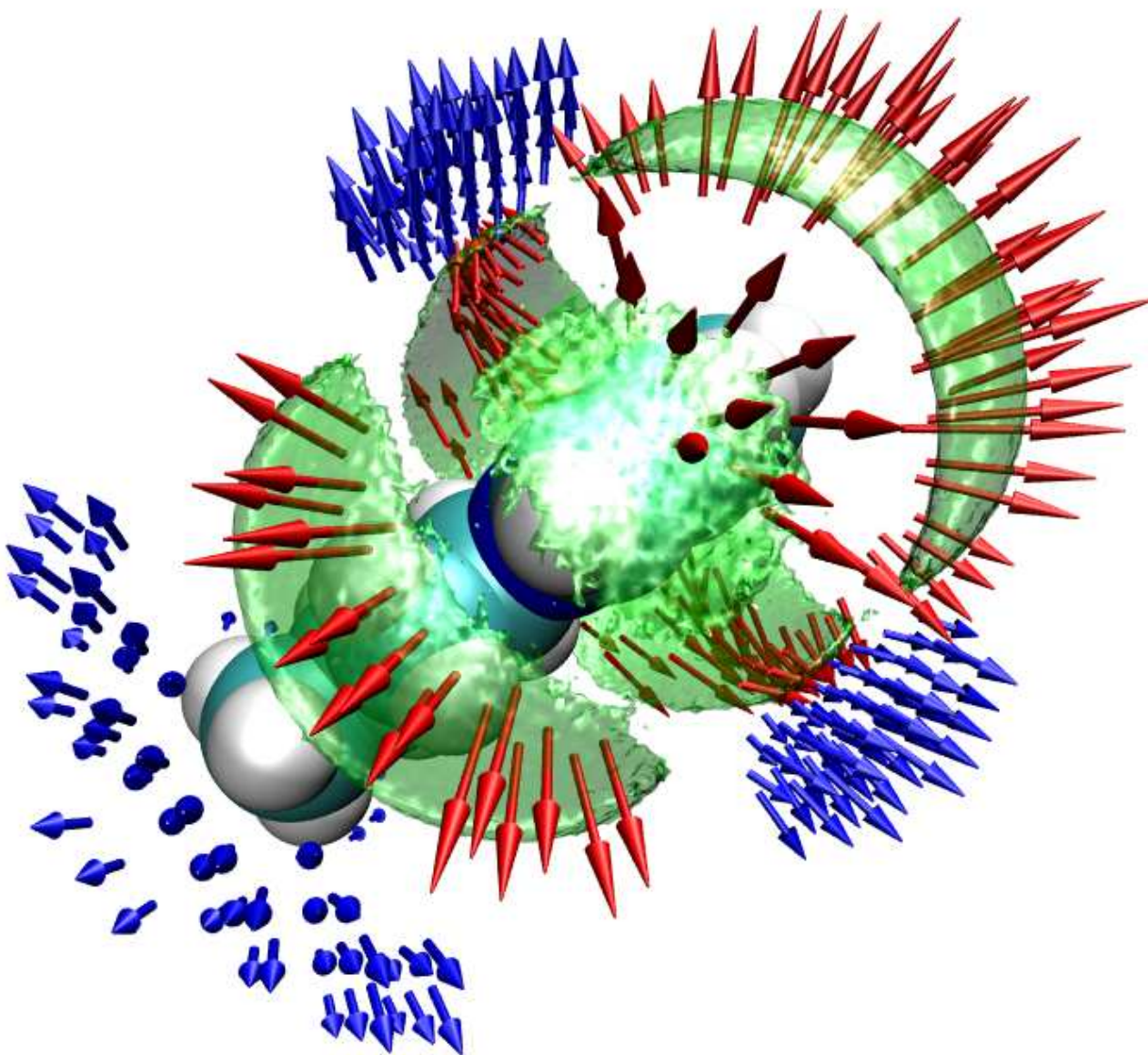


Figure 2.9: Schematic 3D vector diagram of the positioning and orientation of BMIM⁺ (blue), BF₄⁻ (green) and water (red) in the neighbourhood of a reference BMIM⁺. By construction the arrows represent the dipole moments averaged over all entries to the bin from which they emanate.

exists no preferred dipolar orientation, the resulting arrow is very small, e.g. some blue arrows in the vicinity of the tail in lower left corner. In the vicinity of the ring, the direction of cationic dipoles is already more pronounced. In case of a preferred dipolar orientation, the direction is confined to an angular cone and the arrow is of high length. This is indeed the case for water dipoles near ring \mathcal{R} (red arrows). Since BF_4^- exhibits small temporarily fluctuating dipoles, only their position is depicted by green surfaces.

Many features discussed on the basis of g -functions can be found in Fig.2.9 again. The preferred positions of cations around the reference BMIM^+ are at the tail region as well as above and under the ring. The cationic dipole vectors point away as showed by the respective $g^{110}(r)$. From the g -coefficient analysis anion and water molecules are located near the imidazolium ring. Interestingly, all three imidazolium hydrogens in the 3D picture 2.9 appear to be beset with anions and water molecules. At first sight, this seems to be in contradiction to the findings from the g -coefficients, which postulate a strong affinity of \mathcal{W} to the imidazolium H_2 . A more detailed inspection, however, shows that the water molecules in the proximity of H_4 and H_5 are not first Voronoi shell members, but belong to the second shell due to the respective negative sign of $\mathcal{R}\text{-}\mathcal{W} g^{101}(r)$ (dashed line in Fig.2.6c). The distance of 6Å appears to be rather short for second neighbours as it demands an interstitial partner. However, this needs not to be a complete molecule. Based on our atomwise tessellation an interstitial fluorine atom of BF_4^- is sufficient. Altogether we arrive at the conclusion that second neighbours to the H_4 and H_5 (curved green surface in front of the BMIM^+ in Fig.2.9) do not deserve the attribute “hydrogen bonded”. The confined angular freedom of the water dipoles is intuitively visible in the long red arrows of Fig.2.9 and is rationally found in the high peak of $\mathcal{R}\text{-}\mathcal{W} g^{011}(r)$ in Fig.2.6b.

2.3.3 The anion–water network

So far our analysis has mainly dealt with the correlations of the cations with the anion-water network.^{[15],[46]} Now, we turn to the anion-water network itself. The RDFs of this network decomposes into two classes: $\mathcal{W}\text{-}\mathcal{W}$ and $\mathcal{A}\text{-}\mathcal{W}$ look similar and show a pronounced peak at 2.8 and 3.7Å, respectively. This peak can be exclusively attributed to the first Voronoi-shell. It increases with decreasing water content which is in accordance with simulation results of dimethylimidazolium chloride mixtures with water.^[47] Beyond this peak the local density is very similar to the global one. The common feature of both interaction pairs is the dipolar interaction, where in case of $\mathcal{A}\text{-}\mathcal{W}$ the anionic dipole is induced by the distortion of the tetrahedral geometry of BF_4^- . This is a consequence of the strong hydrogen bond acceptor role of the anion which is also reflected in a strong negative peak of $g^{011}(r)$

of anion and water (data not shown). On contrast, the $\mathcal{A}\text{-}\mathcal{A}$ RDF is dominated by charge-charge interaction as well as by the larger effective radius of the anion. Consequently, the first Voronoi peak

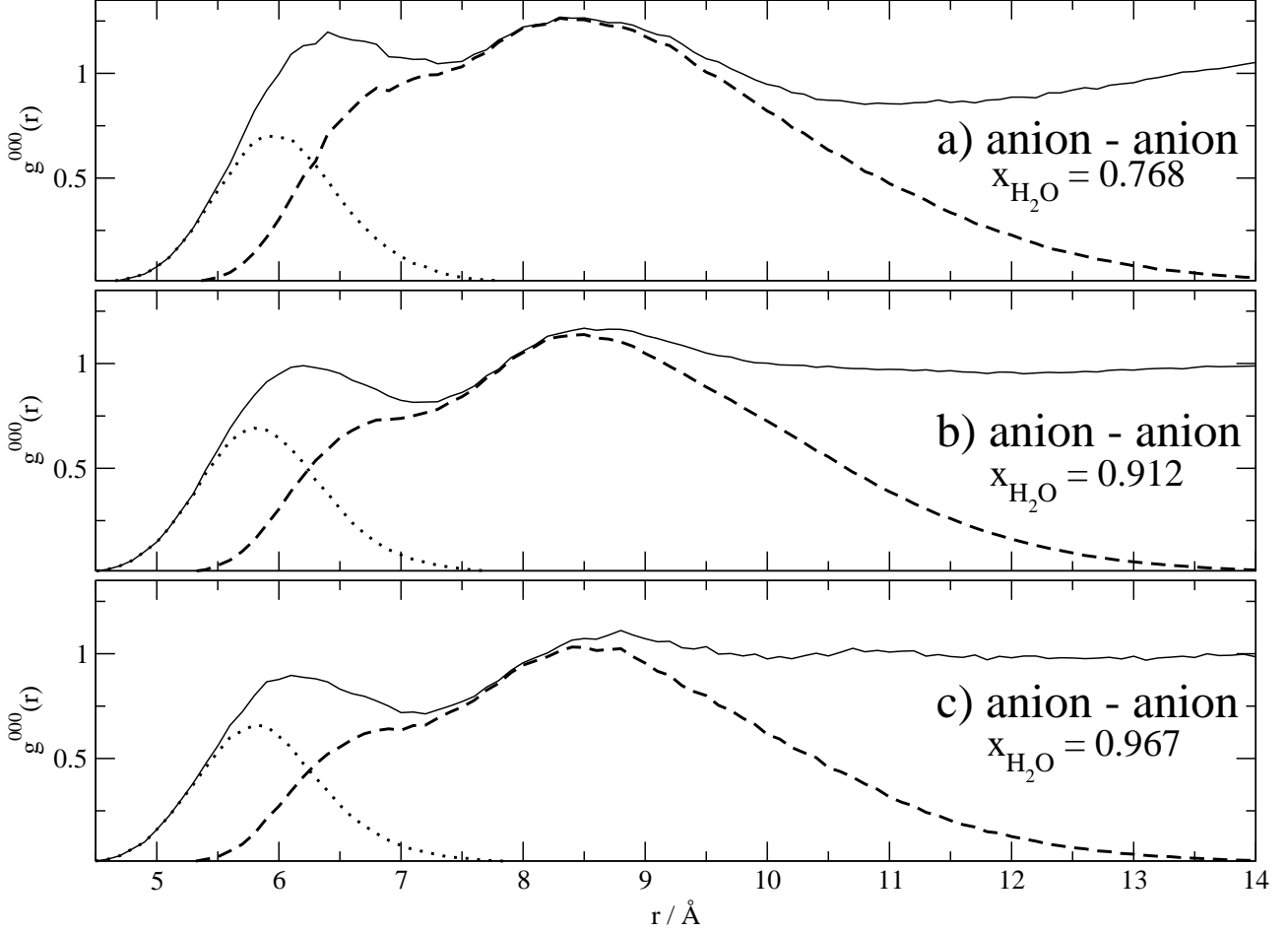


Figure 2.10: Local density of the anions near anions at different $x_{H_2O} = 0.768$ (a), 0.912 (b) and 0.967 (c). Additionally, the first (dotted) and second (dashed) Voronoi shells are visible.

in Fig.2.10a-c is shifted to distances around 5.9\AA . Moreover, the position and the height of this peak does not change with x_{H_2O} . The repulsion of the anions strongly reduces the possibility for direct neighbours. Hence, the first Voronoi peak achieves only 80% of the global density, even the second Voronoi peak exceeds the global density by a mere of 20%. This second peak in Fig.2.10a-c exhibits a double shape which can be attributed to an interstitial neighbour. At short distances, this neighbour is probably water whereas at longer distances cations are possible, too. The double shape of the second peak is also responsible for the peak structure of the overall RDF. While its peak constitutes the second peak of the overall RDF, its shoulder together with the first Voronoi peak gives as a superposition the first peak of the overall RDF which resembles the corresponding $g^{000}(r)$ of NO_3^- in a

simulation of a 1-octyl-3-methyl-imidazolium nitrate/water mixture.^[31]

The first negative peak in $g^{101}(r)$ of $\mathcal{W}\text{-}\mathcal{W}$ occurs at 2.8\AA , i.e. at the same distance as the corresponding RDF $g^{000}(r)$. As this represents the shortest possible oxygen–oxygen distance a neighbouring water molecule can only attack the oxygen site of the reference water and not its hydrogens. The dipole of the reference water points in the direction of the above mentioned hydrogens. Therefore, the angle between this dipole vector and the distance vector pointing to the neighbour waters is almost 180° resulting in the negative sign of the peak in $g^{101}(r)$ (region 3). At larger distance but still in the first Voronoi shell the region around the hydrogens is populated. Consequently, the sign of $g^{101}(r)$ is reversed. This reversal in sign within the very same Voronoi shell can be observed in all three Voronoi shells.

Altogether, the features found and analysed in $\mathcal{A}\text{-}\mathcal{A}$, $\mathcal{A}\text{-}\mathcal{W}$ and $\mathcal{W}\text{-}\mathcal{W}$ correlations are indicative for a strong anion–water network. This network strengthens with decreasing water content. In other words, in the competition of the ions for water, the anion strongly overrules the cations thus expelling the cations from the anion–water network. The cations organise themselves in a hydrophobic $\mathcal{T}\text{-}\mathcal{T}$ network. Consequently, the coordination number of BMIM^+ around BMIM^+ increases from 2.5 at $x_{\text{H}_2\text{O}}=0.967$ to 9.4 at $x_{\text{H}_2\text{O}}=0.768$. This effect is not caused by the stoichiometry since the coordination number of the anions increases much slower from 2.1 to 5.6 in Table 2.2 at the above mentioned water contents. This clustering of tails was already found in the beginning of this discussion, but now we observe even an indirect influence on $\mathcal{H}\text{-}\mathcal{H}$ and $\mathcal{R}\text{-}\mathcal{R}$ correlations. This can be directly seen in Fig. 2.4b for the first Voronoi shell of the $\mathcal{R}\text{-}\mathcal{R}$ packing. The peak raises from 20% to 45% of the global density with decreasing water content and its position is shifted to shorter distances. This behaviour may be interpreted as ring–ring stacking. This stacking is not restricted to mixtures between ionic liquids and water but can be also found in simulations of pure ionic liquids.^{[6],[7]}

2.3.4 Coordination numbers

Traditionally, the coordination number $c.n.$ is calculated by integrating the RDF over spherical shells up to a certain threshold r_{cut} :

$$c.n. = 4\pi \frac{N_s}{V} \int_0^{r_{cut}} g^{000}(r) r^2 dr \quad (2.2)$$

where N_s is the number of molecules of species s and V the total volume of the sample. The basic problem of this procedure is the choice of the threshold r_{cut} : In case of several species, r_{cut} has to be defined for each possible pair. For example, Ref. [48] operates with five r_{cut} in the range from 3.804\AA

to 5.551Å. Even the three imidazolium hydrogens have a spread of 1.2Å. These values are defined as the first minimum of the respective $g^{000}(r)$. Usually the distance spread of Voronoi shells is larger than 5Å as visible in the Fig. 2.4, 2.6, 2.10 and 2.11. As a result, the coordination numbers derived from them are bigger than those computed by the integration of the radial distribution function up to a more or less arbitrary limit.

In this paper we have found numerous examples rendering the r_{cut} -method questionable: First, the first shell does not need to coincide with the first peak in $g^{000}(r)$, e.g. \mathcal{R} - \mathcal{R} in Fig.2.4b. Second, the overlap of two shells on a radial scale may create superposition peaks, e.g. Fig.2.10a-c. Third, in extreme cases the minimum of $g^{000}(r)$ coincides with the maximum of a shell, e.g. the third Voronoi shell in Fig.2.4b. Fortunately, the Voronoi decomposition provides coordination numbers without any further calculation by simply counting the members of the first Voronoi shell. Therefore, it is free of integration errors due to low statistics and/or inappropriate integration limits. Of course, if one applies Eq. 2.2 to the $g^{000}(r)$ of the first Voronoi shell, the very same Voronoi *c.n.* is obtained.

The difference of *c.n.* derived by Voronoi decomposition and r_{cut} -method can be demonstrated for the \mathcal{W} - \mathcal{W} correlation. A threshold of $r_{cut} = 3.5\text{\AA}$ representing the minimum of oxygen-oxygen RDF is well established in the literature.^{[30], [47], [49], [50]} However, the first Voronoi shell for all mixtures $x_{H_2O} = 0.768, 0.912$ and 0.967 is depicted in Fig.2.11a-c as dotted line. The first peak of $g^{000}(r)$ is exclusively made up by the first Voronoi shell, but this shell does not end at 3.5Å. It has a shoulder gradually declining up to 6Å. The second shell starts at approximately 4Å. The traditional threshold r_{cut} ensures that no indirect neighbours are taken into account but misses some direct neighbours: The *c.n.* by means of r_{cut} are 2.0, 3.2 and 3.8 for $x_{H_2O} = 0.768, 0.912$ and 0.967 , respectively. The Voronoi method (including the shoulder) gives alternative values of 4.3, 8.4 and 12.2 in Table 2.2. It is clear that water molecules beyond 3.5Å are not “hydrogen-bonded” to the reference. Nevertheless, they are direct neighbours filling up the free spaces left by the “hydrogen-bonded” neighbours in order to complete the first shell. Although not “hydrogen-bonded” these space-filling direct neighbours still have strong electrostatic interactions with the reference water due to high value of the local partial charges. We emphasise that the used term “hydrogen-bonded” is solely defined on a distance criterion and implies no preferred orientations between water molecules which is normally necessary to completely define an hydrogen bond.

The normalization of *c.n.* by N_s is better suited than the absolute *c.n.* for the comparison of direct neighbourhood and RDFs because the influence of the stoichiometry is ruled out as can be seen in Eq. 2.2. For well equilibrated and sufficiently sampled mixtures a constant ratio $c.n./N_s$ should be

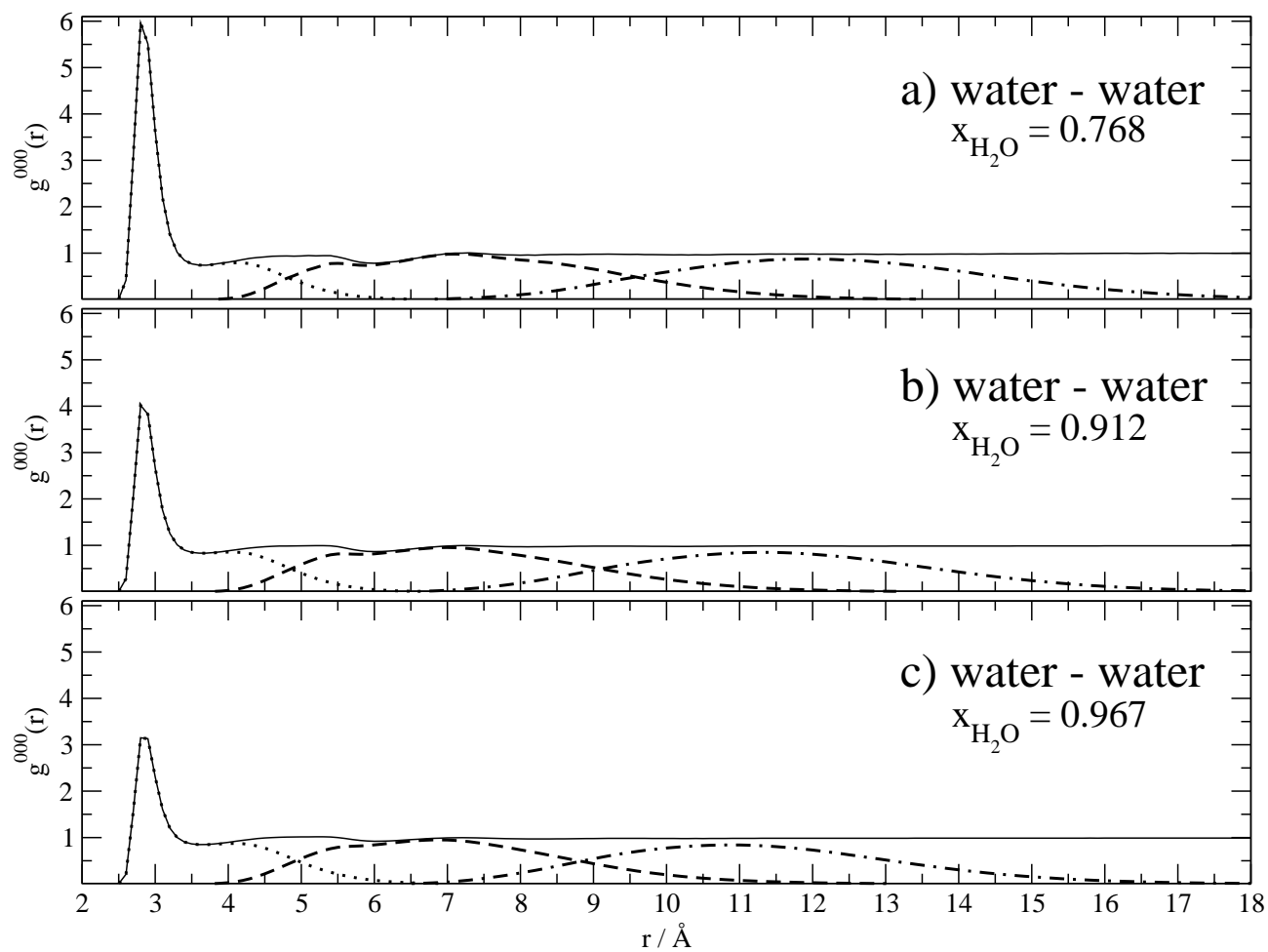


Figure 2.11: Local density of the water near water at different $x_{\text{H}_2\text{O}} = 0.768$ (a), 0.912 (b) and 0.967 (c). Additionally, the first (dotted) and second (dashed) Voronoi shells are visible. The first Voronoi shell lasts longer than the first minimum of the overall $g^{000}(r)$.

achieved. In other words, global particle ratios should be also found on a local scale. Dividing the $c.n.s$ by the total number of water molecules, the Voronoi $c.n.$ values are indeed fairly constant while the r_{cut} values decrease considerably. This rather different behaviour comes from the gradual transformation of “hydrogen-bonded” neighbours to space-filling direct neighbours with increasing water content. In Fig. 2.11a-c a decrement of the first peak height is more or less perfectly balanced by an increment of the accompanying shoulder. With increasing water content the stabilizing effect of the ionic field is weakened. Consequently, “hydrogen bonding” is destabilised in some cases. Nevertheless, the respective water molecules remain direct neighbours.

Table 2.2 gives the absolute $c.n.$ for all pairs of species in order to be comparable with literature data. For example, the cation-anion $c.n.$ of 5.6 at $x_{H_2O}=0.768$ corresponds well to the value reported in Ref. [45] in case of $EMIM^+BF_4^-$. There, the coordination number of six was attributed to a somehow dynamical coordination mechanism. The static quantum-mechanical calculations in Ref. [45] suggest a $c.n.$ of four BF_4^- near the cation which was found by a molecular dynamics study as well.^[46] On the one hand, the higher $c.n.$ in our simulation may be due to our cation $BMIM^+$ which is significantly larger than $EMIM^+$. On the other hand, water is also present in vicinity of the cation. Therefore, repulsive forces of two coordinating anions may be screened by interstitial waters. However, as discussed above, our main interpretation is based on the ratio of $c.n./N_s$. This has the additional benefit that the mutual $c.n.(AB)/N_B$ equals $c.n.(BA)/N_A$. Furthermore, these ratios are rather independent of the water mole fraction x_{H_2O} and comparable to the g -coefficients. From the cations point of view the preferred neighbour is also a cation in accordance with the hydrophobic association of tails. The anionic ratio is approximately 30% higher compared to water. This is to be expected from electrostatic interactions as well as the repulsion of cations from the anion–water network. Since the simpler anions lack an hydrophobic part, their mutual electrostatic repulsion strongly reduces the $c.n./N_s$ ratio. Analogously, electrostatic attraction leads to the highest ratio with anions as reference sites. In case of water as reference site, many neighbours exceed by far the reference water in size. As a result, the favorite position of neighbours changes drastically and may explain the correlation of the $c.n./N_s$ ratios with the square of the peak position of the respective $g^{000}(r)$ of the first Voronoi shell. Therefore, we favor in this case the interpretation in terms of $g^{000}(r)$ rather than mere coordination numbers.

2.4 Conclusion

The Voronoi decomposition is a rational concept of proximity or neighbourhood. It can be applied to a variety of analyses either spatial or temporal. In this pilot study we have combined the Voronoi decomposition with structural g -coefficients: In case of rather disjoint Voronoi shells, the peak pattern of the respective radial distribution function $g^{000}(r)$ is in accordance with the sequence of Voronoi shells. In other words, the first peak of $g^{000}(r)$ mainly consists of molecules in the first Voronoi shell, the second peak is constituted by the second Voronoi shell and so on ... An example for this case is the tail-tail distribution of the cations in this work. In other cases, peaks in the radial distribution function are a superposition of different Voronoi shells. Hence, particles located around this distance are of hybrid nature incorporating two different neighbourhoods, e.g. the first peak of the anion-anion radial distribution. In extreme cases, the first peak of the radial distribution does not represent the direct neighbours of the reference site, but their neighbours, i.e. the second Voronoi shell. A typical example is the ring-ring distribution. This has severe consequence for the interpretation of 3D occupancy plots which are based on a certain threshold. If this threshold is above the maximum of the first Voronoi shell the occupancy plot show only indirect neighbours and misses the direct ones. Lowering the threshold blurs the picture as it mixes direct and indirect neighbours.

Besides these general findings exceeding traditional methods the combined Voronoi- g -coefficient analysis reveals the well established features of ionic liquid/water mixtures: The hydrophobicity of the tails, the enhanced presence of water and anions in the proximity of the ring as well as the existence of a strong anion–water network. In a more detailed view, the first Voronoi shell plays a special role trying to compensate the major part of anisotropy of the reference site. A typical example are water molecules in the direct neighbourhood of a water molecule. Some of these direct neighbours are “hydrogen-bonded” whereas others complete the first shell to a more isotropic body. Hence, the first Voronoi shell is most often characterised by an uniform sign of the respective $g^{011}(r)$ and $g^{101}(r)$ -coefficients. Despite their strong coupling in an anion–water network both species compete for favorable positions in the proximity of the imidazolium hydrogens. While water seems to be restricted to the H_2 region, the anion is more ubiquitous. The most important feature of our coordination number analysis is the independence of the ratio $c.n./N_s$ of the water mole fraction.

Acknowledgement

This work was supported by the project P19807 of the FWF Austrian Science Fund. Furthermore, we would like to thank the Institute of Scientific Computing at the university of Vienna for a generous allocation of computer time.

Bibliography

- [1] P. Debye and E. Hückel, *Physik. Z.* **24**, 185 (1923).
- [2] C. G. Hanke, S. L. Price, and R. M. Lynden-Bell, *Mol. Phys.* **99**, 801 (2001).
- [3] J. K. Shah, J. F. Brennecke, and E. J. Maginn, *Green Chem.* **4**, 112 (2002).
- [4] M. G. Del Pópolo and G. A. Voth, *J. Phys. Chem. B* **108**, 1744 (2004).
- [5] M. G. Del Pópolo, R. M. Lynden-Bell, and J. Kohanoff, *J. Phys. Chem. B* **109**, 5895 (2005).
- [6] B. L. Bhargava and S. Balasubramanian, *Chem. Phys. Lett.* **417**, 486 (2006).
- [7] N. M. Micaelo, A. M. Baptista, and C. M. Soares, *J. Phys. Chem. B* **110**, 14444 (2006).
- [8] O. Borodin and G. D. Smith, *J. Phys. Chem. B* **110**, 11481 (2006).
- [9] A. Bagno, F. D'Amico, and G. Saielli, *J. Mol. Liquids* **131-132**, 17 (2007).
- [10] V. Bitrian and J. Trullas, *J. Phys. Chem. B* **110**, 7490 (2006).
- [11] W. Jiang, T. Yan, Y. Wang, and G. A. Voth, *J. Phys. Chem. B* **112**, 3121 (2008).
- [12] F. Aurenhammer, *ACM Computing Surveys* **23**, 345 (1991).
- [13] A. Okabe, *Spatial tessellations: concepts and applications of Voronoi diagrams* (Wiley, New York, 2000).
- [14] M. Malvaldi and C. Chiappe, *J. Phys. Condens. Matter* **20**, 035108 (2008).
- [15] C. Schröder, T. Rudas, G. Neumayr, S. Benkner, and O. Steinhauser, *J. Chem. Phys.* **127**, 234503 (2007).
- [16] C. Schröder, J. Hunger, A. Stoppa, R. Buchner, and O. Steinhauser, *J. Chem. Phys.* **129**, 184501 (2009).

Bibliography

- [17] K. E. Thompson, *Int. J. Numer. Meth. Engng.* **55**, 1345 (2002).
- [18] D. F. Watson, *The Computer Journal* **24**, 167 (1981).
- [19] M. Gerstein, J. Tsai, and M. Levitt, *J. Mol. Biol.* **249**, 955 (1995).
- [20] G. De Fabritiis and P. V. Coveney, *Comp. Phys. Commun.* **153**, 209 (2003).
- [21] H. Borouchaki, P. L. George, F. Hecht, P. Laug, and E. Saltel, *Finite Elements in Analysis and Design* **25**, 61 (1997).
- [22] H. Borouchaki and S. H. Lo, *Comput. Methods Appl. Mech. Engng.* **128**, 153 (1995).
- [23] I. M. Svishchev, P. G. Kusalik, J. Wiang, and R. J. Boyd, *J. Chem. Phys.* **105**, 4742 (1996).
- [24] P. G. Kusalik and I. M. Svishchev, *Science* **265**, 1219 (1994).
- [25] I. M. Svishchev and P. G. Kusalik, *J. Chem. Phys.* **99**, 3049 (1993).
- [26] O. Steinhauser, *Ber. Bunsenges. Phys. Chem.* **87**, 128 (1983).
- [27] A. J. Stone, *Mol. Phys.* **36**, 241 (1978).
- [28] C. Schröder, T. Rudas, G. Neumayr, W. Gansterer, and O. Steinhauser, *J. Chem. Phys.* **127**, 044505 (2007).
- [29] C. Schröder, T. Rudas, and O. Steinhauser, *J. Chem. Phys.* **125**, 244506 (2006).
- [30] C. Schröder, T. Rudas, S. Boresch, and O. Steinhauser, *J. Chem. Phys.* **124**, 234907 (2006).
- [31] W. Jiang, Y. Wang, and G. A. Voth, *J. Phys. Chem. B* **111**, 4812 (2007).
- [32] C. J. Margulis, H. A. Stern, and B. J. Berne, *J. Phys. Chem. B* **106**, 12017 (2002).
- [33] T. I. Morrow and E. J. Maginn, *J. Phys. Chem. B* **106**, 12807 (2002).
- [34] J. N. A. C. Lopes and A. A. H. Padua, *J. Phys. Chem.* **110**, 3330 (2006).
- [35] L. J. A. Siqueira and M. C. C. Ribeiro, *J. Phys. Chem. B* **111**, 11776 (2007).
- [36] J. N. Canongia Lopes, J. Deschamps, and A. A. H. Padua, *J. Phys. Chem. B* **108**, 2038 (2004).
- [37] J. N. Canongia Lopes, J. Deschamps, and A. A. H. Padua, *J. Phys. Chem. B* **108**, 11250 (2004).

- [38] J. de Andrade, E. S. Böes, and H. Stassen, *J. Phys. Chem. B* **106**, 13344 (2002).
- [39] W. L. Jorgensen, *J. Am. Chem. Soc.* **103**, 335 (1981).
- [40] T. Darden, D. York, and L. Pedersen, *J. Chem. Phys.* **98**, 10089 (1993).
- [41] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, *J. Chem. Phys.* **103**, 8577 (1995).
- [42] J.-P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, *J. Comput. Phys.* **23**, 327 (1977).
- [43] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, and S. Swaminathan, *J. Comput. Chem.* **4**, 187 (1983).
- [44] Z. Liu, S. Huang, and W. Wang, *J. Phys. Chem. B* **108**, 12978 (2004).
- [45] P. A. Hunt, I. R. Gould, and B. Kirchner, *Aust. J. Chem.* **60**, 9 (2007).
- [46] A. R. Porter, S. Y. Liem, and P. L. A. Popelier, *Phys. Chem. Chem. Phys.* **10**, 4240 (2008).
- [47] C. G. Hanke and R. M. Lynden-Bell, *J. Phys. Chem. B* **107**, 10873 (2003).
- [48] C. Spickermann, J. Thar, S. B. C. Lehmann, S. Zahn, J. Hunger, R. Buchner, P. A. Hunt, T. Welton, and B. Kirchner, *J. Chem. Phys.* **129**, 104505 (2008).
- [49] V. Vchirawongkwin, B. M. Rode, and I. Persson, *J. Phys. Chem. B* **111**, 4150 (2007).
- [50] A. E. Garcia and L. Stiller, *J. Comput. Chem.* **14**, 1396 (2004).
- [51] C. Schröder, C. Wakai, H. Weingärtner, and O. Steinhauser, *J. Chem. Phys.* **126**, 084511 (2007).

3 Relaxation of Voronoi shells in hydrated molecular ionic liquids

G. Neumayr, C. Schröder and O. Steinhauser J. Chem. Phys. 131, 174509 (2009)

The relaxation of solvation shells is studied following a twofold strategy based on a direct analysis of simulated data as well as on a solution of a Markovian master equation. In both cases solvation shells are constructed by Voronoi decomposition or equivalent Delaunay tessellation. The theoretical framework is applied to two types of hydrated molecular ionic liquids, 1-butyl-3-methyl-imidazolium tetrafluoroborate and 1-ethyl-3-methyl-imidazolium trifluoromethylsulfonate, both mixed with water. Molecular dynamics simulations of both systems were performed at various mole fractions of water.

A linear relationship between the mean residence time and the system's viscosity is found from the direct analysis independent of the system's type. The complex time behaviour of shell relaxation can be modelled by a Kohlrausch-Williams-Watts function with an almost universal stretching parameter of $1/2$ indicative of a square root time law. The probabilistic model enables an intuitive interpretation of essential motional parameters otherwise not accessible by direct analysis. Even more, incorporating the square root time law into the probabilistic model enables a quantitative prediction of shell relaxation from very short simulation studies. In particular, the viscosity of the respective systems can be predicted.

3.1 Introduction

Decomposition of space is a technique frequently used and necessary in a widespread area of applications covering fields as diverse as natural science, engineering and logistics. A classical tool common to all these areas is the method of Voronoi diagrams^[1] and its dual technique Delaunay tessellation.^[2] In this latter formulation it was applied to mathematical crystallography and thus found its way into

molecular sciences. When applied to the position of atoms Voronoi decomposition associates with each atom that portion of space which is closer to this specific atom than to any other. The shape of this individual volume is a polyhedron. In this way, whole space appears as a collection of irregular honeycombs. In this view neighbourhood is naturally defined by the criterion that two atoms are immediate neighbours if they share a common Voronoi face. The neighbours of the immediate neighbour constitute the second neighbours and so on. Thus, the unique Delaunay tessellation lends itself to a classification of neighbourhood without the use of any parameters. This is in contrast to distance-based definitions of neighbourhood. Uniqueness and the lack of any parameters have made Voronoi decomposition a widely used tool in molecular science. Quite recent examples are given in Refs. [3]–[7].

In the special field of solvation the definition of solvation shells also profits from Voronoi decomposition. Instead of using a somewhat arbitrary set of parameters within a distance-based definition, Delaunay tessellation enables a unique and parameter free classification of solvation shells. This advantage has been extensively exploited in a series of studies performed in our group.^{[8]–[14]} Thereby we could also establish a link to experimental studies of shell specific relaxation as described in detail in the review of Halle.^[15]

Up to now we have focussed on the direct Voronoi analysis of data generated by computer simulation. Due to the vast amount of data to be analysed, alternative ways of interpretation are quite welcome. The present study is based on a probabilistic model which try to mimic molecular motion as a solution of relaxation equations. The two most frequently used classes of models rely either on Markovian processes or on a Master equation or on hybrids of both methods. Typical examples for the combination of simulation based studies and probabilistic models are found in Refs. [16], [17]. Eijnden and Venturoli^[17] combine the technique of milestoning with probabilistic models, while Buchete and Hummer^[16] apply them to peptide folding dynamics. In this paper we develop a probabilistic model providing an interpretation of the relaxation of Voronoi solvation shells. The twofold analysis of shell relaxation is applied to two types of hydrated molecular ionic liquids, 1-butyl-3-methyl-imidazolium tetrafluoroborate and 1-ethyl-3-methyl-imidazolium trifluoromethylsulfonate, both mixed with water. Basic data were extracted from Molecular dynamics simulations of both systems performed at various mole fractions of water.

3.2 Theory

3.2.1 Molecular Dynamics of Voronoi Shells

Tessellation and Voronoi Shells

Given a non-degenerate set of points, a Voronoi decomposition creates an ensemble of space-filling disjunct polyhedra, each containing all space closer to its associated point than to any other point of the given set. The faces of each of these Voronoi polyhedra are constructed by planes perpendicular to the vectors between the associated point and its neighbour points. However, there exists a geometrical duality between this Voronoi decomposition into polyhedra and Delaunay tessellation. In principle, a Delaunay tessellation is a unique partitioning of the point set into “simplices”.^[18] In three dimensional space such a simplex is an irregular tetrahedron and its four vertices are among the set of points under investigation. These four vertices of a tetrahedron lie on the surface of a circumscribed sphere which does not contain any further point. In short this is known as the “Delaunay criterion”. The edges of the tetrahedra are the vectors connecting two points being cut by the orthogonal Voronoi faces midway. A schematic view on the duality of Voronoi decomposition and Delaunay tessellation in two dimensions is given in Fig.3.1.

The direct construction of Voronoi polyhedra is computationally demanding. Therefore, the dual and much simpler Delaunay tessellation has been used extensively in the design of efficient algorithms for interpolation, contouring or mesh generation.^[19] In the literature a plethora of tessellation algorithms can be found,^{[20]–[23]} but in computer simulations of bulk media periodicity is commonly used to emulate infinite systems. Consequently, Delaunay algorithms^{[19],[22]} taking explicitly into account periodicity are of special importance. Thereby, periodicity is an essential part of the algorithm which excludes algorithms working on the primary cell enhanced by its explicit images. Among the periodic Delaunay algorithms “insertion algorithms” have proven particularly successful and were adopted in various ways.^{[19],[20],[24]} These kind of iterative algorithms insert one point at a time into the current tessellation the sequence of points being arbitrary.

In the final tessellation of a trajectory frame the Delaunay distance d is used to define Voronoi shells S_d . Thereby, the reference particle itself is denoted by S_0 to be formally consistent with a zero Delaunay distance. Here, the term Delaunay distance is meant in a graph-theoretical sense and measures the length of the shortest path between two vertices in the tessellation. A direct or “nearest neighbour” of Delaunay distance $d = 1$ may be alternatively defined as a vertex which shares at least

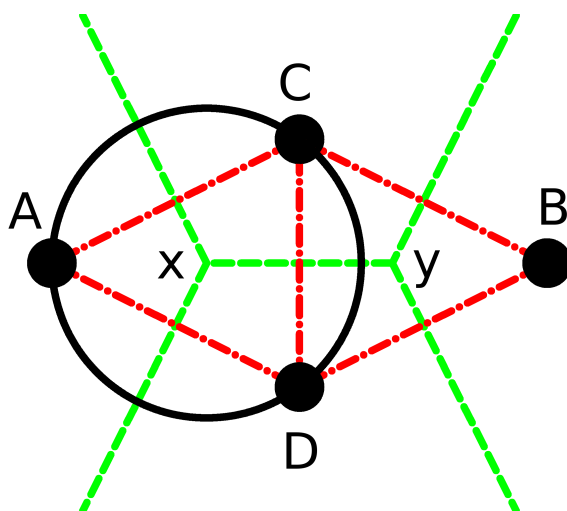


Figure 3.1: Duality of Voronoi- and Delaunay tessellations. In this simplistic scheme, four points (A,B,C,D) are connected via their Delaunay (red, dash-dotted) and Voronoi (green, dashed) diagrams. The two diagrams are dual, i.e. can be converted into each other without any further information. By connecting all points that share a common Voronoi-face, the Delaunay diagram can be drawn. For example, points C and D share the common Voronoi face x-y (green) and are thus connected in the Delaunay diagram (red). The Voronoi nodes (x,y), in turn, coincide with the centers of Delaunay circumcircles. Given a Delaunay tessellation, the Voronoi nodes, e.g. x and y in the figure, are obtained in the following way: In two dimensions for each triangle, e.g. A-C-D, a circumcircle can be constructed. Its center coincides with the vertex x of a Voronoi polygon. In the three dimensional case a Voronoi vertex is identical to the center of the Delaunay tetrahedron's circumsphere.

one simplex with the reference vertex. The set of all these neighbouring vertices constitutes the first voronoi shell S_1 of the reference vertex. The second shell S_2 is comprised of all vertices that are in the first shell of S_1 . In the Voronoi picture two polyhedra are first neighbours if they have one common face. Again, the first shell comprises all these first neighbours and subsequent shells are defined as the set of neighbours of the preceding shell. In Fig.3.1 the minimal distance corresponds to the minimal number of lines connecting two particles symbolised as filled circles.

Timeseries and Time Correlation Functions

As shells are not static but evolve in time we have introduced a binary residence function for a single particle i

$$n_i^d(t) = \begin{cases} 1 & : i \in S_d \\ 0 & : i \notin S_d \end{cases} \quad (3.1)$$

depending on whether particle i is a member of shell S_d or not at time t . We emphasise that the molecular identity is conserved, i.e. reoccurrence of a particle is not recognised as the entrance of a new particle. Here, $\{n_i^d(t)\}$ is a time series characteristic of shell dynamics. In order to get a concise measure of such time series one traditionally uses time correlation functions (TCF). For the present study the essential correlation function is given by

$$C_n^d(t) = \frac{\sum_{i=1}^N \langle n_i(0)n_i(t) \rangle}{CN} \quad (3.2)$$

One term in the above sum represents the memory of a specific particle to be still a member of shell S_d after some time interval t . Summation over all particles N surrounding the reference molecule is done for statistical accuracy. In other words, $C_n^d(t)$ describes the memory of the average particle. The normalizing factor also called coordination number $CN = \sum_{i=1}^N \langle n_i^2(0) \rangle$ gives the average number of particles which are member of the shell S_d . As opposed to many correlation functions which decay to zero in the long-time limit $C_n^d(t)$ approaches a steady state $\lim_{t \rightarrow \infty} C(t) = C_{\text{steady}}$.

$$C_n^d(t) = \{C_n^d(t) - C_{\text{steady}}\} + C_{\text{steady}} \quad (3.3)$$

The characteristic time to reach this steady state is given by the average correlation time

$$\langle \tau \rangle = \int_0^\infty \{C_n^d(t) - C_{\text{steady}}\} dt \quad (3.4)$$

which may be interpreted as a mean residence time (MRT).

Representation of TCF

Quite generally,^[25] TCFs $C(t)$ obey

$$\frac{dC}{dt} = - \int_0^t K(t')C(t-t')dt'. \quad (3.5)$$

For a Markovian or δ -memory kernel

$$K(t') = \lambda\delta(t') \quad (3.6)$$

Eq. (3.5) leads to the special case of a mono-exponential correlation function $C(t) = Ae^{-\lambda t} = Ae^{-\frac{t}{\tau}}$, typical for simple systems. In order to cope with complex dynamics the concept of mono-exponential relaxation is usually generalised to a distribution $g(t')$ of relaxation times

$$C(t) = \int g(t')e^{-\frac{t}{\tau'}}dt'. \quad (3.7)$$

Its discrete version is a superposition of exponential relaxations

$$C(t) = \sum_i A_i e^{-\frac{t}{\tau_i}}. \quad (3.8)$$

From the general definition of Eq. (3.4) the average correlation time may be easily computed in this case

$$\langle \tau \rangle = \frac{\int_0^\infty C(t)dt}{C(0)} = \frac{\sum_i A_i \tau_i}{\sum_i A_i} \quad (3.9)$$

The individual relaxation times τ_i may be either obtained by various fitting methods or more systematically by inverse Laplace transform.^{[26],[27]}

A frequently used representation of complex relaxation behaviour is the stretched exponential or Kohlrausch-Williams-Watts (KWW) function

$$C(t) = A_1 + A_2 e^{(-\frac{t}{\tau})^\beta} \quad (3.10)$$

which introduces a single additional parameter β .^{[28],[29]} Concrete examples for various β -values are shown in Fig.1 of Ref.[30]. For special values of β an analytical expression for this distribution can be given. In particular for $\beta = 1/2$ one gets

$$g(t') = \frac{e^{-\frac{t'}{4\tau}}}{2\sqrt{\pi\tau t'}}. \quad (3.11)$$

From this special case one can see that the distribution of relaxation times is modelled by a single parameter β . A smaller value of β corresponds to a broader distribution. This mapping of a complex

dynamical behaviour to a single parameter is the main reason for the wide-spread use of KWW functions. In addition, a simple analytic formula holds for the average correlation time

$$\langle \tau \rangle = \frac{\tau}{\beta} \Gamma\left(\frac{1}{\beta}\right) \quad (3.12)$$

where Γ denotes the Gamma-function representing the generalization of the factorial to real numbers.

3.2.2 Probabilistic Approach

Markovian Master Equation

So far we have deduced TCFs from the constitutive equation (3.5) based on statistical mechanical averaging over a trajectory of explicit molecular motions. Alternatively to this averaging over molecular dynamical processes, one may study the temporal behaviour of the corresponding "average process" modelled by transitions between different states. These states are defined by the subdivision of participation in a solvation shell according to the proximity to a specific site of the reference molecule as illustrated in Fig.3.2. Therefore, the state space is the collection of labels or numbers of sites $\{1,2,3,...,M\}$.

The analogue of explicit molecular motion is now a walk in state space described by a random variable $X(t)$ specifying the actual state at time t . Like the explicit molecular motion, the sequence or "trajectory" of visited states along this walk, has a memory as well. In other words, the probability \Pr of transition from a state i to a state j at time t depends on preceding transitions.

$$\Pr(i \rightarrow j | t = n \cdot \Delta t) = \Pr(X_n = j | X_{n-1} = i, X_{n-2} = k_{n-2}, \dots, X_0 = k_0) \quad (3.13)$$

A binary or first order Markov chain ignores the complete history prior to the transition from i to j , i.e. all states labelled k_m are omitted. This offers the possibility of a matrix notation with elements

$$W_{ij} = \Pr(i \rightarrow j | t = n \cdot \Delta t) = \Pr(X_n = j | X_{n-1} = i) \quad (3.14)$$

This transition matrix W_{ij} is the essential device for the generation of walks in state space. When averaging over an ensemble of walks, the population of states may be characterised by their probability $p_i(t)$ at time t . The time evolution of this probability in terms of the transition rates W_{ij} within a time interval Δt is given by

$$p_i(t + \Delta t) = p_i(t) \left(1 - \Delta t \sum_j W_{ij} \right) + \Delta t \sum_{j \neq i} W_{ji} p_j(t). \quad (3.15)$$

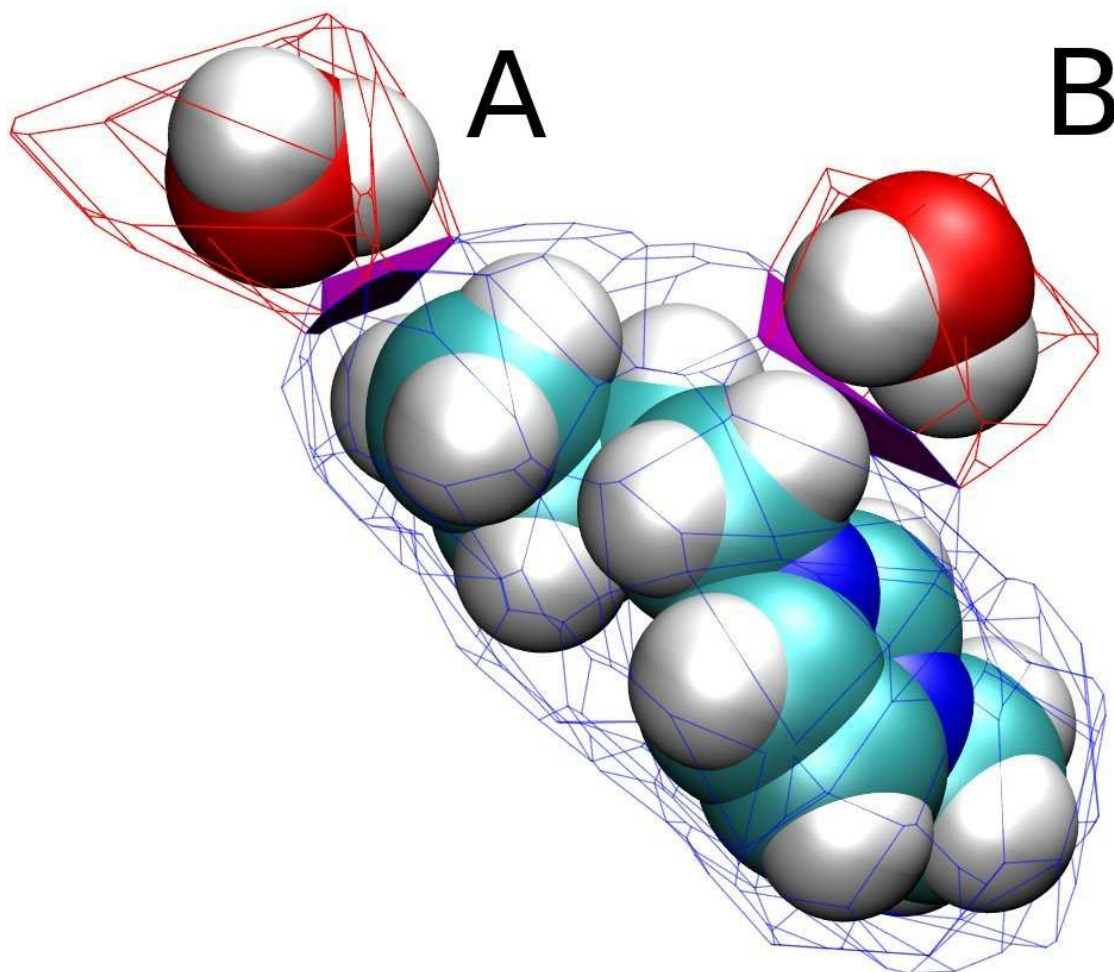


Figure 3.2: Proximity criterion. The two water molecules are both members of the first shell of the BMIM molecule, but are assigned two different states. The distance between water molecule A and the BMIM molecule is shortest at the butyl-terminal methyl group, thus molecule A is populating state seven of an eight state C-W matrix. Water molecule B is populating state four, as its proximity is maximal with respect to butyl hydrogen H_7 .

Assuming a continuous-time Markov jump process^[17] we can take the limit $\lim_{t \rightarrow 0}$ transforming the above rate equation to a differential equation

$$\frac{dp_i(t)}{dt} = \sum_{j \neq i} W_{ji} p_j - \left(\sum_j W_{ij} \right) p_i \quad (3.16)$$

where the right part of the equation is the balance of incoming and outgoing transitions. Introducing the row sum

$$Z_i = \sum_j W_{ij} \quad (3.17)$$

equation (3.16) may be rewritten as

$$\frac{dp_i(t)}{dt} = \sum_j (W_{ji} - Z_i \delta_{ij}) p_j \quad (3.18)$$

or finally

$$\frac{dp_i(t)}{dt} = \sum_j V_{ji} p_j. \quad (3.19)$$

As the transition matrix \mathbf{W} is a stochastic matrix with row sums $Z_i = 1$ the elements of \mathbf{V} are given by

$$V_{ji} = W_{ji} - \delta_{ij} \quad (3.20)$$

and the row sum of \mathbf{V} equals zero.

The formal solution of the Markovian master equation (3.19) is

$$\mathbf{p}(t) = e^{\mathbf{V} \cdot t} \mathbf{p}_0 \quad (3.21)$$

where we have collected the state probabilities in a vector $\mathbf{p}(t) = (p_1(t), p_2(t), \dots)$ with the initial value $\mathbf{p}(t=0) = \mathbf{p}_0$. With the aid of right $\mathbf{V} \mathbf{q}_n^R = \lambda_n \mathbf{q}_n^R$ and left $\mathbf{q}_n^L \mathbf{V} = \lambda_n \mathbf{q}_n^L$ eigenvectors Eq. (3.21) takes the explicit form^[16]

$$\mathbf{p}(t) = \sum_n \frac{\mathbf{p}_0 \cdot \mathbf{q}_n^R}{\mathbf{q}_n^L \cdot \mathbf{q}_n^R} e^{\lambda_n \cdot t} \mathbf{q}_n^L \quad (3.22)$$

This spectral expansion shows that the population of a state is a superposition of exponential functions where the relaxation times $\tau_n = 1/\lambda_n$ are the inverse of the eigenvalues of the transition matrix. As the transition matrix \mathbf{W} is stochastic, it follows from Frobenius' theorem that one eigenvalue of \mathbf{V} is zero and all others must be negative. Therefore, all exponential terms vanish after sufficiently long time and the probabilities reach a steady state.

Probabilistic analogue of TCF

In order to compare the results of explicit molecular motion we need a common level of description. In other words, we need the probabilistic analogue of the correlation function Eq. (3.2). It may be found by the following consideration:

The residence function (3.1) only discriminates between a status IN (being a member of the shell) or OUT (being not a member). The state space introduced in the probabilistic approach further discriminates with respect to proximity to the reference molecule leading to a diversification of IN states. In order to reduce this more detailed information to the binary value IN we have to sum over all populations of IN states, respectively. The OUT state, the last entry in the vector of populations $\mathbf{p}(t)$ is left unchanged. Therefore, we consider the product

$$\left\{ \sum_{i \in S_d} p_i(0) \right\} \left\{ \sum_{i \in S_d} p_i(t) \right\} \rightarrow C_n^d(t) \quad (3.23)$$

as the analogue of the residence function Eq. (3.2). Starting from a population of pure IN states

$$\sum_{i \in S_d} p_i(t=0) = \sum_{i \in S_d} p_{0,i} = 1 \quad (3.24)$$

we have in this case

$$\sum_{i \in S_d} p_i(t) \rightarrow C_n^d(t) \quad (3.25)$$

where the properties of the spectral expansion Eq. (3.22) again give a steady state.

Simple example

In order to illustrate the general principle we give an explicit calculation for the simple case of a probabilistic IN/OUT model. In other words, we abandon the diversification with respect to proximity and only refer to the status being a member of the solvation shell or not.

For a two-state model a very general form of the frequency matrix is

$$\mathbf{F} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}. \quad (3.26)$$

We have assumed a symmetric transition rate b between IN and OUT. As the migration of atomic or molecular volumes is subjected to a homeostasis the frequency matrix is essentially symmetric. In fact a typical set of 500 timesteps selected from the complete trajectory is sufficient to calculate a fairly symmetrical frequency matrix whose elements are of reliable statistical accuracy.

The entries a and c representing the frequency of residence in the IN and OUT state, however are different in general.

According to Eq. (3.32) the stochastic transition matrix \mathbf{W} is derived which in its turn gives

$$\mathbf{V} = \mathbf{W} - \mathbf{I} = \begin{pmatrix} \frac{a}{a+b} - 1 & \frac{b}{a+b} \\ \frac{b}{b+c} & \frac{c}{b+c} - 1 \end{pmatrix}. \quad (3.27)$$

The first eigenvalue of \mathbf{V} is $\lambda_1 = 0$ with the right eigenvector $\mathbf{q}_1^R = (1, 1)$ and left eigenvector $\mathbf{q}_1^L = (-\frac{b+c}{a+b}, 1)$. For the second, non-zero eigenvalue one finds $\lambda_2 = -b\frac{a+2b+c}{(a+b)(b+c)}$ with eigenvectors $\mathbf{q}_2^R = (\frac{a+b}{b+c}, 1)$ and $\mathbf{q}_2^L = (-1, 1)$. Inserting these values into Eq. (3.22) yields

$$p_1(t) = A_1 + A_2 e^{-b\frac{N}{CN(N-CN)}t} \quad (3.28)$$

$$p_2(t) = 1 - p_1(t) \quad (3.29)$$

with the amplitudes

$$A_1 = \frac{CN}{N} = \frac{a+b}{a+2b+c} \quad (3.30)$$

and

$$A_2 = \frac{N-CN}{N} = \frac{b+c}{a+2b+c}. \quad (3.31)$$

The quantities CN and N have an essential meaning: They are the coordination number and the total number of all potential neighbours.

3.3 Methods

3.3.1 Implementation of Tesselation

For the actual tessellation we adapted a periodic version of an insertion algorithm.^[19] In our case the initial tessellation is constructed from a randomly chosen point and seven of its images forming a cube. This cube is subdivided into six tetrahedra serving as initial simplices. Insertion of a new point is performed in four steps: First, the simplex containing the new point or one of its images, the so-called “base” must be found. Second, all neighbouring simplices failing the Delaunay criterion constituting the “core” are detected in a recursive manner. Third, a visibility check is performed to ensure a “star-shaped” core region. Fourth, the core tetrahedra are replaced by newly constructed ones containing the inserted point. In special cases the periodic boundary conditions necessitate a recentering of the “core” in order to ensure that every tetrahedron has at least one vertex in the primary cell. However,

the original algorithm of recentering is prone to persistence of degenerate tetrahedra steadily removed from the core by the visibility check and therefore escaping deletion. This is overcome by replacing the persistent tetrahedron whose circumsphere contains the insertion point by one of its periodic images. We found out, however, that choosing the image closest to the insertion point improves the algorithm considerably despite of the fact that its circumsphere may not contain the primary image of the insertion point. Further details about the implementation of the Voronoi algorithm may be found in Ref.[31].

Each trajectory frame was tessellated at the atomic level. A coarser graining representing the whole molecule by its center of mass was considered too crude, since essential features of anisotropy are lost in this way. As a consistency check the sum of all molecular volumes had to equal the total volume of the simulation box.

Additionally, average molecular volumina in these liquid mixtures are provided. In combination with the particle numbers, the occupancy of each species (cation, anion and water) of the total simulation box can be computed. As the mole fraction x_{H_2O} only counts particle numbers irrespective of the molecular volumes, we consider this occupancy a more intuitive measure of the composition. This is most evident in case of $x_{H_2O} = 0.768$ where the cations cover one half of the total volume leaving a quarter to water which in terms of mole fractions contributes three quarters. Seen in this way the mixtures studied are far from diluted systems as the might look at first sight.

3.3.2 Details of Simulation

Altogether, we have studied eight mixtures of molecular ionic liquids (MIL) with water: Three of them refer to 1-butyl-3-methyl-imidazolium (BMIM⁺) tetrafluoroborate (BF₄⁻) and five to 1-ethyl-3-methyl-imidazolium (EMIM⁺) trifluoromethylsulfonate (TRIF⁻). The composition of all this mixtures and the respective mole fractions of water are summarised in Table 3.1. The entries for the occupancies in this table differ slightly from those given previously in Ref.[31]. This difference comes from the degree of graining in tessellation being now fully atomistic but restricted to center of mass in Ref.[31].

The force field for the BMIM⁺ cation and the BF₄⁻ anion were taken from Ref. [32], [33] and Ref. [34]. Based on the experience of Ref. [35] the intramolecular force field parameters of EMIM⁺ are taken from Ref. [32] including the corrections.^[33] However, the partial charges of EMIM⁺ were changed to the values reported in Ref.[36] obtained by a distributed multipole analysis^[37] at the MP2/6-31G** level. This change in the charge set was motivated by the improvement of the computed value of the viscosity which was found to be a universal scaling parameter of a variety of translational and

Viscosity	η	0.9	1.4	2	2.2	4.5	6.4	7.5	22.5
Water content	x_{H_2O}	0.967	0.932	0.912	0.899	0.852	0.768	0.785	0.65
Cation	C	BMIM	EMIM	BMIM	EMIM	EMIM	BMIM	EMIM	EMIM
Anion	A	TFB	TRIF	TFB	TRIF	TRIF	TFB	TRIF	TRIF
Particle numbers	N_C, N_A	55	400	111	500	600	166	700	800
	N_{H_2O}	1629	5489	1147	4467	3444	548	2561	1504
Occupancy	$[N_C \langle V_C \rangle] / V$	21.5%	28.2%	41.1%	35.1%	42.0%	61.4%	48.2%	54.9%
	$[N_A \langle V_A \rangle] / V$	4.4%	13.9%	8.9%	17.6%	21.5%	14.2%	25.0%	29.2%
	$[N_{H_2O} \langle V_{H_2O} \rangle] / V$	74.1%	57.9%	50.0%	47.3%	36.5%	24.4%	26.8%	15.9%
Voronoi Volumina	$\langle V_C \rangle / \text{\AA}^3$	285.3	212.0	270.2	211.2	210.7	270.1	207.0	206.5
	$\langle V_A \rangle / \text{\AA}^3$	58.5	104.6	58.8	105.9	107.6	62.7	107.3	109.7
	$\langle V_{H_2O} \rangle / \text{\AA}^3$	33.2	31.7	31.9	31.8	31.9	32.5	31.5	31.8
Voronoi Surfaces	$\langle S_C \rangle / \text{\AA}^2$	283.3	228.5	275.4	228.6	229.1	277.5	227.2	227.7
	$\langle S_A \rangle / \text{\AA}^2$	96.3	139.1	95.2	139.8	140.6	97.4	140.0	141.4
	$\langle S_{H_2O} \rangle / \text{\AA}^2$	60.5	58.3	58.4	58.3	58.3	58.8	57.6	57.7
reference site	neighbour	coordination number							
cation	cation	2.5	3.2	5.5	4.5	5.8	9.4	7	8.3
	anion	2.1	3.7	3.8	4.5	5.3	5.6	6	6.8
	water	39.2	27.6	28.3	23.2	18.3	14	13.6	8
anion	cation	2.1	3.7	3.8	4.5	5.3	5.6	6	6.8
	anion	0.3	1.5	0.6	1.9	2.4	0.9	2.8	3.3
	water	15	15.5	10.6	12.7	9.8	5.3	7.2	4.3
water	cation	1.3	2	2.7	2.6	3.2	4.2	3.7	4.3
	anion	0.5	1.1	1	1.4	1.7	1.6	2	2.3
	water	12.2	9.7	8.4	7.9	6.1	4.3	4.4	2.5

Table 3.1: Composition and coordination of the simulated MIL/water mixtures.

rotational correlation times, both, collective and single particle.^[38] The anion TRIF⁻ was modelled by the force field given in Ref. [39]. The TIP3P force field parameters^[40] were used for both types of MILs, BMIM⁺BF₄⁻ and EMIM⁺TRIF⁻.

Coulomb interactions were calculated by the Particle-Mesh Ewald method,^{[41],[42]} using a 10 Å cutoff and a κ of 0.41 Å⁻¹ for the real-space part interactions. All bond lengths were kept fixed by the SHAKE algorithm,^[43] whereas bond angles and dihedrals were left flexible. Trajectories were generated with CHARMM^[44] under constant volume with a boxlength of 41.8 Å (BMIM⁺BF₄⁻H₂O) and 67 Å (EMIM⁺TRIF⁻-H₂O). The average temperature was $T = 300$ K and the average pressure close to 1 atm. The trajectories were generated with a time increment of $\Delta t = 2$ fs. The length of the BMIM⁺BF₄⁻-H₂O trajectories was 62 ns and those of EMIM⁺TRIF⁻-H₂O 30 ns.

3.3.3 Construction of Transition Matrix **W**

In principle, the transition matrix **F** was constructed from the frequencies of transition F_{ij} between states i and j . Initially, the system was decomposed into three solvation shells as described in Ref.[31].

Within each shell the proximity with respect to the reference molecule was resolved at the atomic level. This full atomistic resolution within three solvation shells gives the maximum number of states $3N+1$ where N is the number of atoms of the reference molecule used to classify proximity. The additional state stands for transitions to the bulk. To give a typical example the first solvation shell of BMIM⁺ comprises 25 states, 15 of which correspond to a proximity to the hydrogens of BMIM⁺ and show a rapid exchange as compared to the other states. Thus the initial frequency matrices are of dimension 76x76. As compared to the inner states the bulk state implicitly comprises a great number of frequently occupied OUT states.

Having the frequency matrices **F** at full resolution the granularity may be reduced by contracting selected states. For example the three hydrogens of the methyl group may be united or the methyl, ethyl or butyl side chains may be represented by a single state. Computationally this corresponds to a summation of corresponding elements F_{ij} resulting in a frequency matrix of lower dimension.

In order to convert the frequency matrix **F** to a stochastic matrix of transition probabilities **W** every row vector of **F** is normalised to unity

$$W_{ij} = \frac{F_{ij}}{\sum_j F_{ij}}. \quad (3.32)$$

The initial population was also derived from the frequency matrix

$$p_i(0) = \frac{\sum_{j=1}^M F_{ij}}{\sum_{i=1}^{M-1} \sum_{j=1}^M F_{ij}}. \quad (3.33)$$

It should be noted that the first sum in the denominator lacks the final bulk state. This comes from the following thinking: A row of the frequency matrix stands for a splitting or distribution of the population of state i to all other states which are encountered by a transition $i \leftarrow j$. In other words transitions are seen in a reversed way.

3.4 Results and Discussion

The auto-correlation function defined in Eq. (3.2) representing the particle's memory of being still a member of a shell after some time t can be calculated for a variety of combinations between the reference molecule and its surrounding peers. As our system is composed of three species cation, anion and water 3 x 3 combinations of reference molecules and peers are possible. Immanent symmetry, however, reduces this number to six: cation-cation (C-C), cation-anion(C-A), cation-water(C-W), anion-anion(A-A), anion-water(A-W) and water-water(W-W). A contraction to cation-all (C-all) neglecting the identity of peers was also computed. Fig.3.3 shows a representative example of C-all for the BMIM⁺BF₄⁻H₂O system with $x_{H_2O} = 0.768$.

3.4.1 Motional Parameters of Voronoi Dynamics

Due to the vast abundance of TCFs calculated a graphical representation is impractical. Therefore, they were all fitted to KWW functions defined in Eq. (3.10). From the quadruple set of parameters $\{\tau, \beta, A_0, A_1\}$ the first two are collected in Table 3.2 together with the average correlation time $\langle\tau\rangle$ defined in Eq. (3.12). The two amplitudes A_0 and A_1 are presented in Table 3.3.

Mean Residence Times

In order to find systematic relations and trends in this large set of data a compact characterization of each system by a few or even a single parameters is necessary. From previous studies we know that the viscosity has the power to incorporate the essential dynamic features of a system.^[38] Therefore, as a first attempt, we tried to find a relationship between $\langle\tau\rangle$ and the viscosity η .

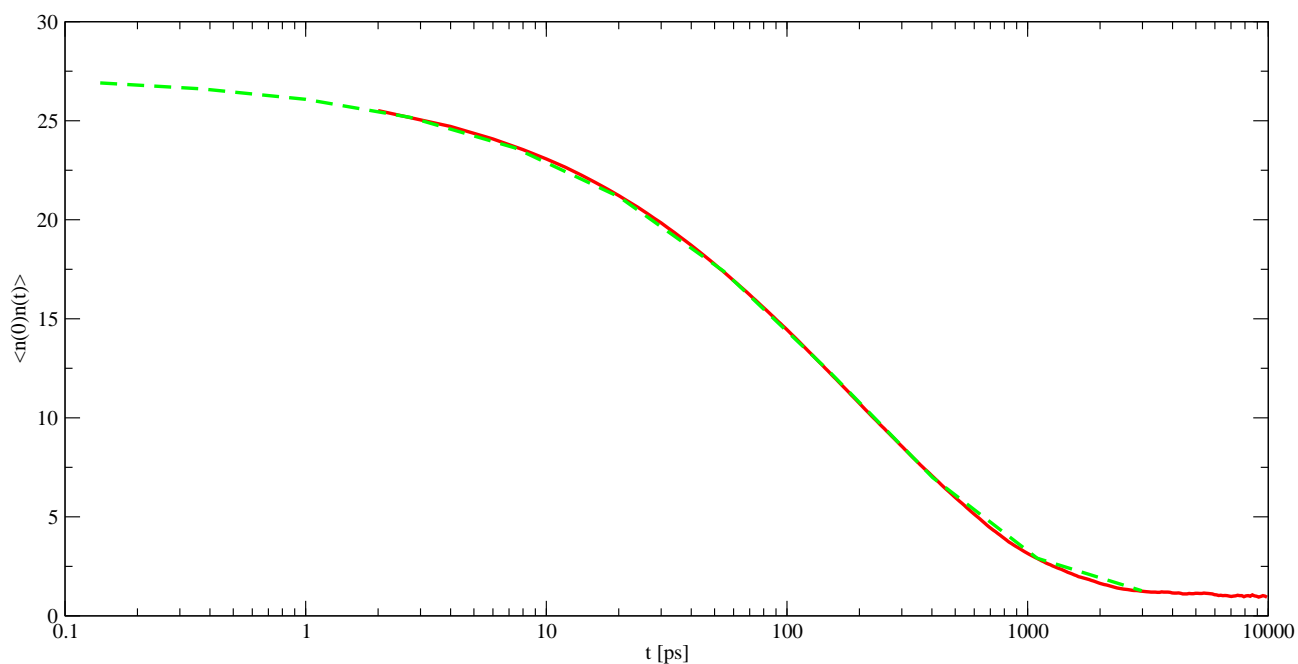


Figure 3.3: Residence correlation function $\langle n(0)n(t) \rangle$ for C-all of the system with $x_{H_2O} = 0.768$ (red solid line). The respective fit to a KWW function is given as a green dashed line. Please note the logarithmic time scale. The short time ($t=0$) and long time ($t=10$ ns) limits are 26.3 and 1.02. This means that in the beginning the cation is surrounded by 26 other molecules of all three species. After 10 ns 25 particles have migrated from the cation, but one is still resident.

Viscosity	η	0.9	1.4	2	2.2	4.5	6.4	7.5	22.5
Water content	x_{H_2O}	0.967	0.932	0.912	0.899	0.852	0.768	0.785	0.65
Cation	C	BMIM	EMIM	BMIM	EMIM	EMIM	BMIM	EMIM	EMIM
Anion	A	TFB	TRIF	TFB	TRIF	TRIF	TFB	TRIF	TRIF
Particle numbers	N_C, N_A	55	400	111	500	600	166	700	800
	N_{H_2O}	1629	5489	1147	4467	3444	548	2561	1504
Species		Motional Parameters							
C- all	$\langle \tau \rangle [ps]$	22.5	66.4	72.9	113.6	233.7	324.8	588.6	1805.6
	$\tau [ps]$	17.2	34.6	43.1	54.7	114.6	202.2	289.3	969.0
	β	0.586	0.512	0.553	0.490	0.495	0.572	0.495	0.520
C- C	$\langle \tau \rangle [ps]$	57.9	114.7	162.8	220.6	393.5	557.3	978.7	2310.8
	$\tau [ps]$	41.6	55.3	105.6	96.9	199.3	426.4	572.0	1437.0
	β	0.639	0.490	0.589	0.468	0.504	0.678	0.548	0.571
C- A	$\langle \tau \rangle [ps]$	64.0	181.8	147.7	287.6	511.6	454.8	1051.6	2728.6
	$\tau [ps]$	47.8	150.5	119.9	216.5	382.0	374.1	752.6	2005.5
	β	0.662	0.737	0.720	0.667	0.662	0.732	0.637	0.652
A- A	$\langle \tau \rangle [ps]$	13.2	86.0	31.3	131.5	224.3	121.8	490.3	1598.7
	$\tau [ps]$	5.9	46.9	13.0	72.3	94.9	47.7	245.7	536.7
	β	0.473	0.525	0.456	0.528	0.460	0.444	0.501	0.417
C- W	$\langle \tau \rangle [ps]$	23.7	49.6	52.3	74.5	130.6	143.7	269.6	625.3
	$\tau [ps]$	16.1	28.9	35.0	43.6	80.6	98.6	169.4	397.6
	β	0.610	0.547	0.604	0.549	0.568	0.615	0.576	0.581
A- W	$\langle \tau \rangle [ps]$	16.2	38.4	34.1	57.1	98.8	97.1	203.4	508.6
	$\tau [ps]$	11.2	22.5	23.8	35.0	61.9	68.2	119.8	340.7
	β	0.618	0.550	0.625	0.566	0.575	0.627	0.551	0.604
W- W	$\langle \tau \rangle [ps]$	9.8	18.0	19.9	27.6	47.3	58.4	97.1	221.0
	$\tau [ps]$	6.1	9.5	12.6	15.0	27.5	38.9	53.4	134.9
	β	0.570	0.515	0.576	0.523	0.547	0.601	0.528	0.564

Table 3.2: Relaxation KWW parameters as obtained from the fit of the residence autocorrelation function $\langle n(0)n(t) \rangle$.

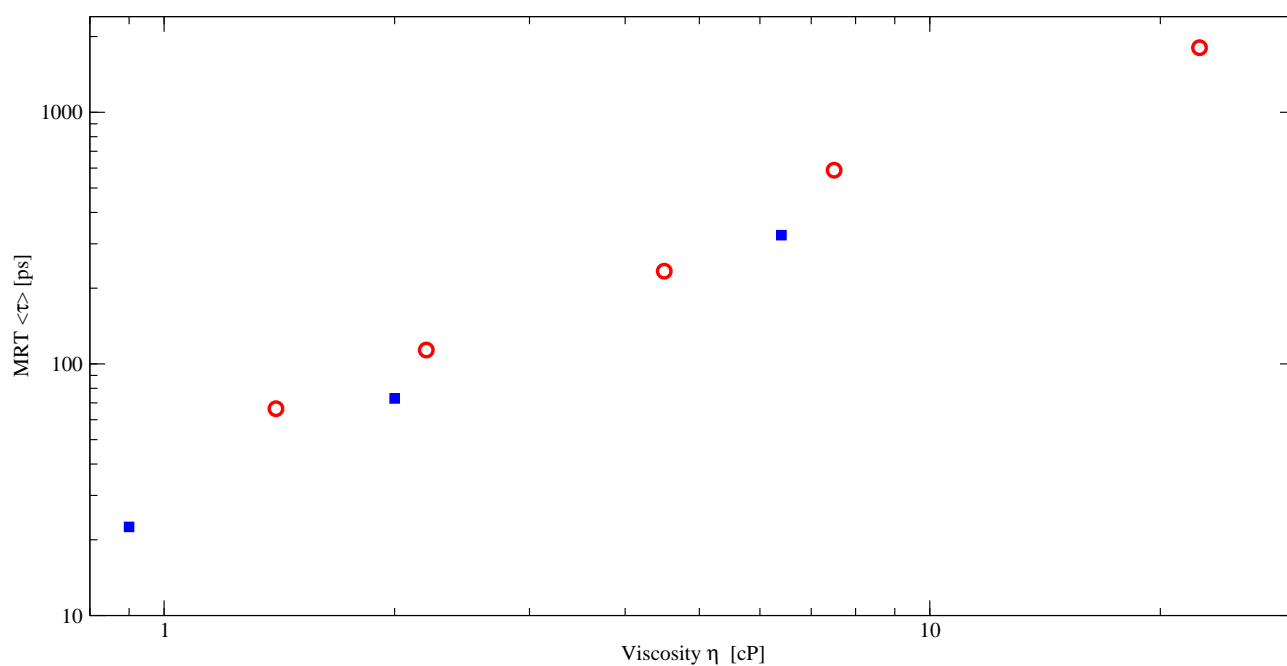


Figure 3.4: 3.4

Plot of the system specific MRT $\langle \tau \rangle$ versus the respective viscosity η . The logarithmic stretching of both axes was chosen in order to cope with the wide spread of viscosity values. The three BMIM⁺BF₄⁻H₂O systems are denoted by red squares, the EMIM⁺TRIF⁻H₂O systems by blue circles.

Fig.3.4 correlates the system specific $\langle\tau\rangle$ values with the respective viscosities in the case of C-all. A linear regression

$$\langle\tau\rangle = k\eta + d \quad (3.34)$$

with $k = 83.4$ and $d = -90.5$ yields an almost perfect fit with a correlation coefficient of 0.991 . We emphasise that this fit is over all systems, irrespective of the type of cation and anion: Each system is solely represented by its viscosity. Only a minimal, systematic spread between the $\text{BMIM}^+\text{BF}_4^-\text{H}_2\text{O}$ and $\text{EMIM}^+\text{TRIF}^-\text{H}_2\text{O}$ systems is observed. We found that the anion is the major source of this spread. This can be further elucidated by analysing in detail all six TCFs listed above. In fact, TCFs involving the anion show a larger spread as compared to others. This demonstrates once again the central role of the anion in determining properties of MILs.

The linear dependence of $\langle\tau\rangle$ on the viscosity is not restricted to the specific example given here. In fact, it is found for all six combinations C-C to W-W. It is even valid for the different systems $\text{BMIM}^+\text{BF}_4^-\text{H}_2\text{O}$ and $\text{EMIM}^+\text{TRIF}^-\text{H}_2\text{O}$. The viscosity is indeed a key quantity for structural relaxation. It seems that the viscosity comprises the characteristic features of relaxation of a specific system irrespective of further details.

We know from previous studies that hydrodynamic laws are no more applicable to MILs in the strict sense.^[45] The only relation which survives as a remnant feature of hydrodynamics is the linear dependence of relaxation times on viscosity. This enables an interpolation or even prediction of the relaxation behaviour of systems not explicitly studied.

So far we have discussed the MRT $\langle\tau\rangle$ as a linear function of viscosity η . This relation also helps in grouping $\langle\tau\rangle$ for the six combinations C-C to W-W. One finds a first group of linear slopes $k = 102$, $k = 88$ and $k = 61$ for C-C, C-A and A-A i.e. for the combinations within the ionic network. This sequence of slopes correlates fairly with the absolute relaxation times having the same order. The second group of slopes $k = 23$ and $k = 19$ gives the right order of the $\langle\tau\rangle$ for C-W and A-W representative for the coupling of the ionic network with the hydrogen-bond water network W-W. The latter shows the smallest slope $k = 8$ as well as the shortest $\langle\tau\rangle$. This splitting into three groups goes along with the stronger interactions of charged species in the ionic network in contrast to the coupling in a polar hydrogen-bonded network. The coupling between these two networks should be of intermediate strength. Indeed, corresponding relaxation times fall in the second group.

The order within each group can be explained by the following arguments: Within the ionic network C-A coupling is, of course, strongest. The additional interaction of hydrophobic tails favours C-C in

comparison to A-A. The longer relaxation time of C-W as compared to A-W is not easy to explain at first sight.

As the surface of the cation is considerably larger than that of the anion (see Table 3.1), this behaviour points to a migration process along the various sites of the respective ionic surface which offers an alternative to the essential possibilities of entering or leaving the solvation cage of the reference ion.

Distribution of Relaxation Times

Quite general, the spread of relaxation times as measured by the parameter β is almost independent of viscosity as a representative of the respective system. However, there is a shift between the $\text{EMIM}^+\text{TRIF}^-\text{H}_2\text{O}$ and the $\text{BMIM}^+\text{BF}_4^-\text{H}_2\text{O}$ systems. This slight shift is specific to the combination of species. For C-C, C-W, A-W and W-W the $\text{EMIM}^+\text{TRIF}^-\text{H}_2\text{O}$ systems stay close to the canonical value $1/2$, while the $\text{BMIM}^+\text{BF}_4^-\text{H}_2\text{O}$ systems show a tendency towards but still below $2/3$.

The β values for both types of systems coincide to $1/2$ for A-A correlations. The strong coupling between cation and anion leads to the highest β values of $2/3$ or more. In other words, the pair with the strongest coupling shows the simplest time dependence closer to a mono-exponential behaviour.

Under the plausible assumption that the difference in the cationic species is of minor influence the slight shifts observed in β are mainly caused by the different anions BF_4^- and TRIF^- . While the latter exhibits a pronounced dipole moment, the former is completely apolar. Therefore, one would expect a stronger coupling to the other species for TRIF^- as compared to BF_4^- . The analysis of β values shows that this stronger coupling leads to a simpler time dependence, i.e. to a larger β value.

In summary, the stronger the coupling, the simpler the time behaviour: The variability of relaxation channels in accordance with a multiple minima energy landscape is reduced by strong coupling mechanisms overruling less pronounced energy minima. As a general rule $\beta=1/2$ is a reasonable choice to describe the spread of relaxation times in MIL-water mixtures.

3.4.2 Probabilistic Approach

So far, we have derived information on shell dynamics from the fit of the simulated residence correlation function Eq. (3.2). As worked out in the theory section, an alternative route may be a probabilistic approach. Starting from the correspondence equation (3.25) and inserting the spectral expansion

(3.22) one gets a residence function as a superposition of exponentials with amplitudes

$$A_i = \frac{\mathbf{p}_0 \cdot \mathbf{q}_i^R}{\mathbf{q}_i^L \cdot \mathbf{q}_i^R} \sum_{j \in S_d} \mathbf{q}_i^L(j) \quad (3.35)$$

where $\mathbf{q}_i^L(j)$ is the j^{th} component of the i^{th} left eigenvector of \mathbf{V} .

Reduction of spectral expansion to a 2x2 system

Sorting the eigenvalues λ_i in size one finds two dominant amplitudes A_1, A_2 corresponding to $\lambda_1 = 0$, the steady state value, and the largest non-zero eigenvalue λ_2 . All other eigenvalues make a marginal contribution only. In other words, the resolution with respect to proximity seems to reduce to a pseudo IN/OUT model. In fact, if one contracts the full transition matrix to the 2x2 case IN/OUT the resulting two eigenvalues are pretty close to those dominating the full spectral expansion. Furthermore, the simple example given in the theory section enables an interpretation of the two dominating amplitudes in terms of the coordination number CN and a total number of all potential neighbours N. (Eqs. (3.28) to (3.31))

Thus the sole dominant non-zero eigenvalue is given by $\lambda_2 = -b \frac{N}{CN(N-CN)}$. Consequently, the time dependence of the residence function is essentially determined by the transition rate b between IN and OUT states. Therefore the probabilistic description of shell dynamics boils down to single a transition rate, or sole eigenvalue of the transition matrix.

As an illustration of the general findings described above we give one example for the $\text{BMIM}^+\text{BF}_4^-\text{H}_2\text{O}$ system and one for the $\text{EMIM}^+\text{TRIF}^-\text{H}_2\text{O}$ system. For 55 $\text{BMIM}^+\text{BF}_4^-$ in 1629 H_2O the \mathbf{V} matrix for C-all, which is not fully atomistic but rather depicts a more crude graining reads

$$\mathbf{V} = \begin{pmatrix} -0.156 & 0.032 & 0.041 & 0.001 & 0.005 & 0.001 & 0.004 & 0.070 \\ 0.173 & -0.390 & 0.013 & 0.081 & 0.045 & 0.023 & 0.018 & 0.036 \\ 0.058 & 0.004 & -0.179 & 0.030 & 0.013 & 0.007 & 0.006 & 0.061 \\ 0.003 & 0.038 & 0.054 & -0.244 & 0.041 & 0.037 & 0.019 & 0.054 \\ 0.014 & 0.025 & 0.027 & 0.045 & -0.285 & 0.055 & 0.069 & 0.051 \\ 0.003 & 0.012 & 0.012 & 0.038 & 0.051 & -0.302 & 0.127 & 0.060 \\ 0.005 & 0.004 & 0.005 & 0.009 & 0.028 & 0.057 & -0.182 & 0.074 \\ 5 \cdot 10^{-4} & 4 \cdot 10^{-5} & 3 \cdot 10^{-4} & 1 \cdot 10^{-4} & 1 \cdot 10^{-4} & 2 \cdot 10^{-4} & 4 \cdot 10^{-4} & -0.002 \end{pmatrix}$$

In any case, the fully atomistic matrix shows the same behaviour in respect of its eigenvalues and their amplitudes. Here, the eigenvalues are $\lambda_n = (0, -0.064, -0.126, -0.200, -0.241, -0.319, -0.360, -0.430)$

with the respective amplitudes $A_n=(0.025, 0.972, 6.3 \cdot 10^{-5}, 1.3 \cdot 10^{-3}, 1.2 \cdot 10^{-3}, 1.4 \cdot 10^{-4}, 1.4 \cdot 10^{-6}, 1.6 \cdot 10^{-4})$. Only the two amplitudes A_1 and A_2 are significantly greater than zero. Contraction of the above Matrix to only one IN and one OUT state, leads to

$$\mathbf{V} = \begin{pmatrix} -0.063 & 0.0631 \\ 0.002 & -0.002 \end{pmatrix}$$

with $\lambda_n=(0, -0.065)$ and $A_n=(0.025, 0.975)$ which shows virtually the same time behaviour as the bigger matrix. The second example is a C-all matrix for the EMIM⁺TRIF⁻H₂O system with 400 MILs. Before contraction, the matrix looks like this:

$$\mathbf{V} = \begin{pmatrix} -0.142 & 0.040 & 0.045 & 0.002 & 0.008 & 0.046 \\ 0.166 & -0.378 & 0.013 & 0.085 & 0.090 & 0.024 \\ 0.060 & 0.004 & -0.167 & 0.035 & 0.026 & 0.042 \\ 0.005 & 0.041 & 0.053 & -0.248 & 0.109 & 0.039 \\ 0.010 & 0.024 & 0.022 & 0.062 & -0.167 & 0.049 \\ 9 \cdot 10^{-5} & 1 \cdot 10^{-5} & 6 \cdot 10^{-5} & 4 \cdot 10^{-5} & 8 \cdot 10^{-5} & -3 \cdot 10^{-4} \end{pmatrix}$$

with eigenvalues $\lambda_n=(0, -0.044, -0.137, -0.200, -0.297, -0.424)$ and amplitudes $A_n=(0.006, 0.993, 1.3 \cdot 10^{-5}, 5.4 \cdot 10^{-5}, 2.2 \cdot 10^{-4}, 1.7 \cdot 10^{-4})$. After contraction, the overall picture stays the same:

$$\text{The matrix } \mathbf{V} = \begin{pmatrix} -0.043 & 0.043 \\ 2.8 \cdot 10^{-4} & -2.8 \cdot 10^{-4} \end{pmatrix}$$

has eigenvalues $\lambda_n=(0, -0.044)$ and the corresponding amplitudes are $A_n=(0.006, 0.994)$;

Comparison of KWW and Markov Amplitudes

Table 3.3 collects all KWW amplitudes as well as the dominant Markov amplitudes A_1 and A_2 for all combinations of species including C-all. In order to get a more vivid impression of the numbers, we have multiplied the normalised amplitudes by the respective coordination number CN representing the number of immediate neighbours surrounding the reference molecule. From the contraction of the general case to the 2x2 model we know that the Markov amplitudes are given by CN/N and $(N-CN)/N$, respectively. Multiplying by CN yields $A_1 = CN^2/N$ and $A_2 = CN(N - CN)/N$. These are the values listed in 3.3 for the Markov amplitudes. The agreement of the A_2 values with the corresponding

Viscosity		η	0.9	1.4	2	2.2	4.5	6.4	7.5	22.5
Species	Method	Relaxation Amplitudes								
C-all	KWW	A_1	1.141	0.285	1.089	0.274	0.268	1.019	0.223	0.263
		A_2	42.2	35.5	36.1	33.7	30.2	26.3	23.2	23.2
	Markov	A_1	1.103	0.190	1.029	0.190	0.187	0.957	0.176	0.174
		A_2	42.7	34.3	36.5	32.0	29.3	28.1	26.0	23.1
C-C	KWW	A_1	0.127	0.036	0.276	0.028	0.050	0.493	0.071	0.153
		A_2	2.2	3.2	5.0	4.4	5.7	7.9	6.4	7.6
	Markov	A_1	0.114	0.026	0.275	0.040	0.056	0.542	0.071	0.088
		A_2	2.4	3.2	5.2	4.4	5.8	8.9	7.0	8.3
C-A	KWW	A_1	0.086	0.068	0.119	0.061	0.092	0.171	0.107	0.099
		A_2	1.9	3.3	3.4	4.3	5.1	5.0	5.9	6.6
	Markov	A_1	0.079	0.034	0.130	0.040	0.047	0.191	0.052	0.059
		A_2	2.0	3.6	3.7	4.4	5.3	5.4	6.0	6.8
A-A	KWW	A_1	0.001	0.002	0.003	0.008	0.009	0.005	0.024	0.021
		A_2	0.2	1.4	0.4	1.8	2.5	0.6	2.5	3.2
	Markov	A_1	0.001	0.005	0.003	0.007	0.009	0.005	0.011	0.014
		A_2	0.3	1.5	0.6	1.9	2.4	0.9	2.8	3.3
C-W	KWW	A_1	0.924	0.180	0.682	0.152	0.108	0.356	0.089	0.052
		A_2	37.2	28.0	26.3	23.5	17.8	12.7	13.7	8.2
	Markov	A_1	0.945	0.139	0.694	0.121	0.098	0.354	0.073	0.043
		A_2	38.3	27.5	27.5	23.2	18.3	13.6	13.6	8.0
A-W	KWW	A_1	0.140	0.057	0.095	0.060	0.039	0.055	0.025	0.025
		A_2	13.2	14.9	9.2	12.1	8.9	4.8	6.9	3.8
	Markov	A_1	0.138	0.044	0.097	0.036	0.028	0.051	0.020	0.012
		A_2	14.8	15.5	10.5	12.7	9.8	5.2	7.2	4.3
W-W	KWW	A_1	0.093	0.023	0.060	0.020	0.014	0.035	0.009	0.007
		A_2	11.6	9.8	7.8	7.9	6.0	3.9	4.6	2.5
	Markov	A_1	0.092	0.017	0.062	0.014	0.011	0.033	0.008	0.004
		A_2	12.1	9.6	8.3	7.9	6.1	4.2	4.4	2.5

Table 3.3: Collection of Relaxation Amplitudes: The results from the KWW fit are compared to those from the Markovian master equation.

KWW entries is fairly good. With very few exceptions, the deviation is below 10%. Deviations above 30% occur exclusively for TFB-TFB, while the anion TRIF⁻ is not exceptional.

On a percentage scale the steady state amplitudes A_1 show much larger deviations. However, their absolute magnitude is almost marginal compared to that of A_2 . Therefore, the absolute discrepancies play a minor role. This marginality of steady state amplitudes is already visible in the two state model: The huge number of potential neighbours N (see 3.1) overrules the actual number of immediate neighbours CN . Although CN values differ only slightly between BMIM⁺BF₄⁻H₂O and EMIM⁺TRIF⁻-H₂O systems, their rather different size makes the steady state amplitudes differ by a factor of 4.

The steady state amplitudes for TFB-TFB are a category of their own. The strong charge-charge repulsion combined with the pronounced steric exclusion causes fairly small amplitudes. Therefore, their high percental discrepancy is misleading.

Consistency Check

In giving the Markovian master equation in the theory section we explicitly discriminated between the discrete Eq. (3.15) and the continuous Eq. (3.16) form. This distinction is a reminder that the choice of Δt is important. Therefore, we consider transition matrices in two rather different time regimes. First, we focus on the short time behaviour $\Delta t = 0.1$ ps. Later, we will also discuss transition times comparable to the actual KWW times. For a true Markov process, the transition matrix for long times should be constructible from its short time analogue.

Previously, we found a linear dependence of the KWW $\langle \tau \rangle$ on the viscosity (see Eq. (3.34)). An analogous relation can be derived for the dominating Markov time $\tau_2 = 1/\lambda_2$. It turns out, however, that the relation is not linear but rather logarithmic:

$$\tau_2^{0.1ps} = K \ln \eta + D \quad (3.36)$$

Typical values for C-all are $K = 0.52$ and $D = 1.59$. Fig. 3.5 shows the actual fit which matches both types of systems BMIM⁺BF₄⁻-H₂O and EMIM⁺TRIF⁻-H₂O simultaneously with a correlation coefficient of 0.998. This excellent agreement between data points and fit enables an extrapolation or even prediction of viscosity values from an average of only hundred 0.1 ps frames(!). In fact, inversion of formula (3.36) predicts viscosity values accurate to a few percent over a viscosity range of a factor 25, almost two orders of magnitude. Since a linear relation between the average KWW residence time $\langle \tau \rangle$ and the viscosity η was shown in Eq. (3.34), the logarithmic dependence of τ_2 found here, would imply a logarithmic relation between τ_2 and the KWW residence time. Such a consistency check

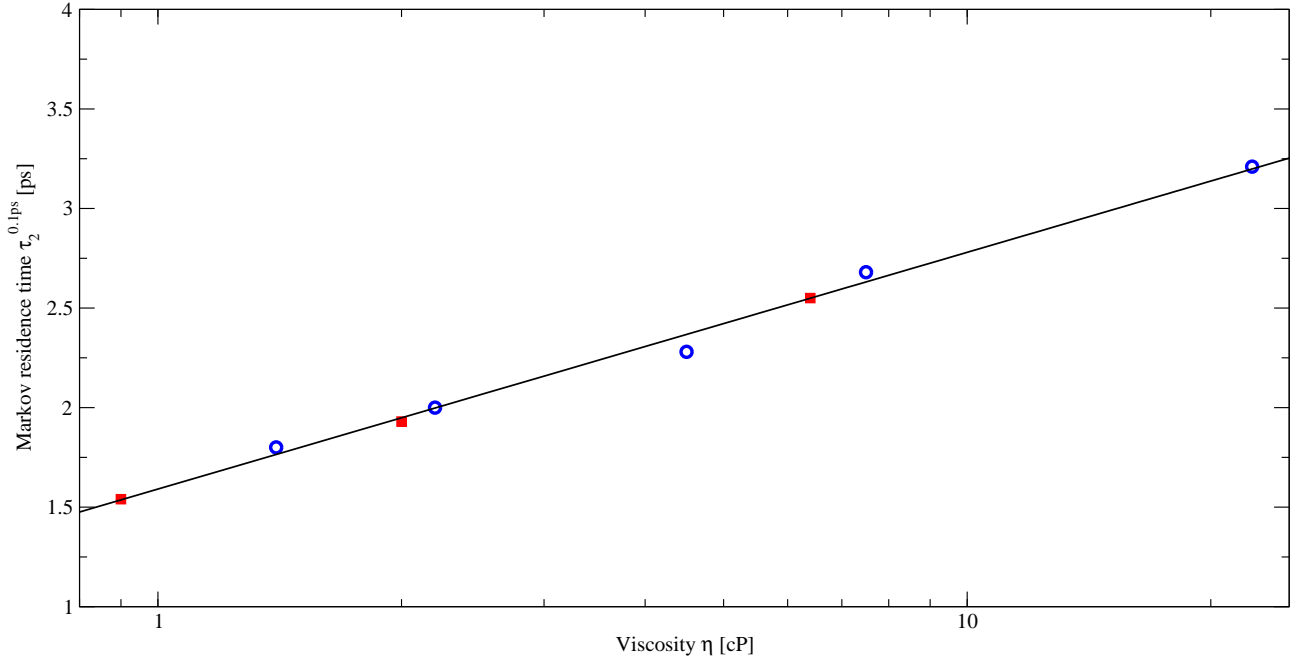


Figure 3.5: The dominant C-all Markov Residence Time $\tau_2^{0.1ps}$ is plotted versus the viscosity η as a representative of the respective system. The tree BMIM⁺BF₄⁻H₂O systems are denoted by red squares, the EMIM⁺TRIF⁻H₂O systems by blue circles. The straight line symbolises the relation $0.52 \ln \eta + 1.59$. Due to the wide spread of viscosity values a logarithmic scale stretching is used.

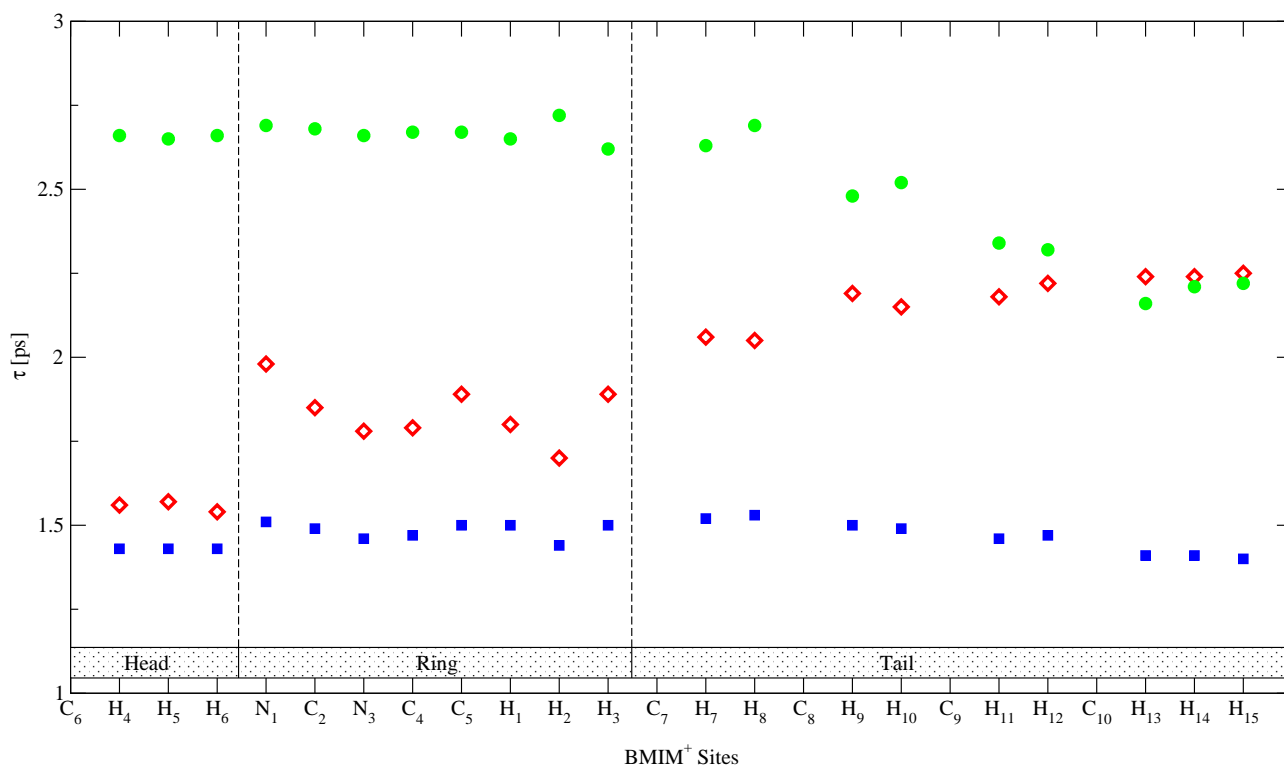


Figure 3.6: The dominant Markov Residence Time $\tau^{0.1ps_2}$ resolved for all 25 proximity sites of BMIM^+ ($x_{H_2O} = 0.967$). Three combinations BMIM-BMIM (red diamonds), BMIM-TFB (green circles) and BMIM-WATER (blue squares).

enhances the reliability of the fitting procedures. Indeed, we could find a relation

$$\tau_2^{0.1ps} = 0.39 \ln \langle \tau \rangle + 0.24 \quad (3.37)$$

for C-all. This consistency prevails over all other combinations C-C to W-W.

Proximity Resolved Populations

So far our focus was on the total population or occupation of a shell. The probabilistic approach, however, permits a finer grained resolution: The population $p_i(t)$ of each site or state can be followed separately in time. Quite generally, from the spectral expansion Eq. (3.22) one would again expect a superposition of exponentials. In order to test this we have fitted a KWW function to each $p_i(t)$. The corresponding β values were found pretty close to unity such that $p_i(t)$ can be best described by a mono-exponential function. The distribution of the site-specific exponential times is displayed in Fig. 3.6. While C-all and C-W show a uniform distribution, C-C and C-A exhibit an opposite trend. Following the classification Head (methyl group), Ring, Tail (butyl- or ethyl group) relaxation times

for C-C are longer at the Tail and shorter at the Head. For C-A the trend is reversed. This is in accordance with previous studies (Figure 7. in Ref.[31]) of the static spatial distribution of cations and anions around a reference cation. The higher concentration of cations in the vicinity of the tail and to some extent close to the ring goes along with the long or medium residence around these parts of the reference molecule found here. Simultaneously, the anion is crowded around the Head and Ring in accordance with the longer residence times. The strict correlation between occupation and residence times found for the ions comes from their strong interaction between charges. The weaker dipolar interaction possible for the water molecule leads to shorter and uniformly distributed residence times. The interactions between hydrophobic tails in the $\text{BMIM}^+\text{BF}_4^-\text{H}_2\text{O}$ systems, in turn, cause a rather non-uniform distribution which is more flat for $\text{EMIM}^+\text{TRIF}^-\text{H}_2\text{O}$ as a consequence of the shorter alkyl chain. The site specificity of residence times, however, does not result in a complex time behaviour of the total population $\sum p_i(t)$. As we have learned above $\sum p_i(t)$ is essentially mono-exponential. A spread in relaxation times as reflected by a KWW exponent β is not really evident.

Matching Molecular Dynamics and Probabilistic Approach

When setting up the Markovian master equation we emphasised the importance of the time interval Δt . So far we have worked with transition matrices in the short time regime $\Delta t = 0.1\text{ps}$. As an alternative, we now analyse transition matrices for a Δt comparable to the characteristic KWW times of $\langle n(0)n(t) \rangle$. Indeed, an enhancement of Δt brings the correlation functions computed in the probabilistic approach closer to the time regime of the residence functions $\langle n(0)n(t) \rangle$ directly computed from simulated data. Even more one could select a specific Δt to achieve a reasonable agreement of both curves. However, we aim at a universal model of residence functions. Therefore, the choice of a case specific Δt is not appropriate and we use a uniform value of $\Delta t = 102.4\text{ps} = 0.1\text{ps} \times 2^{10}$ as the different values for Δt are organised in powers of two. This value of roughly 100 ps goes along with the typical KWW values found in simulation data of the systems studied here.

The choice of a uniform Δt is not really in accordance with the spread of viscosities among the systems. To compensate for this, introduction of a parameter independent of viscosity would be more appropriate. In fact, the KWW parameter β was found to be almost independent of viscosity and had an almost universal value of $\beta = 1/2$. Therefore we propose to retain the uniform Δt but to stretch the time scale by the introduction of β . In other words, we suggest to change the spectral expansion Eq. (3.22) from a superposition of exponentials to a superposition of KWW functions with $\beta = 1/2$. The result of our proposal can be seen in Fig.3.7. We have explicitly used a logarithmic time scale

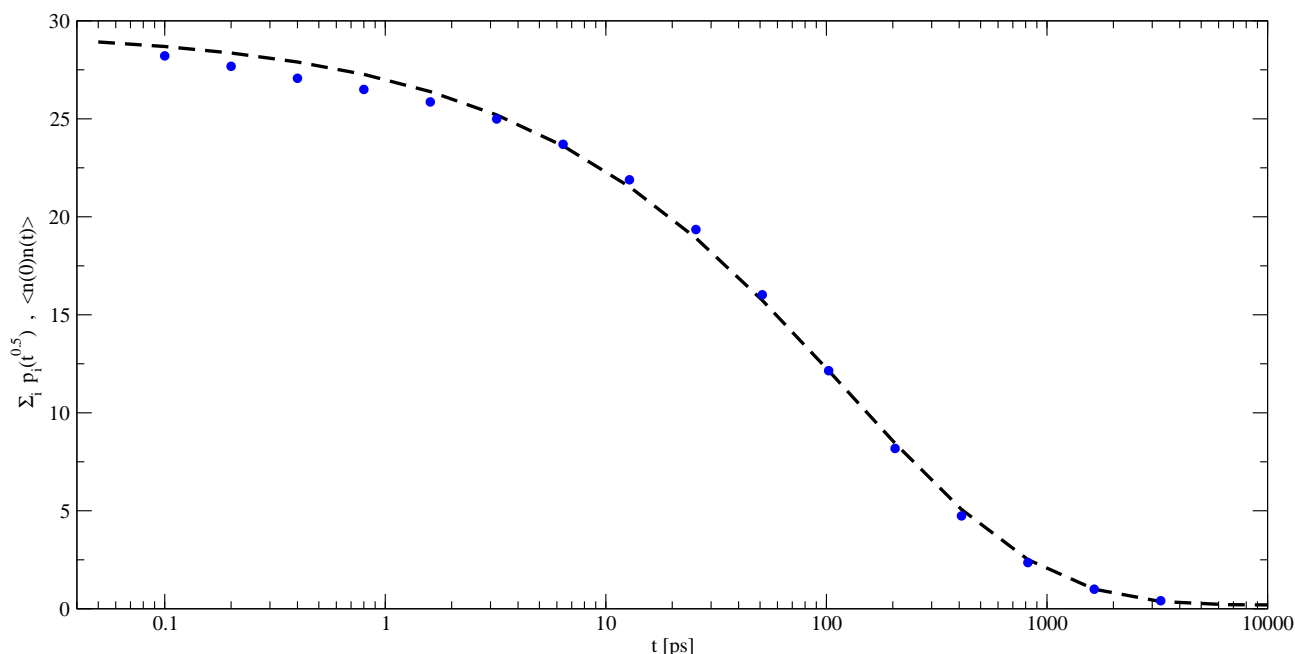


Figure 3.7: Modelling the simulated $C_n(t) = \langle n(0)n(t) \rangle$ by the probabilistic approach for the case of EMIM⁺TRIF⁻H₂O with $x_{H_2O} = 0.852$. The solid green line stands for the time evolution of the net shell population $\sum_{i \in S_d} p_i(\sqrt{t})$. The underlying transition matrix is for $\Delta t = 102.4$ ps. The universal stretch parameter is $\beta = 1/2$. The residence correlation function $C_n(t) = \langle n(0)n(t) \rangle$ is displayed as blue circles.

in order to emphasise the long-time regime, because agreement for short times can be achieved quite easily. The agreement shown in Fig. 3.7 for C-all of the system EMIM⁺BF₄⁻H₂O with 800 MILs is representative for all other residence functions. It would be tempting to use the specific β value of the respective $\langle n(0)n(t) \rangle$ which indeed further improves the agreement. However, this would be against our intent of universality.

3.5 Conclusion

In this study we have followed a twofold strategy to investigate the structural relaxation of Voronoi solvation shells in hydrated molecular ionic liquids. From the combination of a direct analysis of simulated data and the application of a specifically developed probabilistic model a deeper insight into characteristic features of structural relaxation could be gained.

The direct analysis of the time evolution of neighbourhood relations revealed and confirmed two fundamental relations: First, the mean residence times derived from the residence correlation functions were found to be linearly related to the system's viscosity. This linear relationship is not only valid

within systems of the same type, but even applies universally to other system types as well. In this respect it is important to note that the term "different system type" means an exchange of both, the cation and the anion. As it is generally known that the properties of molecular ionic liquids depend strongly on the type of the anion, this inter-system relationship is quite remarkable. In other words, the concept of viscosity scaling already found for single particle and collective dynamics in molecular ionic liquids^[38] applies to the relaxation of Voronoi shells as well. Second, when representing the residence correlation by a Kohlrausch-Williams-Watts function its complex time behaviour can be described by a fractional time which almost universally follows a square root law.

While the direct analysis of site-specific relaxation would require an extremely high numerical effort it is easily accessible - at least in a qualitative way - from our probabilistic approach. In this way one gets a distinct spread of relaxation times which is not visible in the overall relaxation due to compensatory effects. Even more, the latter can be described by a simple two state IN/OUT model whose essential parameters are the frequency of IN/OUT transitions and the coordination number. In particular, an intuitive interpretation of the initial and asymptotic value of the residence correlation function is possible otherwise not accessible by direct analysis. As both, the frequency of IN/OUT transitions and the coordination number, can be derived from an inexpensive short time simulation the probabilistic approach enables an easy description of overall relaxation. The dominant relaxation time is found to be a logarithmic function of the viscosity as well as of the mean residence time derived from the direct analysis. As the latter was found to be a linear function of the viscosity the twofold logarithmic dependence of the dominant relaxation time derived from the probabilistic approach represents a consistency check.

Adopting the square root time law from the direct analysis the probabilistic approach can be tuned to permit an even quantitative description of overall relaxation. This demonstrates that our probabilistic model yields quantitative results solely from the frequency of transitions extracted from very short simulation studies covering a few pico seconds. In this way one can get results which in a direct analysis would require long-term simulations of several nano seconds.

Acknowledgement

This work was supported by the project P19807 of the FWF Austrian Science Fund. Furthermore, we would like to thank the Institute of Scientific Computing at the university of Vienna for a generous allocation of computer time.

Bibliography

- [1] G. F. Voronoi, *J. Reine Angew. Math.* **134**, 198 (1908).
- [2] B. N. Delaunay, *Bulletin of Academy of Sciences of the USSR* **7** **6**, 793 (1934).
- [3] B. Bouvier, R. Grunberg, M. Nilges, and F. Cazals, *Proteins* **76**, 677 (2009).
- [4] B. Kirchner, J. Hutter, I.-F. W. Kuo, and C. J. Mundy, *Int. J. Mod. Phys. B* **18**, 1951 (2004).
- [5] P. F. Goncalves and H. Stassen, *J. Chem. Phys.* **123**, 214109 (2005).
- [6] B. Chaudhuri, F. Pederiva, and G. V. Chester, *Phys. Rev. B* **60**, 3271 (1999).
- [7] M. Schaefer, C. Bartels, fabrice Leclerc, and M. Karplus, *J. Comp. Chem.* **15**, 1857 (2001).
- [8] M. Neumann, F. J. Vesely, O. Steinhauser, and P. Schuster, *Mol. Phys.* **35** (1978).
- [9] M. Neumann, F. J. Vesely, O. Steinhauser, and P. Schuster, *Mol. Phys.* **37** (1979).
- [10] R. Abseher, H. Schreiber, and O. Steinhauser, *Proteins* **25**, 366 (1996).
- [11] G. Löffler, T. Mager, C. Gerner, H. Schreiber, H. Bertagnolli, and O. Steinhauser, *J. Chem. Phys.* **104** (1996).
- [12] S. Boresch, S. Ringhofer, P. Höchtel, and O. Steinhauser, *Biophys Chem.* **78**, 43 (1999).
- [13] S. Boresch, P. Höchtel, and O. Steinhauser, *J. Phys. Chem. B* **104** (2000).
- [14] P. Höchtel, S. Boresch, and O. Steinhauser, *J. Chem. Phys.* **112** (2000).
- [15] B. Halle, *Philosophical Transactions of The Royal Society* **359**, 1207 (2004).
- [16] N.-V. Buchete and G. Hummer, *J. Phys. Chem. B* **112**, 6057 (2008).
- [17] E. V. Eijnden and M. Venturoli, *J. Chem. Phys.* **130**, 194101 (2009).

Bibliography

- [18] A. Okabe, *Spatial tessellations: concepts and applications of Voronoi diagrams* (Wiley, New York, 2000).
- [19] K. E. Thompson, Int. J. Numer. Meth. Engng. **55**, 1345 (2002).
- [20] D. F. Watson, The Computer Journal **24**, 167 (1981).
- [21] M. Gerstein, J. Tsai, and M. Levitt, J. Mol. Biol. **249**, 955 (1995).
- [22] G. De Fabritiis and P. V. Coveney, Comput. Phys. Commun. **153**, 209 (2003).
- [23] H. Borouchaki, P. L. George, F. Hecht, P. Laug, and E. Saltel, Finite Elements in Analysis and Design **25**, 61 (1997).
- [24] H. Borouchaki and S. H. Lo, Comput. Methods Appl. Mech. Engng. **128**, 153 (1995).
- [25] B. J. Berne, J. P. Boon, and S. A. Rice, J. Chem. Phys. **45**, 1086 (1966).
- [26] S. W. Provencher, Comput. Phys. Commun. **27**, 213 (1982).
- [27] S. W. Provencher, Comput. Phys. Commun. **27**, 229 (1982).
- [28] R. S. Anderssen, S. A. Husain, and R. J. Loy, Anziam J. **45**, C800 (2004).
- [29] E. W. Montroll and J. T. Bendler, J. Stat. Phys **34**, 129 (1984).
- [30] C. Schröder, T. Rudas, S. Boresch, and O. Steinhauser, J. Chem. Phys. **124**, 234907 (2006).
- [31] C. Schröder, G. Neumayr, and O. Steinhauser, J. Chem. Phys. **130**, 194503 (2009).
- [32] J. N. Canongia Lopes, J. Deschamps, and A. A. H. Padua, J. Phys. Chem. B **108**, 2038 (2004).
- [33] J. N. Canongia Lopes, J. Deschamps, and A. A. H. Padua, J. Phys. Chem. B **108**, 11250 (2004).
- [34] J. de Andrade, E. S. Böes, and H. Stassen, J. Phys. Chem. B **106**, 13344 (2002).
- [35] C. Schröder, M. Haberler, and O. Steinhauser, J. Chem. Phys. **128**, 134501 (2008).
- [36] C. G. Hanke, S. L. Price, and R. M. Lynden-Bell, Mol. Phys. **99**, 801 (2001).
- [37] A. J. Stone and M. Alderton, Mol. Phys. **56**, 1047 (1985).
- [38] C. Schröder and O. Steinhauser, J. Chem. Phys. **128**, 224503 (2008).

- [39] J. N. Canongia Lopes and A. A. H. Padua, J. Phys. Chem. B **108**, 16893 (2004).
- [40] W. L. Jorgensen, J. Am. Chem. Soc. **103**, 335 (1981).
- [41] T. Darden, D. York, and L. Pedersen, J. Chem. Phys. **98**, 10089 (1993).
- [42] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, J. Chem. Phys. **103**, 8577 (1995).
- [43] J.-P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, J. Comput. Phys. **23**, 327 (1977).
- [44] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, and S. Swaminathan, J. Comput. Chem. **4**, 187 (1983).
- [45] C. Schröder, C. Wakai, H. Weingärtner, and O. Steinhauser, J. Chem. Phys. **126**, 084511 (2007).

4 Global and Local Voronoi Analysis of Solvation Shells of Proteins

G. Neumayr, T. Rudas and O. Steinhauser submitted to J. Chem. Phys. (April 2010)

This paper presents the structure and dynamics of hydration shells for the three proteins ubiquitin, calbindin and phospholipase. The raw data derived from molecular dynamics simulations are analysed on the basis of fully atomistic Delaunay tessellations. In order to cope with the high numerical effort for the computation of these Voronoi shells we have implemented and optimised an intrinsically periodic algorithm.

Based on this highly efficient Voronoi decomposition a variety of properties is presented: Three dimensional water and ion nuclear densities as well as the geometrical packing of water molecules is discussed. Thereby, we develop VISA, the Voronoi analogue of the well known solvent accessible surface area (SASA). The traditional radial distribution functions are resolved into Voronoi shells as a transient device to the new concept of shell grained orientational order. Thus we analyse the donor-acceptor property as well as the amount of dielectric screening. Shell dynamics is described in terms of mean residence times. In this way a retardation factor for different shells can be derived and was compared to experimental values. All these results and properties are presented both, at the global protein level as well as at the local residue level.

4.1 Introduction

Solvation or hydration of proteins plays a functional role in important fields like protein folding,^[1] protein architecture,^[2] conformational stability,^[3] protein dynamics,^[4] ligand binding^[5] and the selectivity of specific interactions.^[6] The complexity of this phenomenon requires a combined investigation, both by experiment^[7] and simulation.^[8] Since the solvent is organised in subsequent layers around

the protein, a proper definition of solvation shells is a key device.

Traditionally, distance-based radial shell definitions are frequently used which necessitate the choice of a set of parameters inferred from the extrema of radial distribution functions. Such an approach does not properly account for the inherent spatial anisotropy of solvation shells and strictly speaking is specific to the type of the reference atom chosen. The method of Voronoi decomposition circumvents these problems. Even more, it permits a parameter-free, unambiguous definition of solvation shells.

Restricted to the case of static structure, an increasing number of Voronoi analyses have elucidated characteristic features like volume, density and packing, pockets, cavities and voids as well as empirical potentials as summarised in the review of Poupon.^[9] Recent studies have extended this approach from single proteins to protein-protein interaction.^{[10], [11]} In the field of static solvation structure, first trials date back to Voronoi studies of simple systems like hydrated anions (PO_4^-)^[12] and zwitterions (glycyl).^[13]

With increasing computer power the Voronoi analyses were extended from single to multiple frames generated by molecular dynamics simulations. The target systems were simple and non-biologic composed of small molecules like silicate glasses,^{[14], [15]} water^[16] and ethanol.^[17] For another set of 30 small solutes ranging from ammonia to thiophenol dissolved in water the Voronoi method was used in order to compute the so-called cavitation contribution to the solvation free energy.^[18] A quite interesting but in the present context rather exotic application of the Voronoi concept was developed by Espanol who transferred simulations of molecular fluid particles to the mesoscopic dynamics of Voronoi fluid particles.^{[19], [20]}

A combination of all above mentioned aspects, namely dynamical aspects of protein solvation computed by molecular dynamics simulations applying the Voronoi method first appeared in a preliminary but seminal study of ubiquitin.^[21] This Voronoi analysis at the molecular level was followed by a paper on mesoscopic, dielectric properties.^[22] At that stage of available software and hardware the high numerical effort restricted the analysis to a short trajectory i.e. to a very limited number of frames. In addition one had to use auxiliary devices like a distance pre-selection. The greatest obstacle, however, was the inherent non-periodicity of the tessellation algorithm. This is the reason why the extensive study of the three proteins ubiquitin, calbindin and phospholipase was still based on radial shells.^{[23], [24]}

Further development and implementation of an intrinsically periodic algorithm^[25] enables us to perform a complete Voronoi analysis of long term simulation data of these three solvated proteins. Thereby we can rely on a fully atomistic Delaunay tessellation. A variety of properties derived from

this newly available Voronoi decomposition will be presented in the following.

4.2 Theory

4.2.1 Tesselation and Voronoi Shells

Voronoi decomposition creates an ensemble of space-filling disjunct polyhedra for a given non-degenerate set of points. Each polyhedron contains all space closer to its associated point than to any other point of the given set. The faces of each of these Voronoi polyhedra are constructed by planes perpendicular to and bisecting the vectors connecting the associated point and its neighbour points. Generally, a geometrical duality between this Voronoi decomposition into polyhedra and Delaunay tessellation exists.

In principle, a Delaunay tessellation is a unique partitioning of the point set into “simplices”.^[26] In three dimensional space such a simplex is an irregular tetrahedron and its four vertices are among the set of points under investigation. These four vertices of a tetrahedron lie on the surface of a circumscribed sphere which does not contain any further point. In short this is known as the “Delaunay criterion”. The edges of the tetrahedra are the vectors connecting two points being cut by the orthogonal Voronoi faces midway. A schematic view on the duality of Voronoi decomposition and Delaunay tessellation in two dimensions is given in Fig.4.1.

As direct construction of Voronoi polyhedra is computationally demanding, the dual and much simpler Delaunay tessellation has been used extensively in the design of efficient algorithms for interpolation, contouring or mesh generation.^[25] In the literature a plethora of tessellation algorithms can be found,^{[27]–[30]} but in computer simulations of bulk media periodicity is commonly used to emulate infinite systems. Consequently, Delaunay algorithms^{[25],[29]} taking explicitly into account periodicity are of special importance. Thereby, periodicity is an essential part of the algorithm which excludes algorithms working on the primary cell enhanced by its explicit images. Among the periodic Delaunay algorithms “insertion algorithms” have proven particularly useful and were adopted in various ways.^{[25],[27],[31]} These kind of iterative algorithms insert one point at a time into the current tessellation the sequence of points being arbitrary.

In a graph-theoretical sense the Delaunay distance d is defined as the shortest path between two vertices in the tessellation. This Delaunay distance d is used to define Voronoi shells S_d where the reference particle itself is denoted by S_0 to be formally consistent with a zero Delaunay distance. A direct or “nearest” neighbour of Delaunay distance $d = 1$ may be alternatively defined as a vertex

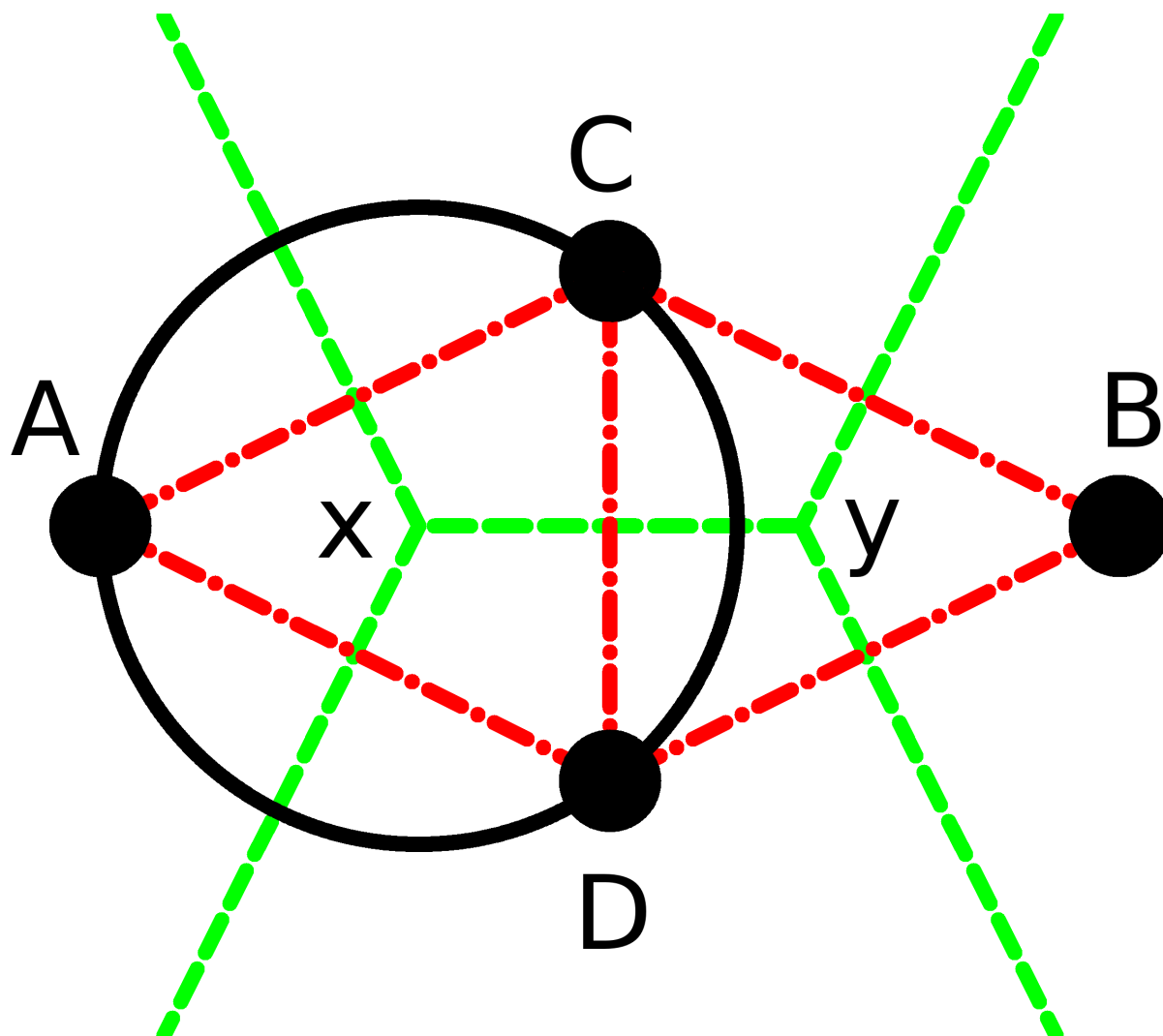


Figure 4.1: Duality of Voronoi- and Delaunay tessellations. This two dimensional scheme shows four points (A,B,C,D) and their Delaunay (red, dash-dotted) and Voronoi (green, dashed) diagrams. The Delaunay graph is the dual of the Voronoi graph which, in the planar, two dimensional case, means that it has a) a vertex for each Voronoi-face and b) an edge for each Voronoi-edge joining two neighbouring regions. The duality property is symmetric, thus Delaunay and Voronoi diagrams can be converted into each other without further information. Each Voronoi vertex or node (x,y) coincides with the Delaunay circumcenters which is indicated by a black circle. As an additional condition, Delaunay- and Voronoi edges are perpendicular to each other. In the three dimensional case a Voronoi vertex is identical to the center of the Delaunay tetrahedron's circumsphere.

which shares at least one simplex with the reference vertex. The set of all these neighbouring vertices constitutes the first Voronoi shell S_1 of the reference vertex. The second shell S_2 is comprised of all vertices that are in the first Shell of S_1 and are not already part of S_0 . In the Voronoi picture two polyhedra are first neighbours if they have one common face. Again, the first shell comprises all these first neighbours and subsequent shells are defined as the set of exterior neighbours of the preceding shell. In Fig.4.1 the minimal distance corresponds to the minimal number of lines connecting two particles symbolised as black dots.

4.2.2 Radially Resolved and Shell Grained Spatial Distribution Functions

The structure of simple liquids lacking molecular orientation is traditionally characterised by the radial distribution function (RDF). It can be generalised to molecular systems taking the reference point \mathbf{r}_i and the point of interest \mathbf{r}_j as the respective center-of-mass:

$$g^{000}(r) = \frac{1}{\rho \, 4\pi r^2 dr} \sum_{j=1}^N \langle 1 \cdot \delta(r - |\mathbf{r}_{ij}|) \rangle. \quad (4.1)$$

Here, $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ is the vector joining the reference site i and the point of interest j . $4\pi r^2 dr$ is the volume of a *spherical* shell of thickness dr and ρ is a global particle density. The product of both, volume and density, corresponds to an average particle number in that spherical shell. The center-of-mass RDF $g^{000}(r)$ measures the local deviation from that average particle number. In practice, $g^{000}(r)$ is computed as a histogram accumulating entries of 1 into a bin from r to $r + dr$. The bin selection is the computational analogue of the mathematical δ -function.

In order to account for the intrinsic orientational order of molecular liquids the center-of-mass RDF $g^{000}(r)$ has to be extended. This can be done by sampling orientational functions representing the mutual orientation of a molecular pair instead of the mere delta function. This leads to the so-called g-coefficient analysis.^[32] We will not give a detailed description here, but rather select the two specific g-coefficients

$$g^{011}(r) = \frac{1}{\rho \, 4\pi r^2 dr} \sum_{j=1}^N \left\langle \frac{\mathbf{r}_{PW} \cdot \boldsymbol{\mu}_W}{|\mathbf{r}_{PW}| \cdot |\boldsymbol{\mu}_W|} \cdot \delta(r - |\mathbf{r}_{ij}|) \right\rangle \quad (4.2)$$

and

$$g^{110}(r) = \frac{1}{\rho \, 4\pi r^2 dr} \sum_{j=1}^N \left\langle \frac{\boldsymbol{\mu}_P \cdot \boldsymbol{\mu}_W}{|\boldsymbol{\mu}_P| \cdot |\boldsymbol{\mu}_W|} \cdot \delta(r - |\mathbf{r}_{ij}|) \right\rangle \quad (4.3)$$

because of their special meaning as donor-acceptor and dielectric screening functions.

As a first step on the transition from a radial description to a pure Voronoi analysis the RDF and all higher g-coefficients may be resolved into contributions from individual Voronoi shells:^[32]

$$g_{S_d}^{000}(r) = \frac{1}{\rho \, 4\pi r^2 dr} \sum_{j \in S_d} \langle 1 \cdot \delta(r - |\mathbf{r}_{ij}|) \rangle. \quad (4.4)$$

As opposed to the full RDF, the set of neighbours j is restricted to a specific Voronoi shell S_d . Thus, $g^{000}(r)$ is decomposed into a sum over all these Voronoi shell specific contributions:

$$g^{000}(r) = \sum_d g_{S_d}^{000}(r). \quad (4.5)$$

We shall see later on (Sec. 4.4.1) that the Voronoi components $g_{S_d}^{000}(r)$ can be described to a good approximation by Gaussian functions

$$g_{S_d}^{000}(r) = A \cdot \exp \left(-\frac{1}{2} \left(\frac{r - r_O}{\sigma} \right)^2 \right) \quad (4.6)$$

as shown by Fig.4.4.

In order to find a formulation of g-coefficients completely within the Voronoi framework we introduce the concept of “shell graining” by considering “one shell as a bin”. This means, we replace the fine grained radial bin r to $r + dr$ by the complete shell S_d . This requires a further adaptation: The bulk coordination number of a spherical shell $4\pi r^2 dr \cdot \rho$ is replaced by the actual Voronoi coordination number $CN(S_d)$. Thus the shell-grained g-coefficients are given by

$$g^{110}(S_d) = \frac{1}{CN(S_d)} \sum_{j \in S_d} \frac{\boldsymbol{\mu}_P \cdot \boldsymbol{\mu}_W}{|\boldsymbol{\mu}_P| \cdot |\boldsymbol{\mu}_W|} \quad (4.7)$$

and

$$g^{011}(S_d) = \frac{1}{CN(S_d)} \sum_{j \in S_d} \frac{\mathbf{r}_{PW} \cdot \boldsymbol{\mu}_W}{|\mathbf{r}_{PW}| \cdot |\boldsymbol{\mu}_W|}. \quad (4.8)$$

and are measures of the shell-grained orientational order in the form of a weighted “sign pattern”. Shell-grained g-coefficients may be viewed alternatively as a decomposition of the radial g-coefficients given in Eqs. (4.2) and (4.3) into Voronoi shells analogously to Eq.(4.5) followed by a subsequent integration over a single Voronoi shell. In a fully atomistic tessellation neighbouring molecules usually have multiple common Delaunay edges. For the computation of the coordination number CN the shortest Delaunay edge representing the strongest interaction of neighbouring molecules is selected. One may further discriminate whether this shortest Delaunay edge points to a water hydrogen or to an oxygen. This offers an alternative way to describe donor-acceptor properties by splitting the coordination number CN into contributions from CN_h and CN_o .

4.2.3 Time Series and Time Correlation Functions

As shells are not static but evolve in time we have introduced a binary residence function for a single particle i

$$n_i^d(t) = \begin{cases} 1 & : i \in S_d \\ 0 & : i \notin S_d \end{cases} \quad (4.9)$$

depending on whether particle i is a member of shell S_d or not at time t . We emphasise that the molecular identity is conserved, i.e. reoccurrence of a particle is not recognised as the entrance of a new particle. Here, $\{n_i^d(t)\}$ is a time series characteristic of shell dynamics. In order to get a concise measure of such time series one traditionally uses time correlation functions (TCF). For the present study the essential correlation function is given by

$$C_n^d(t) = \frac{\sum_{i=1}^N \langle n_i(0)n_i(t) \rangle}{\text{CN}} \quad (4.10)$$

One term in the above sum represents the memory of a specific particle to be still a member of shell S_d after some time interval t . Summation over all particles N surrounding the reference molecule is done for statistical accuracy. In other words, $C_n^d(t)$ describes the memory of the average particle. The normalising factor also called coordination number $\text{CN} = \sum_{i=1}^N \langle n_i^2(0) \rangle$ gives the average number of particles which are member of the shell S_d . As opposed to many correlation functions which decay to zero in the long-time limit, $C_n^d(t)$ approaches a steady state $\lim_{t \rightarrow \infty} C(t) = C_{\text{steady}}$.

$$C_n^d(t) = \{C_n^d(t) - C_{\text{steady}}\} + C_{\text{steady}} \quad (4.11)$$

The characteristic time to reach this steady state is given by the average correlation time

$$\langle \tau \rangle = \int_0^\infty \{C_n^d(t) - C_{\text{steady}}\} dt \quad (4.12)$$

which may be interpreted as a mean residence time (MRT).

In order to cope with complex dynamics the concept of mono-exponential relaxation is usually generalised to a distribution $g(t')$ of relaxation times

$$C(t) = \int g(t') e^{-\frac{t}{t'}} dt'. \quad (4.13)$$

A frequently used representation of complex relaxation behaviour is the stretched exponential or Kohlrausch-Williams-Watts (KWW) function

$$C(t) = A_1 + A_2 e^{(-\frac{t}{\tau})^\beta} \quad (4.14)$$

which introduces a single additional parameter β .^{[33],[34]} A smaller value of β corresponds to a broader distribution. This mapping of a complex dynamical behaviour to a single parameter is the main reason for the wide-spread use of KWW functions. In addition, a simple analytic formula holds for the average correlation time

$$\langle \tau \rangle = \frac{\tau}{\beta} \Gamma\left(\frac{1}{\beta}\right) \quad (4.15)$$

where Γ denotes the Gamma-function representing the generalisation of the factorial to real numbers.

So far we have shown that the two parameters τ and β allow a compact description of the diversity of neighbourhood dynamics. In order to give an analogous intuitive interpretation for the other pair of parameters, namely the static amplitudes A_1 and A_2 , we start from Eq.(4.11). For the special case of a KWW function the steady state C_{steady} is identical to amplitude A_1 , because the second term in Eq.(4.14) vanishes in the asymptotic limit. Similarly, the initial value $C_n^d(0) = \sum_{i=1}^N \langle n_i^2(0) \rangle / CN = 1$ has to be identified with $A_1 + A_2$. So we have a sum rule $A_1 + A_2 = 1$ for the amplitudes. Separate expressions for the two amplitudes can be derived from the following asymptotic consideration. In a finite system of N particles, CN of which are coordinated to or in the shell of a reference particle, the probability for asymptotic residence is given by the ratio of favourable and all possibilities, i.e. by CN/N . This quotient CN/N must be a lower limit for the steady state. In KWW terms this means $A_1 \geq CN/N$ and $A_2 \leq 1 - CN/N$. This relation between the amplitudes can be also deduced from a two-state model of a Markovian master equation for neighbourhood dynamics.^[35]

4.3 Methods

4.3.1 Simulation and System Description

Our analysis of Voronoi shells is based on molecular dynamics (MD) simulations of the three proteins ubiquitin,^{[36],[37]} bovine apo-calbindin D_{9K} ,^[38] and the C-terminal SH2 domain of phospholipase $C - \gamma 1$.^{[39],[40]} We refer to these three systems by their entry code in the protein data bank (PDB): ubiquitin (PDB entry 1UBQ), apo-calbindin (1CLB) and the SH2 domain of phospholipase $C - \gamma 1$ (2PLD). Some details concerning the structure are summarised in Table I of Ref.[41]. For methodic reasons electrostatically neutral systems are essential in MD simulations. Therefore, counter ions were added in order to achieve charge neutrality. In order to compensate the odd net charge of 1CLB we have used monovalent Na^+ ions instead of the genuine divalent Ca^{++} . This was done for two reasons: First, charge neutrality is essential for the correct handling of electrostatic forces in our simulation.

Therefore, we could not compensate an odd protein net charge by divalent counter ions. Second, sodium ions are well established as counter-ions in simulations as there exist reliable parameters to describe their interaction with water and proteins. Three Cl^- ions were added to the 2PLD system. The details of the simulations may be found in Sec. B of Ref.[41].

4.3.2 Implementation of Delaunay Tessellation

As outlined in the theory section we have implemented an algorithm based on iterative point insertion^[25] which can be briefly outlined as follows: Starting from an initial tessellation, the simplex containing the new point or one of its images, the so-called “base” must be found. Second, all neighbouring simplices failing the Delaunay criterion constituting the “core” are detected in a recursive manner. Third, a visibility check is performed to ensure a convex “star-shaped” core region. Fourth, the core tetrahedra are replaced by newly constructed ones containing the inserted point. In special cases the periodic boundary conditions necessitate a recentering of the “core” in order to ensure that every tetrahedron has at least one vertex in the primary cell. For the actual tessellation we adapted this algorithm in two respects: First, our choice of initial conditions is different. We start from only one randomly chosen point and seven of its images forming a cube. This cube is subdivided into six tetrahedra serving as initial simplices. The second adaptation refers to the recentering of the “core” and is described in more detail in Ref.[32] Each trajectory frame was tessellated at the atomic level. A coarser graining representing the whole molecule by its center-of-mass was considered too crude, since essential features of anisotropy are lost in this way. A typical result of such a Voronoi analysis of solvation shells is shown in Fig.4.2 and represents a snapshot of solvated ubiquitin out of our simulation.

4.3.3 Data Analysis and Organisation

The Delaunay tessellation and corresponding derived quantities like volume, surface etc. as well as the orientation algorithm used for nuclear density 3D histograms were implemented and computed by a self written program. For example the Voronoi interface surface area (VISA) was computed in the following way: By rotating through the data structure whose entries are generated by Delaunay tessellation all tetrahedra participating in the buildup of a Voronoi surface area are found. The circumcenters of these tetrahedra are the vertices of a co-planar set of voronoi vertices forming a contiguous three dimensional polygon. The surface enclosed by this polygon is obtained by calculating and rescaling

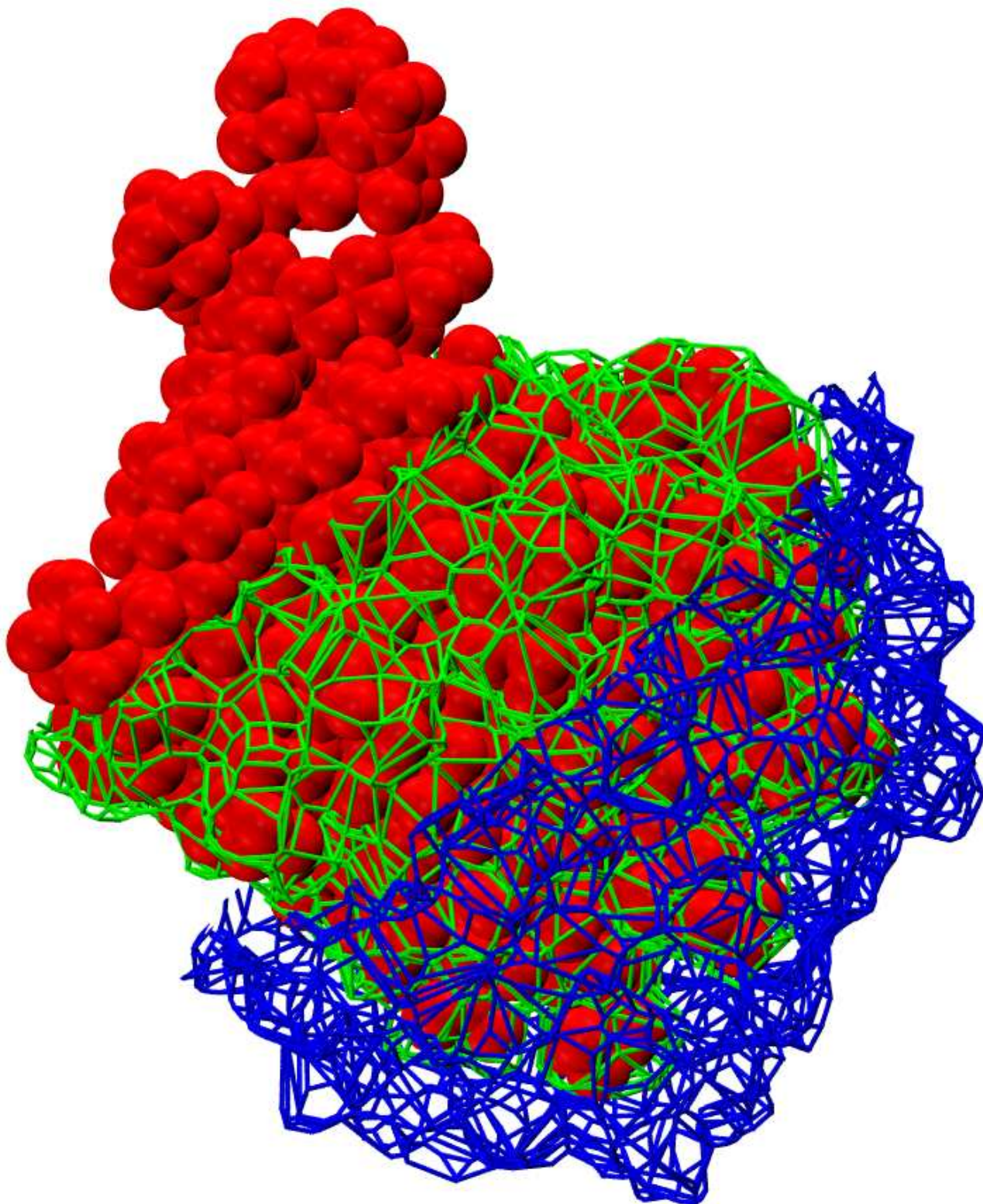


Figure 4.2: This picture shows a snapshot of our ubiquitin trajectory and its Voronoi tessellation. The red balls represent the atoms of the zero-order shell S_0 , i.e. the protein ubiquitin. The green and blue wireframes depict the Voronoi surface between S_0 and S_1 (green) or S_1 and S_2 (blue), respectively. Here, S_d is a shorthand notation for the hydration shell of order d .

the area of its two dimensional projection. Summing the area of all individual surfaces constituting a Voronoi polyhedron the net surface area is obtained. one has to keep in mind that there are inner (intramolecular) and outer (intermolecular) surface areas. The VISA is only built up by the latter ones. This quantity is important for grouping amino acid residues with respect to their solvent accessibility within the Voronoi framework. Since unpolar residues show a large diversity of solvent accessibility one has to introduce a threshold which discriminates between residues participating in the water interface and those being buried in the interior of the protein. Practical analysis has shown that a VISA threshold of 50 \AA^2 is appropriate. The volume of a Voronoi polyhedron was computed by summing the volumes of all pyramids being formed by the aforementioned polygons and the common central reference point according to the basic formula surface area times normal distance divided by three.

The construction of 3D histograms necessitates an efficient algorithm for orientational superposition. We have implemented the Horn algorithm^[42] providing a closed form solution of absolute orientation using unit quaternions. The only non self written program we have used was the DSSP package^[43] in order to compute the solvent accessible surface area (SASA) and to classify secondary structure elements.

Irrespective of the origin of the data we collected and organised them in a postgresQL database. In this way we were able to manipulate and to correlate data at different levels of granularity ranging from the atomistic level over functional groups and whole residues to the full protein level. Voronoi volumes, VISA and coordination numbers were computed and organised at the atomistic level. Mean residence times, SASA and secondary structure information were held at the residue level. In a “master view” both data sets of differing granularity were united at residual resolution which allows an interpretation of data at any level equal or coarser than that of residues.

4.4 Results and Discussion

A typical slice out of the averaged three dimensional nuclear density as shown in Fig.4.3 serves as a starting point for the principle structure of this section. This nuclear density may be seen as the analogue of the familiar electron density where the broadening comes from averaging over nuclear motion. In this slice of the nuclear density the formation of a first shell can be clearly detected. Even more, a close correspondence of the first Voronoi shell (green) and structured region in the nuclear density (black) is observed. This shows that the rational device of the Voronoi algorithm is in

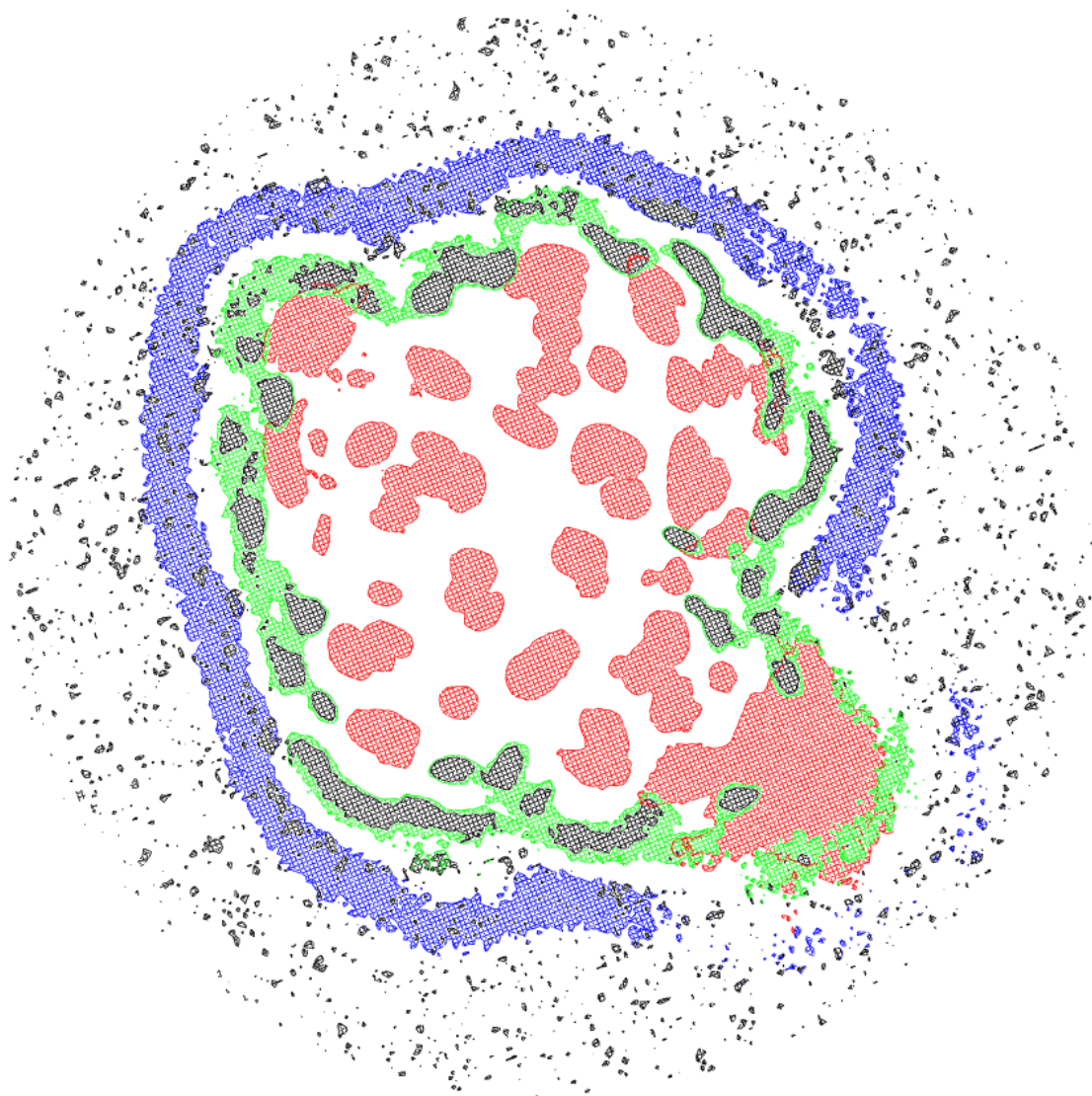


Figure 4.3: A slice out of the three dimensional nuclear density map of 1UBQ (red, density threshold 1.0) and the surrounding water (gray, density threshold 1.2). Again, the first Voronoi shell is displayed in green and the second shell in blue, both at a density threshold of 0.75. The lower right part shows a region of high local mobility where the shell structure is smeared out.

accordance with an intuitive view of solvent structure. A second shell, however, is hardly discernible in the nuclear density. In fact, it needs the Voronoi labelling (blue) to identify an integer second shell. This is due to the beginning homogeneity in certain regions. It is almost impossible to figure out further shells from the nuclear density. Therefore this can only be done in the framework of the Voronoi algorithm. In addition to this blurring of distant shells the nuclear density is plagued with the problem of thresholds. In fact, a change of the threshold value changes the overall picture and as a consequence the resolution of solvation shells. As we have learned that the nuclear density and the Voronoi algorithm match quite closely in the first shell, the parameter-free Voronoi algorithm appears as a general rational device to classify solvation structure. It is not only a natural extension of the threshold dependent nuclear density, but offers the possibility to derive a variety of properties characterising solvent structure in a compact way. Thereby anisotropy, inhomogeneity and flexibility are important features. Obviously, anisotropy of the protein induces anisotropic hydration shells exceeding the simple picture of spherical shells. While there is a general trend for homogenisation with increasing shell number there are still inhomogeneities in the second shell S_2 . The still higher inhomogeneity of the first shell S_1 is a result of the specific hydrogen-bond pattern of the protein water interface and already justifies a more detailed analysis at the local level of individual amino acid residues. This distinction between a global and a local analysis is even more accentuated when considering the influence of flexibility. In certain regions, especially in the highly flexible C-terminal part (lower right corner of Fig.4.3), a considerable dynamic broadening is observed in all three shells S_0 to S_2 . This may have two sources: First, the dynamics is enhanced on an absolute scale. Second, the choice of the set of reference points is inappropriate for the special dynamics occurring in this specific region. This again points to the necessity of a local zooming. The composition of each protein in terms of specific amino acids provides a further argument for an additional local analysis. Combining all arguments a decomposition into a global protein centred and a local residue resolved analysis seems reasonable.

4.4.1 Global Protein Analysis

Geometrical Properties and Packing

The traditional geometrical measure of solvation is the so-called solvent accessible surface area (SASA). It is interesting to compare this frequently used solvation parameter to the Voronoi interface surface area (VISA) between the protein S_0 and the first shell S_1 . Table 4.1 shows that these two values

		1UBQ	1CLB	2PLD
Voronoi interface surface area of S_0	VISA [\AA^2]	4942	5221	8358
solvent accessible surface area	SASA [\AA^2]	5011	5347	8056
	SASA/VISA	1.014	0.976	0.964
coordination number	CN	503	525	793
	VISA/CN [\AA^2]	9.83	9.94	10.54

Table 4.1: The newly defined Voronoi interface surface area (VISA) between the protein S_0 and the first solvation layer S_1 is in good accordance with the well known SASA. In fact the actual values differ by a few percent only. The ratio of VISA and the coordination number CN shows a trend within a certain range which is discussed in the text in more detail.

are pretty close for all three proteins differing by a few percent only. Furthermore, correlating the individual SASA or VISA of all residues in all proteins gives a correlation coefficient close to 1. A further interesting correspondence between SASA and VISA is provided by the coordination number. According to Kabsch and Sander^[43] the ratio of the SASA and the estimated average number of water molecules W is given by $\text{SASA}/W = 9.65$. The actual value calculated as VISA/CN differ from the empirical value 9.65 by less than 9% (see Table 4.1). Within this small variation the value for 1UBQ (9.83) is lowest, followed by a slightly higher value for 1CLB (9.94) and an elevated value for 2PLD (10.54). This behaviour correlates with the sequence of compactness of the three proteins. While the volume scales with the number of amino acids, the VISA of 1CLB is higher than that of 1UBQ, although it consists of 75 instead of 76 residues. The elevated value for 2PLD can be explained by the free C-terminal end. It remains to be seen if the more detailed analysis at the local level offers an alternative explanation, e.g. the ratio of hydrophobic and hydrophilic amino acids. Altogether, this shows that VISA is a much more general measure of global solvation, but resembles the behaviour of SASA for the first solvation shell S_1 . In contrast to SASA, however, which is restricted to the first shell, VISA can be computed for the complete sequence of solvation shells as a direct result of the Voronoi decomposition. These values are given in Table 4.4.1 together with the volume of the solvation polyhedra VSP and their number of faces, i.e. the coordination number CN. The “generalised” Kabsch-Sander ratio $\text{VISA}(n)/\text{CN}(n)$ given in Table 4.4.1 increases monotonically with the shell number. Indeed, a linear regression gives an offset of 9.45 and a slope of 0.45 for 1UBQ, an offset of 9.66 and a slope of 0.29 for 1CLB and for 2PLD the respective values are 10.28

	shell index	1UBQ	1CLB	2PLD
protein volume [\AA^3]	0	11253	10969	15918
	1	14885	15015	23598
	2	22016	21458	30938
shell volume [\AA^3]	3	31539	30697	41557
	4	42653	41539	53935
	5	52617	53708	63600
VISA [\AA^2]	0-1	4942	5221	8358
	1-2	7669	7480	11106
	2-3	11325	11018	15232
VISA [\AA^2]	3-4	15706	15228	20110
	4-5	20328	20188	25306
	5-6	23309	24263	26143
	1	503	525	793
	2	734	728	1029
CN	3	1049	1039	1373
	4	1410	1415	1799
	5	1735	1807	2180
	1	9.83	9.94	10.54
	2	10.45	10.27	10.79
VISA/CN [\AA^2]	3	10.80	10.60	11.09
	4	11.14	10.76	11.18
	5	11.72	11.17	11.61
	1	1.011	1.046	1.005
	2	0.997	1.015	0.995
density [g/cm^3]	3	0.995	1.013	0.988
	4	0.989	1.019	0.998
	5	0.986	1.006	1.025

Table 4.2: This table lists the volumes, VISA and coordination numbers CN of the three proteins 1UBQ, 1CLB and 2PLD (S_0), the suprasolute S_0 plus first Voronoi shell S_1 , and higher suprasolutes. In addition, the ratio VISA/CN and the density of the first five solvation layers are given.

and 0.25. This is in accordance with the slight decrease of the shell density also given in Table 4.4.1. Both facts show that the ordering influence of the protein decreases with increasing shell index. The principal architecture of the protein also influences the coordination number: While helical regions show a minimum average coordination number of $CN = 8.63$ per residue, the coordination number for beta sheets is slightly enhanced to $CN = 9.52$. This behaviour may be explained by the stronger intramolecular hydrogen-bonded network in case of helical structures which reduces the possibilities for intermolecular hydrogen bonds. Hydrogen bonds not bridging adjacent strands are free for solvent hydration. For residues not participating in these two fundamental secondary structure elements, a higher average coordination number of $CN = 10.47$ is found.

radial distribution resolved into Voronoi shells

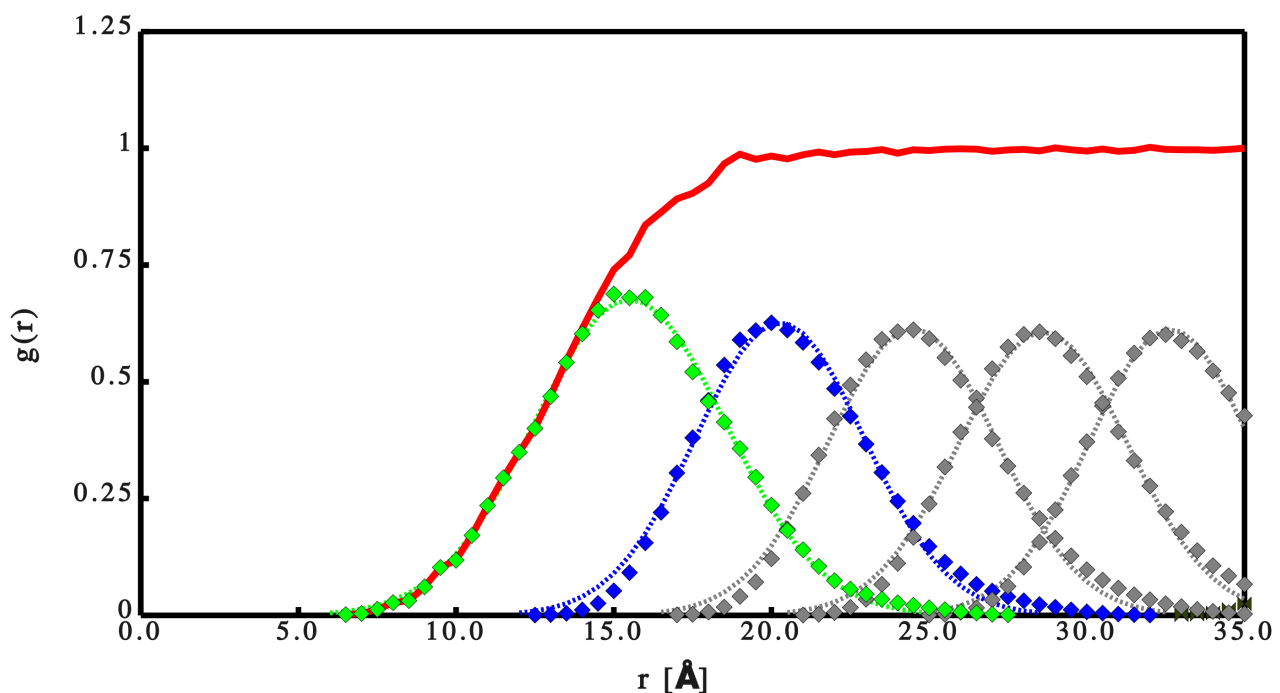


Figure 4.4: Radial distribution function $g^{000}(r)$ of water oxygen around the center-of-mass of 1CLB (red solid line) and its decomposition according to Eqs. (4.4) and (4.5) (diamonds). The decomposition $g_{S_1}^{000}(r)$ for the first Voronoi shell is depicted in green while its second shell analogon $g_{S_2}^{000}(r)$ is blue and further shells are displayed in gray. The dotted lines show the respective Gaussian fits (Eq.(4.6)).

Asymptotically, the shape and charge distribution of a protein is more and more screened by the solvent and thus becomes more and more spherical. In the vicinity of the protein, of course, quite the

opposite is true. In order to judge the extent of anisotropy we have decomposed the traditional radial distribution function (Eq.(4.1)) into Voronoi shells (Eqs. (4.4) and (4.5)) as depicted in Fig.4.4. Apart from an initial rise the RDF is completely unstructured because the detailed Voronoi shells compensate each other upon superposition. Neglecting the slight skewness of the Voronoi shells they can be fitted by Gaussian functions mapping the detailed radial distribution of a shell to three parameters: The amplitude A , the radial offset r_O and the spread σ collected in Table 4.4.1. With the exception of the distinguished first shell the spread σ is almost constant whereas the peak height decreases slightly for 1UBQ and 1CLB. This is in accordance with the decrease of solvent density already observed in Table 4.4.1. For 2PLD the opposite trend is found: The peak height or amplitude A as well as the solvent density increases for distant shells. The slight deviation from monotony goes along with the slight variation of the spread σ . This is again a consequence of the free C-terminal end.

From the radial offset of the Voronoi shells r_O one can derive a shell thickness Δr_O taking the difference of the r_O values of two adjacent shells. It is interesting to compare them to the differences in the effective spherical radii r_{eff} obtained by equating the total volume of a Voronoi shell and its subshells to the volume of a sphere. These two sets of values are given in the last two blocks of Table 4.4.1. Although both sets of values are shifted by one half of the shell thickness their high similarity supports the concept of a hydrodynamic shell diameter.

Shell grained orientational order

The essential modes of motion of small solvent molecules are translation and rotation. While translation is responsible for the packing in solvation shells, rotation effects the orientational ordering of the solvent molecules with respect to the solute. Two important features closely related to orientational ordering are dielectric screening and the donor-acceptor property. A measure for dielectric screening may be the average cosine of the angle between the dipole moment of the protein μ_P and the respective solvent water μ_W . According to the formalism worked out in the theory section we use the term $g^{110}(S_d)$ for the value of $\cos(\angle(\mu_P, \mu_w))$ averaged over shell S_d . The donor-acceptor property may be characterised by $g^{011}(S_d)$ the shell averaged value of $\cos(\angle(\mathbf{r}_{PW}, \mu_w))$ representing the angle between the water dipole and the vector joining water with the protein center. Fig.4.5 compares the donor-acceptor property of the three proteins differing with respect to their overall charge. While 1UBQ is neutral, 1CLB and 2PLD carry a net charge of -7e and +3e, respectively. For the negatively charged 1CLB the averaged angle between the water dipole μ_w and the vector joining molecular centres \mathbf{r}_{PW} is beyond 90 degrees. In other words the two vectors are antiparallel on the average typical for a

	shell index	1UBQ	1CLB	2PLD
amplitude A	1	6.55	7.57	9.47
	2	5.76	6.16	6.86
	3	5.67	6.02	6.36
	4	5.65	5.98	6.22
	5	5.59	5.9	5.86
offset r_O [Å]	1	15.53	15.31	16.86
	2	19.85	20.07	22.69
	3	23.96	24.28	27.22
	4	28.07	28.42	31.5
	5	32.15	32.51	35.53
spread σ	1	3.76	4.3	6.55
	2	3.51	3.77	5.08
	3	3.57	3.76	4.58
	4	3.65	3.79	4.39
	5	3.66	3.72	4.03
effective radius r_{eff} [Å]	1	4.51	4.59	5.52
	2	4.16	4.08	4.49
	3	4.13	4.06	4.28
	4	4.1	4.05	4.19
	5	3.9	4.02	3.89
Δr_O [Å]	1	4.32	4.75	5.83
	2	4.11	4.22	4.53
	3	4.11	4.14	4.27
	4	4.08	4.08	4.03
	5	3.63	3.56	3.52

Table 4.3: Geometrical properties and packing using Voronoi decomposition of radial distribution. Fit parameters A, r_O and σ are used to define a mesoscopic effective radius r_{eff} as well as a shell thickness Δr_O . All values are given for the three proteins 1UBQ, 1CLB and 2PLD.

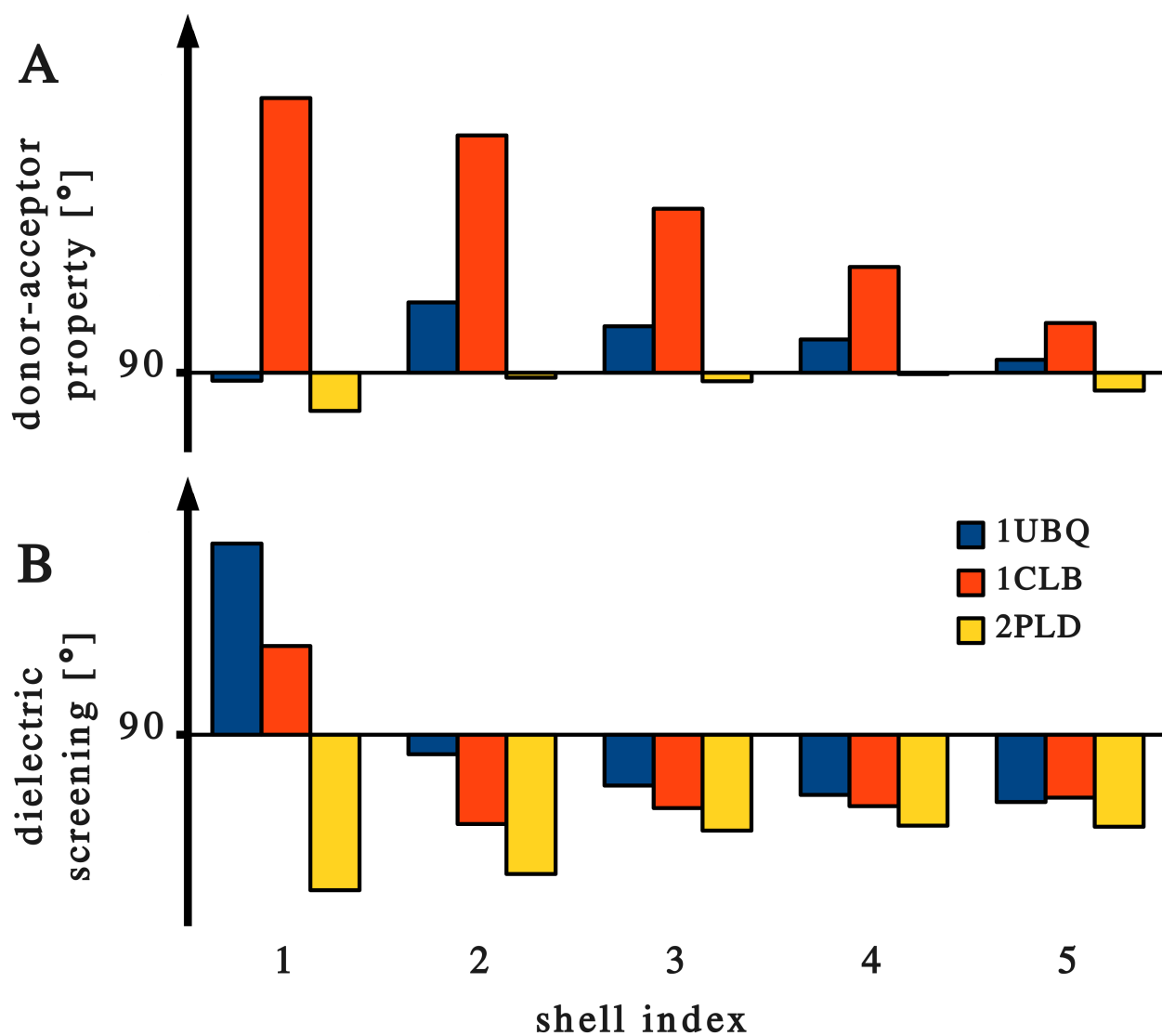


Figure 4.5: Structural orientation of water at the global protein level. The blue (1UBQ), red (1CLB) and yellow (2PLD) bars indicate parallel (angle below 90 degrees) and antiparallel (angle above 90 degrees) orientation of the first five solvation shells. The donor-acceptor property A) resembles the proteins' charge distribution. Part B) illustrates the special role of the first solvation layer in dielectric screening.

donor configuration of the water molecules pointing their positively charged hydrogens towards the negatively charged protein. This pronounced donor configuration in the first shell is followed by a donor configuration of second shell water molecules pointing towards the negative oxygen end of the first shell. This cascade of donor configuration exists up to the fifth shell and beyond. It demonstrates the long-range ordering influence of the protein on the solvent structure. Changing the charge status of the protein this should be reversed to an acceptor cascade shifting the average angle below 90 degrees. This is indeed observed for 2PLD although the effect is less pronounced due to motional averaging of the C-terminus. The neutral 1UBQ holds a middle position between these two charged proteins.

While the donor-acceptor property is uniform in all shells - with a clear damping of distant shells - the dielectric screening or average dipole orientation (see Fig.4.5 part B) shows a peculiar short-range effect: Water dipoles in the first solvation shell of 1UBQ and 1CLB are antiparallel to the protein dipole, thus quenching the net dipole moment of the “suprasolute” protein plus first shell. In subsequent shells water dipoles are mainly orientated in parallel, although the effect is small due to the quenched influence of the suprasolute. Again, 2PLD is an exception in that water dipoles in its first shell are parallel to the protein on average. This pattern in sign, negative for 1UBQ and 1CLB and positive for 2PLD, correlates with the dielectric spectra evaluated on the basis of spherical shells in a previous work (see Table IV of Ref. [24]). The transition from the clear sign pattern of the first shell to the slightly positive distant shell differs between Voronoi and radial shells: With the Voronoi method the transition to positive values is already observed in the second shell, but it seems to be postponed to the third shell when using spherical shells. Altogether we observe that shell-grained as compared to radially resolved orientational order leads to identical conclusions, at least for donor-acceptor properties and dielectric screening. This confirms the general concept: less is more.

Ion density

So far we have focussed exclusively on the water solvent. Due to the charge status of 1CLB and 2PLD information on the distribution of counterions is interesting, especially for calbindin because of its intrinsic biochemical function as a ion binding protein. Analogous to the 3D nuclear density already introduced above, ion residence can be described by a specific 3D ion density. In contrast to the water 3D density the variation within this ion density is extremely large with differences up to two orders of magnitude. This renders the selection of one specific threshold impracticable. Rather one has to screen a range of thresholds to figure out general features of ion distribution. Starting with a high

threshold dominant regions are selected. Upon reduction of the threshold more and more “islands” of concentrated ion density emerge. Roughly speaking, this set of islands may be divided into three groups corresponding to the ratio of thresholds 200:4:1. In literal terms these three thresholds might be described as resident, transient and freely moving. It is interesting to note, that the resident region coincides with the two (ASP 54 and ASP 58) of the four binding sites^[44] of calbindin. At closer inspection, this resident region splits up into two sub regions: One corresponding to the extremely high residence and an adjacent one characterised by one third of the density only. This splitting into two sub regions of different residence may be explained by the substitution of the divalent calcium ions by the monovalent sodium ions which was done for methodological reasons as described in the method section. Sodium ions occupy a lot of space in the binding site but cannot deliver the desired amount of charge for compensation. Therefore the sodium charges in the binding site have satellites outside. These satellites provide the amount of compensating charge but for steric reasons can only reside in the proximity of the binding site. The difference between the transient ions and the freely moving ones is due to the directing influence of the terminal groups of the charged residues GLU17, GLU11, ASP19, GLU51, GLU4, GLU26, GLU27, GLU48 and GLU35. Using the criterion of proximity to the binding site Denisov and Halle have also addressed a set of charged residues.^[44] Despite of these rather different criteria, the set of residues are in qualitative agreement. So far we have focussed on the ion density in the immediate neighbourhood of the protein, so to say in the “first shell”. In analogy to the pronounced shell structure of the water solvent one might ask for a second or third shell in ion density too. Thereby, the question whether a high ion density in the first layer is correlated with a secondary sparse region, is of interest. This question can be answered in an indirect way, the ion density in the second layer is spread over a vast spatial region, but lacks typical “holes” or sparse regions.

Mean Residence Time (MRT)

As the relaxation functions from which mean residence times are deduced can be excellently fitted to Kohlrausch-Williams-Watts functions (for the actual parameters see Table 4.4), the behaviour of global shell relaxation can be described concisely by the average relaxation time $\langle\tau\rangle$ as well as by the parameter β responsible for the complexity of relaxation processes: The more distant the shell, the greater the diversity of relaxation times, or equivalently the smaller the parameter β . In other words distant shells seem to offer a greater diversity of relaxation channels. Despite of the rather different nature of the three proteins, the parameter β monitoring the spread in relaxation times is pretty constant. The inverse average relaxation time $\lambda = 1/\tau$ is an almost linear of the shell index.

	shell index	1UBQ	1CLB	2PLD
$CN \cdot A_0$	1	31	33	63
	2	61	62	103
	3	127	126	190
	4	234	232	321
	5	364	380	447
$CN \cdot A_1$	1	457	473	694
	2	672	673	924
	3	917	911	1181
	4	1161	1160	1454
	5	1396	1414	1654
τ [ps]	1	47.5	51.2	59.6
	2	11.8	12.0	11.2
	3	6.9	7.0	6.5
	4	5.2	5.1	4.8
	5	3.9	3.9	4.2
β	1	0.62	0.62	0.59
	2	0.42	0.42	0.41
	3	0.41	0.42	0.42
	4	0.44	0.44	0.43
	5	0.39	0.38	0.38
$\langle \tau \rangle$ [ps]	1	68.3	73.8	92.1
	2	34.9	34.8	34.4
	3	21.3	20.9	18.6
	4	13.7	13.3	13.6
	5	14.1	14.9	16.8

Table 4.4: Relaxation of solvation in Voronoi shells S_1 to S_5 . In order to allow for an intuitive interpretation, the amplitudes A_0 and A_1 are multiplied by the coordination number. Complexity of relaxation, reflected by the parameter β , is lowest for closer solvation layers (higher β). The average residence time $\langle \tau \rangle$ shows an inverse linear correlation with the shell index leading to a retardation factor of 5 for the fifth shell.

Alternatively, one might say that the product of the average relaxation time and the shell index is constant for the first five shells. The agreement is reasonable for 1UBQ, 1CLB and less accurate for 2PLD. A possible explanation might be that the directing influence of the protein dipole decreases with the inverse third power of the distance while the number of interacting water dipoles increases with the second power of the distance. Altogether one might have a dependence on the inverse distance. For a sequence of shells of constant thickness the average distance is a linear function of the shell index. This might explain the linear dependence on the shell index.

In experiments the slowing down of solvent motion in shells close to the protein is usually described by a so-called retardation factor measuring the enhancement of relaxation times within the first solvation shells as compared to the bulk. For a set of proteins Halle^[7] found retardation factors ranging from 5 to 8. The linear dependence of the MRT on the shell index found in this study would imply a retardation factor of 5 between the first and fifth shell. Our shell resolved donor-acceptor function shows that the directing influence of the protein extends beyond the fifth shell. Therefore, an enlargement of the retardation factor is to be expected and a factor of 8 seems plausible. This finding has an important implication for the system size used in practical simulations: The system size, i.e. the number of water molecules, should be chosen as to permit the undisturbed formation of at least five, better eight solvation shells in order to approach bulk behaviour.

4.4.2 Local Residue Analysis

While the global analysis presented above already gave a general view, it also indicated a considerable increase of information when resolving the protein-solvent interaction into residue specific contributions. When zooming the solvation structure and dynamics in this way, we will focus on three types of properties: Residue-specific VISA, shell-grained packing and orientational order as well as residue specific mean residence times.

Residue specific VISA

Our definition of the VISA being based on an atom tessellation is additive by construction. In other words, the complete VISA of a solvation shell is the sum of the area of all faces bisecting and being perpendicular to the vector joining a pair of a protein and a water atom (first shell) or between two atoms of a pair of water molecules (second and higher shells) in two different shells. In the following these atomic faces will be called “outer faces”. Thus, one can uniquely define an atom specific VISA

as the sum of all areas of outer faces a specific atom. The next step to define a residue specific VISA is quite simple. One has to sum over all atoms constituting the residue. A typical time series of

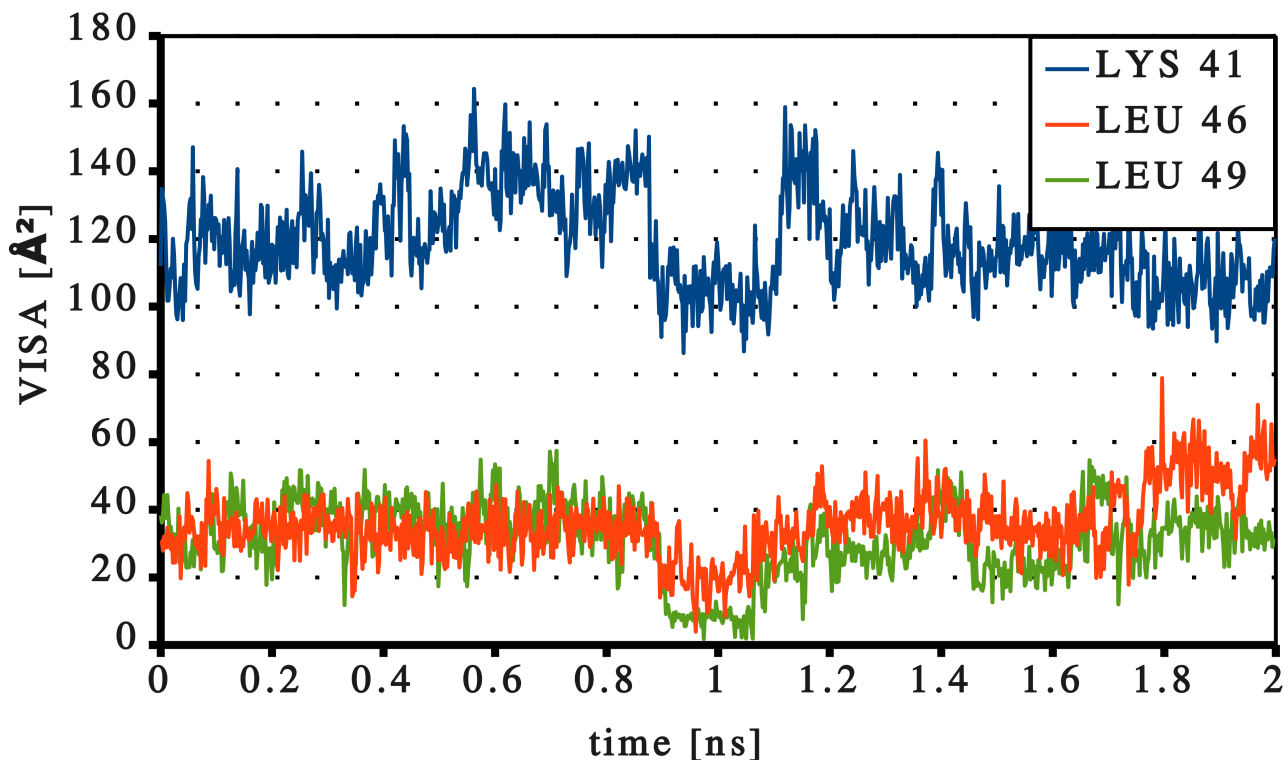


Figure 4.6: Structural orientation of water at the local amino acid level for the first five solvation shells. The donor-acceptor property (A) and dielectric screening (B) are shown for one representative of each charge and polarity group. Negatively charged (ASP, blue) positively charged (ARG, red) and polar (ASN, yellow) amino acids show different behaviour. (see text for details)

a set of residues LYS 41, LEU 46 and LEU 49 of 1CLB are displayed in Fig.4.6. The hydrophilic and hydrophobic character is clearly visible from the difference in size of the residue specific VISA. Nevertheless, the VISA of all three amino acids flip in a cooperative manner within the time window of 880 to 1120 ps. In fact, the actual transition from a high to a low VISA and vice versa occurs within a few ps. The observed cooperativity of the VISA flip goes along with a spatial proximity of these three amino acids. The change of the VISA as such has to be interpreted as the loss or gain of solvent contacts or equivalently as the submergence or emergence of amino acids as a result of local protein dynamics and its coupling to solvent motion.

The cooperative flip-flop in Fig.4.6 is a typical example of the time evolution of residue specific VISA. There exists a variety of individual residues with a flip of the VISA between distinct levels

(states). Amino acid LEU 40 in 1CLB flips from a level of 80 \AA^2 to a level of 50 \AA^2 and after a short period of 240 ps flops back to 80 \AA^2 . This flip-flop is only one type of observed motion. There also occur single flips or flops. For example, LYS 72 in 1CLB flips down from 110 \AA^2 to 80 \AA^2 , while ASP 19 in 1CLB flops up from 60 \AA^2 to 80 \AA^2 . A double flip is observed for ARG 18 in 2PLD from 190 \AA^2 to 170 \AA^2 followed by a second jump to 150 \AA^2 .

It is interesting to see that the relative jump in all these cases is 20 or 30 \AA^2 which correlates with the number of hydrogen-bond acceptors or donors, respectively. Lysine with its three hydrogen donors flips by $3 \cdot 10 \text{ \AA}^2$ whereas the two hydrogen acceptors of aspartate lead to relative flip of $2 \cdot 10 \text{ \AA}^2$. The common factor of 10 \AA^2 is the typical contribution of a water molecule to the protein water interface. The rapid flip, i.e. the change of the VISA within a few ps, points to a collective disruption or formation of hydrogen bonds. The time series of residue specific VISA provides a highly detailed information of side chain dynamics. A much simpler and more compact parameter would be the static, time-averaged VISA of a residue. In order to enhance the statistical accuracy we have further averaged over all amino acids of the same type and over all three proteins. These VISA values are listed in the third column of Table 4.5 which also gives the average coordination numbers of an amino acid in the fourth column. The ratio of both quantities, i.e. the number of water molecules per unit VISA is given in the last column. This represents a resolution of the overall Kabsch and Sander^[43] factor into amino acid specific values. Although a diversity of values is observed, they can be collected into three groups: Charged and polar, usually termed hydrophilic, small size hydrophobic and large size hydrophobic. The borderline between hydrophobic and hydrophilic amino acids is represented by histidine. Values of VISA/CN range from 8.5 to 9.9 for hydrophilics, from 10.25 to 14.6 for small sized hydrophobics and 17.9 to 19.8 for the third group containing large ILE, PHE and TRP. Cysteine with the largest ratio of 20.15 is exceptional because only a single residue appears in all three proteins. The appearance of proline in the group of the hydrophilic amino acids is fortuitous and goes along with its exceptional position among amino acids. The lower ratios for the hydrophilic amino acids originates from their higher attractivity for water molecules thus limiting the space available at the surface.

Averaging the ratio VISA/CN weighted by occurrence gives a mean value of 11.28. This offers an alternative way for selecting amino acids into two groups with a ratio above or below this threshold of 11.28.

The number of amino acids belonging to one of these two groups is somewhat different in the three proteins: In the small ratio group we have 50 for 1UBQ, 36 for 1CLB and 52 for 2PLD. The

residue type	occurrence	VISA [\AA^2]	CN	VISA/CN [\AA^2]
ASN	9	98.3	11.6	8.5
GLY	18	52.4	6.2	8.5
GLU	29	87.8	10.1	8.7
ASP	12	70.8	8.0	8.9
THR	11	69.2	7.3	9.5
PRO	11	85.7	9.0	9.5
GLN	14	84.4	8.8	9.6
LYS	24	120.6	12.4	9.7
SER	19	61.8	6.3	9.8
ARG	13	131.4	13.3	9.9
HSD	6	85.4	8.3	10.2
ALA	10	47.7	3.9	12.2
LEU	29	40.7	3.0	13.4
MET	5	82.4	6.0	13.6
TYR	8	71.4	5.0	14.2
VAL	12	29.4	2.0	14.6
ILE	14	32.9	1.8	17.9
PHE	10	51.3	2.7	19.1
TRP	1	45.3	2.3	19.8
CYS	1	34.1	1.7	20.1

Table 4.5: The amino acid specific ratio VISA/CN suggests an explanation for the trend observed at the global protein level. A protein containing more large hydrophobic amino acids and less charged or polar amino acids shows a higher global VISA/CN ratio.

complementary numbers are 26, 39 and 53. This explains why the ratio VISA/CN for the whole protein (cf. last row of Table 4.1) is smallest for 1UBQ, still below the mean value for 1CLB and close to the mean value for 2PLD.

Shell grained orientational order and solvent contact analysis

Analogously to the global protein analysis we have not resolved orientational order on a fine-grained radial scale. Rather we preferred the concept of shell graining again considering “one shell as a bin”. Like in the previous section, data were again averaged over all residues as well as over all three

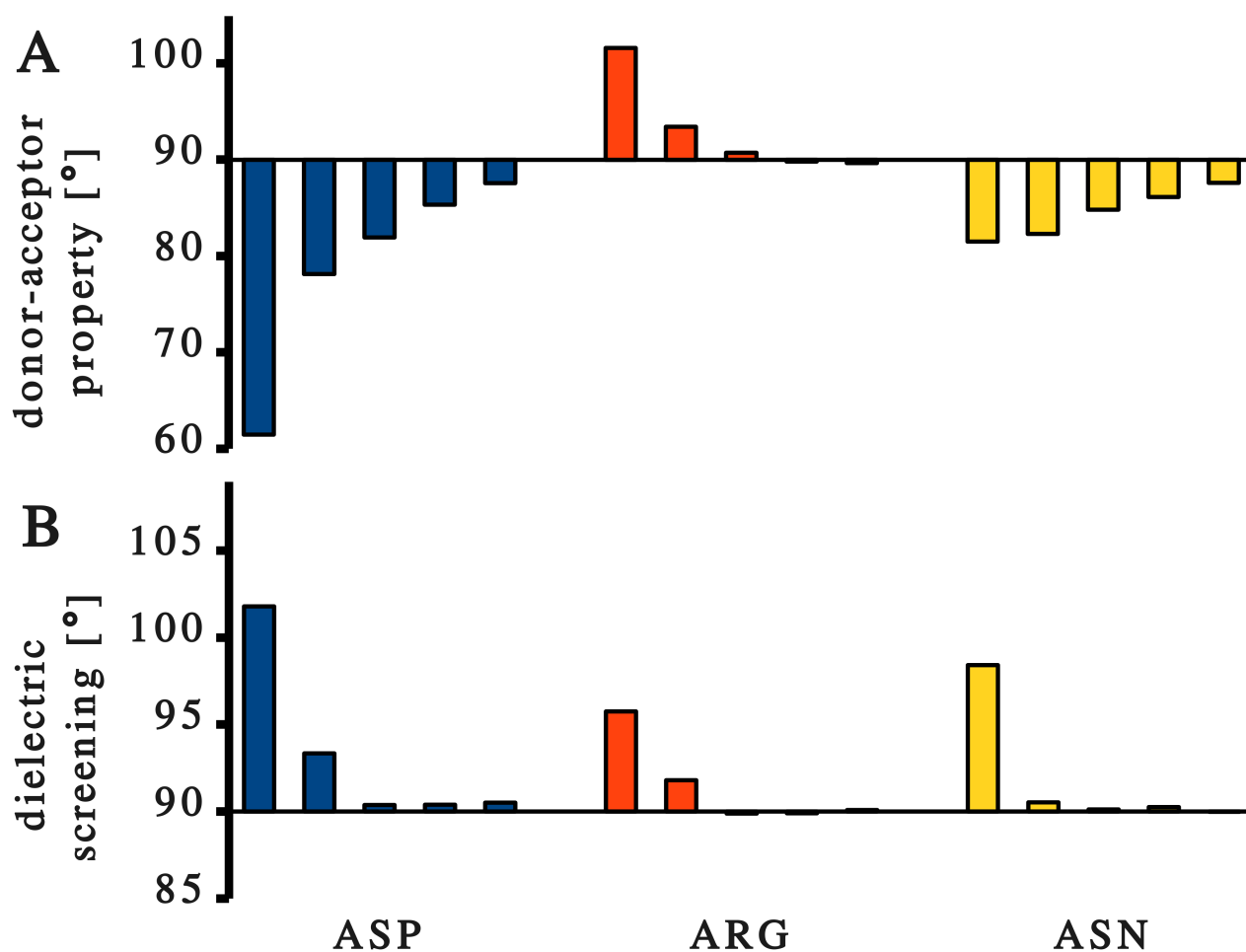


Figure 4.7: An example for a 220 ps VISA flip-flop. The three spatially close amino acids LYS 41 (blue), LEU 46 (red) and LEU 49 (green) of 1CLB show simultaneous changes in their residue specific VISA at the region around 1 ns.

proteins. Comparing the two parts of Fig.4.7 it is generally found that the donor-acceptor function

$g^{011}(S_d)$ shows stronger correlations than the dielectric screening function $g^{110}(S_d)$. The characteristic behaviour of the shell-grained donor-acceptor function $g^{011}(S_d)$ and the dielectric screening function $g^{110}(S_d)$ can be divided into four groups. The negatively (ASP, GLU) and positively (ARG, LYS) charged amino acids form two distinct groups. Polar amino acids (ASN, SER, THR, GLN, TYR) may be collected into a third group. For each of these three groups one representative (ASP, ARG, ASN) are given in Fig.4.7. The remaining group of unpolar amino acids is not given because one would expect no preferred correlations. However, this can only be true for an isolated unpolar side chain. Actually, neighbouring polar or charged amino acids strongly influence the solvent structure and this environmental effect indirectly causes correlations for unpolar amino acids too. Of course, these indirect correlations are small. The rare occurrence of CYS, HSD and MET does not allow a meaningful calculation of shell-grained orientational order for these amino acids.

The strongest donor-acceptor effect is observed for the negatively charged amino acids. In particular, the first shell is rather pronounced. The subsequent shells behave similar to the polar group. The dominant role of the first shell is also visible for the positively charged residues but correlations in the subsequent shells are marginal in this case. A possible explanation for the stronger correlations exerted by negatively charged amino acids as compared to positive ones might be the linear geometry of the hydrogen bond. In case of ASP and GLU the carbonyl oxygens of the amino acid are part of a linear O..H-O arrangement which limits the rotational mobility of the water molecule. On the contrary, the linear N-H..O arrangement typical for positively charged amino acids leaves more rotational freedom for the solvent hydrogens facing away. Once this special effect of hydrogen bonding has come into action, the adduct of the carbonyl group and the OH water bond seems to behave as an effective dipole for the subsequent shells. This cascading influence seems to prevail even at the global protein level already discussed in Sec. 4.4.1.

When considering the dielectric screening function $g^{110}(S_d)$, the exceptional role of the first shell is recognised as a common feature with the donor-acceptor function. Moreover, this dominance of the first shell is found for all groups not only for charged amino acids. A second uniform feature is the preferred parallel alignment of the residue and the solvent dipoles. This shows that the dielectric screening of the whole protein is not brought about by a small set of amino acid types but seems to be a cooperative action of all charged and polar amino acids. For a perfectly screening first shell one would expect marginal dipolar correlations in the subsequent shells. This is indeed the case for polar amino acids as visible in Fig.4.7 whose residue dipole is perfectly screened by the first layer of the solvent. The directing influence of charged amino acids cannot be completely compensated by the first

solvent layer and some remnant features are observed. As an alternative to the angular criteria of the

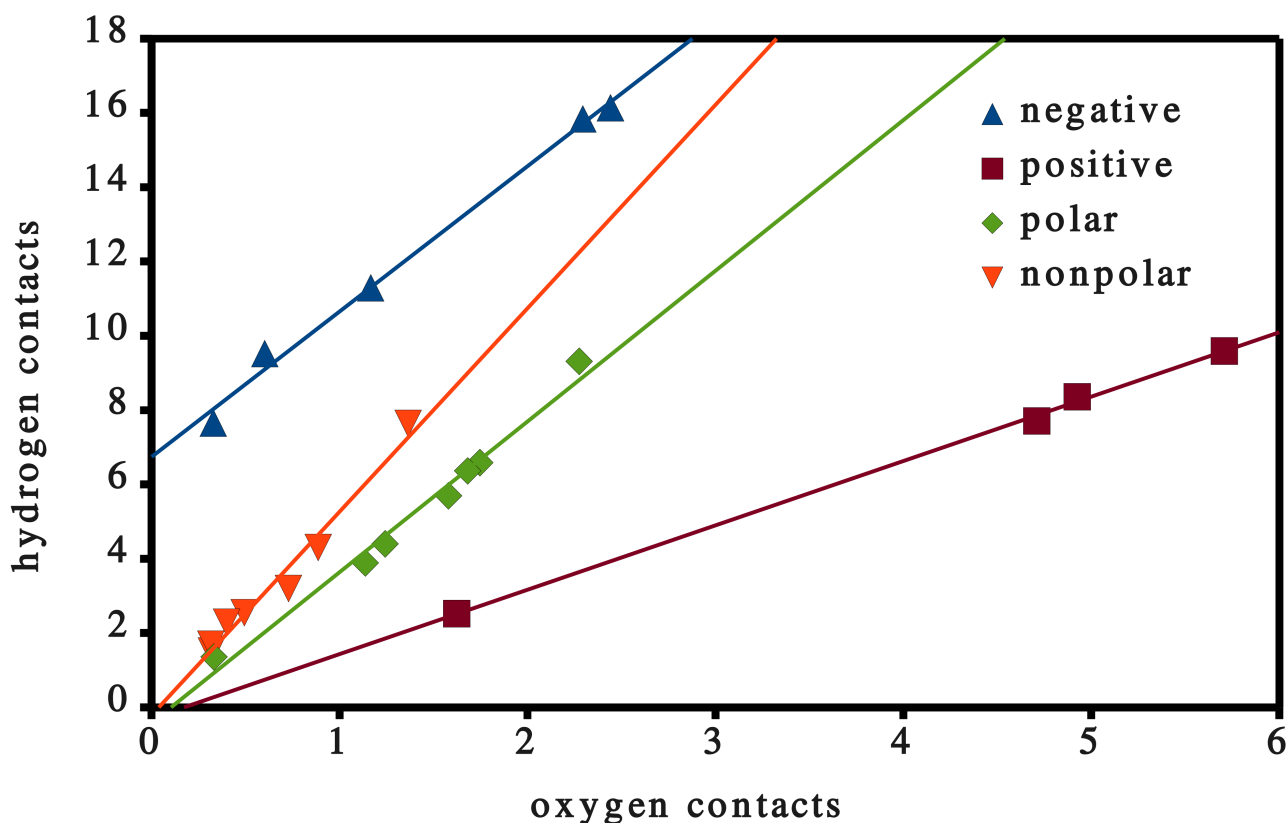


Figure 4.8: Hydrogen/oxygen contact analysis. Amino acids of different charge and polarity status separate almost perfectly on straight lines. Negatively charged amino acids (blue triangles) show more hydrogen contacts indicating their hydrogen bond acceptor role while water molecules residing at positive amino acids (brown squares) are more oxygen oriented. The ratio CN_H/CN_O is less articulate for polar (green diamonds) and nonpolar (orange triangles) amino acids.

donor-acceptor function, one may split the residue specific coordination number CN of the first shell into hydrogen CN_H and oxygen CN_O contacts as already outlined in the theory section. In Fig. 4.8 every amino acid is represented by its CN_H and CN_O “coordinates”. As visible by inspection the four groups already encountered appear once again as all peers of a group lie on a straight line and can be easily fitted by linear regression. Therefore the ratio CN_H/CN_O is a characteristic parameter for each group with values clustering about 1.64 (positively charged), 3.76 (polar), 5.06 (unpolar) and 12.45 (negatively charged). In this context we emphasise that in this analysis we have included the rarely occurring amino acids as well as those at the C or N terminus with a modified charge status. Only MET and TRP were not considered.

Mean Residence Time (MRT)

Analogously to the global protein analysis the residence correlation function was fitted to a KWW function (see Eq.(4.14)) We start our discussion with the static parameters, i.e. with the amplitudes A_1 and A_2 , and analyse the time behaviour and dynamic diversity expressed by the relaxation time τ and the exponent β afterwards. The lower limit $A_1 \geq \text{CN}/N$ and the corresponding upper limit $A_2 \leq 1 - \text{CN}/N$ determined in the theory section provide a guideline to classify the vast amount of data for the amplitudes in five solvation shells around 20 amino acids. Only in the first shell these limits are not reached, in all subsequent shells the amplitudes are already close to the limits. In the fourth and fifth shells the limits are perfectly reached. This increasing agreement between the actual amplitudes and their theoretical limits for higher shell numbers may be explained in the following way: First, the higher the shell number the larger the corresponding coordination number CN which then becomes comparable in its statistical weight to the total number of particles N. Second, each particle has a much higher probability to become a member of the outer shells than of the inner shells. This again favours a statistical bias of the amplitudes. Third, there is a dynamical argument too: The retardation of migration in the inner shells slows down the exchange with outer shells. Static aspects of residence of solvent residence were dealt with in the preceding paragraph. The applicability of the KWW model to residence function permits a projection of residence dynamics to the pair of parameters τ and β . In Fig.4.9 the shell dependence of this pair is shown taking again one representative (ASP, ARG, ASN, LEU) for the four groups of negatively charged, positively charged, polar and unpolar amino acids. The latter group is characterised by a high diversity of residence dynamics which correlates with the accessibility by the solvent as reflected by the size of the VISA. In order to exclude potentially or artificially resident water molecules a VISA threshold of 50 \AA^2 was introduced for all amino acids. This is inspired by the picture that resident water molecules reside in clefts which in turn are characterised by a small interface area or VISA between respective amino acids and water. The solvent exposed charged and polar residues with their high VISA are hardly effected by the introduction of this threshold.

Several features can be observed simultaneously for both, τ and β . The most striking feature of Fig.4.9 is the exceptional role of the first shell, whereas subsequent shells behave rather similar, i.e. their τ and β values are almost independent of the type of amino acids. Thus, the features characteristic for the amino acid type are largely eliminated in subsequent shells. A hint for this almost perfect screening of the first shell was already found when analysing the shell-grained orientational

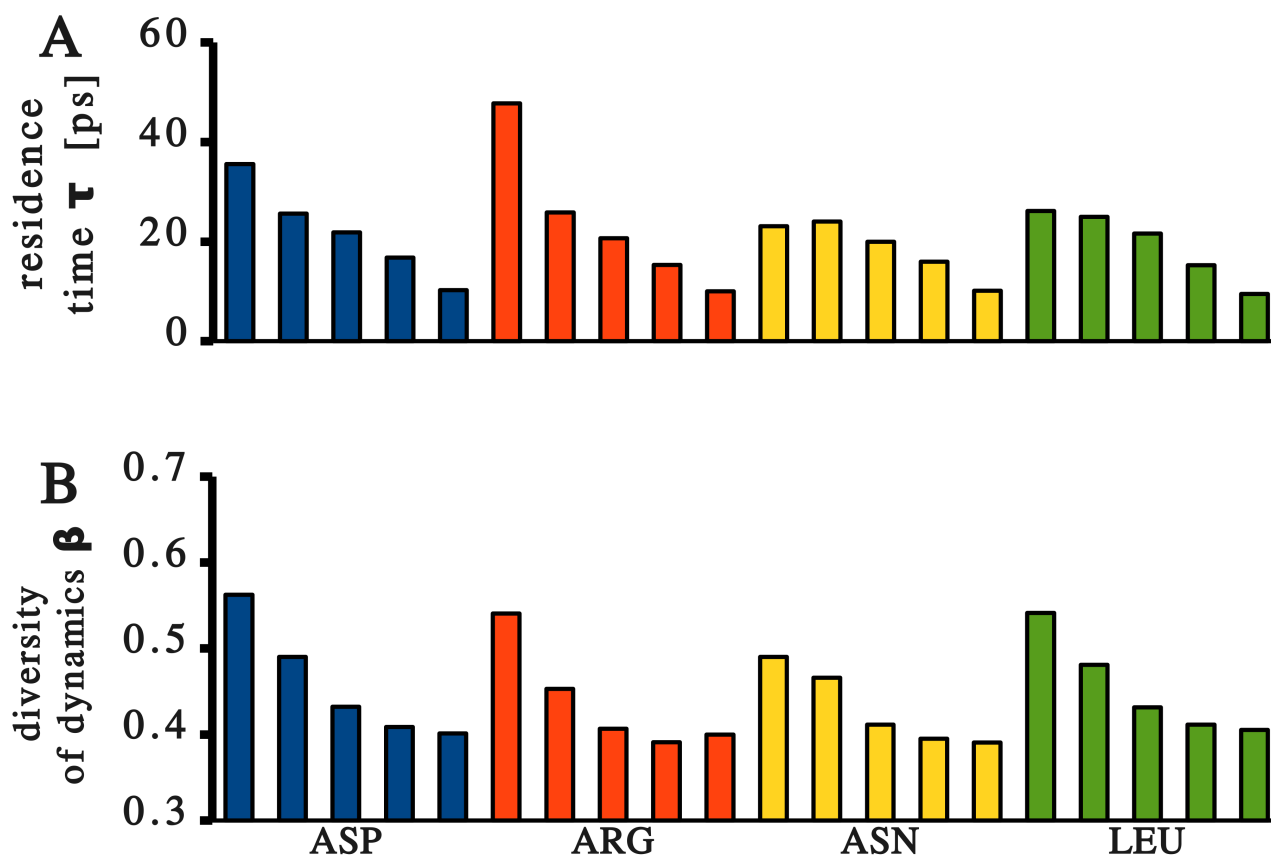


Figure 4.9: Time behaviour (A) and complexity of relaxation (B) on a local residue level for the first five solvation shells. Again, one representative is given for the four groups of negatively charged (ASP, blue), positively charged (ARG, red), polar (ASN, yellow) and unpolar (LEU, green) amino acids.

order. On the one hand, this perfect screening of the first shell is caused by the high polarity and mobility of the water molecules, on the other hand the environmental effect of neighbouring amino acids becomes stronger for more distant shells. A further common feature of τ and β is the stronger influence of charged amino acids as compared to uncharged ones. This leads to elevated relaxation times τ and at the same time enhances the parameter β . In other words, stronger electrostatic couplings lead to a retardation and simultaneously reduce the diversity of residence dynamics. According to Eq.(4.15) the average relaxation time $\langle\tau\rangle$ is essentially given by the ratio τ/β . As both, τ and β are monotonically decreasing functions of the shell index they counteract each other when taking the ratio. Consequently, the shell dependence of the local $\langle\tau\rangle$ is lesser than that of the global one.

4.5 Conclusion

Studying the structural and dynamical aspects of solvation is inevitably connected to a decomposition of extrasolute space into shells. For a small and spatially isotropic solute spherical shells are appropriate and frequently used. In this study we have developed a natural extension of this concept of spherical shells to the case of large and anisotropic solutes, thereby avoiding the introduction of parameters. This parameter-free classification of solvation shells is offered by the method of Voronoi/Delaunay tessellation of space. Generally, spherical shells lend themselves to a resolution of structural and dynamic properties on a radial scale. Therefore it is consistent to abandon a radial resolution when releasing spherical shells in favour of a parameter-free Voronoi shell definition. Rather we consider a whole Voronoi shell as a “bin”, i.e. the smallest piece of granularity. This makes it possible to characterise structure and dynamics of solvation shells by a few numbers in a compact way.

We show in this study that this higher granularity does not lead to fewer information, but in some sense sharpens and compacts it. For example the few numbers derived from an analysis of shell-grained orientational order yield the same information as fully resolved orientational correlation functions. Within the framework of the Voronoi method many properties are directly accessible by counting and summation instead of radial integration between parametrised limits. As the polyhedron enclosing all space nearer to an atom or molecular moiety than to any other one is the elementary building block of Voronoi decomposition, the volume of a shell is simply obtained by summing the volume of elementary polyhedra participating in this shell. Summing up all outer faces gives the Voronoi interface surface area (VISA), while the number of neighbouring polyhedra equals the coordination number. We have applied these general principles to three solvated proteins 1UBQ, 1CLB and 2PLD

differing in charge status (uncharged, net charge $-7e$ and net charge $+3e$) and composition of secondary structure elements. From the elementary polyhedra a variety of properties was derived and analysed. This was first done on a global level and further resolved and explained on a local scale of individual amino acids. In the following we briefly summarise essential findings.

The newly defined Voronoi interface surface area (VISA) proved to be a natural generalisation of the frequently used solvent accessible surface area (SASA). While SASA is restricted to the first shell, VISA can be computed for arbitrary shells. For the first shell both properties agree quite well. The VISA per solvent molecule, $VISA/CN$, is in the vicinity of the traditional Kabsch and Sander value of 9.65.^[43] However, subtle variations between different proteins and different shells are observed and can be interpreted consistently. This is one example that findings at the global protein level can be explained by local residue level analysis. The difference between proteins has its origin in different amino acid frequencies going along with local residue specific VISA fluctuations. The variation over shells correlates with the respective solvent density. The time evolution of residue specific VISA reveals two principle mechanisms. On the one hand, a gradual change of the VISA is observed. On the other hand, sudden changes of the type of a flip flop mechanism are found. This means that in the latter case all hydrogen bonds between the terminal functional group of a residue and water are disrupted or created simultaneously. The outstanding position of the first shell is manifested several times in different properties on the global and on the local scale: This refers to static properties like donor-acceptor function or dielectric screening, as well as to the dynamic properties relaxation rate and diversity of residence dynamics. In the view of a suprasolute representing the union of solute and first solvent layer subsequent shells behave somewhat more uniformly. In other words, the first solvent layer compensates the protein specific features. Although the first layer veils the protein's character it is completely unable to quench its long range directing influence. In fact, the donor-acceptor function as well as the shell specific mean residence time (MRT) show a clear variation up to and beyond the fifth shell. More concrete, the MRT is a linear function of the shell index thus predicting a typical retardation factor of five. Due to the long range influence, however, a retardation by a factor of eight is plausible. This retardation by a factor five to eight is in good agreement with the experimental findings of Halle.^[7] At the more detailed local level the directing influence is visible in static and dynamic properties, too. The VISA per solvent molecule $VISA/CN$ has a value typical for the group (charged, polar, unpolar) to which the specific residue belongs. Furthermore, the strength of influence correlates with the rate of relaxation as well as with its dynamical diversity.

Acknowledgement

This work was supported by the project P19807 of the FWF Austrian Science Fund.

Bibliography

- [1] V. P. Denisov, B. H. Jonsson, and B. Halle, *Nat Struct Biol* **6**, 253 (1999).
- [2] R. Loris, U. Langhorst, S. De Vos, K. Decanniere, J. Bouckaert, D. Maes, T. R. Transue, and J. Steyaert, **36**, 117 (1999).
- [3] U. Langhorst, J. Backmann, R. Loris, and J. Steyaert, *Biochemistry* **39**, 6586 (2000).
- [4] M. Tarek and D. Tobias, *Biophys. J.* **79**, 3244 (2000).
- [5] J. Janin, *Structure Fold. Des.* **7**, R277 (1999).
- [6] A. Palomer, J. J. Pérez, S. Navea, O. Llorens, J. Pascual, L. García, and D. Mauleón, *J. Med. Chem.* **43**, 2280 (2000).
- [7] B. Halle, *Philosophical Transactions of The Royal Society* **359**, 1207 (2004).
- [8] A. R. Bizzarri and S. Cannistraro, *J. Phys. Chem. B* **106**, 6617 (2002).
- [9] A. Poupon, *Current Opinion in Structural Biology* **14**, 233 (2004).
- [10] F. Cazals, F. Proust, R. Bahadur, and J. Janin, *Protein Science* **15**, 2082 (2006).
- [11] B. Bouvier, R. Grunberg, M. Nilges, and F. Cazals, *Proteins* **76**, 677 (2009).
- [12] E. E. David and C. W. David, *J. Chem. Phys.* **76**, 4611 (1982).
- [13] E. E. David and C. W. David, *J. Chem. Phys.* **78**, 1459 (1983).
- [14] O. Gedeon and M. Liska, *Journal of Non-Crystalline Solids* **303**, 246 (2002).
- [15] O. Gedeon, *Journal of Non-Crystalline Solids* **351**, 1139 (2005).
- [16] Y. I. Jhon, K. T. No, and M. S. Jhon, *Fluid Phase Equilibria* **244**, 160 (2006).

Bibliography

- [17] J. T. Fern, D. J. Keffer, and W. V. Steele, J. Phys. Chem. B **111**, 13278 (2007).
- [18] P. F. Goncalves and H. Stassen, J. Chem. Phys. **123**, 214109 (2005).
- [19] P. Espanol, *A fluid particle model* (1997), URL doi:10.1103/PhysRevE.57.2930.
- [20] M. Serrano, G. D. Fabritis, P. Espanol, E. G. Flekkoy, and P. V. Coveney, J. Phys. A: Math. Gen. **35**, 1605 (2002).
- [21] R. Abseher, H. Schreiber, and O. Steinhauser, Proteins **25**, 366 (1996).
- [22] S. Boresch, S. Ringhofer, P. Höchtel, and O. Steinhauser, Biophys Chem. **78**, 43 (1999).
- [23] C. Schröder, T. Rudas, S. Boresch, and O. Steinhauser, J. Chem. Phys. **124**, 234907 (2006).
- [24] T. Rudas, C. Schröder, S. Boresch, and O. Steinhauser, J. Chem. Phys. **124**, 234908 (2006).
- [25] K. E. Thompson, Int. J. Numer. Meth. Engng. **55**, 1345 (2002).
- [26] A. Okabe, *Spatial tessellations: concepts and applications of Voronoi diagrams* (Wiley, New York, 2000).
- [27] D. F. Watson, The Computer Journal **24**, 167 (1981).
- [28] M. Gerstein, J. Tsai, and M. Levitt, J. Mol. Biol. **249**, 955 (1995).
- [29] G. De Fabritiis and P. V. Coveney, Comput. Phys. Commun. **153**, 209 (2003).
- [30] H. Borouchaki, P. L. George, F. Hecht, P. Laug, and E. Saltel, Finite Elements in Analysis and Design **25**, 61 (1997).
- [31] H. Borouchaki and S. H. Lo, Comput. Methods Appl. Mech. Engng. **128**, 153 (1995).
- [32] C. Schröder, G. Neumayr, and O. Steinhauser, J. Chem. Phys. **130**, 194503 (2009).
- [33] R. S. Anderssen, S. A. Husain, and R. J. Loy, Anziam J. **45**, C800 (2004).
- [34] E. W. Montroll and J. T. Bendler, J. Stat. Phys **34**, 129 (1984).
- [35] G. Neumayr, C. Schröder, and O. Steinhauser, J. Chem. Phys. **131**, 174509 (2009).
- [36] S. Vijay-Kumar, C. E. Bugg, and W. J. Cook, **194**, 531 (1987).

- [37] B. P. Monia, D. J. Ecker, and S. T. Crooke, *Biotechnology* **8**, 209 (1990).
- [38] N. J. Skelton, J. Kördel, and W. J. Chazin, **249**, 441 (1995).
- [39] Q. Ji, A. Chattopadhyay, M. Vecchi, and G. Carpenter, *Mol. and Cell. Biol.* **19**, 4961 (1999).
- [40] S. M. Pascal, A. U. Singer, G. Gish, T. Yamazaki, S. E. Shoelson, T. Pawson, L. E. Kay, and J. D. Forman-Kay, *Cell* **77**, 461 (1994).
- [41] C. Schröder, T. Rudas, and O. Steinhauser, *J. Chem. Phys.* **125**, 244506 (2006).
- [42] B. K. P. Horn, *J. Opt. Soc. Am.* **A4**, 629 (1987).
- [43] W. Kabsch and C. Sander, *Biopolymers* **22**, 2577 (1983).
- [44] V. P. Denisov and B. Halle, *J. Am. Chem. Soc.* **117**, 8456 (1995).

5 Summary

The central thesis of this work as introduced in the introduction holds true: It is possible to find a parameter-free (thus system independent) methodology to unambiguously describe solvation structure and relaxation of large and anisotropic molecules. The key is to base the development of these methods on Voronoi/Delaunay decomposition of space and a recursive definition for solvation shells. It was crucial for the progress of this work, to satisfy algorithmic preconditions. Without the adaptation, optimisation and implementation of the Delaunay tessellation algorithm, all following steps would have been rendered void.

The overall layout of this dissertation can be described at two different levels: The system level and the methodic level. At the system level, the work begins with analysing ionic liquids of medium molecular size in water and advances to the domain of large biomolecules. The methodic progress starts with analysing the structural aspects using a Voronoi/Delaunay decomposition of the traditional g-functions. Next, the relaxation behaviour is described based on Voronoi shells. Thereby, following a twofold strategy, mean residence times are obtained by residence correlation functions and a new probabilistic model. Subsequently, as a third step, the g-functions are being disengaged and replaced by a full Voronoi/Delaunay description of packing and orientation.

The general features and quantities obtained by this approach are in good accordance with both, experimental and computational methods and shed a new light on structure and dynamics of solvation. Thus, they provide confirmation for the underlying theoretical framework. Moreover, new findings emerge from the studies conducted that could not have been obtained by traditional computational methods, at least not in such clarity. From the application of the Voronoi methods to computer simulations of hydrated ionic liquids and proteins, the following can be learnt in general:

- Voronoi analyses sharpen the overall picture of solvation. Regarding protein and molecular ion solvation, solute anisotropy is an important feature that cannot be described by traditional g-functions. This is reflected in and can best be shown by Voronoi decomposition of the radial distribution function (RDF). The first solvation shell does not necessarily coincide with the first

5 Summary

RDF peak. Furthermore, the radial overlapping creates peaks, that appear as a second or third radial shell, but indeed arise from superposition of two adjacent solvation shells.

- In general, Voronoi based solvation analysis reveals the impact of molecular interactions and forces on structure and dynamics. The cation-cation relaxation times are shorter at the head and longer at the tail. This arises from the hydrophobicity of the tail. The opposite is true for cation-anion relaxation times which are dominated by relatively strong charge-charge interactions. In contrast, water residence times, that are influenced mainly by weaker dipolar interaction, are shorter and uniformly distributed. In this context, the following principle applies broadly to the complexity of relaxation: The stronger the coupling, the bigger β and, thus, the less diverse the relaxation times. Hence, the strong cation-anion interactions lead to relatively uncomplex relaxation while water-water relaxation seems to include more relaxation channels. The same principle can also be verified in hydrated protein systems: Water in the immediate neighbourhood of charged amino acids shows less relaxation complexity than water adjacent to uncharged residues. Even on a global scale this rule holds true: the closer the water to the protein, the stronger the influence, the smaller the diversity of relaxation times.
- The anion plays a central role in hydrated ionic liquids influencing both, structure and dynamics. Although the charge-charge dominated anion-anion interaction changes only little with water mole fraction, the anion-water network, dispelling the cation, strengthens with increasing water concentration. A square-root time-law for relaxation is shown to be valid with only a minimal systematic spread between different ionic liquids. However, the observation can be made that the anion causes the most spread.
- Different specific features of the solute and the solvent allot a special role to the first solvation shell. This can already be seen in hydrated ionic liquids but more clearly in hydrated protein systems. The imidazole ring acts as a hydrogen-bond donor for the first water shell. This donor behaviour is reversed to a weaker acceptor role in the second shell. In proteins, dielectric screening analysis shows a peculiar short range effect. The first hydration shell seems to be sufficient for hiding most of the protein's specific local features.
- Despite the importance of the first solvation layer, long-range effects on structure as well as dynamics of protein solvation can be observed. For example, donor/acceptor analyses reveal a cascade that reaches beyond five shells. This cascading effect is strongest for negatively charged

amino acids. Dynamics in the first shell is retarded by a factor of 5 compared to bulk dynamics which is in good agreement with experiments. This retardation factor is still above 1 in shell four. By this means, Voronoi analysis allows to describe how and to what extent the influence of the protein decreases with increasing shell index.

- This work delivers some insight into the relation between structure and dynamics. The viscosity has been shown to be a central parameter influencing overall dynamics. A linear relation of viscosity and mean residence times has been shown. This linear dependence is even stronger for residence times obtained from the newly developed Markov model. Thus, the probabilistic Model allows prediction of viscosity from very short simulations. Additionally, it provides an intuitive interpretation of the initial and asymptotic values of residence correlation functions. If the viscosity, as the central parameter influencing overall dynamics, is enhanced in a system, the structure is generally sharpened.
- Based on volume and surface calculations of the Voronoi polyhedra further methodic advanced were made: The concept of occupancy, based on Voronoi volumes, is an appropriate measure for prominence of different molecular species, exceeding the concept of concentration. The Voronoi interface surface area (VISA), as a generalization of the well known solvent accessible surface area (SASA), can be computed not only for the innermost but for any solvation layer. Observation of the VISA's time behaviour reveals two mechanisms of surface solvation: On one hand, a gradual change in the VISA is observed. On the other hand, collective disruption and formation of several hydrogen-bonds is reflected in flip-flop like almost instantaneous changes of the VISA over time.

6 Outlook

Where to go from here? The most obvious application of the Voronoi-based methods developed in this work lies within the context of molecular solvation. Preliminary studies of molecular ionic liquids (MIL) as solvents are already being used as a basis for further analyses regarding proteins solvated in MIL at the *Institute of Computational Biological Chemistry*. One example of a zinc-finger solvated in hydrated EMIM⁺ TRIF⁻ is given in the User Guide chapter (8). However, a great advantage of Voronoi-based methods is their universal applicability. At an abstract level, the structural and dynamic quantities that have been developed in this work could be applied to other scientific fields than molecular sciences.

Markovian Master equations could be used in order to investigate relaxation more deeply in terms of relaxation channels. The framework that has been developed, especially the transition matrix W within this framework allows for a vast multitude of state definitions of any kind. Currently, the shell membership function $n(t)$ and solute side proximity have been used to define these states. This results in values of $\beta = 1.0$ for relaxation complexity. Preliminary investigations suggest that incorporation of concepts like “connectivity” can lower this value to about $\beta = 0.8$ and below, which means, complexity of relaxation could be better reflected by a connectivity based state definition. By this means, orientation would be accommodated and its role in relaxation processes can be analysed. Eventually, fractional time behaviour could be explained directly by distinct modes of molecular motion. Thus, time behaviour could be predicted and explained even better in terms of complexity by incorporation of single particle observables like orientation, distance, velocity or energy. This could lead to a deeper understanding of relaxation mechanisms.

Concepts like promiscuity and connectivity have been defined and used for preliminary studies only. They could be further developed to yield universally applicable methods for modelling and, finally, prediction of dynamics. Thereby, promiscuity seems promising as a universal, system independent, measure of viscosity. Preliminary studies have shown that a Voronoi-based, thus parameter-free, hydrogen-bond definition might be possible. Maybe this could be achieved by combination of several

6 Outlook

different contact types. Voronoi analysis lends itself to other interesting fields like voids and cavities that have not been studied to a greater extent in this institute up to now. Other preliminary tests have shown that the Voronoi-decomposed, shell specific g-functions could be fitted by multiple Gaussian functions, providing a parametric description of packing and orientation.

Altogether, the Voronoi approach seems to be fruitful but technically demanding. One way to cope with this issue is the concept of parallel computation. This feature has already been included in GEPETTO at the level of “embarrassingly parallel” computation. This means, only little communication between separate tasks is necessary when computations refer to independent frames. Two further levels of parallelity could be introduced into GEPETTO in order to improve performance: At the post-processing level, calculation of the correlation functions or mean square displacements could be parallelised using MPI. Finally, data parallelity could be implemented and used to analyse even larger systems.

7 GEPETTO - Implementation

Molecular dynamics yields trajectories, which describe the temporal evolution of molecular motion. The large volume of these data necessitates efficient tools for analysis, which led to the decision to develop the “Grid Enabled Parallel Enhanced Trajectory TOol”(GEPETTO). Given N AtomGroups (i.e. molecules, segments, residues or atoms) and t Frames (time steps), single particle observables are of data complexity $O(tN)$, distance based methods are of complexity $O(tN^2)$ and collective time series are of complexity $O(t)$. One key idea of efficient calculation is to establish an elegant way of internal data handling, calculating multiple tasks at once, grouping calculations by their complexity, thereby avoiding computational redundancy and exploiting the compilers vectorization capabilities. Furthermore, efficiency is achieved by the optimised implementation of suitable algorithms. The initiation of this development dates back to the beginning of the author’s work at the *Institute of Computational Biological Chemistry*. However, the branch of GEPETTO that has been used to carry out most of the analysis in this work, has been developed in cooperation with Thomas Taylor and Michael Haberler: Some parts have been developed by pair programming. Other parts result from efforts that every contributor has made by himself. Information on the origin of every herein described part of the software is given to the best knowledge of the author. Functionality designed by the author of this dissertation is not specifically marked.

Basically, GEPETTO follows a design that allows for both, a flexible and secure input or scripting mechanism and at the same time efficient calculation of results. Flexibility and high performance are typical antagonists in software design in a way that often one is favoured at the expense of the other. That being said, it becomes evident that an uncompromising solution comes at the cost of a more complex design as well as higher effort in implementation and maintenance of the code.

On one hand, simple and secure usage demands the input to be structured intuitively and results-oriented. Efficient calculation, on the other hand, includes reuse of intermediate results for multiple calculations. In order to achieve both, the program is divided in a so-called “instruction space” (IS) and a “calculation space” (CS). The first contains classes that cope with user interaction, program flow

and optimization of execution plans, the second pools algorithms for calculation. Particular concepts like histograms or time series are present in both spaces, although in different occurrence. Having a partially duplicate structure was considered a tolerable tradeoff in order to decouple instructions and algorithms and thus allow for both flexible and efficient calculation.

Tasks, the central concept of IS reflect mathematical functions that are to be calculated by GEPETTO using given spatial and temporal selections of a trajectory. The majority of tasks obey the following design concept: An observable, represented as a scalar, a vector or a matrix is calculated for a set of frames and stored in a task list object. Based on this time series, other properties like correlation functions, mean square displacement or frequency distributions are produced.

On the contrary, the so-called calculation flag groups are the central concept of CS. Each task has calculation flags indicating necessary intermediate results. Via these calculation flags, one task is mapped to several calculation flag groups while one calculation flag group, in turn, can organise requirements and intermediate data of multiple tasks. The process of generating CS objects out of IS objects is called “optimization” in our terminology.

It might be helpful to give an example at this point. Let’s assume a simultaneous calculation of the radial distribution function (RDF) $g^{000}(r)$ and the orientational correlation function (OCF) $g^{110}(r)$ of a set of atom groups represented by the center-of-mass (atom group selection or space selection), over a set of frames (temporal selection). Two tasks are created, one for $g^{000}(r)$ and one for $g^{110}(r)$, both specifying the same (!) set of selections. After calculation is finished, each task writes its own output file. That is how instruction space is structured. In calculation space things are different: The center-of-mass is a prerequisite for both the RDF $g^{000}(r)$ and the OCF $g^{110}(r)$. In addition, $g^{110}(r)$ needs the electric dipole of each atom group to be calculated. Thus, a calculation flag group is created, holding a non-redundant set of atom groups of both tasks and setting the center-of-mass (COM) and dipole flags (DIP) accordingly. This saves computation time in two different ways: First, the center-of-mass, although needed in two different tasks is calculated only once per atom group. Second, sorting and grouping the atom groups by their calculation flags provides the possibility to use one optimised function for the combined calculation of the center of mass and the dipole, if appropriate. Given N atom groups, the calculation flag group described so far corresponds to a computational complexity $O(N)$ and copes with the prerequisites. A second calculation flag group of complexity $O(N^2)$ is generated, which is being used for calculation of minimum center-of-mass distances (NEEDRDF, NEEDDIST flag), and dipole/dipole correlation (NEEDHMUMU flag). By this means, a non-redundant calculation of distances is warranted and, again, computation time is saved.

Auxiliary classes contain code for parsing and translation of user scripts and reading trajectories. The general program flow, as it is defined in "source/program/main.cpp", is described in the following Section. Further Sections describe concepts that occur in instruction space or calculation space. Owing to the efforts of Thomas Taylor and Michael Haberler, a more detailed documentation of GEPETTO's class structure can be generated directly from the code using the documentation generator **doxygen**. This chapter is not intended to replace the **doxygen** documentation in any way. A table-like structured documentation, like the one generated by **doxygen**, is a concise source of information and, at the same time, much more complete in terms of classes, objects, methods and their interaction. This chapter rather constitutes the attempt to give an overview of the whole program and background information about design decisions having been made during development.

7.1 General Program Flow

At the very general level, GEPETTO follows a predefined sequence of function calls that is encoded in source/program/main.cpp. This sequence contains the following steps:

- The first step is **parsing** and **translating** the script that has been provided by the user and tells GEPETTO what to calculate. This step also includes consistency checks of user input. Some details about the current version of the script parser and translation can be found in Sec.7.2.2.
- As a next step, **selections** of atoms, residues, functional groups or molecules defined in the userscript are created as IS objects (see Sec.7.2.3).
- Based on selection and further information from the script, tasks are **optimized** for calculation. Thereby, CS data structures are created from IS data structures (see Sec.7.2.4).
- During **calculation**, GEPETTO iterates over all frames of all user specified trajectories and stores results in histograms, time series, matrices and other data structures that have been defined in the script as described in Sec.7.3.
- **Postprocessing**, in the current version, is basically a synonym for normalization or correlation of data, that occurs after main calculation (see Sec.7.2.5).
- As a last step, the **output** is written to files (see Sec.7.2.5).

A detailed overview is given in Figure 7.1. Single items in this diagram describe concepts implemented in GEPETTO. It can be seen as a mixture of a data flow diagram and a rudimentary entity

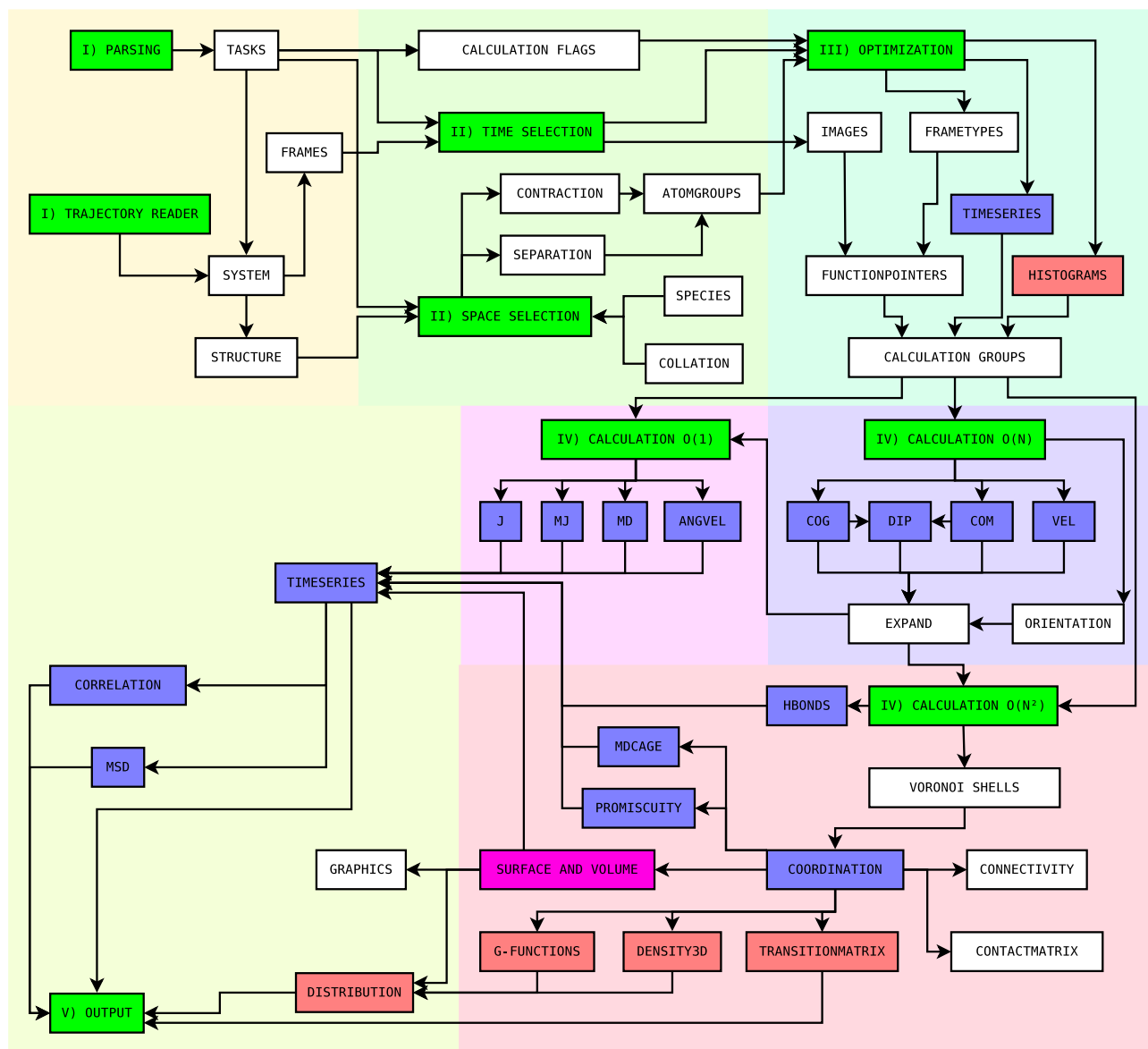


Figure 7.1: GEPETTO overview. Items belonging to the same step are marked by a common background color, IS being coloured green and yellow and CS red and blue. Green boxes depict the coarse program flow, red ones are related to histograms and the blue boxes symbolise time series-related concepts.

relationship diagram. Each subsection lists the source files associated with the respective concept like script parsing, periodic boundaries or transition matrices, to name but a few. All paths are given relative to the directory "source/". Thereby, files that implement the particular concept are printed using a **boldface font**, files containing auxiliary classes used by the concept are printed normally and files using the particular concept are printed *italically*. In order to make a distinction between classes and their instances (objects), the former are printed **bold** and the latter normal. Namespaces, like classes, are printed **bold**. Member functions or methods are given in the form `class::method(...)` where the ellipsis indicate the method's arguments.

7.2 Instruction Space (IS)

7.2.1 Flow Control

The general program flow, as described above, is being controlled by the class **TaskList**. Like **TaskResource** it is implemented as singleton which means, only one object instance can be created. All functions being called by the function `main()` in "source/program/main.cpp" are implemented by the **TaskList** singleton: First, the function `TaskList::parse()` calls the script parser for each **ScriptLocation**, saves the parsed scripts and translates them into Task objects containing all further objects. Second, `TaskList::select()` starts the selection process where **AtomGroup** objects are created and O(N)-flags as well as unique identifiers are assigned. Furthermore, this function checks for empty selections. Then, optimization is controlled by `TaskList::optimize()`. Here, **Histogram**, **Timeseries** and **FrameType** objects are being prepared for calculation. The method looping over all needed **Frame** objects thereby invoking the respective `FrameType::doCalculations(...)` is called `TaskList::calcStuff()`. The two final steps are implemented in `TaskList::postProcess()` and `TaskList::storeResults()`.

instruction/	tasklist.h, tasklist.cpp, taskresource.h
--------------	-------------------------------------------------

7.2.2 Program Input

Script Parsing

The script parser has been developed by Thomas Taylor using **boost::spirit**. For a more detailed description of parser internals a closer look at the **spirit** documentation is recommended. The parser,

7 GEPETTO - Implementation

having been invoked by the **TaskList** singleton (see Sec.7.2.2), reads the user script files and translates them to objects that can be handled by GEPETTO. The overall process is performed in two passes, both being invoked by the **TaskList** class and mediated by the **Script** class:

First, the script is split into tokens, that are linked into a hierarchical parser tree (class **ScriptTree**) whose nodes and leafs (class **Node**) contain values of different basic data types. They can be of type `int`, `double`, `std::string` or `enum`. For instance, the `std::string` data type is used for identifiers like trajectory names or the names of structural selections. The resulting **ScriptTree** resembles the script file in an already expanded way. That means, if a script item (e.g. a structural selection) is defined once but reused several times within one script, the tree holds fully descriptive copies of the subtree describing that item for every occurrence. This first pass is the actual “parsing phase”.

The following second pass is called “translation” in our terminology. During this process, the parser tree is used as a template for the actual hierarchy of IS objects. The file `script/gep/gep.cpp` contains a method called `Script_Gep::translate()` which is the core of the translation process. It iterates over all task nodes in the parser tree and creates all necessary objects by instantiating classes like **Histogram**, **MDsystem**, **AtomGroupSelection**, **SystemSelection**, **Shell** definitions and, of course, **Task**. The class **TaskResource** has been introduced as a singleton in order to prevent redundancies. Single objects of any type are registered by **TaskResource** ensuring that objects that have been specified multiple times are created only once.

script/	gepetto.h, node.h, node.template.cpp, helpers.h
script/gep/	gep.h , gep.cpp , grammar.h , token.h , token.cpp , interface.cpp, script.h, script.cpp, functor.h
instruction/	<i>tasklist.cpp</i> , <i>taskresource.h</i>

Trajectories

Generally, molecular dynamics trajectory data describe molecular motion over time. Typically, data structures to hold such information are split into two parts: The first part describes chemical and physical properties that are constant over time, like charge, mass of atoms or covalent bonds between them. This structural aspects can be imported by GEPETTO as XPLOR formatted `".psf"` files. Coordinates and velocities changing over the temporal evolution of the system are read in as `charmm` files. They usually have the extensions `".dcd"` and `".vel"`, respectively. This results from the prac-

tical constraints of charmm currently being the most frequently used software package for simulation in our group. However, the trajectory framework developed by Thomas Taylor anticipates future changes by separating a more abstract layer comprising the base classes **Structure**, **Trajectory** and **Frame**, where the latter inherits from **coor3d<frame_precision>**. The charmm specific classes **Psf** and **charmm::Trajectory** inherit from these base classes and are the two most important members of the class **charmm::MDSsystem** which combines both aspects of a trajectory: static (psf) and dynamic (dcd). The external dcd reader can be found in the directory "source/external/io".

source/mdsystem/	mdsystem_charmm.h, mdsystem_charmm.cpp, mdsystem_charmm_helper.h, mdsystem_charmm_helper.cpp
source/mdsystem/trajectory/	charmm_dcd_external.h, charmm_dcd_external.cpp, frame.h, frame.cpp, trajectory.h, trajectory.cpp, structure.h, structure.cpp
source/mdsystem/structure/	charmm_psf_external.h, charmm_psf_external.cpp
source/instruction/	<i>frametype.h, tasklist.h, taskresource.h</i>

Task

The central class in instruction space is **Task** which is a virtual abstract base class. Multiple derived classes inherit from **Task**. As the name implies, each derived class defines one distinct “job” or calculation. The member variables **TaskType** and **Observables** specify the kind of calculation and the observables to be calculated. Further important members are **Histograms** and **Timeseries**, both mainly coping with data and results organization. As mentioned at the beginning of this chapter, task-specific **CalculationFlags** (see Sec.7.2.4) are set in this class, as well as **Shell** indices (see Sec.7.3.3). Every **Task** object has a temporal selection called **TimeSelection** of type **SystemSelection** and a spatial selection called **SpaceSelection**, which is of class type **AtomGroupSelection**. These topics are dealt with in Sec.7.2.3. Finally, the **Task** class handles postprocessing and output (see Sec.7.2.5). At the moment, the following derived classes exist: **Task_Timeseries**, **Task_Correl**, **Task_Distribution**, **Task_MSD**, **Task_gFunction**, **Task_Density3d**, **Task_Hbond** and **Task_TransitionMatrix**.

instruction/	task.cpp, task.h, tasklist.cpp, tasklist.h, taskresource.cpp, taskresource.h
instruction/correl/	task_correl.h, task_correl.cpp
instruction/density3d/	task_density3d.h, task_density3d.cpp
instruction/distribution/	task_distribution.h, task_distribution.cpp
instruction/gfunction/	task_gfunction.h, task_gfunction.cpp
instruction/msd/	task_msd.h, task_msd.cpp
instruction/hbond/	task_hbond.h, task_hbond.cpp
instruction/timeseries/	task_timeseries.h, task_timeseries.cpp
instruction/transitionmatrix/	task_transitionmatrix.h, task_transitionmatrix.cpp

7.2.3 Selection

Structural selections are used to choose distinct structural components like atoms, residues or molecules for calculation. Temporal selections define a time range within the trajectory as well as a temporal graining.

Structural Selections

GEPETTO's atom selection functionality is implemented straightforwardly. Yet, some general selection related concepts have to be explained. Basically, the member variable **SpaceSelection** of class **Task** collects a choice of atoms (**StructureAtoms**) from the **Structure** of a **MDSsystem** that match specific **SelectionPattern** objects and are grouped into **AtomGroup** objects that aggregate the contained atoms' properties. In method **MultiSelection::create(...)** only those **iAtoms** of the **ReferenceStructure** are selected, that match specific **SelectionPatterns**. According to ".psf" file specifications, the Patterns are defined in terms of atom name, atom number, residue name, residue number and segment name (see Fig.7.2). If a selection pattern specifies a segment name, all atoms belonging to that segment are being selected. The only wildcard character supported at the moment is the asterisk (*) indicating all items: Specification of **ATOMTYPE *** in the script matches all atom types. The grouping into functional groups, residues, segments, molecules or molecular complexes is a further, independent step called "separation". GEPETTO knows the following six different granularities of **AtomGroup** objects: **SeparateByAtomId** creates a separate **AtomGroup** ob-

PSF					Example 1)				Example 2)				Example 3)							
Atom Number	Segment Name	Residue Number	Residue Name	Atom Name	Unique Residue Id	Species Id	Collation Id	Tessellation Selection	Core Selection	Surround Selection	Species Id	Collation Id	Tessellation Selection	Core Selection	Surround Selection	Species Id	Collation Id	Tessellation Selection	Core Selection	Surround Selection
1	TPEP	1	MET	N	0	0	0				0	0				0	0			
2	TPEP	1	MET	HT1	0	0	0				0	0				0	0			
3	TPEP	1	MET	HT2	0	0	0				0	0				0	0			
4	TPEP	1	MET	HT3	0	0	0				0	0				0	0			
5	TPEP	1	MET	CA	0	0	0				0	0				0	0			
6	TPEP	1	MET	HA	0	0	0				0	0				0	0			
7	TPEP	1	MET	CB	0	0	0				0	0				1	1			
8	TPEP	1	MET	HB1	0	0	0				0	0				1	1			
9	TPEP	1	MET	HB2	0	0	0				0	0				1	1			
10	TPEP	1	MET	CG	0	0	0				0	0				1	1			
11	TPEP	1	MET	HG1	0	0	0				0	0				1	1			
12	TPEP	1	MET	HG2	0	0	0				0	0				1	1			
13	TPEP	1	MET	SD	0	0	0				0	0				1	1			
14	TPEP	1	MET	CE	0	0	0				0	0				1	1			
15	TPEP	1	MET	HE1	0	0	0				0	0				1	1			
16	TPEP	1	MET	HE2	0	0	0				0	0				1	1			
17	TPEP	1	MET	HE3	0	0	0				0	0				1	1			
18	TPEP	1	MET	C	0	0	0				0	0				0	2			
19	TPEP	1	MET	O	0	0	0				0	0				0	2			
20	TPEP	2	GLN	N	1	0	1				0	1				0	3			
21	TPEP	2	GLN	HN	1	0	1				0	1				0	3			
22	TPEP	2	GLN	CA	1	0	1				0	1				0	3			
23	TPEP	2	GLN	HA	1	0	1				0	1				0	3			
24	TPEP	2	GLN	CB	1	0	1				0	1				1	4			
25	TPEP	2	GLN	HB1	1	0	1				0	1				1	4			
26	TPEP	2	GLN	HB2	1	0	1				0	1				1	4			
27	TPEP	2	GLN	CG	1	0	1				0	1				1	4			
28	TPEP	2	GLN	HG1	1	0	1				0	1				1	4			
29	TPEP	2	GLN	HG2	1	0	1				0	1				1	4			
30	TPEP	2	GLN	CD	1	0	1				0	1				1	4			
31	TPEP	2	GLN	OE1	1	0	1				0	1				1	4			
32	TPEP	2	GLN	NE2	1	0	1				0	1				1	4			
33	TPEP	2	GLN	HE21	1	0	1				0	1				1	4			
34	TPEP	2	GLN	HE22	1	0	1				0	1				1	4			
35	TPEP	2	GLN	C	1	0	1				0	1				0	5			
36	TPEP	2	GLN	O	1	0	1				0	1				0	5			
37	TPEP	3	GLY	N	2	0	2				0	2				0	6			
38	TPEP	3	GLY	HN	2	0	2				0	2				0	6			
39	TPEP	3	GLY	CA	2	0	2				0	2				0	6			
40	TPEP	3	GLY	HA1	2	0	2				0	2				0	6			
41	TPEP	3	GLY	HA2	2	0	2				0	2				1	7			
42	TPEP	3	GLY	C	2	0	2				0	2				0	8			
43	TPEP	3	GLY	OT1	2	0	2				0	2				0	8			
44	TPEP	3	GLY	OT2	2	0	2				0	2				0	8			
45	WAT	1	TIP3	OH2	3	0	3				1	3				2	9			
46	WAT	1	TIP3	H1	3	0	3				1	3				2	9			
47	WAT	1	TIP3	H2	3	0	3				1	3				2	9			
48	WAT	2	TIP3	OH2	4	0	4				1	4				2	10			
49	WAT	2	TIP3	H1	4	0	4				1	4				2	10			
50	WAT	2	TIP3	H2	4	0	4				1	4				2	10			
51	WAT	3	TIP3	OH2	5	0	5				1	5				2	11			
52	WAT	3	TIP3	H1	5	0	5				1	5				2	11			
53	WAT	3	TIP3	H2	5	0	5				1	5				2	11			
54	WAT	4	TIP3	OH2	6	0	6				1	6				2	12			
55	WAT	4	TIP3	H1	6	0	6				1	6				2	12			
56	WAT	4	TIP3	H2	6	0	6				1	6				2	12			
57	WAT	5	TIP3	OH2	7	0	7				1	7				2	13			
58	WAT	5	TIP3	H1	7	0	7				1	7				2	13			
59	WAT	5	TIP3	H2	7	0	7				1	7				2	13			
60	WAT	6	TIP3	OH2	8	0	8				1	8				2	14			
61	WAT	6	TIP3	H1	8	0	8				1	8				2	14			
62	WAT	6	TIP3	H2	8	0	8				1	8				2	14			
63	WAT	7	TIP3	OH2	9	0	9				1	9				2	15			
64	WAT	7	TIP3	H1	9	0	9				1	9				2	15			
65	WAT	7	TIP3	H2	9	0	9				1	9				2	15			
66	WAT	8	TIP3	OH2	10	0	10				1	10				2	16			
67	WAT	8	TIP3	H1	10	0	10				1	10				2	16			
68	WAT	8	TIP3	H2	10	0	10				1	10				2	16			
69	WAT	9	TIP3	OH2	11	0	11				1	11				2	17			
70	WAT	9	TIP3	H1	11	0	11				1	11				2	17			
71	WAT	9	TIP3	H2	11	0	11				1	11				2	17			
72	WAT	10	TIP3	OH2	12	0	12				1	12				2	18			
73	WAT	10	TIP3	H1	12	0	12				1	12				2	18			
74	WAT	10	TIP3	H2	12	0	12				1	12				2	18			

```

1) TASK {
    SELECTION {TYPE TESSELATION
      AG(SEGMENTTYPE *) }
    SELECTION {TYPE CORE
      AG(SEGMENTTYPE TPEP) }
    SELECTION {TYPE SURROUND
      AG(SEGMENTTYPE WAT) }
}

```

```

2) TASK {
    SELECTION {TYPE NEIGHBOURMAP
      SEPARATE RESIDUE
      AG(RESIDUENAME MET, GLN, GLY) }
    SELECTION {TYPE NEIGHBOURMAP
      AG(SEGMENTTYPE WAT) }
    SELECTION {TYPE TESSELATION
      AG(SEGMENTTYPE *) }
    SELECTION {TYPE CORE
      SEPARATE ATOMGROUP
      CONTRACT COG
      AG {RESIDUETYPE MET
        ATOMTYPE CE, HE1, HE2, HE3}
      AG {RESIDUETYPE GLN
        ATOMTYPE NE2, HE21, HE22 } }
    SELECTION {TYPE SURROUND
      AG(ATOMTYPE OH2) }
}

```

```

3) AG backbone {SEGMENTTYPE TPEP
  ATOMTYPE CA, HA, HA1, C, O, N, HN,
    OT1, OT2, HT1, HT2, HT3}
AG sidechain {SEGMENTTYPE TPEP
  ATOMTYPE HA2, CB, HB1, HB2, CG, HG1,
    HG2, CD, SD, CE, OE1, NE2,
    HE1, HE2, HE3, HE21, HE22}
AG water {SEGMENTTYPE TIP3}

TASK {
  SELECTION {TYPE NEIGHBOURMAP
    SEPARATE FUNCTIONAL AG backbone}
  SELECTION {TYPE NEIGHBOURMAP
    SEPARATE FUNCTIONAL AG sidechain}
  SELECTION {TYPE NEIGHBOURMAP
    SEPARATE RESIDUE AG water}
  SELECTION {TYPE TESSELATION
    AG backbone AG sidechain}
  SELECTION {TYPE TESSELATION
    CONTRACT COG
    SEPARATE RESIDUE AG water}
  SELECTION {TYPE CORE
    SEPARATE FUNCTIONAL AG sidechain}
  SELECTION {TYPE SURROUND AG water}
}

```

Figure 7.2: Spatial Selections

ject for each single atom while, analogously, `SeparateByResidueId` and `SeparateBySegmentId` yield residue-grained and segment-grained **AtomGroup** objects. If `SeparateNone` is specified, all selected atoms matching the pattern are put into one large **AtomGroup** object, while `SeparateFunctional` and `SeparateExplicit` allow for an arbitrary grouping of atoms. Concretely, `SeparateFunctional` applies the same grouping to every single residue matched by the selection pattern. Examples are the classification of backbone and sidechain shown in Fig.7.2 or the “Head”, “Ring”, “Tail” classification as done in Chapter 2. On the contrary, `SeparateExplicit` allows for maximal flexibility when selecting atoms and explicitly creates one single **AtomGroup** object for each “AG” clause in the script. This means, using `SeparateExplicit`, any combination of atoms can be collected into an **AtomGroup** object, which includes the possibility to create an group consisting of atoms that don’t even belong to the same molecule or segment. Such a grouping of atoms seems useless at first sight, but it emphasises the freedom of `SeparateExplicit`. As atoms are being grouped, they can be represented by a common center-of-mass, a common dipole and the like. The distinction between center-of-mass and center-of-geometry is made by the additional parameter `Contraction` which can take the three different values `ContractCOM`, `ContractCOG` and `ContractNone`. The latter uses the coordinates of the first atom in the **AtomGroup** object in order to locate the whole group. In most cases, `ContractNone` is used in combination with `SeparateAtom`.

For an example, let’s assume a protein structure file like the one depicted in the leftmost block of Fig.7.2. It contains the tripeptide MET-GLN-GLY solvated in 10 water molecules. If the center-of-mass of each amino acid is to be selected, first of all a matching pattern has to be created that matches every atom of these three residues. The simplest way to achieve this in our example is to select the whole segment TPEP (see yellow example in Fig.7.2). The next step is the specification of the `Separation` parameter. Choosing `SeparateByResidueId` yields three distinct **AtomGroup** objects, each containing all atoms of one of the three amino acids. As we want each **AtomGroup** to be represented by its center-of-mass, we choose the `ContractCOM` value for member variable `Contraction`.

The result of this selection process is being collected in a **SingleSelection** object. GEPETTO allows to combine multiple **SingleSelection** objects that emanate from different selection patterns, separations and contractions into an object of class type **MultiSelection**. These **MultiSelection** objects are used as input for specific calculations. Currently, each spatial selection has up to four different types of **MultiSelection** objects, two of which are needed for every $O(N^2)$ calculation. They are called `CoreSelection` and `SurroundSelection` and define the set of reference points and test points, respectively. For these two selections, the defaults are `SeparateByResidueId` and `ContractCOM`

as specified in the source file `"/source/script/gep/gep.cpp"`. The third selection is needed for Delaunay tetrahedralization. It is called `TessellationSelection` and by default is separated by atom number (`SeparateByAtomId`) and thus not contracted (`ContractNone`) leading to an “all atom” or atom-grained tessellation.

The fourth **MultiSelection** type being called `NeighbourMapping` is special in a way that its `AtomGroups` have no representation and therefore it ignores contraction. As the name indicates, the `NeighbourMapping` is used to relate the other three **MultiSelection** types to each other. In the code two member variables hold information about these relations: `SpeciesId` and `CollationId`. Every **SingleSelection** object of `NeighbourMapping` defines a distinct species (e.g backbone or sidechain). Iterating through all atoms in the structure, a new `CollationId` is assigned to each **AtomGroup** object, every time either the `ResidueId` or the `SpeciesId` change. If no `NeighbourMapping` is specified in the script, a default mapping is being created that is separated “by residue”. That means, every selected residue is part of the same single species. The values of `SpeciesId` and `CollationId` both are being set in all four `MultiSelections` as they are defined by `NeighbourMapping`. Thus, two constraints must be met. First, this special `MultiSelection` has to contain the superset of all `StructureAtoms` defined in `Core`, `Surround` or `Tessellation`. Second, its graining (i.e. Separation) must be equally coarse as or coarser than that of the other selections.

Example 1 in Fig. 7.2 shows the minimum requirements for a tessellation task in the GEPETTO script syntax (yellow box to the right). Furthermore, it shows the impact on the respective **AtomGroup** objects’ variables `collationId` and `speciesId` (yellow block in the table). Actually, if no tessellation is required, the corresponding `TESSELTATION` selection can be omitted. For no `NeighbourMapping` has been specified, the default is only one species with `SpeciesId`= 0 as described above. Therefore, the `CollationId` is identical to the `UniqueResidueId` both starting at a value of 0 and being incremented every time the original psf residue number changes. The `TessellationSelection` explicitly matches all atoms (`SEGMENTTYPE *`). The default `SeparateByAtomId` yields one `AtomGroup` for each atom (the grey bordered boxes). The `CoreSelection` specifies all residues of the tripeptide (`SEGMENTTYPE TPEP`) while the `SurroundSelection` refers to all water molecules (`SEGMENTTYPE WAT`). Both `MultiSelections` consisting of one `SingleSelection` each have the default value `ContractCOM`. Example 2 is somewhat more elaborate. It shows how to select the center-of-geometry of the two terminal functional groups of MET and GLN as reference points (`CORE`) and the water oxygens as test points (`SURROUND`). Because two explicit `SingleSelection` objects of type **NeighbourMapping** are specified, two distinct values for `SpeciesId` are assigned. Finally, Example 3 defines three species,

7 GEPETTO - Implementation

the first being the peptide's backbone, the second being the side chains of each residue and the third one, again, being water. For more examples, please read the User Guide chapter (8). During the optimization process (see Sec. 7.2.4) **FrameType** objects collect all the needed **MultiSelection** objects in their member variable `SpaceSelection` which is of type `AtomGroupSelection`.

instruction/	selection.h, selection.cpp , <code>atomgroup.cpp</code> , <code>atomgroup.h</code> , <code>frametype.h</code> , <code>tasklist.cpp</code> , <code>taskresource.h</code> , <code>pattern.cpp</code> , <code>pattern.h</code> , <code>timeseries.h</code> , <code>timeseries.cpp</code> , <code>task.h</code>
mdsystem/	<code>structure.h</code> , <code>structure.cpp</code>

Temporal Selections

The temporal selection framework (class **SystemSelection**), as implemented by Thomas Taylor, allows for two different types of time selection specification. One is reflected by the `RegularTimeLocations` while the other one is called `ExtraTimeLocations` and both are members of the **SystemSelection** class. However, only `RegularTimeLocations` are supported throughout the code of GEPETTO. Actually, it defines a set of 3-tuples of the form: { first, last, increment}. Here, first and last are the first and last frame under consideration, while increment denotes the spacing of frames. In GEPETTO script syntax, first is called `FIRSTFRAME` and increment is called `GRAINING`. It has to be emphasised, that the parameter `MAXFRAME` in the user script does not denote the last frame but the total (maximum) number of frames to be considered. An increment of 12 means that every twelfth frame is used for calculation, all the other frames being ignored. Note that increment refers to saved frames rather than simulated frames. If every 10th simulated frame is stored in the `dcd` file, an increment of 10 means every 100th simulated frame is considered. For a more detailed description, please have a look at the **doxygen** documentation and read the User Guide chapter for some examples on this topic.

mdsystem/	mdselection.h, mdselection.cpp , <code>mdtime.h</code> , <code>mdtimeselector.h</code>
instruction/	<code>frametype.h</code>

7.2.4 Optimization

Technically, the term “optimization” names the process of creating CS objects out of IS objects in GEPETTO terminology. Thereby, objects of IS classes (**Task**, **Histogram**, **Timeseries**, **Shell**, **AtomGroupSelection**...) are used as templates for the creation of the respective CS classes (**calculationFlagGroup**, **caHistogram**, **caTimeseries**, **caShell**, **Block**...). The **ca...** classes are described in Sec.7.3 while this section concentrates on the optimization itself.

In user scripts (see Section 7.2.2), the task is the most prominent concept. The user is free to enter tasks in any order regardless of redundancy and potential reuse of intermediate results. Optimization reorganises the tasks at three hierarchical levels in a way that they can be computed efficiently. The first level is the tessellation selection level, the second one copes with core and surround selections ($O(N^2)$, see Sec.7.2.3) and the third one is the level of $O(N)$ calculations. By this means, all calculations defined by **Task** objects that share the same **TessellationSelection** are grouped into one object of type **calculationFlagGroupTessellation** which in turn contains multiple **calculationFlagGroupDS** (see Sec.7.2.4). Flag groups of these types are member variables of the **FrameType** class (see Sec.7.2.4).

In order to allow for vectorization and loop optimization as well as for efficient communication needed for parallelism, most data structures in use are guaranteed to be contiguous. This is reflected by the code in two different ways: First, spatial selections might overlap in terms of their atom groups, so the results of $O(N)$ calculations are expanded into sequential blocks of data for $O(N^2)$ calculations via the **caExpandMask** class explained in Sec.7.2.4. Second, the two main data containers **Histogram** and **Timeseries**, being used to store intermediate and final results, are organised consecutively in CS (see Sec.7.2.4).

./	globals.h
calculation/	caAtomGroup.h, caAtomGroup.cpp, caHistogram.h, caHistogram.cpp, caShell.h, caStructure.h, caTimeseries.h, caTimeseries.cpp, eff.h, caExpandMask.h, ptrVector.h
instruction/	frametype.h, frametype.cpp, tasklist.cpp

Calculation Flags and Groups

CalculationFlagGroup (CFG) objects work like functors that contain all the information necessary for calculation. They present the central interface for algorithm invocation in GEPETTO. During optimization, the CFG objects are created using the information of each **Task**'s CalculationFlags. They are divided into five different types that differ in computational or data complexity. For each of these types special flags indicate what functions have to be calculated (see namespace mode in the file "source/globals.h").

The creation of respective CFG objects is implemented in the following optimization functions:

```
FrameType::optimize_N(), FrameType::optimize_N2(...),
FrameType::optimize_1(), FrameType::optimize_TesselationGroups() and FrameType::optimize_-
HbondTimeseries(...).
```

Each of them contains information on the observables (CalculationFlags) that have to be calculated by applying special functions collected in a `std::vector` of function pointers. These function points are of type `fpAlgo` which is defined in "source/algorithm/algorithm_pre.h". A Selection in IS becomes one or more **Blocks** in CS.

Currently, the following classes are implemented:

- The interface for time series calculations of complexity $O(1)$ is defined by the class **calculationFlagGroup1**. It has pointers to time series and a `calculationFlagGroup1::setTime()` function. This function is called once for each frame and sets the pointers to memory addresses corresponding to the particular time. Generally, these time series are data dependent on $O(N)$ or $O(N^2)$ calculations. This part of GEPETTO was developed by Michael Haberler.
- The class **calculationFlagGroup** is used for $O(N)$ calculations. In addition to the CalculationFlags it contains a `std::vector` of type `algoN::fpAlgo` pointing to distinct Algorithms. There is only one object of type **Block** called A containing the indices of the first and last `caAtomGroup` item corresponding to one specific **MultiSelection**.
- Objects of class type **calculationFlagGroupDS** mediate $O(N^2)$ calculations, so two **Block** objects named A and B are defined. Thereby, A stands for the core selection and B for the surround selection. The algorithms function pointer vector is of type `std::vector<algoN2::fpAlgo>`. Additionally, the class **calculationFlagGroupDS** contains pointers to **caTimeseries** that are set to the current time position by the member function `calculationFlagGroupDS::setTime()`

for each frame. Here, DS means double selection which describes a pair of a CoreSelection and a SurroundSelection used for $O(N^2)$ calculations. Pointers to transition matrices and histograms are called TransitionMatrix and histos or histos3d, respectively.

- Invocation of Delaunay tessellation is wrapped by the class **calculationFlagGroupTessellation**. It contains only one **Block** named C representing the tessellation selection. Usually, neighbour information is needed at another level of granularity than tessellation (see Sec.7.2.3). In most cases, neighbour interaction is derived for residues or molecules instead of atoms. This is the reason for the presence of the coarse2fine and fine2coarse mapping at this point. The larger number calculationFlagGroupTessellation::N of fine (i.e. atom-resolved) tessellation points is mapped to the smaller number calculationFlagGroupTessellation::n of coarse neighbours (i.e. residues or molecules). As tessellation is the most computationally intensive calculation, it has been put to a higher hierarchy level within this calculation group concept. This is expressed by the member variable std::vector<Srp<calculationFlagGroupDS> > calcGroups of **calculationFlagGroupTessellation**. Each **calculationFlagGroupDS** object belongs to exactly one **calculationFlagGroupTessellation** object which allows multiple $O(N^2)$ calculations to use the same tessellation. This also holds true if tessellation is not needed. In this case a **calculationFlagGroupTessellation** object is created with the flag NEEDSHELL set to false.
- A separate interface for hydrogen bonds has been implemented by Michael Haberler. It is called **calculationFlagGroupHB** and contains a special set of vectors that are being used to store donor and acceptor data as well as hydrogen bond information.

./	globals.h
algorithm/	<i>algorithm_1.h, algorithm_1.cpp,</i> <i>algorithm_N2_base.h, algorithm_N2_base.cpp,</i> <i>algorithm_N.h algorithm_N.cpp</i>
calculation/	caShell.h, eff.h
instruction/	<i>frametype.h, frametype.cpp, task.h, task.cpp,</i> <i>tasklist.cpp</i>

Frame index:	1	2	3	4	5	6	7	8	9	10
FrameType:	F_{12}		F_1	F_2	F_1		F_{12}		F_1	F_2
Task T_1:	x		x		x		x		x	
Task T_2:	x			x			x			x

Table 7.1: The Principle of **FrameType**

Frame Types

One central concept of optimization is the so-called **FrameType**. It has been introduced as a container for the set of calculations that are to be done for a specific frame. If two tasks called T_1 and T_2 are to be performed on the same MD system but with a different time graining (see Section 7.2.3), three different **FrameType** objects are constructed: F_1 containing all calculations for T_1 , F_2 for T_2 and a third one for frames that are needed by both tasks T_1 and T_2 which shall be named F_{12} . Assuming a graining of 2 for T_1 and a graining of 3 for T_2 , the sequence of frames and their respective frame types would look like depicted in Table 7.1. The frames (1,7...) are of type F_{12} because they are needed for both tasks, while frames (3,5,9...) and (4,10...) are of respective **FrameType** F_1 and F_2 . All other frames can be ignored and are actually not even read by the trajectory reader.

FrameType objects are instantiated in the function `TaskList::optimize_FrameTypes()`. Thereby, a table is being created listing all required properties or observables (columns) of all tasks (rows). The properties can be divided into three groups: (1) Keys identifying the task including general properties, (2) the start and end values of the temporal selection defining which frames are concerned and (3) binary flags indicating which observables are to be calculated. During **FrameType** optimisation, the table, holding one row for each task at the beginning, is being reduced successively according to the congruence in the key values (1). Frame selection (2) and binary flags (3) are being contracted thereby using aggregate functions. Actually, “first frame” is the only property that is being minimised, while all others are being maximised. The key fields (1) are `traj`, `inc`, `shell`, `core` and `surr` each containing an index of the respective objects in the **TaskResource** collection. All tasks sharing the same trajectory, shell definition and tessellation selection at the same temporal graining are grouped together into one **calculationGroupTessellation** that is registered to all respective **FrameType** objects. This whole process results in a more or less diverse collection of **Frame** objects being annotated a **FrameType** object ready for calculation.

instruction/	<code>frametype.h, frametype.cpp, tasklist.cpp,</code> <code>taskresource.h</code>
--------------	---------------------------------------------------------------------------------------

Histograms and Time Series

The result container classes **Histogram** and **Timeseries** have been designed with parallelisation in mind. A sequential organisation of data leads to a performance gain whenever data has to be transmitted or broadcast between multiple machines over a network, because the transmitting functions (e.g. MPI) work more efficient on larger contiguous blocks of data reducing communication overhead. This is the reason why the memory necessary to store the CS counterparts **caHistogram** and **caTimeseries** is allocated in cohesive blocks. These data containers can get quite large in size, which renders a double allocation impractical. The solution used here is to create IS objects with pointers to CS allocated memory. This way, the actual data is contiguous and, at the same time, IS objects provide a more or less elegant data accessibility for postprocessing and output. In CS, both **caHistogram** and **caTimeseries** have headers that contain information about memory size, dimensions, bin width, upper and lower bounds and so on. The headers are documented in the code files "source/calculation/caHistogram.h" and "source/calculation/caTimeseries.h". The memory that is needed for the header is attached directly at the beginning of the respective data blocks.

Timeseries data can be of one of three types: fully expanded, compressed binary and compressed integer. The more general integer compression style is used for the neighbour function $n(t)$ offering the possibility for non-binary neighbourhood relations. Hydrogen-bond time series are coded using binary compression.

Note that **Histogram** objects need to be specified explicitly in the script file, using at least the parameters BINWIDTH and BINCOUNT while **Timeseries** do not. Rather they are defined by the **Trajectory** objects' time selection. GEPETTO allows only one format per task, but multiple Timeseries and Histogram objects.

instruction/	histogram.h , histogram.cpp , timeseries.h , timeseries.cpp , <i>tasklist.h</i> , <i>tasklist.cpp</i> , <i>taskresource.h</i>
calculation/	caHistogram.h , caHistogram.cpp , caTimeseries.h , caTimeseries.cpp , <i>eff.h</i>
instruction/distribution/	<i>task_distribution.cpp</i> , <i>task_distribution.cpp</i>
instruction/gfunction/	<i>task_gfunction.cpp</i> , <i>task_gfunction.cpp</i>
instruction/timeseries/	<i>task_timeseries.cpp</i> , <i>task_timeseries.cpp</i>
instruction/correl/	<i>task_correl.cpp</i> , <i>task_correl.cpp</i>

Expand Mask

Sequentiality of data does not only have advantages in conjunction with parallelity. It allows for an efficient way of data access in loops and provides the possibility of loop vectorization which can yield performance gains of orders of magnitude. As described above, calculation is split into a $O(N)$ and an $O(N^2)$ part, where the latter is data dependent on the former. In other words, results of $O(N)$ calculations are needed as input for $O(N^2)$ calculations.

Now, $O(N)$ calculations are applied to the object `postN_AGV` of type **caAtomGroup** being a non-redundant list for all tasks and all selections of one MD system. By this means, every $O(N)$ calculation is done only once per atom group. This improves $O(N)$ performance. However, atom groups of all selections are scattered in `postN_AGV` which is suboptimal for $O(N^2)$ calculations. To overcome this, the results in `postN_AGV` are being expanded into a new **caAtomGroup** object called `preN2_AGV` using an expand mask (**caExpandMask**). The resulting list `preN2_AGV` is organised in a way, that all data belonging to the same single selection are contiguous. Contrary to `postN_AGV` which is non-redundant in terms of atom groups but possibly scattered as far as selections are concerned, `preN2_AGV` can be redundant in terms of atom groups but the atom groups of one selection are stored one after the other. Technically, the **caExpandMask** is a map of atom group indices in `postN_AGV` and the respective indices in `preN2_AGV` which is being created only once but used for every frame. This way, loops in the **algoN2** namespace can iterate over the data by efficient pointer incrementation instead of having to apply a more complicated and thus more expensive pointer arithmetics.

In the scheme displayed in Fig.7.3, for two selections different $O(N)$ and $O(N^2)$ calculations are to be performed. For Selection I, the center-of-mass (COM) and the dipole (DIP) are required while Selection

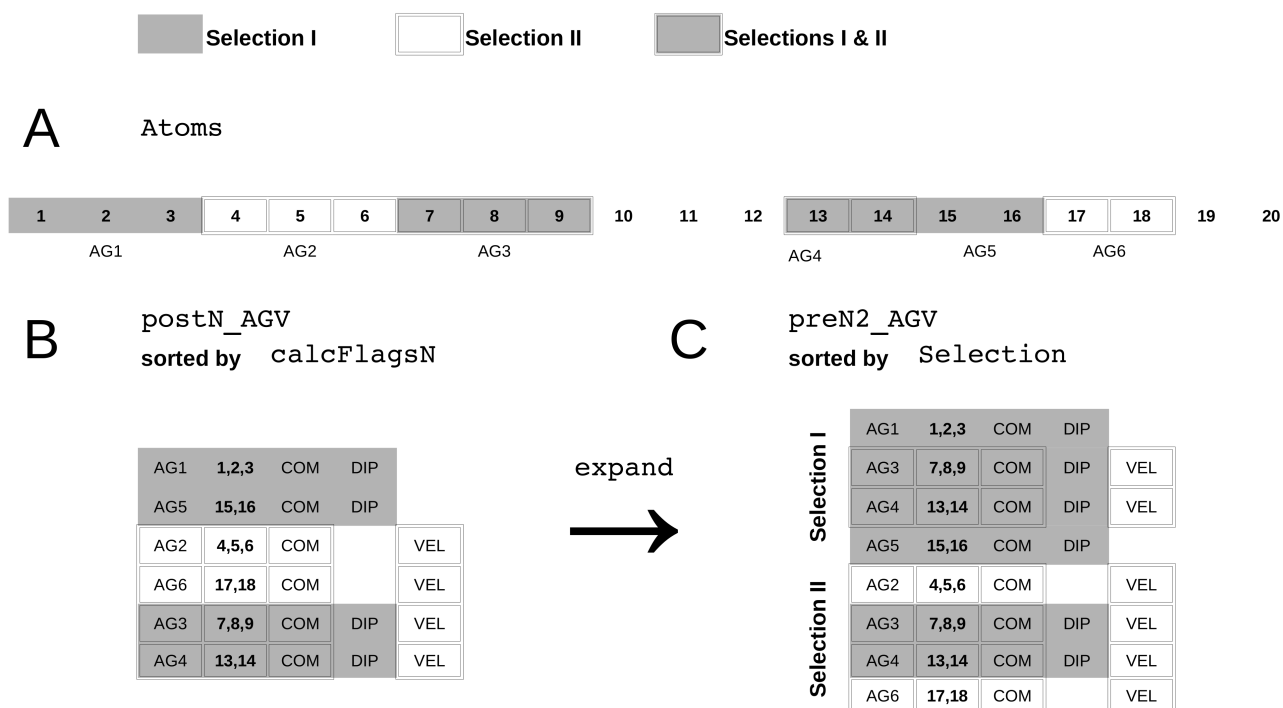


Figure 7.3: Expanding $O(N)$ results to $O(N^2)$ input: A) Selected atoms are displayed shaded (Selection I) or double-bordered (Selection II). B) The object `postN_AGV` collects results from $O(N)$ calculations for each atom group (AG1-AG6). C) For $O(N^2)$ calculations, `postN_AGV` is expanded to `preN2_AGV`.

7 GEPETTO - Implementation

II needs COM and velocity of the center-of-mass (VELCOM). Part A in the picture shows 20 hypothetical atoms from which atoms 1,2,3,7,8,9,13,14,15 and 16 are selected by Selection I and 4,5,6,7,8,9,13,14,17 and 18 are selected by Selection II. Atoms 7,8,9,13 and 14 occur in both selections, thus the two selections are overlapping. As both selections need the COM to be calculated, it is possible to save computation time by computing it only once per atom group. Therefore, the **caAtomGroup** object `postN_AGV` is used. It is sorted by the atom groups' calculation flags: For atom groups 1 and 5 the flags `mode::COM` and `mode::DIP` are set, for atom groups 2 and 6 `mode::COM` and `mode::VEL` is needed. Atom groups 3 and 4 are members of both selections, thus `mode::COM`, `mode::DIP` and `mode::VEL` is to be calculated. This sorting, allow the compiler to vectorise memory access. The second advantage of this organisation is the possibility to use combined calculation functions for COM+DIP, COM+VEL and COM+DIP+VEL, again, potentially saving computation time. Of course, in real systems these lists consist of many atom groups each but, for the sake of clarity, the example contains only two atom groups per calculation flag group. After the $O(N)$ calculations for a distinct frame are finished, `postN_AGV` is expanded to the selection-ordered and redundant `preN2_AGV` using the predefined **caExpandMask**. Again sequentiality is the key, but this time, atom groups belonging to one selection are organised in a contiguous and ordered way. Each **FrameType** object has a member object of type **caExpandMask** called `expandMask`. It is initialised in the member function `FrameType::optimize_N2()`. During calculation, the `algoN::expand(...)` method is invoked by the `FrameType::doCalculations(...)` method. It creates a deep copy of each item in `postN_AGV` and stores them in `preN2_AGV` using the `expandMask`.

instruction	<i>frametype.h, frametype.cpp</i>
calculation	expand_mask.h

7.2.5 Program Output

Postprocessing

In most cases, intermediate results have to be further processed in one of several ways which is referred to as “postprocessing”. The necessary information for different kinds of postprocessing (e.g. normalization) is stored in the headers of **caHistogram** and **caTimeseries**. After calculations are finished, the singleton **TaskList** iterates over all **Task** objects and invokes the respective `...::postProcess()` method. Histograms of g-functions need to be normalised which is done in

`Task_gFunction::postProcess()`. Histograms used for calculations other than g-functions are normalised in `Task_Distribution::postProcess()`. Three dimensional histograms are being normalised in `Task_Density3d::postProcess()`. Time series don't need postprocessing in general, but they need to implement the virtual base class method `Task::postProcess()` which leads to an empty method corpse of `Task_timeseries::postProcess()`. Another type of postprocessing is the correlation of time series. It is conducted by `Task_Correl::postProcess()` which differentiates between compressed and non-compressed time series. Hydrogen-bonds combine both, time series as well as correlation which is performed in `Task_Hbond::postProcess()`. Thereby, a very efficient encoding of binary time series has been used. `Task_MSD::postProcess()` calculates the mean square displacement of time series.

<code>instruction/</code>	<code>tasklist.h, tasklist.cpp</code>
<code>instruction/gfunction/</code>	<code>task_gfunction.h, task_gfunction.cpp</code>
<code>instruction/distribution/</code>	<code>task_distribution.h, task_distribution.cpp,</code>
<code>instruction/timeseries/</code>	<code>task_timeseries.h, task_timeseries.cpp</code>
<code>instruction/correl/</code>	<code>task_correl.h, task_correl.cpp</code>
<code>instruction/hbond/</code>	<code>task_hbond.h, task_hbond.cpp</code>
<code>instruction/msd/</code>	<code>task_msd.h, task_msd.cpp</code>
<code>calculation/</code>	<code>caTimeseries.h, caTimeseries.cpp, caHistogram.h,</code> <code>caHistogram.cpp</code>

Reporting and Output

One can distinguish between reporting of results occurring after completion of postprocessing, on the one hand, and output being produced constantly at runtime giving meta-information on parsing, optimization, calculation and postprocessing, on the other hand. Reporting of results is managed by the **Task** class and its child classes. Generally, each class derived from **Task** implements its virtual `Task::dataOut()` function. Output files are specified in the script and checked during the parsing process (see Sec.7.2.2). After calculation and postprocessing are finished, `TaskList::storeResults()` iterates over all **Task** objects and invokes the respective output function. These `Task::dataOut()` functions, in turn, iterate over results either stored in a **Histogram** or **Timeseries** object and print formatted results to the file "outfilename.<SUFFIX>". The <SUFFIX> and the actual file formats

7 GEPETTO - Implementation

depend on the task type and are described in the User Guide in Sec.8.

Output generated at runtime is supported by output streams having been designed by Thomas Taylor and implemented consequently all over the code by Michael Haberler. They are declared in the "source/globals.h" and defined in "source/program/main.cpp". Currently, the following streams exist: `out::err`, `out::warn`, `out::info`, `out::log`, `out::stat`, `out::debug` and `out::parserdebug` which are all redirected to `std::cout`. Each output stream can be turned on and off according to the output levels `VERBOSE_OFF`, `VERBOSE_DEATH`, `VERBOSE_CRITICAL`, `VERBOSE_STRONG`, `VERBOSE_STANDARD`, `VERBOSE_WEAK`, `VERBOSE_SLIGHT`, `VERBOSE_EXTRA` and `VERBOSE_ALL`, given in descending order of selectivity. Every message that is printed can be assigned an output level using the `out::level()` method. Assigning an output level to a stream in "source/program/main.cpp" using the `out::show()` method causes this stream to print all messages of at least that level.

./	globals.h
instruction/	tasklist.cpp
instruction/correl/	task_correl.cpp
instruction/density3d/	task_density3d.cpp
instruction/distribution/	task_distribution.cpp
instruction/gfunction/	task_gfunction.cpp
instruction/msd/	task_msd.cpp
instruction/hbond/	task_hbond.cpp
instruction/timeseries/	task_timeseries.cpp
program/	main.cpp

7.3 Calculation Space (CS)

The Calculation Space collects all algorithms and necessary data structures for computation. Therefore, the directory "source/calculation" contains calculation flag group classes along with CS analoga of IS classes (see Sec.7.2.4): The **Shell** analogon in CS is **caShell**. What is a **AtomGroupSelection** in IS becomes a **Block** in CS. Further classes are **caTimeseries**, **caHistogram**, **caAtomGroup** and **caStructure**. In CS, the `collationId` and other information can be accessed via pointers. The names of these pointers include one of the three markers A, B and C which stand for `CoreSelection`, `SurroundSelection` and `TesselationSelection`, respectively. For example, the `collationId` of

AtomGroup objects of the SurroundSelection can be accessed via `COLLB[index]`, with `index` being an integer indicating the respective **caAtomGroup** item within **Block B** which is actually the SurroundSelection.

The calculation space is structured by data complexity. Results for single atom groups like center-of-mass, center-of-geometry, orientation matrix, velocity of center-of-mass, angular velocity, or dipole are of complexity $O(N)$ and collected in the **algoN** namespace. Analogously, calculations that yield one single value or a set of values per frame are found in the **algo1** namespace. Computations of higher complexity are summarised in the **algoN2** namespace. This includes distances or tessellation ($O(N^2)$). A further namespace `algoImage` has been created that contains functions coping with periodic boundaries. This section is structured by these namespaces and describes the implementation of respective algorithms. Generally, the concept of function pointers is used quite frequently in CS which has two major advantages: First, the code is clear and more easy to maintain because there is only one call to a function pointer instead of a lengthy conditional expression with several calls to appropriate functions. Second, the function pointers can be set at the earliest possible moment during runtime, which means, that compared to multiple conditional expressions like nested `if` or `switch` blocks within loops, the decision has to be made less frequently leading to a better overall performance.

algorithm/	algorithm_1.h, algorithm_1.cpp, algorithm_N2_base.h, algorithm_N2_base.cpp, algorithm_N2_geometry.h, algorithm_N2_geometry.cpp, algorithm_N2_neighbour_interaction.h, algorithm_N2_neighbour_interaction.cpp, algorithm_N2_probability.h, algorithm_N2_probability.cpp, algorithm_N2_solvation_shell.h, algorithm_N2_solvation_shell.cpp, algorithm_N.h, algorithm_N.cpp, algorithm_pre.h, images.h, images.cpp, images_pre.h
/calculation/	eff.h

7.3.1 Periodic Boundaries

Two different types of periodic boundaries have been implemented by Thomas Taylor that differ in their geometry. A primary cell or **Crystal** can either have the geometric form of a cube or that of a truncated octahedron. The type of crystal is automatically determined by the member function `Crystal::identifyType()` using the information encoded in the trajectory header. All functions for calculation of toroidal shifts are declared and defined in the **algoImage** namespace. It defines a global pointer to a **Crystal** object called `currentCrystal` that can be accessed from any point in CS and is updated from the ".dcd" file every time `TaskList::calcStuff()` reads a new frame. Thus, GEPETTO can analyse both, NVT and NPT ensembles. Generally, the **algoImage** namespace provides functions to calculate minimum image distances (`algoImage::getFunctionDist(...)`), to translate points to a distinct image (`algoImage::getFunctionTranslate(...)`) and to apply rotations to a point and fold the resulting point into the primary image (`algoImage::getFunctionTranslateRotateFold(...)`). As these functions are invoked very frequently throughout a GEPETTO run, it pays off to optimise them according to two criteria: First, as already mentioned, there are two different types of crystals for both of which a separate set of specific functions are implemented. Second, for each of these two crystal types, **algoImage** provides different modes of calculation, leading to the following combinations: CUBE functions can be of type "coordinate transform" (CT), "coordinate transform vectorizable" (CTV) or "cycling boundary conditions" (CBC, default). OCTA functions can also be of type CT (default) or CBC but there is a different third type called "best transformation" (BT). CT means, the vector in cartesian coordinates is transformed into a vector in system specific coordinates and, after all the math is done, it is transformed back. Depending on the compiler and the hardware, the vectorizable version (CTV) might be beneficial. CBC is regarded as the "safest". All boundary conditions are checked repeatedly. BT, as the name indicates, checks all possible transformations before the best is applied. Generally, it is advisable to use the defaults CBC for cubic systems and CT for truncated octahedra.

algorithm/	images.h, images.cpp, delaunay_periodic.cpp, <i>algorithm*</i>
mdsystem/trajectory/	crystal.h, crystal.cpp

Orientation

Some analysis require superimposition of molecules. This functionality has been integrated using the “Closed-form solution of absolute orientation using unit quaternions” by Berthold K. P. Horn. Implementation and citation can be found in the file "source/algorithm/BertholdHorn.cpp". The interface function `BertholdHorn::GetRotation(...)` returns the rotation matrix which superimposes the test point set onto the reference point set. Here, test point describes variable atom coordinates of the core selection, while the reference point set is constant and has to be specified in an additional ".pdb" file. The coordinates in this file have to be centered at the origin, which means their center-of-geometry must be at (0/0/0) in terms of cartesian coordinates. A second feature of this file is the residue number. If it is set to a value of '-1' the coordinates apply to all respective atoms of that residue type. On the contrary, a value greater than 0 matches atoms of one distinct residue. During the selection process, these orientation coordinates are read from the file (`SingleSelection::readOrientationCoordinates()`) and assigned to respective **AtomGroup** objects (`AtomGroup::addOrientationCoordinates(...)`). During calculation, the rotation matrices (`caAtomGroup::AgRotation`) are computed by the function `algoN::calcOri(...)`. This already indicates that one rotation matrix exists for each selected **AtomGroup** object in each **Frame**. Additionally, a translation vector (`caAtomGroup::AgTranslation`) joining the center-of-mass and the origin is stored for each rotation matrix.

algorithm/	<code>algorithm_1.h</code> , <code>algorithm_N2_base.h</code> , <code>algorithm_N.h</code> , <code>algorithm_N.cpp</code> , <code>BertholdHorn.h</code> , <code>BertholdHorn.cpp</code>
calculation/	<code>caAtomGroup.h</code> , <code>caStructure.h</code> , <code>eff.h</code>
instruction/	<code>atomgroup.h</code> , <code>atomgroup.cpp</code> , <code>selection.h</code> , <code>selection.cpp</code>

7.3.2 Single Particle Observables - $O(N)$

We already encountered the center-of-mass (COM), velocity of center-of-mass (VEL), dipole (DIP) and the center-of-geometry (COG). These values are used to represent an atom group which consists of multiple atoms. Their actual calculation is implemented in namespace **algoN** and the results are being stored contiguously in an object of class type **caAtomGroup**. Pointers like `caAtomGroup::AgCom` or `caAtomGroup::AgVelX` can be used to access these values from within calculation space. Again,

optimization is mediated by function pointers. The method `algoN::GetFunctions(...)` returns a vector of function pointers according to the respective `CalculationFlags`. GEPETTO has two different kinds of functions for the calculation of observables: If an atom group is composed of more than one atom, the `algoN::calc...` functions are used. Properties of atom groups containing one single atom are simply copied from the atom's properties using the `algoN::copy...` functions.

algorithm/	<code>algorithm_N2_base.h</code> , <code>algorithm_N.h</code> , <code>algorithm_N.cpp</code>
calculation/	<code>caAtomGroup.h</code>

7.3.3 Neighbourhood Calculation - $O(N^2)$

All $O(N^2)$ calculations are based on the results of $O(N)$ calculations. The pointers described in the previous subsection (e.g. `caAtomGroup::AgCom`) allow a direct access to the expanded results (see Sec.7.2.4). However, there exists a second set of pointers that enables a more simple and potentially more efficient access to single **caAtomGroup** observables which is used in $O(N^2)$. These pointers are shortcuts for the somewhat lengthy **caAtomGroup** syntax. For example, the `collationId` of core selection atom groups is represented by the pointer `COLLA` which is set to `calcAtomGroup->AgCollationId[currentCalcGroup->A.first]`. Further examples are `COMxC` (the x component of the center-of-mass of atom groups selected for tessellation) or `DIPyB` (the y component of the electric dipole of atom groups in the surround selection). These assignments are being made by the functions `algoN2::setPointers(...)` and `algoN2::setPointersShell(...)` which are defined in the files "`algorithm/algorithm_N2_base.cpp`" and "`algorithm/algorithm_N2_solvation_shell.cpp`", respectively. If a neighbourhood definition based on Voronoi/Delaunay tessellation is needed, first of all, the frame has to be tessellated. As a second step, neighbour relations are inferred from the tessellation.

Voronoi Tessellation

The tessellation or tetrahedralization process is implemented in the class **Tessellation** which is, together with the auxilliary class **Tetrahedron**, defined in the file "`delaunay_periodic.h`" and implemented in "`delaunay_periodic.cpp`". The original algorithm and data structure has been adapted in three ways: First, it has been optimised for computational efficiency rather than memory efficiency, which means that the data structure has been extended to hold not only information about the tetrahedral network but also about coordinates of centers of circumcircles and other geometrical infor-

mation that are recalculated more often in the original algorithm, apparently due to memory saving issues. Therefore, in addition to the original datastructure (`Tetrahedron::pts[]`), a further array (`Tetrahedron::det[]`) has been introduced which contains information about geometrical features. The second adaptation concerns the initial conditions. Thompson suggests an initial tetrahedralization consisting of 40 tetrahedra. These tetrahedra are formed by eight points closest to the respective quadrant in the coordinate system. The author of this dissertation found, however, that an initial tessellation consisting of only 6 tetrahedra being formed by only one arbitrary point and its images is sufficient and more practical. On one hand, smaller systems can be tessellated using 6 tetrahedra as starting conditions. On the other hand, it can be directly applied not only to cubic boundary conditions but also to truncated octahedra. The third adaptation refers to the recentering of the “core” and is explained in more detail below.

Generally, the algorithm has the important advantage of computational efficiency. However, if systems are too small or contain many coplanar atoms, there is a certain probability that the tessellation process is not successful. It is to be emphasised that this is not implementation specific but rather algorithm related. In order to fix these shortcomings, cases where something went wrong are recognised and the whole tessellation is rejected. Using a new random order of insertion points, the process of tessellation is iterated until success. In order to avoid infinite looping, the loop is broken after 1000 iterations. Both, random seed and this value of 1000 iterations can be set in the interface function `Tessellation::tetrahedralize(...)`. A second basic parameter influencing the tessellation process is called `MIN_QUALITY`. It is a threshold for tetrahedron quality, based on the volume/surface ratio. Degenerate (e.g. coplanar) tetrahedra have a lower quality than nondegenerate ones. In the constructor of class **Tessellation**, an expected number of tetrahedra in a tessellation of N points is set to $6.77 \cdot N$ according to Baudson and Klein [Berechnung und Visualisierung von Voronoi-Diagrammen in 3D, März 2006]

The interface method `Tessellation::tetrahedralize(...)` takes three pointers to the three cartesian coordinate sets of points (vertices), the parameter regarding the minimum quality of intermediate tetrahedra (`MIN_QUALITY`) and a flag indicating if graphical debug output is to be produced (`DEBUGPRINT`). It iteratively calls `Tessellation::InsertVertices(...)` which takes a pointer to a `std::vector` of randomly ordered vertex indices and reflects the actual insertion algorithm: For every new vertex (or point), the “base” has to be identified (`Tessellation::findBase(...)`). This means, the tetrahedron containing the new point or one of its images has to be found. If the data structure is corrupted, which can happen in certain cases explained above, the whole tessellation process is stopped.

7 GEPETTO - Implementation

On success, the “core” is constructed recursively by `Tesselation::constructCoreRecursive(...)`. Please note, that “core” has a different meaning here than in the selection context. In the context of this algorithm, “core” denominates all tetrahedra whose Delaunay circumsphere criterion is broken by the newly inserted point. In other words, the “core” contains all tetrahedra of the preexisting tessellation that need to be removed and replaced by a new partial tessellation. The third adaptation of the original algorithm mentioned above has to do with the correction of the “core”. Each newly identified “core” is required to be convex or “star-shaped”. In the original algorithm this is obtained by considering those images of “core” tetrahedra that contain the primary location of the newly inserted point. This criterion is found to be less efficient than taking those images whose center-of-geometry is closest to the primary location of the insertion point which has been implemented in GEPETTO. Two additional functions have been implemented that can be used for debugging: `Tesselation::checkDelaunay()` and `Tesselation::checkConnections()`. The first one performs a complete global check of the Delaunay criterium, which claims that within the circumsphere of a tetrahedron no vertex exists other than the four constituting vertices. The second function inspects the data structure and prints warnings if it is corrupted. Since execution of these two functions is rather time consuming, it is recommended for debugging purposes only.

There exists some experimental code for the graphic representation of Delaunay tetrahedra and Voronoi polyhedra that can be used for visual debugging or illustration purposes. Delaunay representation is implemented for a single tetrahedron in `Tetrahedron::print(...)`. The method takes two arguments: The first is an integer that takes values 0 to 3 indicating the respective tetrahedron face. The second argument is a line thickness defaulting to 1. In order to use Delaunay graphics it is necessary to prepend the print-function call with a call to `vmdnewmol(...)`, which is defined in the file “source/globals.h”. If multiple Delaunay tetrahedra should appear as a single molecule in **vmd**, this can be achieved by a single call to `vmdnewmol(...)` followed by multiple invocations of `Tetrahedron::print(...)`. Voronoi graphics is more elaborate as it allows to draw images of whole shells rather than single polyhedra and includes not only periodic boundaries but also rotation. The three central methods of the interface are `Tesselation::clearGraph()`, `Tesselation::addFaceToGraph(...)` and `Tesselation::printGraph(...)`. The names already indicate the principle: A graph is implemented as a bifid resident data structure that stores vertices (points) and edges (connections). A Voronoi face can be added by a call to `Tesselation::addFaceToGraph(...)`. After all faces have been registered this way, `Tesselation::printGraph(...)` writes one of two possible representations of the graph to `std::cerr`, both of which can be displayed

using **vmd**. One type is a ".pdb" file containing pseudo-atoms and the keyword CONNECTION which is usually used to define covalent bonds. The other one is a **vmd** Tcl/Tk script like the one produced by the Delaunay graphics framework. For a large number of vertices, the pdb version is recommended because it is displayed much more efficiently than the Tcl/Tk version in **vmd**. For more details, the reader is referred to the source code.

algorithm/	algorithm_N2_solvation_shell.h,
	algorithm_N2_solvation_shell.h
algorithm/	delaunay_periodic.h, delaunay_periodic.cpp

Shell Definition

Once the tessellation process is finished, the resulting data structure can be used to define solvation shells. This is done in `algoN2::calcDelaunayShells()` defined in the file "algorithm_N2_solvation_shell.cpp". The member function takes a single integer argument indicating the maximum shell number. The reason for this lies in the recursive definition of delaunay shells. Each shell is defined on the basis of shells having a lower shell index which means for shell *d* all shells 0 up to *d*-1 are required before shell *d* can be assigned. The subroutine `algoN2::calcDelaunayShells()` creates three different data structures holding neighbourhood information. All three are defined in the file "source/algorithm/algorithm_N2_solvation_shell.h". The first one is called `neighbours` and contains collationId-resolved (by default residue-resolved) information about neighbourhood. Here, `neighbours[j]` is a set of integer values indicating the neighbour **AtomGroup** objects of **AtomGroup** *j*. The second data structure called `neighbourAtoms` is of the same type, but contains atom-resolved neighbourhood information, given that the `TessellationSelection` has the value `SeparateByAtomId` set (see Sec.7.2.3). Both, `neighbours` and `neighbourAtoms` can be used to directly infer interaction partners. They are also used to fill the third data structure, the integer matrix `delaunay_shells`. Given a tessellation of *N* atoms belonging to *n* atom groups, `delaunay_shells` is a *n* × *n* matrix and element (*i*,*j*) is the shortest path in the delaunay graph between **AtomGroup** *i* and **AtomGroup** object *j*. At the same time, this is the definition for *j* being in the `delaunay_shells[i,j]th` shell of *i* and vice versa. Generally, the calculation of radial shells is prepared but neither has it been used nor tested. However, analogously to the `delaunay_shells` matrix, a matrix called `radial_shells` exists. The shells being defined here are used in many parts of the code, mostly in **algoN2**.

```
algorithm/  algorithm_N2_base.cpp,  algorithm_N2_geometry.cpp,
           algorithm_N2_neighbour_interaction.h , algorithm_
           N2_neighbour_interaction.cpp,      algorithm_N2_
           probability.h, algorithm_N2_probability.cpp
           , algorithm_N2_solvation_shell.h,
           algorithm_N2_solvation_shell.cpp,
           delaunay_periodic.h , delaunay_periodic.cpp
```

7.3.4 Pair Properties $O(N^2)$

Pair properties, one of GEPETTO's main areas of application are implemented in the **algoN2** namespace. They describe properties of one particle or particle type (test points) relative to another one (reference point). In IS, the test points are selected via a SurroundSelection, while reference points are specified using a CoreCollection. As already mentioned before, in CS the corresponding selections are coded in objects of class type **Block** and named A (core) or B (surround). The **algoN2** namespace has been split into multiple source files due to its large extent. General features can be found in "algorithm_N2_base.h" and "algorithm_N2_base.cpp". The files "algorithm_N2_solvation_shell.h" and "algorithm_N2_solvation_shell.cpp" contain Voronoi shell specific features. Coordination, contact matrix, hydrogen-bonds, MDcage, and dipole decomposition are declared in "algorithm_N2_neighbour_interaction.h" and implemented in the corresponding ".cpp" file. Volume and surface related quantities are coded in "algorithm_N2_geometry.h" and "algorithm_N2_geometry.cpp", while probability related functionality like transition matrices and the promiscuity went to "algorithm_N2_probability.h" and "algorithm_N2_probability.cpp".

Coordination

A very intuitive relation of molecules is the average number of direct interaction partners as described by the Delaunay/Voronoi based coordination number CN. Three subroutines calculate the coordination number and related values. First, `algoN2::calcCN()` computes the coordination number for core and surround species as defined by the **MultiSelection** NeighbourMap and cumulates the values in a **caTimeseries** and/or **caHistogram** object. Second, the function `algoN2::calcNSHELL()` creates the time series $n(t)$ used for mean residence time calculation. Michael Haberler developed `algoN2::calcContactMatrix()` based on Voronoi coordination. As the name implies, it stores the

atom group resolved coordination number in a matrix.

```
algorithm/  algorithm_N2_neighbour_interaction.h,
            algorithm_N2_neighbour_interaction.cpp
```

Connectivity

Separating the coordination number by contact type yields the so-called connectivity. It must be said, that at the time this is being written, the concept of connectivity is not fully implemented but rather work in progress and thus to be seen as “experimental”. There exists a function called `algoN2::getConnectivity(...)` that can distinguish between atom type related contact types. The resulting decomposition of CN yields a set of additive coordination numbers: CN_{HH} , CN_{HO} , CN_{CH} , CN_{CO} , CN_{NH} , CN_{NO} , CN_{OH} , CN_{OO} , CN_{SH} and CN_{SO} . CN_{HO} and CN_{OH} as well as CN_{NH} and CN_{HN} account for possible hydrogen bonds. The first index marks the type of the core atom involved in the intermolecular Delaunay edge, while the second index stands for the surround atom (currently only water). In order to use connectivity, the function can be called anywhere as long as the matrix `delaunay_shell[]` is available. The function `algoN2::getConnectivity(...)` takes two atom indices `i` and `j` and an integer array `data[]` to return the output. The format of this array is as follows:

<code>data[0-9]</code>	The number of different atom-atom contacts. Thereby <code>data[0]</code> stands for hydrogen-hydrogen contacts (CN_{HH}) and <code>data[9]</code> contains the number of sulfur-oxygen contacts (CN_{SO})
<code>data[10]</code>	The proximity variable taking one of the values <code>[0...9]</code> indicating the type of the shortest delaunay edge between particle <code>i</code> and <code>j</code> . Again, a value of 0 means hydrogen-hydrogen contact.
<code>data[11]</code>	The length of the shortest Delaunay edge between particle <code>i</code> and <code>j</code> in Å.
<code>data[12]</code>	The distance between the COM of particle <code>i</code> and the COM of particle <code>j</code> in Å.
<code>data[13]</code>	A flag indicating if particle <code>i</code> and particle <code>j</code> form a hydrogen-bond. This distinction is based on Delaunay contacts and not to be confused with the traditional hydrogen-bond criteria. (work in progress)
<code>data[14-17]</code>	Internal data that can be used for output.

algorithm/	<code>algorithm_N2_neighbour_interaction.h,</code> <code>algorithm_N2_neighbour_interaction.cpp</code>
------------	-----------------------------------------------------------------------------------------------------------

MD Cage

Cage is a term often used to describe the innermost solvation layer or first Voronoi shell. Actually, the collective rotational dipole moment (M_D) of a cage is not really a pair property but rather a collective property of data complexity $O(N)$ and its ensemble average is of data complexity $O(1)$. However, the computational complexity is $O(N^2)$ as it necessitates the calculation of Delaunay tessellations and includes nested loops over both the core and the surround selection. It is implemented in the **algoN2** namespace and discussed here. The implementation by Michael Haberler is well embedded in the GEPETTO framework. Thus, the specialities of this feature are concentrated in the single function `algoN2::calcMDCage()`. Altogether there are three different observables being calculated in this function: The collective rotational dipole moment of selected Delaunay shells is called MDCAGE. The other two observables are complementary to it and named MDCORE, which is actually the same observable as DIP, namely the average dipole of the core selection, and MDBULK, which is the collective dipole of atom groups in the surround selection, which are not member of the cage.

algorithm/	<code>algorithm_N2_neighbour_interaction.h,</code> <code>algorithm_N2_neighbour_interaction.cpp,</code> <code>algorithm_N2_solvation_shell.h,</code> <code>algorithm_N2_solvation_shell.cpp</code>
------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Surface and Volume

Computation of the SURFACE and VOLUME properties are implemented in the **algoN2** namespace for the same reasons given for MDCAGE. The file "algorithm/algorithm_N2_geometry.cpp" comprises wrapper functions for these observables while the actual computation is performed in the functions `Tessellation::calcVoronoiVolumes(...)` and `Tessellation::calcVoronoiSurfaces(...)` that can be found in the file "delaunay_periodic.cpp". Surface areas are evaluated atom-wise as follows: Loop through all tetrahedra in the tessellation structure and check if the constituting vertices correspond to one item in the set of selected Atoms. For the sake of clarity, let's call this atom

A_1 . Member function `Tesselation::rotateAboutEdge(...)` takes three arguments indicating the found tetrahedron (T_1) and two vertices defining the Delaunay edge connecting A_1 to a neighbouring atom A_2 . This Delaunay edge E_{12} is orthogonal to the Voronoi face F_{12} whose surface area is to be calculated. `Tesselation::rotateAboutEdge(...)` “rotates” through the tessellation data structure, thereby visiting all tetrahedra sharing edge E_{12} . The circumcenters of these tetrahedra are identical to the Voronoi face’s vertices and thus collected in the vertex array `V[]`. The function `calcArea(...)` computes the area of the three dimensional planar polygon defined by `V[]` describing face F_{12} . In order to save computer resources, the area of a two dimensional projection of F_{12} is calculated and scaled according to the projection to obtain the actual 3D face’s surface area. Finally, `Tesselation::rotateAboutEdge(...)` returns this surface area and it is stored in a map of a pair of integers indicating the respective two atoms A_1 and A_2 , and the area of F_{12} encoded as double. This map is called `surfaces_n2` and represents the basic atom-resolved surface matrix for further use in GEPETTO. Volumes are calculated by the function `Tesselation::calcVoronoiVolumes(...)` by summation over all pyramids built by the involved faces and the atom A_1 . The base area of the pyramids are equal to the constituting surfaces F_{12} and the height is identical to the normal distance between this base area and atom A_1 and corresponds to half the length of edge E_{12} .

algorithm/	algorithm_N2_geometry.h, algorithm_N2_geometry.cpp, delaunay_periodic.h, delaunay_periodic.cpp
------------	---------------------------------------------------------------------------------------------------------------------------

Transition Matrix

For calculation of transition matrices, two functions are implemented in GEPETTO:

`algoN2::calcTransitionMatrix(...)` and `algoN2::calcTransitionMatrix2(...)`. The first one is used by default (see `algoN2::GetFunction(...)` in `"algorithm_N2_base.cpp"`). The difference between these two functions is the so-called Markov order M denoting the number of time steps that the memory of the process can overlook. In other words, a process that has no memory reaching further back than the previous M time steps is defined to be of Markov order M . Now, `algoN2::calcTransitionMatrix()` models a process of $M = 1$ while `algoN2::calcTransitionMatrix2()` has a variable called `ORDER` that can take arbitrary integer values for M . Be aware that the memory needed for matrices of S states and Markov order M is of data complexity $O(S^{2M})$ which directly results

from the encoding of higher order matrices: Each higher order state is actually a sequence of states describing the history of the system. Therefore, the default (`algoN2::calcTransitionMatrix(...)`) is recommended which obeys the following design: A so-called statemap is computed for each reference atom group (core selection). This map assigns states to every surrounding atom group (surround selection) with respect to the reference atom group. All statemaps for the current frame are collected in the vector `currentStates`. Two state definitions exist, that differ in the way states are assigned according to the homogeneity of the core selection. In the case of all selected core atom groups having the same residue type (e.g. all TIP3 or all EMIM), Delaunay interaction with a core atom type is regarded a distinct state (`currentCalcGroup->OneAtomOneState`). If, on the other hand, the core selection contains multiple different residues (e.g. all residues of a protein) then interaction to a residue as a whole defines a state. Once the states have been assigned, the second part of the function compares the `currentStates` to the previous states (`currentCalcGroup->previousStates[mat]`) which are stored for each matrix in the calculation flag group.

algorithm/	<code>algorithm_N2_probability.h,</code> <code>algorithm_N2_probability.cpp</code>
------------	---------------------------------------------------------------------------------------

Promiscuity

The new concept of promiscuity $dL(t)/dt$ has not been published yet and was used for preliminary tests only. In infinite systems, $L(t)$ denotes the non-redundant number of pairs, cumulated over time and is given by the equation

$$L(t) = \sum_i \sum_j 1 - \prod_{t_k \leq t} (1 - n_{ij}(t_k)). \quad (7.1)$$

In other words, every time a new pair of directly interacting atoms - one that has not been encountered before - is registered, $L(t)$ is incremented by one. For infinite systems, the value $dL(t)/dt$ is the more or less constant slope of the function $L(t)$. However, as we are coping with finite systems of N particles, $L(t)$ is converging to $N*(N-1)$, thus the promiscuity is asymptotically 0. To overcome this problem, the actual implementation differs from the theoretical assumptions above. A former interaction partner is seen as a new contact if the delaunay edge between the two involved atoms was interrupted in the meanwhile. Thus, in the code two counters exist: Every existing pair at time t is inserted into the `std::set neibcount`. If the pair is disrupted (`delaunay_shells[MA][MB]==-1`) then all entries of the pair are removed from `neibcount`. The number of removed items is added to the `static_count`.

The implemented version of function $L(t)$ thus is the sum of `static_count` and `neibcount.size()` and increasing in an almost linear fashion. Hence, the running average of the slope of the “corrected” $L(t)$ converges.

Preliminary tests have shown that there are at least three factors that influence the promiscuity:

- viscosity η
- time step Δt
- equilibration state: The more equilibrated the simulation is, the more ‘viscous’ the system seems.

Calculation of the promiscuity is defined in `algoN2::calcPromiscuityCorrected()` in the file `"algorithm_N2_probability.cpp"`.

algorithm/	<code>algorithm_N2_probability.h,</code> <code>algorithm_N2_probability.cpp</code>
------------	---------------------------------------------------------------------------------------

Hydrogen-bonds

The functionality to compute hydrogen-bonds, developed by Michael Haberler, uses two criteria to identify such interactions: The first one is an angular criterium, which is based on the idea that hydrogen bonds have a linear geometry and deviate less than a certain angle from that linear arrangement. The second criterion is distance-based and implemented as a minimum and a maximum threshold. The implementation is to be seen as “experimental”. The crude layout is following a time series like concept yet functions are settled at a higher hierarchical level. This means, for example, the data management is separated from the rest of the time series. It is done by the function `FrameType::optimize_HbondTimeseries(...)`. The main function calculating the hydrogen-bond property is `algoN2::calcHbonds()` and can be found in the file `"algorithm_N2_neighbour_interaction.cpp"`.

algorithm/	<code>algorithm_N2_base.h,</code> <code>algorithm_N2_neighbour_interaction.h,</code> <code>algorithm_N2_neighbour_interaction.cpp</code>
------------	------------------------------------------------------------------------------------------------------------------------------------------------

Radial and 3D Density and Orientation

In order to exploit synergies, the traditional, radially resolved g-functions are implemented in the same function as 3D nuclear density maps. This central function is called `algoN2::calcgFunction()` and defined in "algorithm_N2_solvation_shell.cpp". Basically, it consists of a nested double loop over all core atom groups (**Block A**) and all surround atom groups (**Block B**). Within this loop, according to the requirements of Voronoi or radial shells, the **caHistogram** objects are filled by continuous incrementation at the appropriate position. The position is calculated from the minimum distance vector connecting the particular core (**Block A**) and surround atom groups (**Block B**). Thereby, the one dimensional position (`pos1D`) or bin in the histogram corresponds to the length of the distance vector, while the three dimensional position (`pos3Dx`, `pos3Dy`, `pos3Dz`) is calculated directly from the vector components. The local array `Increment[7]` in function `algoN2::calcgFunction()` is used to map histograms as defined in the ".gep" script file to the respective g-function. This works as follows: The class **calculationN2Histograms** in file "source/calculation/eff.h" defines an array, also called `Increment`, which gets its values for each task from `Task_gFunction::getCalculationValueIndex()` during optimization (`FrameType::optimize_N2(...)`). Thus, each object of type **calculationN2Histograms** "knows" which g-function to accumulate. In order to save computation time, the 3D nuclear density maps are calculated in the same subroutine. This allows reuse of the minimum distance vector. For normalization, the number of core-surround pairs is counted and stored in the header of **caHistogram** (see Sec.7.2.4). Normalization takes place in the postprocessing step (see Sec.7.2.5).

algorithm/	algorithm_N2_solvation_shell.h, algorithm_N2_solvation_shell.cpp
calculation/	caHistogram.h, caHistogram.cpp, eff.h

7.3.5 Collective Properties $O(1)$

The collective properties have mainly been implemented by Michael Haberler. In GEPETTO, collective properties are (weighted) sums or averages of single particle properties originating from $O(N)$ and $O(N^2)$ calculations. In functions, being prefixed `algoN1::rec...` and `algoN1::calc...` the summation is being performed over all core atom groups (`cGFlag->coreCollIdSize`). The difference between calculation and recording of observables lies in the resolution of results. `calc...` functions produce sums over the whole selection (i.e. actual collective properties), while `rec...` functions

directly store single particle properties, thus producing much more output.

Like it is the case in **algoN** and **algoN2**, all functions in **algo1** have the same signature - i.e. they take the same arguments and have the same return type - allowing to reference them by function pointers which optimises calculation. The most basic functions are `algo1::recCOM(...)` and `algo1::recCOG(...)`, responsible for logging the center-of-mass and center-of-geometry, respectively. Five electric dipole related functions exist at the moment: `algo1::recDIP(...)`, `algo1::recDIPCOG(...)`, `algo1::calcMD(...)`, `algo1::calcMJ(...)` and `algo1::calcJ(...)`. The first two can be used to record single particle dipoles, one being based on the center-of-mass, the other one on the center-of-geometry. The third function calculates the collective rotational dipole moment. Analogously to `algo1::calcMD(...)`, `algo1::calcMJ(...)` and `algo1::calcJ(...)` compute the collective translational dipole moment and the collective current. Two methods exist for angular velocity handling: `algo1::calcAngVel(...)` and `algo1::recAngVel(...)`. While `algo1::recAngVel(...)` records the angular velocities as calculated in `algoN::calcANGVEL(...)`, `algo1::calcAngVel(...)` stores the rotation matrix resulting from superposition (`algoN::calcOri(...)`). On the basis of this rotation matrix the angular velocity is calculated during postprocessing. They can be seen as alternatives. The velocity of the center-of-mass is recorded in `algo1::recVELCOM(...)`.

algorithm/	algorithm_1.h, algorithm_1.cpp
------------	---------------------------------------

8 GEPETTO - User Guide

Basically, a GEPETTO script file is organised in a declarative manner. This means the logic of computation is expressed without describing the control flow, unlike an imperative script that describes not only what to calculate but also how to do it. On one hand, this makes it easier for users who are not used to imperative programming to use GEPETTO. On the other hand, it is more easy for GEPETTO to do what it can do best: calculate fast. The following general remarks apply to all ".gef" scripts: All keywords are case insensitive and all whitespace characters are treated equally. This means, the actual file formatting can be chosen conveniently. However, each item has to be in the correct scope, a scope being a region within braces. The order of items within a scope is not important and can be chosen freely. Note that GEPETTO appends file extensions automatically corresponding to the task type. All three selections, CORE, SURROUND and NEIGHBOURMAP, default to CONTRACT COM and SEPARATE RESIDUE. The TESSELATION selection, however, defaults to CONTRACT NONE and SEPARATE ATOM. If no explicit NEIGHBOURMAP is given, GEPETTO will automatically generate a NEIGHBOURMAP specification according to: `SELECTION {TYPE NEIGHBOURMAP SEPARATE RESIDUE AG {SEGMENTTYPE *}}`. For a description of the selection process, please read Sec. 7.2.3. Comment lines start with the character "#" in GEPETTO script syntax.

All script examples in this chapter use a zinc-finger system simulated by Michael Haberler. It is a simulation of a zinc-finger polypeptide (pdb code 1ZNF) solvated in hydrated molecular ionic liquids. In total, it contains 2557 TIP3 molecules, 200 TRIF anions and 200 EVOT cations. In GEPETTO scripts, the system can be chosen as shown in the short script snippet below:

```
1 TRAJECTORY znf
2 {
3     FILELIST { FILETYPE DCD_CHARMM FILEPATH "5znf_evot_trif*.dcd" }
4     FILELIST { FILETYPE PSF FILEPATH "5znf_evot_trif.psf" }
5     FIRSTFRAME 10
6     GRAINING 5
7     MAXFRAMES 10000
8 }
```

A TRAJECTORY, being named znf, is declared on line 1 of this snippet and defined on lines 3-7. The first FILELIST item specifies the coordinate file, the second specifies the protein structure file (psf).

Generally, GEPETTO supports pathname expansion, also called globbing. This means, files can be specified in the same way as they are specified using the bash, including wildcard characters (like the asterisk on line 2). Three additional parameters can be chosen, that influence the statistics for calculations and, at the same time, specify time series dimensions. FIRSTFRAME can be used to set the time origin to a specific frame. The term frame denominates the stored coordinates of a single time step in a trajectory. In this example, GEPETTO starts all calculations at the 10th frame, relative to the first frame in the first file matching "5znf_evot_trif*.dcd". The GRAINING parameter specifies the interval of frames. Here, every fifth frame is considered, and all other frames are skipped. MAXFRAMES is the total number of frames under consideration. This means, GEPETTO takes the frames {10, 15, .. 50005, 50010}. In order to use this trajectory definition in a ".gep" script file it is sufficient to paste the whole code block into the file at any convenient position at the outermost scope. Most of the following examples assume that block at the beginning of the respective code snippets.

8.1 Single Particle Properties

This section introduces the three task types TIMESERIES, MSD and CORRELATION on the basis of four single particle observables. The first one is the center-of-mass (COM), the second one is the electric dipole (DIP). Both can be calculated from coordinate files (".dcd"). The other two observables, velocity-of-center-of-mass (VELCOM) and angular velocity (ANGVEL), need velocity files (".vel") in addition to the coordinate files.

8.1.1 Example I: Mean Square Displacement

The most basic observable is the center-of-mass. Often a mean square displacement function (MSD) thereof is used in order to describe general dynamic features like viscosity. The following script shows how to compute a MSD:

```

1 TASK
2 {
3     TYPE MSD OBSERVABLE COM
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE AG {RESIDUETYPE EVOT}}
7     SELECTION {TYPE SURROUND AG {SEGMENTTYPE NONE}}
8
9     REPORT {FILEPATH "evot"}
10 }
```

Line 3 of this script specifies the task type MSD and the observable COM. The following line defines the trajectory which has been predefined (see above) and tagged znf. The CORE selection chooses all

EVOT residues. By default, they are separated by residue and contracted to their center-of-mass. The SURROUND selection is a dummy in this case and not used by GEPETTO. Nevertheless it is included for formal reasons. The resulting file is called "evot.MSD_COM" and contains two columns. The first one is the time in picoseconds, the second one is the observed MSD in Å². Generally, mean square displacements can be calculated for any type of observable. However, not all combinations have been tested up to now.

8.1.2 Example II: Atom Coordinates and Unfolding

This simple example for a time series is also based on the observable COM:

```

1 TASK
2 {
3   TYPE TIMESERIES OBSERVABLE COM
4   TRAJECTORY znf
5
6   SELECTION {TYPE CORE AG {RESIDUETYPE TIP3 RESIDUENUMBER 1173}}
7   SELECTION {TYPE SURROUND AG {RESIDUETYPE TIP3 RESIDUENUMBER 1173}}
8
9   REPORT {FILEPATH "water1173"}
10 }
```

It writes the time in ps, and the three cartesian coordinates (Å) of the center-of-mass of water 1173 into the file "water1173.COM". Thereby it produces one line for each selected time step. The chosen water molecule in this trajectory shows multiple toroidal shifts over the first 1000 frames. To overcome this, the trajectory can be unfolded using the keyword UNFOLD. This keyword could be added directly in the definition of TRAJECTORY znf outside of the TASK scope. Alternatively, it can be added inside the TASK scope as shown below:

```

1 TASK
2 {
3   TYPE TIMESERIES OBSERVABLE COM
4   TRAJECTORY znf {UNFOLD}
5
6   SELECTION {TYPE CORE AG {RESIDUETYPE TIP3 RESIDUENUMBER 1173}}
7   SELECTION {TYPE SURROUND AG {RESIDUETYPE TIP3 RESIDUENUMBER 1173}}
8
9   REPORT {FILEPATH "water1173"}
10 }
```

Generally, GEPETTO allows to add keywords to already defined items. However, it is recommended to use this feature with care.

8.1.3 Example III: Dipole Autocorrelation

The following script computes the autocorrelation of the electric dipole of the first triflate (TRIF) in the protein structure file (".psf"):

```

1 TASK
```

```

2 {
3   TYPE CORRELATION OBSERVABLE DIP, DIP
4   TRAJECTORY znf
5
6   SELECTION {TYPE CORE          AG {SEGMENTTYPE TRIF RESIDUENUMBER 1}}
7   SELECTION {TYPE SURROUND      AG {SEGMENTTYPE TRIF RESIDUENUMBER 1}}
8
9   REPORT {FILEPATH "trif1"}
10 }

```

The first column in the resulting file "trif1.DIP_DIP" contains the time in ps, the second one is the dipole autocorrelation in $(e\text{\AA})^2$.

8.1.4 Example IV: Velocity Autocorrelation Functions

Velocity auto correlation functions (VACF) can easily be obtained:

```

1 TRAJECTORY znf
2 {
3   FILELIST { FILETYPE DCD_CHARMM FILEPATH "5znf_evot_trif_200_2557_1800.vel" }
4   FILELIST { FILETYPE PSF FILEPATH "5znf_evot_trif_200_2557.psf" }
5   FIRSTFRAME 1 GRAINING 1 MAXFRAMES 1000
6 }
7
8 TASK
9 {
10  TYPE CORRELATION PROP VELCOM,VELCOM
11  TRAJECTORY znf
12
13  SELECTION { TYPE CORE          SEPARATE RESIDUE AG {RESIDUETYPE TIP3} }
14  SELECTION { TYPE SURROUND SEPARATE RESIDUE AG {RESIDUETYPE TIP3} }
15
16  REPORT { FILEPATH "tip3" }
17 }

```

Here, the whole script is included because the TRAJECTORY differs from the other examples. In the case of the observable velocity-of-center-of-mass (VELCOM) a ".vel" trajectory containing velocities instead of coordinates has to be loaded (line 3 in the script). As both selections select all water molecules, the resulting VACF is averaged over all water molecules in the simulation box. The REPORT file "tip3.VELCOM_VELCOM" has two columns: Time in ps and VACF in \AA ps^{-1} .

8.1.5 Example V: Angular Velocity

If both, coordinates and velocity files are needed, as it is the case with the ANGVEL property, they have to be listed in arbitrary order in the TRAJECTORY scope:

```

1 TRAJECTORY znf
2 {
3   FILELIST { FILETYPE DCD_CHARMM FILEPATH "5znf_evot_trif_200_2557_1800.dcd" }
4   FILELIST { FILETYPE DCD_CHARMM FILEPATH "5znf_evot_trif_200_2557_1800.vel" }
5   FILELIST { FILETYPE PSF FILEPATH "5znf_evot_trif_200_2557.psf" }
6   FIRSTFRAME 1 GRAINING 5 MAXFRAMES 100
7 }
8
9 TASK

```

```

10 {
11     TYPE TIMESERIES OBSERVABLE ANGVEL
12     TRAJECTORY znf
13
14     SELECTION {TYPE CORE SEPARATE NONE AG {SEGMENTTYPE 5ZNF}
15               ORIENT {FILEPATH "5znf_ca_coororiemass.pdb"}}
16     SELECTION {TYPE SURROUND AG {SEGMENTTYPE TIP3}}
17
18     REPORT {FILEPATH "znf "}
19 }

```

The observable ANGVEL needs the ORIENT clause of the CORE selection to define a ".pdb" file containing a set of comparison coordinates. The leftmost column in the output file "znf.ANGVEL" is the time in ps, the other three columns describe the rotation as the rotation axis in rad/ps. Thereby, the sign of these values indicates the rotation direction.

8.2 Pair Correlation Functions

Currently, GEPETTO can calculate seven different types of g-functions: The radial distribution function $g^{000}(r)$, the three orientational correlation functions $g^{110}(r)$, $g^{011}(r)$ and $g^{101}(r)$ and their second legendre polynomes $g^{220}(r)$, $g^{022}(r)$ and $g^{202}(r)$. Thereby, the minimum image convention is satisfied, which leads to curves that decay at distances corresponding to the borders of the box. All g-functions require at least the two selections CORE and SURROUND as well as a HISTOGRAM specification.

8.2.1 Example VI: Simple g-Functions

A GEPETTO script file to calculate the functions $g^{000}(r)$, $g^{110}(r)$ and the second legendre polynome $g^{220}(r)$ for all water molecules in our znf trajectory around all centers-of-mass of all LYS residues looks like this:

```

1 TASK
2 {
3     TYPE G000, G110, G220
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE AG {RESIDUETYPE LYS}}
7     SELECTION {TYPE SURROUND AG {RESIDUETYPE TIP3}}
8
9     HISTOGRAM {BINWIDTH 0.5 BINCOUNT 40 MAXVALUE 20 MINVALUE 0}
10    REPORT {FILEPATH "water_around_lys"}
11 }

```

The TYPE property of the TASK item above is a list of all required g-functions. Line 2 defines which trajectory to use. The CORE selection specifies all LYS residues while the SURROUND selection matches all water molecules in trajectory znf. Distribution functions generally require that a HISTOGRAM

be specified (line 9). Four parameters need to be given: A BINWIDTH defining Δr in the case of g-functions, a BINCOUNT defining the total number of bins as well as a MAXVALUE and a MINVALUE defining the upper and lower bound for values on the abscissa. In this example, the upper and lower bounds are given in Å. Three files are being written: "water_around_lys.g000", "water_around_lys.g110" and "water_around_lys.g220".

8.2.2 Example VII: Voronoi Decomposition of g-Functions

The g-function script can be easily extended to calculate the RDF and $g^{101}(r)$ of the two innermost Voronoi shells by adding a TESSELATION selection (line 8) and the two shell specifications (lines 10 and 11). As the recursive shell definition process is quite time consuming, it is recommended to be very selective in terms of the shells specified in the script file.

```

1 TASK
2 {
3     TYPE G000, G101
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE AG {RESIDUETYPE LYS}}
7     SELECTION {TYPE SURROUND AG {RESIDUETYPE TIP3}}
8     SELECTION {TYPE TESSELATION AG {SEGMENTTYPE *}}
9
10    SHELL {TYPE DELAUNAY MINVALUE 1 MAXVALUE 1}
11    SHELL {TYPE DELAUNAY MINVALUE 2 MAXVALUE 2}
12
13    HISTOGRAM {BINWIDTH 0.5 BINCOUNT 40 MAXVALUE 20 MINVALUE 0}
14    REPORT {FILEPATH "water_around_lys_voro12"}
15 }
```

This calculation will yield the two files "water_around_lys_voro12.g000" and "water_around_lys_voro12.g101", both having three columns: The radial distance r , the g-function for the first shell and the g-function for the second shell. An additional specification of SHELL {TYPE NOSHELL} leads to a third column containing the total RDF. The order of columns in the REPORT file corresponds to the order of SHELL clauses in the script. Generally, this decomposition is additive, which means that summing up all shell specific columns should yield the total g-function. While the execution of the simple g-function example for 10000 frames should take about 15 seconds on one of the currently available machines, the Voronoi TESSELATION and decomposition for two shells takes more than three hours.

8.3 Nuclear Density Maps

The three dimensional density maps need an orientation file to be specified for the CORE selection. GEPETTO superimposes every atom group in the CORE selection on the structure given in the ori-

entation file. Like g-functions, a HISTOGRAM clause has to be defined but unlike the one dimensional histograms for g-functions or other distributions, the three dimensional histogram needs only the two parameters BINWIDTH and MAXVALUE. The first parameter describes the voxel size, the second is the radius of a sphere which is centered at the coordinate origin and defining the border of the density map. The following two examples use a ".pdb" file for orientation that specifies reference positions for all C_α atoms:

1	ATOM	5	CA	LYS	1	-12.076	-4.963	1.716	1.00	0.00	5ZNF
2	ATOM	27	CA	THR	2	-11.305	-1.390	0.524	1.00	0.00	5ZNF
3	ATOM	41	CA	TYR	3	-8.034	-0.094	1.765	1.00	0.00	5ZNF
4	ATOM	62	CA	GLN	4	-6.145	3.081	1.288	1.00	0.00	5ZNF
5	ATOM	79	CA	CYD	5	-2.866	4.689	2.085	1.00	0.00	5ZNF
6	ATOM	89	CA	GLN	6	-2.657	7.240	4.931	1.00	0.00	5ZNF
7	...										

Every line defines a matching pattern (currently only atom type, residue type and residue number are being used) as well as the x,y and z coordinates. The resulting maps are written in **xplor** electron density map format which can be displayed, for example, using **pymol** or **vmd**. For a visualization in **vmd**, an empty line has to be prepended. In addition to the **xplor** files, GEPETTO creates two files containing the functions $g^{000}(r)$ and $g^{101}(r)$ calculated directly from the density map. These files can be used for a mapping of 3D density “islands” to peaks in one dimensional g-functions.

8.3.1 Example VIII: Nuclear Density Map

The first 3D density example creates a map of centers-of-mass of water molecules oriented at the zinc-finger’s C_α -trace:

```

1 TASK
2 {
3     TYPE DENSITY3D
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE SEPARATE NONE AG {SEGMENTTYPE 5ZNF}
7         ORIENT {FILEPATH "5znf_ca_coororiemass.pdb"}}
8     SELECTION {TYPE SURROUND AG {SEGMENTTYPE TIP3}}
9
10    HISTOGRAM h1 {BINWIDTH 1.0 MAXVALUE 30}
11    REPORT {FILEPATH "water_around_5znf_voro12"}
12 }
```

The CORE selection has two notable features: First, separation has been set to NONE, which creates one single atom group for the whole 5ZNF segment. The second feature is the specification of the orientation file for the CORE selection.

8.3.2 Example IX: Multiple Nuclear Density Maps

Specifying multiple tasks in one single script file has at least two advantageous aspects. Code reuse leads to scripts which are structured more clearly and much easier to maintain. The second aspect has to do with optimised calculation. By calculating multiple tasks at once, the real power of GEPETTO as a fast trajectory analysis tool can be exploited. The aim of this example is to show both aspects. It consists of three tasks that calculate different density maps using the same TESSELATION and NEIGHBOURMAP selections. Analogously to the TRAJECTORY object `znf` used throughout this section, other objects can be tagged and reused too:

```

1 SELECTION co {TYPE CORE SEPARATE NONE AG {SEGMENTTYPE 5ZNF}
2   ORIENT {FILEPATH "5znf_ca_coororiemass.pdb"}}
3 SELECTION t {TYPE TESSELATION AG {SEGMENTTYPE *}}
4 SELECTION n1 {TYPE NEIGHBOURMAP SEPARATE SEGMENT AG {SEGMENTTYPE 5ZNF}}
5 SELECTION n2 {TYPE NEIGHBOURMAP SEPARATE RESIDUE AG {SEGMENTTYPE EVOT,TIP3}}
6
7 SHELL shn {TYPE NOSHELL}
8 SHELL sh1 {TYPE DELAUNAY MINVALUE 1 MAXVALUE 1}
9 SHELL sh2 {TYPE DELAUNAY MINVALUE 2 MAXVALUE 2}
10
11 HISTOGRAM h1 {BINWIDTH 1.0 MAXVALUE 30}

```

This listing defines all selections that our three tasks have in common, excluding the SURROUND selection which differs between the three tasks. By assigning a name to each selection they can be addressed by the respective tasks. The SHELL definitions and the HISTOGRAM specification are being defined in this script header as well. The usage of these predefined objects is quite simple:

```

1 TASK
2 {
3   TYPE DENSITY3D
4   TRAJECTORY znf
5
6   SELECTION co SELECTION t SELECTION n1 SELECTION n2
7   SELECTION {TYPE SURROUND AG {SEGMENTTYPE TIP3}}
8
9   SHELL shn SHELL sh1 SHELL sh2
10
11   HISTOGRAM h1
12   REPORT {FILEPATH "water_around_5znf_voro12"}
13 }
14
15 TASK
16 {
17   TYPE DENSITY3D
18   TRAJECTORY znf
19
20   SELECTION co SELECTION t SELECTION n1 SELECTION n2
21   SELECTION {TYPE SURROUND AG {SEGMENTTYPE EVOT}}
22
23   SHELL shn SHELL sh1 SHELL sh2
24
25   HISTOGRAM h1
26   REPORT {FILEPATH "evot_around_5znf_voro12"}
27 }
28
29 TASK

```

```

30 {
31     TYPE DENSITY3D
32     TRAJECTORY znf
33
34     SELECTION co SELECTION t SELECTION n1 SELECTION n2
35     SELECTION {TYPE SURROUND AG {SEGMENTTYPE TRIF}}
36
37     SHELL shn SHELL sh1 SHELL sh2
38
39     HISTOGRAM h1
40     REPORT {FILEPATH "trif_around_5znf_voro12"}
41 }

```

Each of these three tasks results in three xplor files (one for each SHELL) and two additional files for 3D - 1D mapping. All filenames include an index starting at 0 indicating the SHELL. Again, the order of this index corresponds to the order of SHELL objects in the script. This means, if SHELL 3 is specified before SHELL 2 and SHELL 1, it will be assigned index 0, while SHELL 2 gets index 1 and SHELL 1 is marked by the index 2.

8.3.3 Example X: Nuclear Density Map and Ensemble Average

So far we have calculated the density maps merely for one atom group (in this case the whole zinc-finger) using time averaging. If the ensemble average is required, the ORIENT file has to be adapted. Instead of specifying residues explicitly by listing the residue number in the ORIENT file, ensemble averaging can be achieved by setting all residue numbers to a value of -1 which causes the residue number to be ignored when assigning fixed positions. This example creates the density map of all water hydrogens around all water molecules and uses the following orientation file containing reference positions for all water atoms:

1	ATOM	1	H1	TIP3	-1	0.000	0.780	0.190	1.00	0.00	TIP3
2	ATOM	2	OH2	TIP3	-1	0.000	0.000	-0.380	1.00	0.00	TIP3
3	ATOM	3	H2	TIP3	-1	0.000	-0.780	0.190	1.00	0.00	TIP3

The script file looks like this:

```

1 TASK
2 {
3     TYPE DENSITY3D
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE AG {RESIDUETYPE TIP3}
7         ORIENT {FILEPATH "tip3.pdb"}}
8     SELECTION {TYPE SURROUND AG {RESIDUETYPE TIP3 ATOMTYPE H1, H2}}
9
10    HISTOGRAM h1 {BINWIDTH 1.0 MAXVALUE 30}
11    REPORT {FILEPATH "hydrogen_around_water_voro12"}
12 }

```

Each water molecule is used as a reference point (CORE) while the position of all surrounding water hydrogens is being registered in the 3D matrix (SURROUND). Three files are being created: "hydrogen_

around_water_voro12.0.xplor", "hydrogen_around_water_voro12.0.g0003d" and "hydrogen_around_water_voro12.0.g1013d". As already mentioned, the idea behind the latter two files is a possible mapping between the density maps and the one dimensional g-functions. If a hypothetical nuclear density map shows three distinct regions of high density “islands”, depending on their topology they lead to overlapping peaks in the corresponding one dimensional RDF or OCF. Setting the density of one of these regions to a value of 0 - which can only be done within the GEPETTO source code at the moment - should make the corresponding peak in the g-functions disappear which can provide additional interpretation and visualization possibilities.

8.4 Coordination and Neighbour Interaction

The built-in Delaunay TESSELATION allows some basic neighbour interaction-related features to be calculated. Thereby, the most basic information is reflected by the coordination number CN accounting for the number of immediate neighbours or interaction partners (SURROUND) of a reference atom group (CORE). GEPETTO provides two different OBSERVABLEs directly related to the coordination number that differ in the granularity of results: CN results in a coordination number that is averaged over all CORE atom groups of one species and all SURROUND atom groups of one species. Thus, if a script specifies three species for the CORE selection and the same three species for the SURROUND selection, $3 \times 3 = 9$ species-species specific CN values are calculated per frame. The second OBSERVABLE is called NSHELL and gives $n(t)$. It is mainly used for the $\langle n(0)n(t) \rangle$ autocorrelation describing neighbourhood dynamics. A contact matrix can be created by listing all CN values in a $N \times M$ matrix, N being the number of CORE atom groups and M the number of SURROUND atom groups. If the time dimension is included into this consideration, a transition matrix can be constructed, which consists of transition rates between distinct states. The concept of promiscuity has been implemented but not studied to a great extent up to now. However, it could be used as a Voronoi related alternative to atom position mean square displacement and seems to be correlated to the system’s viscosity. The last example in this section uses angle- and distance-based criteria for identification of hydrogen-bonds. This section shows another example for code reuse: Some of the following examples share specifications for SHELL and SELECTION. Thus, it is useful to define these items once at the global scope and use abbreviations for them within the respective TASK scope. Here are the global definitions:

```
1 SHELL sh1 {TYPE DELAUNAY MINVALUE 1 MAXVALUE 1}
2
3 SELECTION st {TYPE TESSELATION AG {SEGMENTTYPE *}}
4
```



```

5 SELECTION n1 {TYPE NEIGHBOURMAP SEPARATE NONE AG {SEGMENTTYPE 5ZNF}}
6 SELECTION n2 {TYPE NEIGHBOURMAP AG {RESIDUETYPE EVOT}}
7 SELECTION n3 {TYPE NEIGHBOURMAP AG {RESIDUETYPE TRIF}}
8 SELECTION n4 {TYPE NEIGHBOURMAP AG {RESIDUETYPE TIP3}}

```

8.4.1 Example XI: Coordination Number

A distribution of coordination numbers - that is the total number of direct Delaunay neighbours - for the zinc-finger molecule over multiple frames can be calculated as follows:

```

1 TASK
2 {
3   TYPE DISTRIBUTION OBSERVABLE CN
4   TRAJECTORY znf
5
6   HISTOGRAM {MINVALUE 240 MAXVALUE 280 BINWIDTH 1 BINCOUNT 40}
7
8   SELECTION {TYPE CORE SEPARATE NONE AG {SEGMENTTYPE 5ZNF}}
9   SELECTION {TYPE SURROUND AG {SEGMENTTYPE EVOT, TRIF, TIP3}}
10
11  SELECTION {TYPE NEIGHBOURMAP SEPARATE NONE AG {SEGMENTTYPE 5ZNF}}
12  SELECTION {TYPE NEIGHBOURMAP SEPARATE RESIDUE AG {RESIDUETYPE EVOT, TRIF, TIP3}}
13
14  SELECTION st
15  SHELL sh1
16  REPORT {FILEPATH "all_around_znf"}
17 }

```

The example above uses three predefined script items: The `TRAJECTORY znf` at line 4, the `SELECTION st` (line 14) and the shell definition `SHELL sh1` at line 15. Again, in order to run this script, these definitions have to be prepended to the script file. The `CORE` selection at line 8 shows how to select a protein as one single entity or atom group. The `AG` clause matches all atoms having segmenttype 5ZNF and the specification of `SEPARATE NONE` “tells” GEPETTO to handle these atoms as one whole entity. The `NEIGHBOURMAP` is split into the two distinct species zinc-finger (line 11) and all the other compounds in the box (line 12). This, in combination with the `CORE` and `SURROUND` selections automatically leads to one `CORE` species (zinc-finger) and one `SURROUND` species (all others). Like any typical distribution result file, “all_around_znf.CN” has two columns: The first one contains the observable which is CN in this case. The second one is the probability of occurrence.

8.4.2 Example XII: Species Specific Coordination Number

The following example calculates a time series of two residuetype specific coordination numbers, each being averaged over five reference molecules (`CORE` selection). In other words, for every selected time step and every species one averaged value is computed from an ensemble of five molecules:

```

1 TASK
2 {
3   TYPE TIMESERIES OBSERVABLE CN

```

```

4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE AG {RESIDUETYPE EVOT RESIDUENUMBER 1, 2, 3, 4, 5}}
7
8     SELECTION {TYPE SURROUND AG {RESIDUETYPE TRIF, TIP3}}
9
10    SELECTION st SELECTION n1 SELECTION n2 SELECTION n3 SELECTION n4
11    SHELL sh1
12    REPORT {FILEPATH "all_around_evot12345"}
13 }

```

Three major differences to the previous script are to be noted: First, the TASK defines a TIMESERIES instead of a DISTRIBUTION. Second, this CORE selection contains more than one atom group. This might not be immediately evident, but follows from the separation defaults which are SEPARATE RESIDUE for the CORE selection, SURROUND selection and NEIGHBOURMAP. The third difference is the species-splitting into four single NEIGHBOURMAP selections (n1, n2, n3 and n4). Although the zinc-finger is not being used by either the CORE or the SURROUND selection it has to be mapped SELECTION n1 because these atoms are specified in the TESSELATION selection (line 11) and the NEIGHBOURMAP must contain all atoms that are selected by any other selection. Execution of the script results in a file ("all_around_evot12345.CN") having three columns. The first one is the time in ps, the other two columns list the species specific coordination numbers for the two selected species TRIF and TIP3 as defined by the NEIGHBOURMAP selections around the five EVOT molecules. The order of columns in the result file is the order of neighbour mappings in the script file.

8.4.3 Example XIII: Residue Resolved Coordination Number

In order to get one single coordination number for each of the 31 residues of 5ZNF (30 amino acids plus one zinc ion), each residue has to be defined as a single species. This means, one distinct NEIGHBOURMAP item has to be specified for each single residue:

```

1 TASK calcCNAG
2 {
3     TYPE TIMESERIES OBSERVABLE CN
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE AG {SEGMENTTYPE 5ZNF}}
7     SELECTION {TYPE SURROUND AG {RESIDUETYPE EVOT, TRIF, TIP3}}
8     SELECTION {TYPE NEIGHBOURMAP AG {RESIDUETYPE EVOT, TRIF, TIP3}}
9
10    SELECTION {TYPE NEIGHBOURMAP AG {SEGMENTTYPE 5ZNF RESIDUENUMBER 1}}
11    SELECTION {TYPE NEIGHBOURMAP AG {SEGMENTTYPE 5ZNF RESIDUENUMBER 2}}
12    ...
13    SELECTION {TYPE NEIGHBOURMAP AG {SEGMENTTYPE 5ZNF RESIDUENUMBER 30}}
14    SELECTION {TYPE NEIGHBOURMAP AG {SEGMENTTYPE 5ZNF RESIDUENUMBER 31}}
15
16    SELECTION st
17    SHELL sh1
18    REPORT {FILEPATH "residue_specific"}
19 }

```

Again, the order of columns in the result file "residue_specific.CN" corresponds to the order of NEIGHBOURMAP entries in the script file. Up to now, only one CORE species has been specified in the coordination number examples. Of course, the specification of multiple CORE (N) and SURROUND species (M) is possible. Thus, for every frame in the time selection N times M values are calculated and listed in the output file. The results of such a calculation can be seen as a serialised matrix.

8.4.4 Example XIV: Mean Residence Time

Neighbourhood dynamics can be described by the mean residence time (MRT) which, in turn, can be obtained by fitting the autocorrelation function $\langle n(0)n(t) \rangle$ to a Kohlrausch-Williams-Watts function. The observable corresponding to the function $n(0)$ is called NSHELL in GEPETTO scripts. The following example computes the NSHELL autocorrelation of water surrounding the zinc-finger.

```

1 TASK calcMRT
2 {
3     TYPE CORRELATION OBSERVABLE NSHELL, NSHELL
4     TRAJECTORY znf
5     SELECTION {TYPE CORE SEPARATE NONE AG {SEGMENTTYPE 5ZNF}}
6     SELECTION {TYPE SURROUND AG {RESIDUETYPE TIP3}}
7
8     SELECTION st SELECTION n1 SELECTION n2 SELECTION n3 SELECTION n4
9     SHELL sh1
10    REPORT {FILEPATH "water_residence"}
11 }
```

The output file "coordination.NSHELL_NSHELL" contains two columns. The first one lists the time t in ps and the second one is the CN times the probability of still being a neighbour after time t . Furthermore, this example shows that tasks, like any script object, can be assigned names.

8.4.5 Example XV: Contact Matrix

A contact matrix between different atom groups can be calculated as follows:

```

1 TASK
2 {
3     TYPE DISTRIBUTION OBSERVABLE CONTACTMATRIX
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE AG {SEGMENTTYPE 5ZNF}}
7     SELECTION {TYPE SURROUND AG {SEGMENTTYPE 5ZNF}}
8     SELECTION {TYPE TESSELATION AG{SEGMENTTYPE *}}
9
10    HISTOGRAM dummy {BINWIDTH 1 BINCOUNT 1 MINVALUE 1 MAXVALUE 1}
11
12    SHELL {TYPE DELAUNAY MINVALUE 1 MAXVALUE 1}
13    REPORT {FILEPATH "dummy"}
14 }
```

At the moment, this is work in progress and thus to be seen as “experimental”. This example does not explicitly declare a NEIGHBOURMAP selection which leads to automatic generation of a neighbour

mapping with AG {SEGMENTTYPE *} and SEPARATE RESIDUE. A dummy HISTOGRAM and a dummy REPORT have to be given. The resulting matrix is written to `std::cerr` which can be redirected into a file by using the `2>` operator in bash. It is structured as a $N \times M$ matrix with N being the number of CORE atom groups and M being the number of SURROUND atom groups. In our example, CORE and SURROUND are identical, which leads to diagonal values of 1. All matrix values are normalised. This means, the matrix element with indices n and m reflects the probability of CORE atom group n being an immediate neighbour of SURROUND atom group m .

8.4.6 Example XVI: Markovian Transition Matrix

If a CORE selection is homogenous with respect to the selected residue types, single atoms are interpreted as distinct states. Else, each residue is seen as a distinct state. The following example uses a set of all 30 residues of the zinc-finger plus the zinc ion as the - heterogeneous - CORE selection leading to $30+1+1 = 32$ states. The additional state is the so-called “out state”. Running the following script results in the file `"znf_water.TRANSITIONMATRIX"`:

```

1 TASK
2 {
3     TYPE TRANSITIONMATRIX
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE AG {SEGMENTTYPE 5ZNF}}
7     SELECTION {TYPE SURROUND AG {SEGMENTTYPE TIP3}}
8     SELECTION {TYPE TESSELATION AG{SEGMENTTYPE *}}
9
10    SHELL {TYPE DELAUNAY MINVALUE 1 MAXVALUE 1}
11    REPORT {FILEPATH "znf_water"}
12 }
```

According to the 32 states, `"znf_water.TRANSITIONMATRIX"` contains 32 rows and 32 columns. Values describing the aforementioned “out state” are listed in the rightmost column and the bottommost row, respectively. Note, that the resulting matrix is not row- or column- normalised. It can be seen as the frequency matrix \mathbf{F} (see Sec.3). The element with indices i and j indicates how often a transition from state i in frame t to state j in frame $t+\Delta t$ was observed. In this example, the element (1,1) has the value 12.8081. This means that about 13 water molecules are residing at the znf residue 1 at time t and are still there at time $t+\Delta t$. The element (1,2) has the value 0.181818 which means that on average 0.2 water molecules were residing at residue 1 at time t , but changed over to residue 2 after a time of Δt . Row normalization and subtraction of the identity matrix yields the matrix \mathbf{V} which can be used to describe dynamics by a Markovian master equation (Sec.3). As the time difference Δt indicates, each transition matrix has an intrinsic time step which is set by the parameter `GRAINING` in the `TRAJECTORY` scope of the script.

8.4.7 Example XVII: Promiscuity

This example shows how to compute the promiscuity of a system. This viscosity related quantity depends on the chosen time step (GRAINING). An example script shall be given here for sake of completeness:

```

1 TASK
2 {
3     TYPE TIMESERIES OBSERVABLE PROMISCUITY
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE          AG {RESIDUETYPE *}}
7     SELECTION {TYPE SURROUND      AG {RESIDUETYPE *}}
8     SELECTION {TYPE TESSELATION   AG {RESIDUETYPE *}}
9     SELECTION {TYPE NEIGHBOURMAP  AG {RESIDUETYPE *}}
10
11     SHELL {TYPE DELAUNAY MINVALUE 1 MAXVALUE 1}
12
13     REPORT {FILEPATH "all_residues"}
14 }
```

The output file is called "all_residues.pms" and contains five columns. The first one is the time in picoseconds starting at 0. The second column contains the cumulative number of contacts $L(t)$ that an atom group “made” so far. Thus, the quantity $L(t)$ is steadily increasing. The promiscuity $dL(t)/dt$ is given in the third and a running average thereof in the fourth column. For more details on promiscuity see Sec. 7.3.4.

8.5 Voronoi Surfaces and Volumes

Voronoi decomposition lends itself to the calculation of the geometric properties volume and surface. Currently, GEPETTO implements time series and distribution of both quantities.

8.5.1 Example XVIII: Voronoi Volume Time Series of Proteins

The volume of a protein is the sum over all single atomic volumes. A GEPETTO script calculating a time series of this quantity is given below.

```

1 TASK
2 {
3     TYPE TIMESERIES OBSERVABLE VOLUME
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE SEPARATE NONE AG {SEGMENTTYPE 5ZNF}}
7     SELECTION {TYPE SURROUND AG {SEGMENTTYPE EVOT, TRIF, TIP3}}
8
9     SELECTION {TYPE TESSELATION AG {SEGMENTTYPE *}}
10
11     SELECTION {TYPE NEIGHBOURMAP SEPARATE NONE AG {SEGMENTTYPE 5ZNF}}
12     SELECTION {TYPE NEIGHBOURMAP AG {SEGMENTTYPE EVOT, TRIF, TIP3}}
13
14     SHELL {TYPE DELAUNAY MINVALUE 0 MAXVALUE 0}
```

```

15
16     REPORT {FILEPATH "znf"}
17 }

```

The MINVALUE and MAXVALUE parameter defines the so-called Delaunay distance which is the length of the shortest path between two points in a Delaunay tessellation. Thus, a value of 0 corresponds to the reference point or CORE selection atom group itself. A value of 1 would lead to calculation of the first solvation shell's volume. Explicit specification of two NEIGHBOURMAP selections is imperative in this case. This comes from the fact that the default NEIGHBOURMAP is residue grained. Thus, the protein is not interpreted as a monolithic entity unless explicitly defined by specification of SEPARATE NONE. Again, the resulting file "znf.VOLUME" contains two columns where the first one is the time in ps and the second one the volume of the zinc-finger in Å³.

8.5.2 Example XIX: Voronoi Volume Distribution of Solvation Shell

The script above can easily be adapted to calculate the volume of the second voronoi shell and store it into a histogram:

```

1 TASK
2 {
3     TYPE DISTRIBUTION OBSERVABLE VOL
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE SEPARATE NONE AG {SEGMENTTYPE 5ZNF}}
7     SELECTION {TYPE SURROUND AG {SEGMENTTYPE EVOT, TRIF, TIP3}}
8
9     SELECTION {TYPE TESSELATION AG {SEGMENTTYPE *}}
10
11    SELECTION {TYPE NEIGHBOURMAP SEPARATE NONE AG {SEGMENTTYPE 5ZNF}}
12    SELECTION {TYPE NEIGHBOURMAP AG {SEGMENTTYPE EVOT, TRIF, TIP3}}
13
14    HISTOGRAM {BINWIDTH 10 BINCOUNT 500 MINVALUE 25000 MAXVALUE 30000}
15
16    SHELL {TYPE DELAUNAY MINVALUE 2 MAXVALUE 2}
17
18    REPORT {FILEPATH "znf"}
19 }

```

Three changes are needed: First, in line 3, TIMESERIES changed to DISTRIBUTION. This brings about the second change in line 14 where a HISTOGRAM has been specified. Caution! If the borders (MINVALUE and MAXVALUE) are not chosen right, the resulting distribution will show zero values or nan. Finally, SHELL 2 is specified at line 16.

8.5.3 Example XX: Residue Specific VISA

In order to produce a Voronoi interface surface area (VISA) time series, the following script can be used:

```

1 TASK
2 {
3     TYPE TIMESERIES OBSERVABLE SURFACE
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE AG {RESIDUETYPE LYS RESIDUENUMBER 1}}
7     SELECTION {TYPE SURROUND AG {SEGMENTTYPE TIP3}}
8     SELECTION {TYPE TESSELATION AG {SEGMENTTYPE *}}
9
10    SHELL {TYPE DELAUNAY MINVALUE 0 MAXVALUE 0}
11
12    REPORT {FILEPATH "znf"}
13 }

```

It calculates a time series of the Voronoi interface surface area (SURFACE) between the terminal LYS 1 residue and water molecules. The neighbourmapping is generated automatically according to the defaults (see other examples).

8.6 Collective Properties

GEPETTO knows three electric collective observables, that are used in the context of dielectric theory: The collective rotational dipole moment MD, the collective translational dipolemoment MJ and the current J. Additionally, two special OBSERVABLEs have been implemented: The first one is called MDCAGE (see Example XXIV) and allows for the Voronoi shell specific calculation of MD. The second one is named DECOMPOSEDDIP (see Exmaple XXV) and calculates the molecular dipolemoments μ_i of the SURROUND selection with respect to the center-of-mass of the CORE selection and the projection of μ_i to the total dipolemoment M_{tot} .

8.6.1 Example XXI: MD, MJ, J Time Series

Another example of combined computation shall be given here:

```

1 #vel is needed for J
2 #UNFOLD is needed for MJ
3 TRAJECTORY znf
4 {
5     FILELIST { FILETYPE DCD_CHARMM FILEPATH "5znf_evot_trif_200_2557_1800.dcd"}
6     FILELIST { FILETYPE DCD_CHARMM FILEPATH "5znf_evot_trif_200_2557_1800.vel"}
7     FILELIST { FILETYPE PSF FILEPATH "5znf_evot_trif_200_2557.psf"}
8     FIRSTFRAME 1 GRAINING 50 MAXFRAMES 1000
9     UNFOLD
10 }
11
12 #global definition of selections
13 SELECTION cs {TYPE CORE AG {SEGMENTTYPE EVOT}}
14 SELECTION ss {TYPE SURROUND AG {SEGMENTTYPE NONE}}
15
16 TASK
17 {
18     TYPE TIMESERIES OBSERVABLE MD
19     TRAJECTORY znf

```

```

20     SELECTION cs SELECTION ss
21     REPORT {FILEPATH "evot"}
22 }
23
24 TASK
25 {
26     TYPE TIMESERIES OBSERVABLE MJ
27     TRAJECTORY znf
28     SELECTION cs SELECTION ss
29     REPORT {FILEPATH "evot"}
30 }
31
32 TASK
33 {
34     TYPE TIMESERIES OBSERVABLE J
35     TRAJECTORY znf
36     SELECTION cs SELECTION ss
37     REPORT {FILEPATH "evot"}

```

Three files are written on completion of calculation. Each file contains four columns, the first being the time in ps and the other three the x, y, and z component of the particular observable. In the case of MD and MJ, these components are given in $e\text{\AA}$, while in the case of J the unit is $e\text{\AA}ps^{-1}$. As can be seen from the comments, J calculation needs velocity trajectories, and MJ needs the coordinates to be unfolded.

8.6.2 Example XXII: $\langle MD(0)MD(t) \rangle$ Autocorrelation LTC

Analogously to the single particle properties, the task types TIMESERIES, MSD and CORRELATION can be applied to collective observables. An example for the autocorrelation of the collective rotational dipole moment looks like this:

```

1 TASK
2 {
3     TYPE CORRELATION OBSERVABLE MD,MD LTC
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE AG {SEGMENTTYPE EVOT}}
7     SELECTION {TYPE SURROUND AG {SEGMENTTYPE EVOT}}
8
9     REPORT {FILEPATH "evot.ltc"}
10 }

```

The name of parameter LTC (line 4) is an abbreviation for long tail correction. If this parameter is specified, GEPETTO subtracts the average value from the time series before correlation. According to the OBSERVABLE MD, the correlation is given in $(e\text{\AA})^2$.

8.6.3 Example XXIII: $\langle MD(0)J(t) \rangle$ Crosscorrelation

Different observables can not only be autocorrelated but also crosscorrelated:

```

1 TRAJECTORY znf
2 {

```



```

3 FILELIST { FILETYPE DCD_CHARMM FILEPATH "5znf_evot_trif_200_2557_1800.dcd"}
4 FILELIST { FILETYPE DCD_CHARMM FILEPATH "5znf_evot_trif_200_2557_1800.vel"}
5 FILELIST { FILETYPE PSF FILEPATH "5znf_evot_trif_200_2557.psf"}
6 FIRSTFRAME 1 GRAINING 50 MAXFRAMES 1000
7 }
8
9 TASK
10 {
11     TYPE CORRELATION OBSERVABLE MD,J
12     TRAJECTORY znf
13
14     SELECTION {TYPE CORE AG {SEGMENTTYPE EVOT}}
15     SELECTION {TYPE SURROUND AG {SEGMENTTYPE EVOT}}
16
17     REPORT {FILEPATH "evot"}
18 }

```

Again, J requires the velocity trajectory. The units in the REPORT are ps and $(e\text{\AA})^2\text{ps}^{-1}$. For further details please read the examples above.

8.6.4 Example XXIV: MD Cage Correlation

The observable MDCAGE was designed to allow observation of the solvation shell specific collective rotational dipole moment:

```

1 TASK
2 {
3     TYPE CORRELATION OBSERVABLE MDCAGE, MDCAGE
4     TRAJECTORY znf
5
6     SELECTION {TYPE CORE SEPARATE NONE AG {SEGMENTTYPE 5ZNF}}
7     SELECTION {TYPE SURROUND AG {SEGMENTTYPE TIP3}}
8     SELECTION {TYPE TESSELATION AG {SEGMENTTYPE *}}
9
10    SELECTION {TYPE NEIGHBOURMAP SEPARATE NONE AG {SEGMENTTYPE 5ZNF}}
11    SELECTION {TYPE NEIGHBOURMAP AG {SEGMENTTYPE EVOT, TRIF, TIP3}}
12
13    SHELL {TYPE DELAUNAY MINVALUE 1 MAXVALUE 1}
14
15    REPORT {FILEPATH "water_around_znf"}
16 }

```

Like any other OBSERVABLE, MDCAGE can be used to calculate time series or correlations. The specification of a SHELL and a TESSELATION selection is mandatory. In this example, the contribution of water in the first solvation shell of the zinc finger to its M_D^{cage} is computed. In order to select the protein as a whole (i.e. not separated into residues or even atoms), the SEPARATE NONE keyword is recommended. This SEPARATE on line 6 must coincide with a NEIGHBOURMAP separation (line 10). The output file has two columns and is called "water_around_znf.MDCage_MDCage". As the dipole moment has three cartesian components, a calculation of a TIMESERIES of this OBSERVABLE would lead to a 4 column output file, the first of which is the time in ps and the other three are the cartesian components of M_D^{cage} in eÅ. Therefore, the unit of the CORRELATION is $(e\text{\AA})^2$.

8.6.5 Example XXV: Molecular Dipoles Projected to the Total Dipole Moment

A decomposition of a molecular dipole moment into residue specific contributions and the projection of these residue specific contributions to the total molecular dipole moment can be calculated using DECOMPOSEDDIP.

```

1 TASK
2 {
3   TYPE TIMESERIES PROP DECOMPOSEDDIP
4   TRAJECTORY znf
5
6   SELECTION { TYPE CORE          SEPARATE NONE      AG { SEGMENTTYPE 5ZNF }
7             ORIENT {FILEPATH "/home/michael/charmm/5znf/5znf_ca_coororiemass.pdb" }}
8
9   SELECTION { TYPE SURROUND      SEPARATE RESIDUE AG { SEGMENTTYPE 5ZNF } }
10
11  SELECTION { TYPE TESSELATION   SEPARATE ATOM      AG { ATOMTYPE * } }
12
13  SELECTION { TYPE NEIGHBOURMAP SEPARATE NONE      AG { SEGMENTTYPE 5ZNF } }
14  SELECTION { TYPE NEIGHBOURMAP SEPARATE RESIDUE AG { RESIDUETYPE EVOT,TRIF,TIP3 } }
15
16  SHELL { TYPE NOSHELL }
17
18  REPORT { FILEPATH "test"}
19 }
```

The CORE selection (line 6) specifies the whole protein as a single entity while the SURROUND selection separates it into residues. GEPETTO calculates the total dipole of the zinc-finger as the CORE dipole and the single residue specific dipoles as SURROUND dipoles with respect to the CORE center-of-mass. This means, both, CORE and SURROUND dipoles relate to the CORE center-of-mass. Thus all SURROUND dipole contributions add up to the CORE dipole. Although no tessellation is needed, the TESSELATION selection is specified on line 11. It is solely used to provide access to atom coordinates. In order to prevent GEPETTO from doing the expensive tetrahedralization, a SHELL {TYPE NOSHELL} is defined on line 16. The output file has the following format: Column 1: time [ps], Columns 1+(i-1)*5: μ_i^x , μ_i^y , μ_i^z , the angle between μ_i and μ_{tot} and the projection $(\mu_i \cdot \mu_{tot})/|\mu_{tot}|$. Here, μ_i is the contribution of residue i to the total CORE selection dipole μ_{tot} .

Curriculum vitae

Gregor Neumayr

Universität Wien

Institut für

Computergestützte Biologische Chemie

Währingerstrasse 17

1090 Wien

Phone: 0699 10179976

Email: gregor@mdy.univie.ac.at

Personal Information

Born on October 31, 1979.

Austrian Citizen.

Education

03/2006 - now	PhD study of Natural Sciences at the University of Vienna
03/1999 - 04/2006	Studies of Genetics/Microbiology at the University of Vienna
1998	Matura at BG/BRG in Lienz passed with distinction

Teaching

10/2009 - 02/2010	Teaching Assistant , <i>Übungen zu Mathematik für Molekularbiologen</i>
10/2008 - 02/2009	Tutor , <i>Theoretisch-chemische Übungen</i>
10/2006 - 02/2007	Tutor , <i>Übungen zu Mathematik und Statistik für Molekulare Biologen</i>

Employment

01/2009 - now	Teaching/Research Assistant , Institute for Computational Biological Chemistry
06/2007 - 01/2009	Project Staff , Institute for Computational Biological Chemistry
08/2006 - 05/2007	Senior Software Developer , SAIL Labs Technology
04/2004 - 03/2006	Software Developer , Biovertis AG
02/2002 - 04/2004	Software Developer , DOA_Consulting
04/2001 - 06/2001	Software Tester , Sysis

Publications

Journal Articles

Global and Local Voronoi Analysis of Solvation Shells of Proteins. Neumayr G, Rudas T, Steinhauser O submitted to J. Chem. Phys. (April 2010)

Relaxation of Voronoi Shells in Hydrated Ionic Liquids. Neumayr G, Schröder C, Steinhauser O. J Chem Phys. 2009 Nov 7; 131(17):174509

On the collective network of ionic liquid/water mixtures. III. Structural analysis of ionic liquids on the basis of Voronoi decomposition Schröder C, Neumayr G, Steinhauser O. J Chem Phys. 2009 May 21; 130(19):194503

On the collective network of ionic liquid/water mixtures. I. Orientational Structure Schröder C, Rudas T, Neumayr G, Benkner S, Steinhauser O. J Chem Phys. 2007 Dec 21;127(23):234503.

Impact of anisotropy on the Structure and Dynamics of ionic liquids: a computational study of 1-butyl-3-methyl-imidazolium trifluoroacetate. Schröder C, Rudas T, Neumayr G, Gansterer W, Steinhauser O. J Chem Phys. 2007 Jul 28;127(4):044505.

Diploma Thesis

Analysis of biochemical pathways in bacteria by means of bioinformatics