

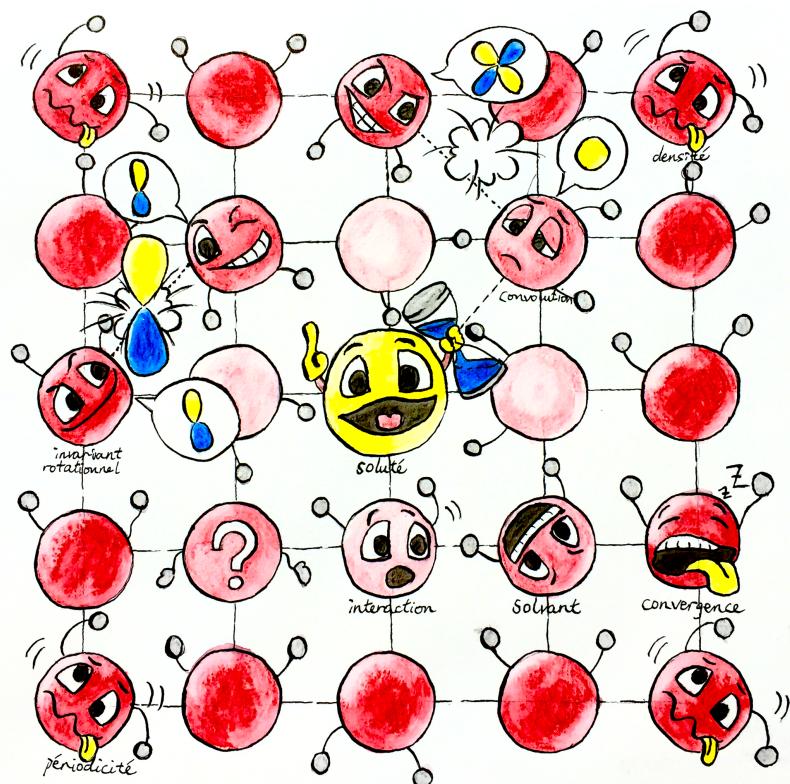
MOLECULAR DENSITY FUNCTIONAL THEORY

UNDER HOMOGENEOUS REFERENCE FLUID APPROXIMATION

LU DING

Under the direction of

DANIEL BORGIS, LUC BELLONI & MAXIMILIEN LEVESQUE



SYSTEMATIC PREDICTION OF SOLVATION PROPERTIES

WITH MOLECULAR-SCALE LIQUID THEORY

ABSTRACT

Solvation properties often play an important role in chemical and bio-chemical issues. The molecular density functional theory (MDFT) is one of the frontier domains to evaluate these properties, in which the free energy functional is minimized for an arbitrary solute in a periodic cubic solvent box. In this thesis, we work on the evaluation of the excess term of the free energy functional under the homogeneous reference fluid (HRF) approximation. Two algorithms are proposed: the first one is an extension of the previous algorithm, which allows to calculate full 3D molecular solvent (depending on three Euler angles) instead of linear solvent (depending on two angles); the other one is a new algorithm that integrates the molecular OZ equation treatment of angular convolution into MDFT, which in fact expands the solvent density and the functional gradient on generalized spherical harmonics (GSHs). It is shown that the new algorithm is much more rapid than the previous one, while the latter is more stable in terms of convergency, specially for negative charged solutes. Both algorithms are successful to predict free energy and structure of ions and small molecules.

RÉSUMÉ

*When you are studying any matter, or considering any philosophy, ask yourself only,
what are the facts and what is the truth that the facts bear out.*

*Never let yourself be diverted either by what you wish to believe, or by what you think
would have beneficent social effects if it were believed.
But look only, and solely, at what are the facts.*

— Bertrand Russell

ACKNOWLEDGEMENTS

First of all, I would like to express my most respectful gratitude to my thesis advisors, Daniel Borgis and Luc Belloni, who have developed the main theories used in this thesis and whose great knowledge of liquid theory as well as the genius way of thinking and explaining have given me a solid guide for doing this research. I would also like to thank them for their tireless work in correcting this manuscript.

I'm also grateful to Maximilien Levesque, the other main developer of MDFT who joined the supervision of my thesis. Thanks for sharing some results and data that have proven useful to my research.

I wish to acknowledge all the great minds willing to evaluate and give ideas about my work: (**noms des jurés**), thank you for agreeing to be part of the jury of this thesis.

As a master student issue from pure chemistry speciality, a lack of knowledge about Informatics brought to me a lot of difficulties. I would like to express my sincere gratitude to my colleagues Pierre Kestener, Matthieu Haefele, and Yacine Ould-Rouis for their huge aid in Informatics and very useful advices during this thesis.

This thesis was produced at *Maison de la Simulation, CEA Saclay*, financially supported by the scholarship **IDEX-CEA**. I acknowledge all the organizations and staff that gave me the chance to have this three-year experience.

I am also grateful to Thomas Wiggins for help in correcting the huge amount of grammar faults in this manuscript.

I'm deeply indebted to my tutors during my bachelor and master, respectively Mr. Hongwei Tan and Mrs. Michelle Gupta, whose clear logic and warm encouragement gave me all that I needed to be in love with theoretical chemistry.

Finally I would like to thank my father, who made the right decision to send me here in France and support me in every aspect.

CONTENTS

1	INTRODUCTION	1
1.1	Modeling of solvent effects,	1
1.2	Scope of this thesis,	5
I	STATE OF THE ART: SOLVATION, MODELS AND METHODS	7
2	MODEL OF SOLUTION SYSTEM	9
2.1	Continuum solvation models,	10
2.1.1	Poisson-Boltzmann methods,	12
2.1.2	Born / Onsager / Generalized Born models,	13
2.2	Model potential of explicit molecules,	14
2.2.1	Interaction of spherical particle,	17
2.2.2	Site-site interactions,	18
2.2.3	Multipole and spherical harmonic expansion,	19
2.2.4	SPC/E water model,	20
2.2.5	Flexible and polarizable models,	22
2.3	Model of solute,	23
3	STATISTICAL MECHANICS OF ATOMIC FLUIDS	25
3.1	Hamiltonian and ensemble properties,	25
3.2	Functional derivatives and distribution functions,	27
3.3	Classical density functional theory,	30

3.4	Integral equation theory,	31
3.5	Equivalence between cDFT and IET for a dilute solution system,	32
4	APPROACH TO MOLECULAR SOLVENTS	35
4.1	Molecular density functional theory,	35
4.1.1	The ideal term,	36
4.1.2	The external term,	37
4.1.3	The excess term,	39
4.2	Angular dependent integral equation theory,	41
4.2.1	Expansion of angular-dependent distribution functions,	41
4.2.2	Blum's reduction of molecular OZ equation,	42
5	CODE MDFT	45
5.1	Supercell discretization,	45
5.2	Minimizer L-BFGS-B,	46
5.3	Treatment to avoid unphysical density,	47
II	THEORY: HRF APPROXIMATION, FOR MOLECULAR SOLVENT	52
6	REDUCTION OF SPATIAL CONVOLUTION BY FFT	53
6.1	Using intermolecular DCF,	54
6.1.1	Zero-order interpolation of DCF,	58
6.1.2	Linear interpolation of DCF,	59
6.2	Direct calculation of DCF from rotational invariant projections,	59
6.2.1	Using projections in form of $\hat{c}_{\mu\nu}^{mn}(k)$,	60
6.2.2	Using projections in form of $\hat{c}_{\mu\nu,\chi}^{mn}(k)$,	61
7	ANGULAR CONVOLUTION, A BETTER ALGORITHM	63

7.1	Angular convolution using Blum's reduction,	63
7.2	Fast generalized spherical harmonic transform,	66
7.2.1	Equivalence of order in angular quadratures and projections,	67
7.2.2	Integration of Φ, Ψ using FFT,	68
7.3	Operational algorithm,	72
7.3.1	Commutativity between operations,	73
7.3.2	Reduction by symmetry,	76
8	FREE ENERGY CORRECTION FOR SINGLE IONS	79
8.1	Correction of type B,	79
8.2	Correction of type C,	80
9	SOLVATION STRUCTURE	83
9.1	Radial distribution function and site-site distribution function,	83
9.2	Radial polarization functions,	84
9.3	Rotational invariant expansion,	85
9.4	Equivalence between the curves,	85
III	IMPLEMENTATION	89
10	ALGORITHMS AND BRANCHES	91
10.1	Branches "naive" ,	92
10.2	Branches "convolution",	92
10.3	Testing branches for $n_{\max}=1$,	93
10.4	Other paths,	93
11	NUMERICAL AND PHYSICAL ACCURACY	95
11.1	Generalized spherical harmonics transform,	95

11.1.1 m_{\max} and n_{\max} of projections,	96
11.1.2 From ρ to γ ,	98
11.2 Comparison between branches,	99
11.2.1 Difference in energy evaluation,	99
11.2.2 A single k-kernel,	100
11.2.3 k-border effect,	101
11.3 Intrinsic variation of free energy,	104
11.4 Series of charged LJ centre,	106
11.4.1 Box length dependance and charge dependance of free energy,	106
11.4.2 Comparison with IET after corrections,	107
11.4.3 Comparison with MD,	110
11.5 Premier conclusion,	111
12 COMPUTING PERFORMANCE OF SEQUENTIAL CODE	115
12.1 FFT,	115
12.2 FGSHT,	117
12.3 k -kernel,	118
12.4 Entire iteration of \mathcal{F}_{exc} evaluation,	119
12.4.1 Comparison between “naive” methods and “convolution_pure_angular”,	
119	
12.4.2 Comparison “convolution_standard” and “convolution_pure_angular”,	
120	
12.4.3 Comparison “convolution_standard” and “convolution_asymm”,	121
121	
12.4.4 Comparison mmax and nmax,	121
12.5 Global view of the sequential code performance,	122

IV APPLICATIONS	123
13 IONS AND SMALL MOLECULES COMPARED TO MD SIMULATION	125
13.0.1 ions, 125	
14 HYDROGEN TRANSFER REACTION OF MN-OXO: ROLE OF SOLVENT IN REACTION SIMULATION	127
14.1 Reinvestigation of the Manganese-oxo , 127	
V CONCLUSION AND PERSPECTIVES	129
15 CONCLUSION	131
16 PERSPECTIVES	133
16.1 Bug report, 133	
16.2 Reduce memory use in MDFT, 133	
16.2.1 Pass to simple precision, 134	
16.2.2 MPI of the L-BFGS-B minimizer, 134	
16.3 Polarizable solute, 134	
16.4 MDFT Viewer, 134	
16.5 Classical SCF method, 135	
16.6 Other branches of development about MDFT, 135	
VI APPENDIX	137
A BASIC CONCEPTS ABOUT COMPUTING PERFORMANCE	139
A.1 Algorithm complexity, 139	
A.2 Roofline model and memory delay, 141	

A.3 Scalability of parallelized code,	143
A.4 Profiling and tracing toolkits,	144
B DIRECT CORRELATION FUNCTION OF WATER	145
B.1 Dipole DCF from molecular dynamics simulation,	145
B.1.1 Principle,	145
B.2 DCF projections from bulk Monte Carlo simulation,	145
B.3 Comparison,	146
B.3.1 Comparison with non-coupling dipole DCF in MDFT ($n_{\max} = 1$),	
146	
B.3.2 Comparison with respect to n_{\max} ,	147
C EQUIVALENCE OF QUADRATURE-PROJECTION ORDER	149
C.1 Gaussian quadrature,	149
C.2 Angular integration in GSHT,	150
D ROTATIONAL INVARIANT EXPANSION	153
D.1 Orthogonality of Φ ,	154
D.2 Rotational invariance of Φ ,	155
D.3 Transform in local frame,	156
D.3.1 Transform between $F_{\mu\nu}^{mn l}(r)$ and $F_{\mu\nu,\chi}^{mn}(r)$,	158
D.3.2 Rotational invariant transform with FFT-3D,	159
D.4 Transform in fixed frame,	160
D.4.1 Expansion of $F(\mathbf{r}, \boldsymbol{\Omega})$ on rotational invariants,	160
D.4.2 Rebuilding of $F(\mathbf{r}, \boldsymbol{\Omega})$ from projections,	162
D.5 Symmetry,	162
D.5.1 Symmetric rules of $F(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$ in intermolecular form,	162

D.5.2 Symmetric rules of rotational invariant projections,	163
E CALCULATION OF ROTATION MATRIX ELEMENTS $R_{\mu\mu'}^m$ BY RECURRENCE	167
E.1 Case of $m_{\max} \leq 1$,	167
E.2 Case of $m_{\max} > 1$,	169
E.2.1 Recurrence relation for $-m + 1 \leq \chi' \leq m - 1$,	169
E.2.2 Recurrence relation for $-m + 2 \leq \chi' \leq m$,	170
E.2.3 Symmetries,	170
F PROPERTIES OF WIGNER 3J-SYMBOL AND GSH	173
F.1 Properties of Wigner 3j-Symbol,	173
F.2 Properties of GSH,	175
F.3 Convention of GSH,	177
G ERROR EVALUATION OF INTERPOLATION STRATEGIES FOR DCF IN LOCAL FRAME	179
H ORIGINAL DATA OF MDFT IMPLEMENTATION	181
H.1 Series of charged CH ₄ center,	181
H.1.1 Branch “naive” result,	181
H.1.2 Branch “standard” result,	181
BIBLIOGRAPHY	183

LIST OF FIGURES

- Figure 1.1 The solvation process, 2
- Figure 2.1 Continuum solvent model, 10
- Figure 2.2 Definition of cavity surfaces, 11
- Figure 2.3 Euler angles, 15
- Figure 2.4 LJ pair potential, 17
- Figure 2.5 Water models, 21
- Figure 2.6 Interactions in a flexible model, 22
- Figure 2.7 Hierarchy of models, 23
- Figure 4.1 Charge density projected onto grids. (a) Solute. (b) Solvent., 38
- Figure 5.1 Main structure of code MDFT, 45
- Figure 6.1 Molecules 1 and 2 in different coordinate systems, 55
- Figure 6.2 Rotation matrices, 56
- Figure 6.3 Rotation to k-frame, 56
- Figure 6.4 $\phi_1 - \phi_2$ distribution, 58
- Figure 7.1 Indices arrangement in a complete forward-backward FFT-2D process of $m' \times m$ elements, 71
- Figure 7.2 Commutativity of operations, 74
- Figure 7.3 Distribution of points to be calculated according to symmetry in a 2D plan, 78
- Figure 8.1 IQ model and summation scheme, 81
- Figure 9.1 Structure of a POINCARE node, 88

- Figure 10.1 Possible algorithms for γ evaluation, 91
- Figure 11.1 The minimum value of $\Delta\rho(\mathbf{r}, \Omega)/\rho_0$ after a forward-backward GSHT process, 98
- Figure 11.2 Schema of a k-kernel test , 100
- Figure 11.3 k -border effect, 102
- Figure 11.4 Maximum and average difference in $\hat{\gamma}(\mathbf{k}, \Omega)$ and $\gamma(\mathbf{r}, \Omega)$, 103
- Figure 11.5 A selection of rotational invariant projections $\gamma_{\mu\nu}^{mn}(r)$ for a 65^3 grid, 104
- Figure 11.6 Space-grid and Ψ dependence of code MDFT, 105
- Figure 11.7 Quadratic charge dependence of free energy of CH₄ centre series, 107
- Figure 11.8 Comparison to IET, without P-scheme correction, 108
- Figure 11.9 Comparison to IET, with P-scheme correction, 108
- Figure 11.10 Comparison to IET, with P-scheme correction, $m_{\max} = 5$, $n_{\max} = 0, \dots, 5$, 109
- Figure 11.11 Comparison to IET. Profile of ρ in rotational invariant projections, L=24, nfft=72., 110
- Figure 11.12 Comparison to IET. Profile of ρ in rotational invariant projections., 111
- Figure 11.13 Free energy (without correction) of charged CH₄ centre (-1.0 to 1.0) with respect to the box length, for **naive_nmax1** method, with $m_{\max} = n_{\max} = 1$, at 300K., 112
- Figure 11.14 Free energy (without correction) of charged CH₄ centre (-1.0 to 1.0) with respect to the box length, for **naive_interpolation** method, with 14 angles of Lebedev quadrature angles for Θ and Φ , 3 for Ψ , DCF of $n_{\max} = 5$, at 300K., 113

- Figure 11.15 Free energy (without correction) of charged CH₄ centre (-1.0 to 1.0) with respect to the box length, for **convolution_standard** method, with $m_{\max} = n_{\max} = 1$, at 298.15K., 113
- Figure 12.1 timing FFT, 116
- Figure 12.2 FFT real to complex, 117
- Figure 12.3 Computing time of GSHT and FGSHT (per 10000 times), between parentheses is the order of symmetry axes s , 117
- Figure 12.4 FGSHT real to complex, 118
- Figure 12.5 Entire iteration of \mathcal{F}_{exc} evaluation: timing overall , 119
- Figure 12.6 comparison of **convolution_standard** and **convolution_pure_angular**, 120
- Figure 12.7 comparison of **convolution_standard** and **convolution_asymm**, 121
- Figure 12.8 comparison of **convolution_standard** for $m_{\max} = n_{\max}$ and $m_{\max} = 5$, 122
- Figure A.1 Function growth, 140
- Figure A.2 The roofline model and performance pattern, 142
- Figure A.3 Memory usage in hardware level, 142
- Figure D.1 Symmetry operations of a 2-molecule system, 163
- Figure G.1 Error of finding $\hat{c}(\mathbf{k}, \Omega_1, \Omega_2)$ by interpolation compared to direct calculation, 179

LIST OF TABLES

Table 1.1	Different solvation theories,	5
Table 2.1	Parameters for SPC and SPC/E water,	21
Table 7.1	Number of FE needed by OZ equation of different form,	66
Table 10.1	Branch option in MDFT,	91
Table 11.1	Maximum absolute error E_a^{\max} introduced by a forward-backward GSHT process,	97
Table 11.2	Free energy calculated during 1 iteration,	99
Table 11.3	Parameters of charged Lenard-Jones centre (modified from CH ₄) ,	
	106	
Table 11.4	Methods and parameters for CH ₄ series test. * Leb is Lebedev quadrature, with is mathematically equivalent with Gauss-Legendre quadrature but only ~2/3 angles.,	106
Table 12.1	Timing [sec] of loop k of “naive_standard”, “naive_interpolation” and “convolution_pure-angular”, and the rest of iteration,	120

NOTATIONS

$\mathcal{F}[\rho]$	Solvation free energy functional [$\text{kJ} \cdot \text{mol}^{-1}$] -
$\rho(\mathbf{r}, \Omega)$	Density of solvent [\AA^{-3}]-
$\mathcal{F}_{\text{id}}[\rho]$	Ideal free energy functional [$\text{kJ} \cdot \text{mol}^{-1}$] -
$\mathcal{F}_{\text{ext}}[\rho]$	External free energy functional [$\text{kJ} \cdot \text{mol}^{-1}$] -
$\mathcal{F}_{\text{exc}}[\rho]$	Excess free energy functional [$\text{kJ} \cdot \text{mol}^{-1}$]-

$\gamma(\mathbf{r}, \boldsymbol{\Omega})$	Gradient of excess free energy functional, []-
q_e	Elementary charge, $q_e = 1.602176565 \cdot 10^{-19}$ [C]
ε_0	Vacuum permittivity, $\varepsilon_0 = 8.854187817 \cdot 10^{-12}$ [C ² · J ⁻¹ · m ⁻¹]
N_A	Avogadro constant, $N_A = 6.02214129 \cdot 10^{23}$ [mol ⁻¹].
f_Q	$f_Q = q_e^2 10^{-3} N_A / (4\pi \varepsilon_0 10^{-10})$, electrostatic potential unit so that $f_Q \cdot q^2/r$ is in [kJ · mol ⁻¹], where q is the number charge without unity, r in [Å].
K_B	Boltzmann constant, $K_B = 1.3806488 \cdot 10^{-23}$ [J · K ⁻¹]
ρ_0	Bulk solvent angular density, $n_0 = \int d\Omega \rho_0 = 8\pi^2 \rho_0$ is the bulk solvent number density, both of unity [Å ⁻³]-
β	$\beta = (K_B T)^{-1}$, reciprocal of the thermodynamic temperature [mol · kJ ⁻¹]

ACRONYMS

DCF direct correlation function

DFT discret Fourier transform, also refer to density functional theory

FE function evaluation

FFT fast Fourier transform

FGSHT fast generalized spherical harmonic transform

GSH generalized spherical harmonic

GSHT generalized spherical harmonic transform

HNC hypernetted-chain (approximation)

HRF homogeneous reference fluid (approximation)

IET integral equation theory

MC Monte Carlo

MD molecular dynamics

MDFT molecular density functional theory

MOZ molecular Ornstein-Zernike (equation)

OZ Ornstein-Zernike (equation)

PCF pair correlation function

PDF pair distribution function

QM quantum mechanics

INTRODUCTION

This thesis aims to develop an original numerical toolkit for physical chemists and structural biologists based on the molecular density functional theory (MDFT), which makes it possible to predict the solvation properties of arbitrary molecular objects in arbitrary molecular solvents (mainly water) efficiently and with microscopic accuracy. This introduction contributes to understand the objective of this thesis, it explains why the theorists are interested in the nature of solvation, what are the present computing trends in solvation simulations, and where our work situates in this frame of solvation theories.

1.1 MODELING OF SOLVENT EFFECTS

Solvation is a fundamental phenomenon in chemistry. The chemical behavior of numerous systems strongly depends on the nature of solvation; this is the case for example for the reaction mechanisms in metal-organic reacting centers [16, 51], or pharmaceutical studies [30, 52, 53]. The solvation properties demanded by scientific studies are highly diverse; these include for example the Gibbs free energy of solvation, solubility, concentration, partition coefficient, saturated vapor pressure, pH value, the 3D solvation structure, etc. Overall, the interest in these solvation properties touches many fields of study such as chemistry, biochemistry, as well as pharmaceutical, environmental, and agrochemical industries. Unlike the well-studied quantum mechanics (QM) for chemical interactions at a

microscopic scale, and the finite element models for macroscopic physical processes, the theories of solvation lie in between these description scales and are still under development, owing to the ambiguous compromise between accuracy and computing cost, and the rapid development of computer hardware which makes complicated calculations more and more accessible. In a word, the studies in this domain are quite vibrant.

To change a phenomenon into a model, we must first understand its process. Solvation is defined as the process of moving a molecule from the gas phase (or vacuum) to a condensed phase (figure 1.1), which builds a stabilizing interaction with the solute (or solute moiety, e. g., residues, interfaces, etc.) [46]. Such interactions are mostly classical interactions, involving electrostatic and van der Waals forces; but also with additional more specific chemical effects such as hydrogen bond formation, and quantum effects for some small solvent molecules whose vibrational or rotational energy states are at the same magnitude as $k_B T$, and maybe other effects etc.

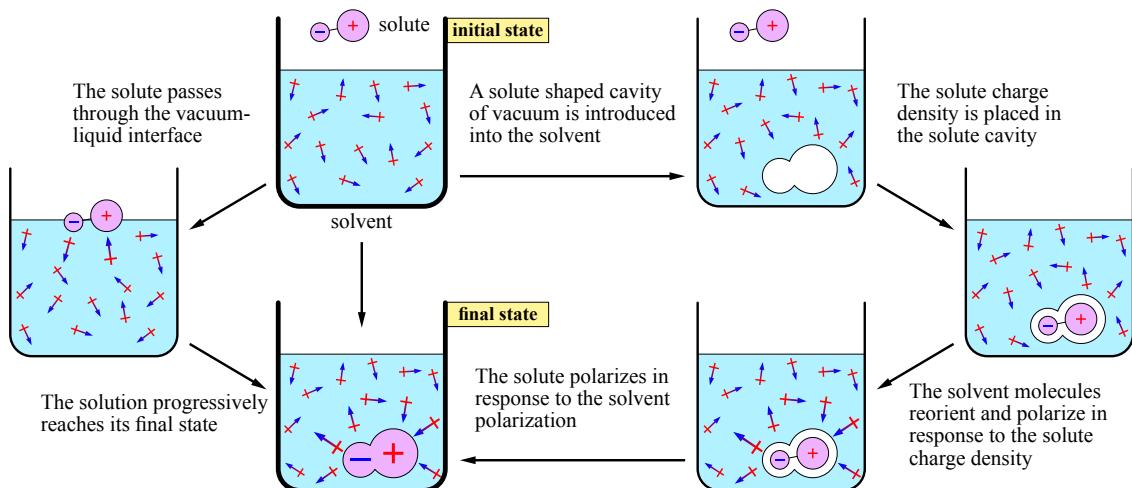


Figure 1.1: The solvation process. A thermodynamic system, whose properties only depend on the initial and final states, can go through different paths. The physical process of solvation (left path) takes the solute from vacuum into bulk solvent, progressively passing through the vacuum-liquid interface. Theoretically, the solvation energy is defined as the energy consumed in such a process. In theoretical studies, the process can be decomposed to some artificial unphysical process (right path), involving the growth of an uncharged solute-sized cavity within the bulk solvent, the transfer of the solute charge distribution from vacuum into cavity, and the interaction between the solute and solvent.

As not all kinds of interactions are important in applications, different models and methods have been developed according to the usage.

For most of the 20th century, the study of solvation effects has been dominated by continuum (implicit) models [11, 36], which mostly rely on the continuum dielectric description of the solvent and are not costly in terms of computation resources. They provide an accurate way to treat the strong, long-range electrostatic interactions which dominate many solvation phenomena, but lack detailed information on the first solvation shell. This information mainly includes the cavity formation energy and solute-solvent van der Waals interactions, that are often rudely treated by introducing an artificial form of cavity that links to the form of solute. The methods for testing electrostatic interactions include the Generalized Born (GB) approximation or, for better estimates, Poisson-Boltzmann (PB) calculations. These are widely integrated within QM calculations by adding extra solvation terms onto the Fock or Kohn-Sham operator [65, 66]. However, the improper treatment of the first shell, where the microscopic interactions are primarily located, often introduces potentially huge errors in free energy evaluation, especially for polar solvents (such as water), despite the accuracy that the QM calculation alone can achieve. Therefore, classical molecular simulations, which describe the individual solvent molecules explicitly (explicit solvent), particularly the molecular dynamics (MD) and Monte Carlo method (MC), have become the alternative solution during the last few decades. They generate trajectories and configurations, and from there estimate free energy changes by statistical mechanical techniques, such as free energy perturbation (FEP) theory or thermodynamic integration (TI) [37]. These calculations are very demanding in computing cost, due to the need for very many (hundreds or thousands) solvent molecules to form a realistic model, and very many configurations (millions) to be statistically significant.

Recently, a third domain of theory to describe solvents based on the statistical mechanics of fluids has been growing rapidly. It mainly involves the integral equation theory (IET), and the classical density functional theory (cDFT) for liquids. These approaches are capable of giving the molecular nature of the first solvation shell, while without calculating all the instantaneous micro-states with respect to time, but rather by integrating over

positions and momenta theoretically. Therefore, they are of orders of magnitude faster than the simulations done by micro-states.

The integral equation theory (IET) focuses on solving the Ornstein-Zernike (OZ) equation with a specific closure equation [22, 26]. It was firstly limited to so-called “simple liquids” - a system of spherical particles. The extension to molecular fluids, composed of polyatomic molecules with non-spherical shapes, was done in two different directions. On one hand, Chandler and Andersen in 1971 [7] developed the reference interaction site model (**RISM!**), which discretizes the distribution and correlation functions into site-site functions, and solves the OZ equation and the closure in a matrix form [28]. On the other hand, Blum [4, 5], Fries and Patey [17] extend the OZ equation into a full molecular form, where the distribution and correlation functions depend on both position and orientation. In their theory, the orientation part of OZ equation is simplified by expanding the distribution and correlation functions onto Wigner generalized spherical harmonics.

The classical density functional theory approach deals with inhomogeneous liquids, and uses the same variation principle and minimization strategy [14, 24, 47] as electronic density functional theory (eDFT) for electron-electron interactions. The latter has received an immense success in computational chemistry. Classical DFT gives the Helmholtz free energy and the equilibrium solvent density by minimizing the free energy functional of the solvent density in the presence of a given external potential. Borgis and collaborators [21, 32–35, 42, 56, 57, 60, 68] have recently generalized it into the molecular case, leading to molecular density functional theory (MDFT), where the solvent density depends on both position and orientation, $\rho(\mathbf{r}, \Omega)$. The main theoretical difficulty lies in the definition of well-fundated and reliable functionals of the excess free energy $\mathcal{F}_{\text{exc}}[\rho]$, accounting for the geometric complexity of the solvent molecule. Some recent research has shown that MDFT is capable to describe linear solvents like acetonitrile, but has still some caveats for the most complex solvent, i. e. water [68]. MDFT can be proven to be mathematically equivalent to the two-component molecular IET.

The majority of work of all these theories has been focused on water, since it is one of the most difficult systems to model due to its molecular geometry, unavoidable multi-body character, quantum effects, and hydrogen bonds, etc. The importance of including instantaneous polarization in potential functions is also an issue [12, 63]. However, since polarizable force fields are not yet in common use, the simulations by micro-states and the liquid theory which feed on force fields also have their own limits, compared to the continuum model which can be totally polarizable. The advantages and disadvantages of each branch of theory are listed in table 1.1.

THEORY	SPEED	LONG-RANGE	FIRST-SHELL	POLARIZABLE SOLVENT
Continuum model	fast	yes	no	fully
Simulation by time	costly	yes	yes	partially, very costly
Theory of liquids	fast	yes	yes	partially

Table 1.1: Different solvation theories

1.2 SCOPE OF THIS THESIS

This thesis aims at developing the theory and the code of MDFT, focusing on the generalization and algorithmic acceleration of the excess free energy functional \mathcal{F}_{exc} evaluation under homogenous reference fluid (HRF) approximation, which will be discussed in detail in latter chapters.

Chapter I reviews a selection of models and methods to describe solvent effects. It includes the implicit and explicit models, the basics of liquid-state theory, as well as its two frontier research domains, IET and MDFT. Some details of the code MDFT, associated to the MDFT approach, on which all the developments of this thesis are based, are also presented.

Chapter II presents all the theory developed and newly used in this thesis. Two algorithms for the excess energy functional evaluation are proposed. One is an extension of the previous algorithm which can be only applied to linear solvent (using only two spherical coordinate angles Θ and Φ as orientation of the solvent particle, instead of three Euler angles), to a full 3D molecular solvent case; while the other is a new algorithm, that combines the molecular OZ equation treatment of angular convolution with MDFT. The solvation properties that the code generates are also presented, mainly containing the free energy and solvent structure.

Chapter III reports all the implementation results, which are divided into two aspects: the “accuracy”, which involves the error evaluations, comparisons between algorithms and with IET and MD results; and the “efficiency”, which evaluates the computing cost, from the parts of the code to the entire branches.

Chapter VI gives applications to ions and molecules. Some works that remain unachieved due to the time limit are put in the perspectives.

Chapter I

STATE OF THE ART: SOLVATION, MODELS AND METHODS

This chapter gives a brief review of all the basic concepts and previous work that this thesis is based on.

In section 2, we begin by introducing the frequent models of solvent in a simulation, from the simplest implicit continuum model to the most complex explicit one. The overview of these models then helps to understand the choice of description scale used in our study, as well as its limits.

Once the model is chosen, all the theories become mathematical problems. Section 3 reviews some basic concepts of statistical mechanics for liquids (theory of liquids), which present some brief formalisms deduced for an atomic solvent model. Two frontier approaches are then introduced with a few deductions: the molecular density functional theory (MDFT) that this thesis works upon, and the integral equation theory (IET). A mathematical equivalence between these two theories is also presented, which gave us later the idea to use the expansion technics in IET to serve MDFT. The following section 4 gives the extension of these theory to molecular solvent case.

Section 5 gives a detailed presentation of the initial code MDFT, which the development in this thesis is based on.

MODEL OF SOLUTION SYSTEM

Computing models of solvents are broadly divided into two types: those treating the solvent as a continuous medium (implicit models) and those describing the individual solvent molecules (explicit models). In the continuum model, the solvent is characterized by the dielectric constant ϵ and contains an artificially shaped cavity. The explicit models can have more specific microscopic scales. Within the scope of classical mechanics, the most detailed (and thus the most expensive) methods involve flexible and polarizable explicit models, while in computational chemistry, less detailed models often have wider usage (for example, the proteins are treated in the unity of residues). As the theory of liquids was initially established for spherical atom-like solvent particles, the model adopted by such a theory is a rigid entity carrying distributed point charges, characterized by their position and orientation, i. e. there is no internal movement considered. This approximation has been proven reasonable [22]. There also exist models in which the scale lies somewhere between the implicit and explicit models; for example, so-called coarse-grained models, which gather groups of atoms into a single interaction site.

In this section, we will give a brief introduction of the implicit model in order to facilitate later discussion on solvation free energy corrections. We will then focus on the rigid solvent models and discuss the limits of such approximations. The flexible and polarizable models will also be briefly mentioned.

2.1 CONTINUUM SOLVATION MODELS

Continuum models [11, 36, 66], which are popular in QM calculations, consider the solvent

The dielectric constant ε is the key parameter characterizing the solvent. It is normally a constant value,

but that can depend on the distance from the solute M . (see

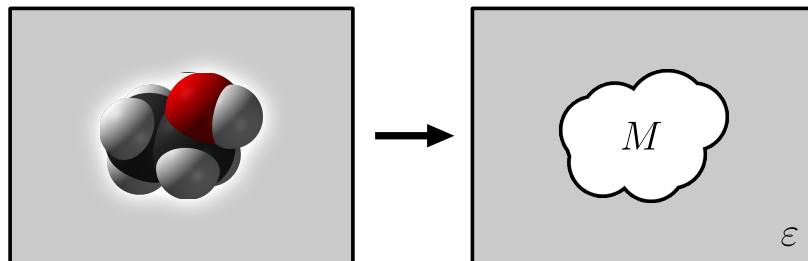


Figure 2.1: Continuum solvent model

The solvation Gibbs free energy according to this model is

§2.1.1)

$$\Delta G_{\text{solvation}} = \Delta G_{\text{cavity}} + \Delta G_{\text{dispersion}} + \Delta G_{\text{elec}} \quad (2.1)$$

where $\Delta G_{\text{cavity}} > 0$ is the energy needed to create a hole in the medium, and $\Delta G_{\text{dispersion}}$ is the dispersion interaction, which is roughly the van der Waals energy $\Delta G_{\text{vdW}} < 0$ between the solvent and solute. In principle, there may also be a repulsive component, and the dispersion term is sometimes denoted dispersion/repulsion. $\Delta G_{\text{elec}} < 0$ is the contribution of electrostatic interactions, introduced by electric charge distribution of M which polarizes the medium, and the action back of the medium on the molecule (reaction field).

The initial two terms in eq. (2.1) are linked to the configuration of the first solvation shell (cavity). The definition of cavity varies from the simplest sphere or ellipsoid to the ensemble of atomic surfaces defined by the van der Waals radii in the solute. It is somewhat reasonable to consider the cavity area as proportional to the number of solvent molecules in the first solvation shell. This number can be calculated as the area passing

through the middle region of first shell solvent. This area, named the solvent-accessible surface area (SASA) [27, 41], can be calculated by adding the radius of the probe solvent ball on the solvent excluded surface area (figure 2.2).

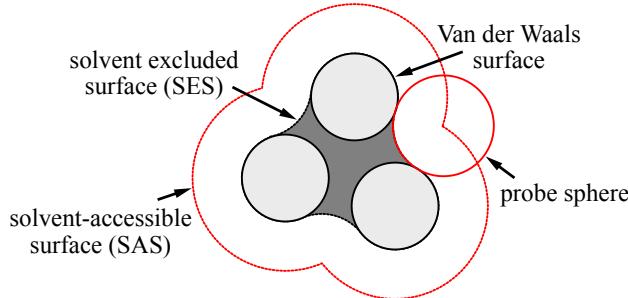


Figure 2.2: Definition of cavity surfaces. The solvent accessible surface (SAS) traced out by the center of the probe representing a solvent molecule. The solvent excluded surface (SES) is the topological boundary of the union of all possible probes that do not overlap with the molecule.

The energy required to create such a cavity and the stabilization due to van der Waals interactions between the solute and solvent, assumed to be proportional to the surface area of the cavity, is expressed as

$$\Delta G_{\text{cavity}} + \Delta G_{\text{dispersion}} = \gamma S_{\text{SASA}} + \beta \quad (2.2)$$

or parameterized by having a constant ξ specific for each atom type, with the ξ parameters being determined by fitting to experimental solvation data:

$$\Delta G_{\text{cavity}} + \Delta G_{\text{dispersion}} = \sum_i^{\text{atoms}} \xi_i S_i \quad (2.3)$$

The models and methods employed to calculate the electrostatic contribution ΔG_{elec} have varied greatly according to their usage. The sections below list the most common examples. On another topic, the integration of continuum models into QM calculations is also a very important field; these developments will not be detailed here as they do not connect yet to our work. Such kinds of methods are called the self-consistent reaction field (SCRF) models, which integrate the calculation of the solute-solvent interaction in

addition to that of the solute wave function by an iterative procedure. Some examples are presented in the list of Gaussian keyword SCRF [20], and the field is well reviewed by, for example, Tomasi [65, 66] and Jensen [36].

2.1.1 Poisson-Boltzmann methods

The Poisson-Boltzmann equation (PBE) [29] makes it possible to calculate the position-dependent electrostatic potential $V_{\text{elec}}(\mathbf{r})$ in the continuum model, such that the electrostatic component of the free energy can be written as

$$\Delta G_{\text{elec}} = \frac{1}{2} \int d\mathbf{r} \rho_q(\mathbf{r}) V_{\text{elec}}(\mathbf{r}) \quad (2.4)$$

where ρ_q is the charge distribution (of the solute?).

The Maxwell-Gauss equation in SI units convention gives

$$\nabla \cdot D(\mathbf{r}) = \frac{\rho_q(\mathbf{r})}{\epsilon_0} \quad (2.5)$$

where $D(\mathbf{r}) = \epsilon_0 E(\mathbf{r}) + P(\mathbf{r})$ is the electric displacement field, $P(\mathbf{r})$ is the system polarization, $E(\mathbf{r})$ the electric field, and ϵ_0 the vacuum permittivity. $D(\mathbf{r})$ can also be expressed in terms of the position-dependent dielectric constant $\epsilon(\mathbf{r})$, $D(\mathbf{r}) = \epsilon(\mathbf{r})E(\mathbf{r})$, which thus gives

$$\nabla \cdot \epsilon(\mathbf{r}) E(\mathbf{r}) = \frac{\rho_q(\mathbf{r})}{\epsilon_0} \quad (2.6)$$

or in terms of electrostatic potential:

$$\nabla \cdot [\epsilon(\mathbf{r}) \nabla V_{\text{elec}}(\mathbf{r})] = -\frac{\rho_q(\mathbf{r})}{\epsilon_0} \quad (2.7)$$

This second-order differential equation (2.7) is called the Poisson equation.

This equation cannot be solved analytically for complex geometries (such as a protein). Therefore it is done numerically using appropriate methods; for example, mentioned in the article of Roux and Simonson [58] or Holst [29]. A density functional approach based on the minimization of the polarization density can also be used to solve this equation [43, 44].

If the solvent is ionic, the Poisson equation can be modified by taking into account a (thermal) Boltzmann distribution of ions in the solvent, i. e.,

$$\rho_{\text{tot}}(\mathbf{r}) = \rho_q(\mathbf{r}) + qn_{\text{ion}} \sinh\left(\frac{qe}{k_B T} V_{\text{elec}}(\mathbf{r})\right) \quad (2.8)$$

for a salt composed of ions of charge $+q$ and $-q$ and of density n_{ion} . Replacing in eq. (2.7) leads to the Poisson-Boltzmann Equation:

$$\nabla \cdot (\varepsilon(\mathbf{r}) \nabla V_{\text{elec}}(\mathbf{r})) - \frac{qn_{\text{ion}}}{\varepsilon_0} \sinh\left(\frac{qV_{\text{elec}}(\mathbf{r})}{kT}\right) = -\frac{\rho(\mathbf{r})}{\varepsilon_0} \quad (2.9)$$

2.1.2 Born / Onsager / Generalized Born models

For simple geometries, the Poisson equation (2.7) can be solved analytically.

The simplest model is a spherical cavity. For a net charge q in a cavity of radius a , the electrostatic free energy of a medium with a dielectric constant of ε is given by the Born formula:

$$\Delta G_{\text{elec}}(q) = -\frac{1}{8\pi\varepsilon_0} \left(1 - \frac{1}{\varepsilon}\right) \frac{q^2}{2a} \quad (2.10)$$

Other similar models include the Onsager model, in which a point dipole (characterized by the dipole moment μ) is put in a spherical cavity. The Kirkwood model refers to a general multipole expansion in a spherical cavity, while the Kirkwood-Westheimer model

arises for an ellipsoidal cavity. Those simplified models are not fully able to predict the solvent behavior in many realistic cases [36].

The generalized Born (GB) model is an empirical model based on the superposition of several net charges in spherical cavities as the Born model describes, with a similar formula:

$$\Delta G_{\text{elec}} = -\frac{1}{8\pi\epsilon_0} \left(1 - \frac{1}{\varepsilon}\right) \sum_i \sum_j \frac{q_i q_j}{f_{ij}} \quad (2.11)$$

where the function f_{ij} depends on the internuclear distance r_{ij} between the centers of atoms i and j and on the Born radii for each pair of atoms a_i and a_j :

$$f_{ij} = \sqrt{r_{ij}^2 - a_i a_j \exp\left(\frac{r_{ij}^2}{4a_i a_j}\right)} \quad (2.12)$$

The key (empirical) point is to be able to attribute an effective Born radius a_i to each atom inside the complex, non-spherical cavity formed by the solute. Once this is accomplished, the GB model provides a very fast method, with an overall accuracy comparable to that of Poisson-Boltzmann calculations. That makes it widely used in computational structural biology to perform structure optimization and molecular dynamics simulations.

2.2 MODEL POTENTIAL OF EXPLICIT MOLECULES

The model potential frequently used in the theory of liquids is a classical, rigid, pairwise additive model [22, 26]. It is based on three assumptions.

1. Firstly, the quantum effects should be ignored. It is assumed that the rotational and transitional motion of solvent particles are continuous and classical, which means the separation of both transitional and rotational states are largely inferior of $k_B T$. For light molecules, that is not always convincing. Some molecules containing hydrogen (e. g. H₂O, NH₃, and particularly H₂) exhibit obvious quantum effects at low

temperature in the liquid state. Gaseous H₂O and NH₃ also need quantum effect corrections. However, for the liquid of most interest to us, H₂O at room temperature, the contribution of this effect is small enough to be neglected. And obviously, there should not be any chemical interaction of the solvent with the solute.

2. Secondly, the intramolecular movement (vibration and internal rotation) should be either independent of transitional and rotational movement or absent. This rigid molecule approximation assumes that the intermolecular potential $\mathcal{U}(\mathbf{r}^N, \Omega^N)$ for N particles only depends on the positions of the N molecular centers $\mathbf{r}^N \equiv (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$ and on their orientations $\Omega^N \equiv (\Omega_1, \Omega_2, \dots, \Omega_N)$, where $\Omega \equiv (\Theta, \Phi, \Psi)$ represents the Euler angles (figure 2.3). The natural choice for the molecular center is the center of mass. This is, however, arbitrary if only equilibrium properties are considered.

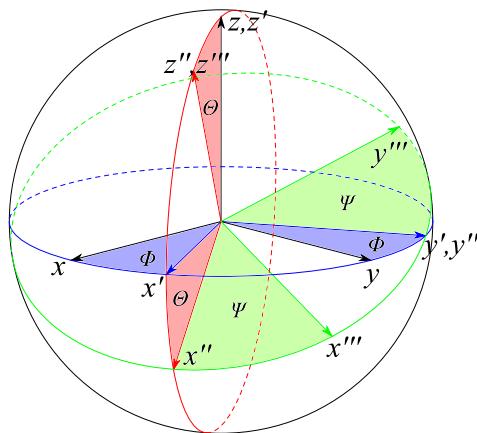


Figure 2.3: Euler angles. The basis vectors of the new orientation are obtained by 3 sequential operations: (1) A rotation ϕ ($0 < \phi < 2\pi$) about the z -axis, bringing the frame of axes from the initial position \mathbf{S} into the position \mathbf{S}' (2) A rotation θ ($0 < \theta < \pi$) about the y -axis of the frame \mathbf{S}' , which is transformed into \mathbf{S}'' (3) A rotation ψ ($0 < \psi < 2\pi$) about the z -axis of the frame \mathbf{S}'' .

The rigid approximation is quite realistic for molecules in which the separation of vibrational states largely exceeds $k_B T$, implying that the molecule stays in its ground vibrational state. This is the case for many small solvent molecules such as N₂, CO₂, C₆H₆, and indeed for the bending and stretching modes of water.

Compared to atomic models that only depend on \mathbf{r}^N , the angular correlations can give influence on both structural and thermodynamic proprieties. That is why our theory is extended to linear case ($\Omega \equiv (\Theta, \Phi)$) then molecular case ($\Omega \equiv (\Theta, \Phi, \Psi)$).

3. Finally, the intermolecular forces have to be assumed as pairwise additive:

$$\mathcal{U}(\mathbf{r}^N, \Omega^N) = \frac{1}{2} \sum_{i \neq j} u(\mathbf{r}_{ij}, \Omega_i, \Omega_j) = \sum_{i < j} u(\mathbf{r}_{ij}, \Omega_i, \Omega_j) \quad (2.13)$$

This means that the model potential only depends on the intermolecular separation \mathbf{r} and on the molecular orientations Ω_1 and Ω_2 . This approximation is quasi-exact for low density gases, where the contribution of the three and more body terms decreases rapidly. But for dense fluids, in most of the cases the multi-body potential cannot be ignored. The complete model potential with higher-order corrections can be written in the form of

$$\mathcal{U}(\mathbf{r}^N, \Omega^N) = \sum_{i < j} u(ij) + \sum_{i < j < k} u(ijk) + \sum_{i < j < k < l} u(ijkl) + \dots \quad (2.14)$$

where $u(ij) = u(\mathbf{r}_{ij}, \Omega_i, \Omega_j)$ and $u(ijk) = u(\mathbf{r}_{ij}, \mathbf{r}_{jk}, \mathbf{r}_{ki}, \Omega_i, \Omega_j, \Omega_k)$, etc. The omission of the three-body and higher-order terms can cause error, for example, in surface tension and surface energy calculation [49]. However the higher-order terms are often accounted for by an effective pair potential (measured by experiments or calculated by simulations), which reduces considerably the computational cost for simulations, or the degree of theory needed. Such models are presented below going from simple to molecular liquids. For the molecular solvent considered in this thesis, water, most publications have stayed at this two-body level of description.

2.2.1 Interaction of spherical particle

The simplest model of a fluid is the hard sphere model. With d the hard-sphere diameter, the pair potential is defined as:

$$u(r) = \begin{cases} \infty & r < d \\ 0 & r > d \end{cases} \quad (2.15)$$

This model is indeed a fundamental reference model in statistical mechanics, and it can represent some physical systems, such as neutral colloidal suspensions. However, the absence of attractive force, which precludes the existence of a liquid-gas transition, makes it too simple for realistic fluids. More realistic neutral particle models, like the Lenard-Jones (LJ) model, exhibit a potential energy curve that has the same shape as the real interaction of rare gas, as shown in figure 2.4.

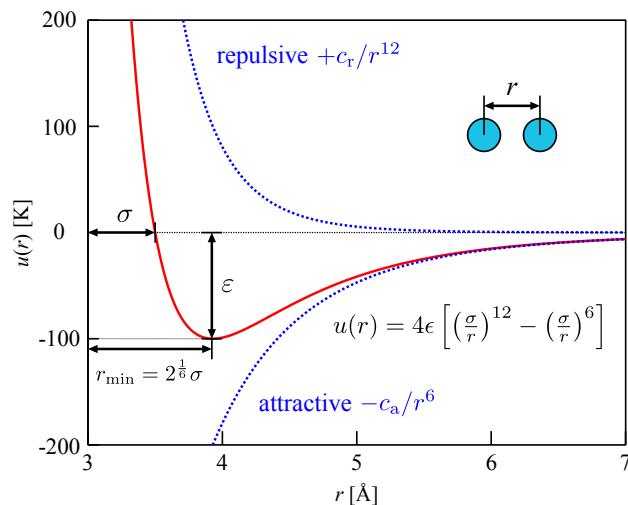


Figure 2.4: LJ pair potential. The plot gives the potential energy $u(r)$ versus internuclear distance r of two particles. At large distances, both attractive and repulsive interactions are small. As the distance between the atoms decreases, the attractive electron-proton interactions dominate, and the energy of the system decreases. At the observed bond distance, the repulsive electron-electron and proton-proton interactions just balance the attractive interactions, preventing a further decrease in the internuclear distance. At very short internuclear distances, the repulsive interactions dominate, making the system less stable than the isolated atoms.

The Lennard-Jones (LJ) interaction gives

$$u_{LJ}(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] \quad (2.16)$$

where r is the distance from centre to centre, σ is the collision diameter or the particles separation where $u(r) = 0$, and ϵ is the well depth of the potential (of unity energy kJ/mol). The well minimum occurs at $r_{\min} = 2^{1/6}\sigma$ and $u(r_{\min}) = -\epsilon$. The parameters σ and ϵ can be extracted from experiments.

Theoretically, all terms in the multipole series represent attractive contributions to the potential. The leading term, varying as r^{-6} , describes the (quantum?) dipole-dipole interaction. Higher-order terms represent dipole-quadrupole (r^{-8}), quadrupole-quadrupole (r^{-10}) interactions, and so on, but these are negligible compared to r^{-6} . The short-range interaction is difficult to define properly, and for the sake of simplicity and numerical efficiency, it is defined as r^{-12} in the LJ model.

If the spherical particles are charged (as in molten salts), the electrostatic interaction between them is described by the Coulomb point charge interaction:

$$u_{\text{Coul}}(r) = \frac{q_1 q_2}{4\pi\epsilon_0 r} \quad (2.17)$$

For such charged simple fluids, the overall pair $u(r)$ is a sum of LJ and Coulomb interactions. Such decomposition can be extended to molecular fluids in terms of site-site interactions, which are discussed in the following section.

2.2.2 Site-site interactions

Indeed, a spherical description of interactions is not sufficient to fully describe molecular fluids. The site-site model is a further extension of atomic models, in which the solvent

molecule is represented by a set of discrete interaction sites. The total potential energy is a sum of spherical interaction potentials:

$$u(1, 2) = \frac{1}{2} \sum_{\alpha} \sum_{\beta} u_{\alpha\beta}(|\mathbf{r}_{2\beta} - \mathbf{r}_{1\alpha}|) \quad (2.18)$$

(the 1/2 ?) where \mathbf{r}_{is} is the coordinates of site s in molecule i , $u_{\alpha\beta}(r)$ the interatomic potential energy of pairs of sites α and β , as discussed above. More specifically, it is generally decomposed into a Lennard-Jones and a Coulombic contribution:

$$u(1, 2) = \frac{1}{2} \sum_{\alpha} \sum_{\beta} \left\{ 4\epsilon_{\alpha\beta} \left[\left(\frac{\sigma_{\alpha\beta}}{r_{12}^{\alpha\beta}} \right)^{12} - \left(\frac{\sigma_{\alpha\beta}}{r_{12}^{\alpha\beta}} \right)^6 \right] + \frac{1}{4\pi\epsilon_0} \frac{q_{\alpha}q_{\beta}}{r_{12}^{\alpha\beta}} \right\} \quad (2.19)$$

where $r_{12}^{\alpha\beta} = |\mathbf{r}_{2\beta} - \mathbf{r}_{1\alpha}|$, $\epsilon_{\alpha\beta}$ and $\sigma_{\alpha\beta}$ are the site-site LJ parameters and q_{α} the partial charge on each site. This model is the most commonly adopted, and it will be so in this thesis.

2.2.3 Multipole and spherical harmonic expansion

To obtain fewer terms in the calculation, the model potential can be presented in convergent series via multipole or spherical harmonic expansion.

For polar liquids, the dipole-dipole interaction should be mainly taken into account. Thus the model considers dipole-dipole interactions in addition to a spherically symmetric Lennard-Jones-like potential:

$$u(1, 2) = u_0(r) - \boldsymbol{\mu}_1 \cdot \mathbf{T}(\mathbf{r}) \cdot \boldsymbol{\mu}_2 \quad (2.20)$$

where \mathbf{r} is the vector separation of the molecular centers, $u_0(r)$ is the spherically symmetric term discussed above, $\boldsymbol{\mu}_i$ is the dipole moment vector of particle i and $\mathbf{T}(\mathbf{r})$ is the dipole-dipole interaction tensor:

$$T(\mathbf{r}) = \nabla^2 \left(\frac{1}{r} \right) = 3\mathbf{rr}/r^5 - \mathbf{I}/r^3 \quad (2.21)$$

and \mathbf{I} is the unit tensor. Note that this model can be made more realistic by including higher-order electrostatic interactions, such as dipole-quadrupole, quadrupole-quadrupole, etc. Such a systematic multipolar approach has been proposed for water [10].

Alternatively, the intermolecular potential can be expanded onto rotational invariants in the form:

$$u(\mathbf{r}_{12}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) = \sum_{mnl\mu\nu} u_{\mu\nu}^{mnl}(r_{12}) \Phi_{\mu\nu}^{mnl}(\hat{\mathbf{r}}_{12}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) \quad (2.22)$$

where the angular basis functions $\Phi_{\mu\nu}^{mnl}(\hat{\mathbf{r}}_{12}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2)$ can be expressed in terms of generalized spherical harmonics (GSHs) [22]. A detailed description of rotational invariant transform is in appendix D.

2.2.4 SPC/E water model

As water cannot be perfectly described by a pair potential (due to multi-body effects, quantum effects, hydrogen bond, etc.), various models have been developed to fit a maximum number of properties. Those models contain several sites, which can be placed possibly elsewhere than at the center of atoms (figure 2.5). The more sites the model has, the more precise it can be. There is a great work done by Martin Chaplin [8] to summarize the most widely used water models.

In this thesis, we use the extended simple point charge model (SPC/E) of water [2] as

It should be noted our solvent model through all this thesis. It is a 3-site model, the electrostatic interaction

that any rigid

solvent model is

compatible with

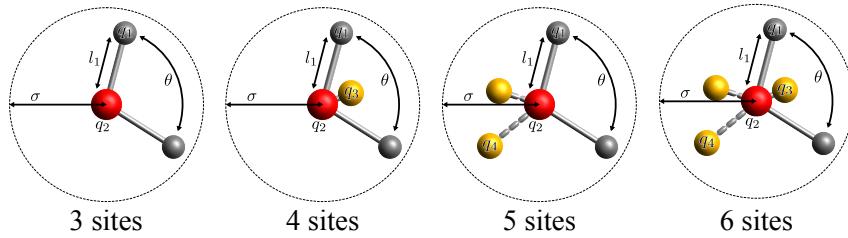


Figure 2.5: Water models

being modeled using Coulomb's Law and the dispersion and repulsion forces using the Lennard-Jones potential, as described above.

With respect to the original SPC model, the SPC/E model adds a polarization correction term to the SPC potential energy in order to better match the experiment:

$$E_{\text{pol}} = \frac{1}{2} \sum_i \frac{(\mu - \mu^0)^2}{\alpha_i}$$

where μ is the dipole of the effective pair model, μ^0 is the dipole moment of an isolated water molecule, and α_i is the isotropic scalar polarizability [2]. The SPC/E model gives a better radial distribution function and diffusion constant than the SPC model. It is the most commonly used model for applications.

Some parameters are listed in table 2.1, compared with its relative SPC model.

MODEL	σ [\AA^6]	ε [$\text{kJ} \cdot \text{mol}^{-1}$]	l_1 [\AA]	q_1 [e]	q_2 [e]	θ [$^\circ$]
SPC [3]	3.166	0.650	1.0000	+0.410	-0.8200	109.47
SPC/E [2]	3.166	0.650	1.0000	+0.4238	-0.8476	109.47
experiment [61]	-	-	0.991	-	-	105.5

(a) Structural parameters

MODEL	MOLAR VOLUME [cm^3]	NUMBER DENSITY	DIELECTRIC CONSTANT	DIPOLE MOMENT [D]
SPC			65 [62]	2.274[2]
SPC/E		[ref? unity?]	71 [2, 40]	2.351[2]
experiment	18.0685 [1006]		78.4	2.95

(b) Calculated physical properties. All the data is at 25 °C and 1 atm.

Table 2.1: Parameters for SPC and SPC/E water

2.2.5 Flexible and polarizable models

Up to this point, molecules were considered as rigid bodies. Flexible models give extra degrees of freedom in vibration and internal rotation. In that case, the interaction potential contains several extra terms, yielding typically five kinds of forces: three for the direct interactions in addition to the two indirect interactions (LJ and Coulomb).

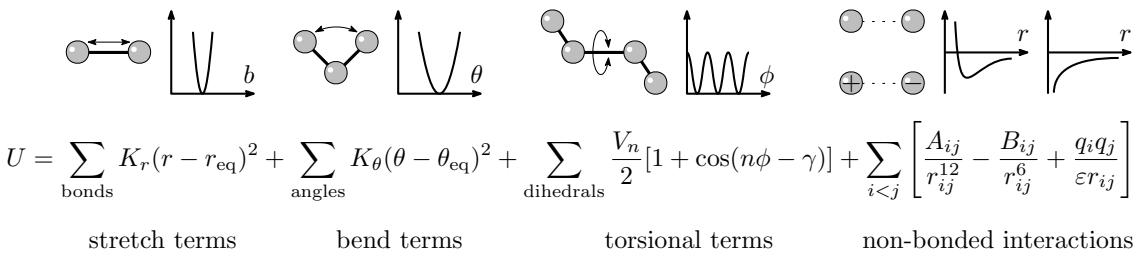


Figure 2.6: Interactions in a flexible model

The flexible yielding can deal with the non-rigidity of the solvent, which is partially polarized owing to the vibrational degrees of freedom (the so-called atomic polarizability). On the other hand, electronic polarizability (the deformation of the molecule electron cloud under the action of the external electric field) can be taken into account even in a rigid model. This polarizability can be described by introducing a modifiable charge distribution, for example by adding an induced dipole at the molecular center of the molecule, or even on each of its atomic sites, and by solving the set of induced dipoles self-consistently. Introducing variable atomic charges is possible too [ref]. Optimizing the induced charges/dipoles has a large computational overhead compared to fixed charges.

Complex models require expensive computing cost, but still can have large fluctuations due to use of imposed small system size. There is a compromise between the choice of model and the choice of system size. For this reason, the rigid models are still nowadays the most popular. On the other hand, computing technologies have greatly developed compared to the theories themselves, which makes it possible to use more and more precise models in computation.

2.3 MODEL OF SOLUTE

The model of solute also have a substantial influence on the predicted energy and structure of solvation. The solute can eventually be treated by QM calculations in terms of wave function and electron density. This is the case for the implicit SCRF method, which for apolar solvents (i.e. toluene) has been proven to work well. There is a clear mismatch, however, between the very refined description of the solute and the rather primitive continuous-medium treatment of the solvent. The compromise to have a better model of solvent or solute is debatable, and should vary according to the applications. On the other extreme, one never uses a quantum solvent model with an implicit solute; this would not be profitable even if the solute is of simple geometry (wall). In the case of molecular solutes, it is consistent to require the solute and solvent to have at least the same scale of description. Within molecular force fields, this leads to the hierarchy of potential models described in figure (2.7).

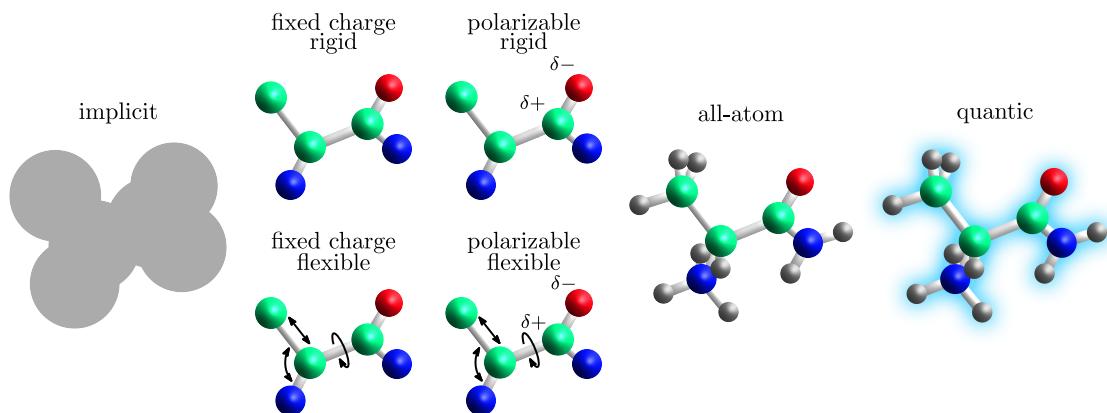


Figure 2.7: Hierarchy of models

In this thesis, our first step will be to use a rigid molecular model to describe the solute. This is coherent with IET, which cannot treat the solvent and solute at different scales of description. Polarizable and/or flexible models of solute, and the coupling of a QM solute to the molecular solvent, will be described in perspective.

In conclusion, the choice of model for the solute/solvent system is a compromise between the required precision according to the application, and the computing cost that the research can afford.

3

STATISTICAL MECHANICS OF ATOMIC FLUIDS

Statistical mechanics serves to deduce thermodynamic quantities from the Hamiltonian of any given system. In this section, we present some basic formalism for a classical atom-like spherical solvent model in grand canonical ensemble (μ, V, T) . Firstly, we introduce the relations between the statistical mechanics and thermodynamic quantities. Then we change the view to the structure of the solvent. The two theories we use in this thesis, here referred to as IET and MDFT, as well as their equivalency, are proven in the following sections. The majority of these sections is based on the book by Hansen & McDonald [25, 26], and the articles and notes of Evans [14, 15, 64].

3.1 HAMILTONIAN AND ENSEMBLE PROPERTIES

Once we define a spherical solvent model, of which the movement only depends on its position and momentum (\mathbf{r}, \mathbf{p}) , the instantaneous state (phase point, micro-state) of an N -particle solvent system is specified by $3N$ coordinates $\mathbf{r}^N \equiv \mathbf{r}_1, \dots, \mathbf{r}_N$ and $3N$ momenta $\mathbf{p}^N \equiv \mathbf{p}_1, \dots, \mathbf{p}_N$. The internal energy of particles in a system is characterized by its Hamiltonian:

$$H_N(\mathbf{r}^N, \mathbf{p}^N) = K_N(\mathbf{p}^N) + V_N(\mathbf{r}^N) + V_N^{\text{ext}}(\mathbf{r}^N) \quad (3.1)$$

where

$$\begin{aligned}
K_N(\mathbf{p}^N) &= \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m} \text{ is the kinetic energy;} \\
V_N(\mathbf{r}^N) &= \sum_{i<j}^N u(|\mathbf{r}_i - \mathbf{r}_j|) + 3 \text{ body} + \dots \text{ is the interatomic potential energy } \mathcal{U}(\mathbf{r}^N); \\
V_N^{\text{ext}}(\mathbf{r}^N) &= \sum_{i=1}^N V_{\text{ext}}(\mathbf{r}_i) \text{ is the potential energy arising from the interaction of the} \\
&\quad \text{particles with the external field (e. g. a solute).}
\end{aligned}$$

The grand potential, characteristic thermodynamic state function for the grand canonical ensemble, which depends on the chemical potential μ , the volume V and the temperature T , is linked with the statistical mechanics quantities with the relation:

$$\Omega(\mu, V, T) = -k_B T \ln \Xi \quad (3.2)$$

where

$$\Xi = \sum_{N=0}^{\infty} \frac{e^{\beta \mu N}}{\hbar^{3N} N!} \int d\mathbf{r}^N d\mathbf{p}^N e^{-\beta H_N(\mathbf{r}^N, \mathbf{p}^N)} \quad (3.3)$$

$$= \sum_{N=0}^{\infty} \frac{1}{N!} \int d\mathbf{r}^N e^{-\beta V_N(\mathbf{r}^N)} \left(\prod_{i=1}^N \frac{e^{\beta V_{\text{int}}(\mathbf{r}_i)}}{\Lambda^3} \right) \quad (3.4)$$

is the grand partition function, with $\Lambda = (2\pi\beta\hbar^2/m)^{-\frac{1}{2}}$ the de Broglie thermal wavelength, and

$$V_{\text{int}}(\mathbf{r}_i) = \mu - V_{\text{ext}}(\mathbf{r}_i) \quad (3.5)$$

the intrinsic chemical potential.

$N = \int d\mathbf{r} \rho(\mathbf{r})$ is We can also define the intrinsic free energy:

the number of

particles in
canonical
ensemble, but the

formulae (3.6)

and (3.8) are

also available for

grand canonical

ensemble.

$$\begin{aligned}
\mathcal{F}_{\text{int}} &= F - \int d\mathbf{r} \rho(\mathbf{r}) V_{\text{ext}}(\mathbf{r}) \\
&= \Omega + \mu N - \int d\mathbf{r} \rho(\mathbf{r}) V_{\text{ext}}(\mathbf{r}) \\
&= \Omega + \int d\mathbf{r} \rho(\mathbf{r}) V_{\text{int}}(\mathbf{r})
\end{aligned} \quad (3.6)$$

where

$$\rho(\mathbf{r}) = \langle \varrho(\mathbf{r}) \rangle = \left\langle \sum_{i=1}^N \delta(\mathbf{r} - \mathbf{r}_i) \right\rangle \quad (3.7)$$

is the density profile of instantaneous density $\varrho(\mathbf{r})$ distribution.

The differential form of \mathcal{F}_{int} is

$$\delta\mathcal{F}_{\text{int}} = -S\delta T + \int d\mathbf{r} \delta\rho(\mathbf{r}) V_{\text{int}}(\mathbf{r}) \quad (3.8)$$

with S the entropy.

The internal energy of the solvent contains two contributions, one due to the kinetic energy of the particles, $K_N(\mathbf{p}^N)$, and the other linked to the interaction between particles, $V_N(\mathbf{r}^N)$. When the fluid is a perfect gas, which means $V_N = 0$, it can be easily derived from eq. (7.55-3.5) that \mathcal{F}_{int} has the following expression:

$$\mathcal{F}_{\text{id}} = \beta^{-1} \int d\mathbf{r} \rho(\mathbf{r}) \left(\ln [\Lambda^3 \rho(\mathbf{r})] - 1 \right) \quad (3.9)$$

When interactions are accounted for, the total expression of \mathcal{F}_{int} can be

$$\mathcal{F}_{\text{int}} = \mathcal{F}_{\text{id}} + \mathcal{F}_{\text{exc}} \quad (3.10)$$

and the form of \mathcal{F}_{exc} will be detailed in later sections.

3.2 FUNCTIONAL DERIVATIVES AND DISTRIBUTION FUNCTIONS

The structure of the solvent in the grand canonical ensemble can be characterized by its n -particle density

$$\rho^{(n)}(\mathbf{r}^n) = \frac{1}{\Xi} \sum_{N=n}^{\infty} \frac{1}{(N-n)!} \int d\mathbf{r}^{(N-n)} e^{-\beta V_N(\mathbf{r}^N)} \left(\prod_{i=1}^N \frac{e^{\beta V_{\text{int}}(\mathbf{r}_i)}}{\Lambda^3} \right) \quad (3.11)$$

which means the probability to find n particles in a volume element $d\mathbf{r}^n$. In particular, the probability to find one particle in a volume element is the solvent density $\rho^{(1)}(\mathbf{r}) = \rho(\mathbf{r})$, that

$$\rho^{(1)}(\mathbf{r})d\mathbf{r} = \langle N \rangle \quad (3.12)$$

where $\langle N \rangle$ is the ensemble average of the number of particles, **that is to say the average number of particles at equilibrium.** $\rho^{(n)}(\mathbf{r}^n)$ becomes ρ^n if the system is homogeneous. It can be proven that

$$\frac{\delta\Omega}{\delta V_{\text{int}}(\mathbf{r})} = -\rho^{(1)}(\mathbf{r}) \quad (3.13)$$

The corresponding n -particle distribution function is defined as:

$$g^{(n)}(\mathbf{r}^n) = \frac{\rho^{(n)}(\mathbf{r}^n)}{\prod_{i=1}^n \rho^{(1)}(\mathbf{r}_i)} \quad (3.14)$$

such that $g^{(n)}(\mathbf{r}^n) \rightarrow 1$ when all pairs of particles becomes sufficiently large.

The two-particle pair distribution function (PDF), $g^{(2)}(\mathbf{r}_1, \mathbf{r}_2)$, is one of the most important quantities in the theory of liquids. Its corresponding pair correlation function (PCF) is defined as:

$$h^{(2)}(\mathbf{r}_1, \mathbf{r}_2) = g^{(2)}(\mathbf{r}_1, \mathbf{r}_2) - 1 \quad (3.15)$$

which vanishes when $|\mathbf{r}_1 - \mathbf{r}_2| \rightarrow \infty$.

For any ensemble. If we define the density-density correlation function as:

$$H^{(2)}(\mathbf{r}_1, \mathbf{r}_2) = \rho^{(1)}(\mathbf{r}_1)\rho^{(1)}(\mathbf{r}_2)h^{(2)}(\mathbf{r}_1, \mathbf{r}_2) - \rho^{(1)}(\mathbf{r}_1)\delta(\mathbf{r}_1 - \mathbf{r}_2) \quad (3.16)$$

which means the correlation [1] between the instantaneous fluctuation of particle density from its ensemble average, it can be proven that

$$\frac{\delta\Omega^2}{\delta V_{\text{int}}(\mathbf{r}_1)\delta V_{\text{int}}(\mathbf{r}_2)} = -\beta H^{(2)}(\mathbf{r}_1, \mathbf{r}_2) = -\frac{\delta\rho^{(1)}(\mathbf{r}_1)}{\delta V_{\text{int}}(\mathbf{r}_2)} \quad (3.17)$$

As an analogue, the direct correlation function (DCF) is defined as the derivative of the excess free energy functional $\mathcal{F}_{\text{exc}}[\rho]$:

$$c^{(1)}(\mathbf{r}) = -\frac{\delta(\beta\mathcal{F}_{\text{exc}}[\rho^{(1)}])}{\delta\rho^{(1)}(\mathbf{r})} \quad (3.18)$$

$$c^{(2)}(\mathbf{r}_1, \mathbf{r}_2) = \frac{\delta c^{(1)}(\mathbf{r}_1)}{\delta\rho^{(1)}(\mathbf{r}_2)} = -\frac{\delta^2(\beta\mathcal{F}_{\text{exc}}[\rho^{(1)}])}{\delta\rho^{(1)}(\mathbf{r}_1)\delta\rho^{(1)}(\mathbf{r}_2)} = c^{(2)}(\mathbf{r}_2, \mathbf{r}_1) \quad (3.19)$$

$$c^{(n)}(\mathbf{r}_1, \dots, \mathbf{r}_n) = \frac{\delta c^{(n-1)}(\mathbf{r}_1, \dots, \mathbf{r}_{n-1})}{\delta\rho^{(1)}(\mathbf{r}_n)} \quad (3.20)$$

According to the definition of F_{int} , as well as the expression of δF_{int} in eq. (3.8), we have

$$\begin{aligned} \beta V_{\text{int}}(\mathbf{r}) &= \beta \frac{\delta F_{\text{int}}[\rho^{(1)}]}{\delta\rho^{(1)}(\mathbf{r})} = \beta \frac{\delta F_{\text{id}}[\rho^{(1)}]}{\delta\rho^{(1)}(\mathbf{r})} + \beta \frac{\delta F_{\text{exc}}[\rho^{(1)}]}{\delta\rho^{(1)}(\mathbf{r})} \\ &= \ln [\Lambda^3 \rho^{(1)}(\mathbf{r})] + c^{(1)}(\mathbf{r}) \end{aligned} \quad (3.21)$$

The functional derivative chain rule leads to

$$\begin{aligned} \int d\mathbf{r}_3 \frac{\delta V_{\text{int}}(\mathbf{r}_1)}{\delta\rho^{(1)}(\mathbf{r}_3)} \cdot \frac{\delta\rho^{(1)}(\mathbf{r}_3)}{\delta V_{\text{int}}(\mathbf{r}_2)} &= \int d\mathbf{r}_3 \frac{\delta V_{\text{int}}[\rho^{(1)}(\mathbf{r}_1)]}{\delta\rho^{(1)}(\mathbf{r}_3)} \cdot \beta H^{(2)}(\mathbf{r}_3, \mathbf{r}_2) \\ &= \delta(\mathbf{r}_1 - \mathbf{r}_2) \end{aligned} \quad (3.22)$$

which in addition to the definition of H in eq. (3.16) gives

$$h^{(2)}(\mathbf{r}_1, \mathbf{r}_2) = c^{(2)}(\mathbf{r}_1, \mathbf{r}_2) + \int d\mathbf{r}_3 \left(c^{(2)}(\mathbf{r}_1, \mathbf{r}_3) \rho^{(1)}(\mathbf{r}_3) h^{(2)}(\mathbf{r}_3, \mathbf{r}_2) \right) \quad (3.23)$$

which is called the Ornstein-Zernike (OZ) equation.

3.3 CLASSICAL DENSITY FUNCTIONAL THEORY

The density functional theory is based on two theorems :

1. For a given choice of V_N , T and μ , the intrinsic free energy functional \mathcal{F}_{int} is a unique functional of the equilibrium one-particle density $\rho^{(1)}(\mathbf{r})$, expressed by $\mathcal{F}_{\text{int}}[\rho^{(1)}]$.
2. Let $n(\mathbf{r})$ be some arbitrary one-particle microscopic density, and define the grand-potential $\Omega[n]$ as

$$\Omega[n] = \mathcal{F}_{\text{int}}[n] - \int d\mathbf{r} n(\mathbf{r}) V_{\text{int}}(\mathbf{r}) \quad (3.24)$$

for a fixed external potential V_{ext} (or intrinsic chemical potential V_{int}), then the variational principle states that

$$\Omega[n] \geq \Omega[\rho^{(1)}] \quad (3.25)$$

with the equal sign takes at $n(\mathbf{r}) = \rho^{(1)}(\mathbf{r})$. The differentiation of eq. (3.24) with respect to $n(\mathbf{r})$ gives

$$\frac{\delta \Omega[n]}{\delta n(\mathbf{r})} \Big|_{n=\rho^{(1)}} = \frac{\delta \mathcal{F}_{\text{int}}[n]}{\delta n(\mathbf{r})} \Big|_{n=\rho^{(1)}} - V_{\text{int}}(\mathbf{r}) = 0 \quad (3.26)$$

with the right hand vanishes as eq. (3.8). (it is the reason or the result?)

These theorems build an approach that, for a given choice V_N , T and μ and an external potential V_{ext} , amounts to minimize the total Helmholtz free energy functional of the solvent

$$\mathcal{F}[n(\mathbf{r})] = \mathcal{F}_{\text{int}} + \mathcal{F}_{\text{ext}} = \mathcal{F}_{\text{id}} + \mathcal{F}_{\text{exc}} + \mathcal{F}_{\text{ext}} \quad (3.27)$$

gives the equilibrium density of solvent $\rho^{(1)}(\mathbf{r})$, $\mathcal{F}_{\text{ext}} = \int d\mathbf{r} n(\mathbf{r}) V_{\text{ext}}(\mathbf{r})$ defined as the external functional. Note that eq. (3.26) is then equivalent to

$$\frac{\delta \mathcal{F}[n]}{\delta n(\mathbf{r})} \Big|_{n=\rho^{(1)}} - \mu = 0 \quad (3.28)$$

The external term \mathcal{F}_{exc} of the Helmholtz free energy functional can be given by defining a reference homogeneous single-particle density ρ_0 of bulk solvent at equilibrium, and the corresponding bulk DCF defined in eq. (3.18) via Taylor expansion:

$$\begin{aligned} \mathcal{F}_{\text{exc}} [\rho^{(1)}] &\equiv \mathcal{F}_{\text{exc}} [\rho_0] + \int d\mathbf{r}_1 \left. \frac{\delta \mathcal{F}_{\text{exc}} [\rho^{(1)}]}{\delta \rho^{(1)}(\mathbf{r}_1)} \right|_{\rho^{(1)}=\rho_0} \Delta \rho(\mathbf{r}_1) \\ &\quad + \frac{1}{2} \int d\mathbf{r}_1 d\mathbf{r}_2 \left. \frac{\delta^2 \mathcal{F}_{\text{exc}} [\rho^{(1)}]}{\delta \rho^{(1)}(\mathbf{r}_1) \delta \rho^{(1)}(\mathbf{r}_2)} \right|_{\rho^{(1)}=\rho_0} \Delta \rho(\mathbf{r}_1) \Delta \rho(\mathbf{r}_2) + \mathcal{O}(\Delta \rho^3) \\ &= \mathcal{F}_{\text{exc}} [\rho_0] - \beta^{-1} \int d\mathbf{r}_1 c^{(1)}(\mathbf{r}) \Delta \rho^{(1)}(\mathbf{r}_1) \\ &\quad - \frac{\beta^{-1}}{2} \int d\mathbf{r}_1 d\mathbf{r}_2 c^{(2)}(\mathbf{r}_1, \mathbf{r}_2) \Delta \rho^{(1)}(\mathbf{r}_1) \Delta \rho^{(1)}(\mathbf{r}_2) + \mathcal{O}(\Delta \rho^3) \end{aligned} \quad (3.29)$$

where $\Delta \rho = \rho^{(1)} - \rho_0$ (how to remove the ρ_0 and $c^{(1)}(\mathbf{r})$ term ?)

If the reference fluid is homogeneous, the DCF only depends on the relative distance, i.e. $c^{(2)}(\mathbf{r}_1, \mathbf{r}_2) = c(r_{12})$, so that

$$\mathcal{F}_{\text{exc}} [\rho^{(1)}] \simeq -\frac{\beta^{-1}}{2} \int d\mathbf{r}_1 d\mathbf{r}_2 c(r_{12}) \Delta \rho(\mathbf{r}_1) \Delta \rho(\mathbf{r}_2) \quad (3.31)$$

This was called the homogenous reference fluid (HRF) approximation. The generalization to a molecular, non-spherical solvent for which orientations matter is described in §4.

3.4 INTEGRAL EQUATION THEORY

As opposed to MDFT which aims to find the equilibrium solvent density ρ and the Helmholtz free energy \mathcal{F} , the integral equation theory (IET) aims to find the pair distri-

bution function g and the gradient of energy γ . Both theories give complete information of solvation energy and structure.

IET is about to solve a pair of integral equations of $h^{(2)}(\mathbf{r}_1, \mathbf{r}_2)$ and $c^{(2)}(\mathbf{r}_1, \mathbf{r}_2)$. One of these equations is the OZ equation shown as eq. (3.23). Other is a closure equation, which can be deduced from eq. (3.26), giving the minimum density

$$\rho^{(1)}(\mathbf{r}_1) = \rho_0 \exp \left(-\beta V_{\text{ext}}(\mathbf{r}_1) + \int d\mathbf{r}_2 \Delta \rho^{(1)}(\mathbf{r}_2) c_0^{(2)}(\mathbf{r}_{12}) \right) \quad (3.32)$$

(where is μ ? external potential?) which gives for example, one of the simplest closure equation, the hypernetted-chain HNC approximation. In the uniform case we can deduce

$$g(1, 2) = \exp [-\beta u(1, 2) + h(1, 2) - c(1, 2)] \quad (3.33)$$

(u?)

The general relation to close the OZ equation is

$$g(1, 2) = \exp [-\beta u(1, 2) + h(1, 2) - c(1, 2) + b(1, 2)] \quad (3.34)$$

where the b is the bridge function. Other closures are also possible, such as Percus-Yevick (PY) approximation or mean-spherical approximation (MSA).

3.5 EQUIVALENCE BETWEEN CDFT AND IET FOR A DILUTE SOLUTION SYSTEM

The generalization of the OZ equation in eq. (3.23) to n components can be written as

$$h^{\circ -}(1, 2) = c^{\circ -}(1, 2) + \rho \sum_{\lambda} x_{\lambda} \int c^{\circ -}(2, 3) h^{\circ -}(1, 3) d3 \quad (3.35)$$

where $x_{\nu} = N_{\nu}/N$ is the number concentration of species $\nu \in [1, n]$.

For a two-component homogeneous solute-solvent mixture, where the solute (M) is infinitely diluted in the solvent (S) ($x_S \rightarrow 1$), the coupled OZ relations are written as

$$h_{SS}(1, 2) = c_{SS}(1, 2) + \rho \int h_{SS}(1, 3)c_{SS}(2, 3)d3 \quad (3.36)$$

$$h_{SM}(1, 2) = c_{SM}(1, 2) + \rho \int h_{SS}(1, 3)c_{SM}(2, 3)d3 \quad (3.37)$$

$$h_{MS}(1, 2) = c_{MS}(1, 2) + \rho \int h_{MS}(1, 3)c_{SM}(2, 3)d3 \quad (3.38)$$

$$h_{MM}(1, 2) = c_{MM}(1, 2) + \rho \int h_{MS}(1, 3)c_{SM}(2, 3)d3 \quad (3.39)$$

Eq. (3.36) is the OZ equation for bulk solvent, and eq. (3.39) has no physical interest as $r_{MM} \rightarrow \infty$. Eqs. (3.37) and (3.38) describe the correlations between the solute and solvents, which are equivalent. From eq. (3.38) we can deduce eq. (3.32) which is the analytical result of eq. (3.26) in the MDFT. And in IET, eq. (3.38) is normally used for two-component solution.

APPROACH TO MOLECULAR SOLVENTS

We now consider the case of a non-spherical molecular solvent, like water. The solvent molecules now carry a molecular structure that is described by a collection of distributed atomic interaction sites (LJ and Coulombic). The two theories mentioned in the previous section are formulated in the molecular picture in which each solvent molecule is considered as a rigid body and characterized by its position \mathbf{r} (e. g. the position of center of mass), and its orientation $\boldsymbol{\Omega}$, defined by the three Euler angles $\boldsymbol{\Omega} \equiv (\Theta, \Phi, \Psi)$. In MDFT, the solvent is characterized by an inhomogeneous position and orientation density $\rho(\mathbf{r}, \boldsymbol{\Omega})$. In IET, an angular dependent form of the pair distribution function $g(\mathbf{X}_1, \mathbf{X}_2)$ ($X \equiv (\mathbf{r}, \boldsymbol{\Omega})$) is proposed, and the molecular OZ equation is expanded on rotational invariants. Another way to extend the IET is the RISM [28] which will not be discussed in this thesis.

4.1 MOLECULAR DENSITY FUNCTIONAL THEORY

Here we can just extend some equations written in the last section to the case of an angular-dependent external potential and density. In molecular density functional theory (MDFT), the grand potential density functional corresponding to an inhomogeneous fluid density

$\rho(\mathbf{r}, \boldsymbol{\Omega})$ is given by (changing notations not to confuse the traditional notation of the grand potential with that of the orientations):

$$\Theta[\rho(\mathbf{r}, \boldsymbol{\Omega})] = \Theta[\rho_0] + \mathcal{F}[\rho(\mathbf{r}, \boldsymbol{\Omega})] \quad (4.1)$$

where $\Theta[\rho_0]$ is the correspondent reference bulk fluid grand potential, and ρ is the fluid density function depending now on 5 or 6 variables, 3 coordinates of the positions, and 2 or 3 for the angular part depending on whether the molecule is linear or not. The homogeneous density is equal to $n_0/4\pi$ for a linear solvent, and $n_0/8\pi^2$ for the general case.

According to the variation principle described above, the equilibrium density can be found by minimizing the free energy functional $\mathcal{F}[\rho]$ regarding to $\rho(\mathbf{r}, \boldsymbol{\Omega})$:

$$\frac{\delta \mathcal{F}[\rho]}{\delta \rho(\mathbf{r}, \boldsymbol{\Omega})} \Big|_{\rho=\rho_0} = 0 \quad (4.2)$$

This functional is defined as a sum of functional contributions:

$$\mathcal{F}[\rho] = \mathcal{F}_{\text{id}}[\rho] + \mathcal{F}_{\text{ext}}[\rho] + \mathcal{F}_{\text{exc}}[\rho] \quad (4.3)$$

4.1.1 The ideal term

The ideal term $\mathcal{F}_{\text{id}}[\rho]$ is deduced from the particle interaction-free condition:

$$\mathcal{F}_{\text{id}}[\rho] = \beta^{-1} \int d\mathbf{r}d\boldsymbol{\Omega} \left[\ln \left(\frac{\mathbf{ae}(\mathbf{r}, \boldsymbol{\Omega})}{\mathbf{ae}_0} \right) - \rho(\mathbf{r}, \boldsymbol{\Omega}) + \rho_0 \right] \quad (4.4)$$

where ρ_0 is the reference bulk density of the pure solvent. The differentiation of $\mathcal{F}_{\text{id}}[\rho]$ will be used for the minimization, discussed later, which has form

$$\frac{\delta \mathcal{F}_{\text{id}}[\rho]}{\delta \rho(\mathbf{r}, \boldsymbol{\Omega})} = \beta^{-1} \ln \left(\frac{\rho(\mathbf{r}, \boldsymbol{\Omega})}{\rho_0} \right) \quad (4.5)$$

4.1.2 The external term

The solute, like the solvent, is described in microscopic detail by a molecular non-polarizable force field involving atomic Lennard-Jones and partial charge parameters, creating at each point in space an external potential $V_{\text{ext}}(\mathbf{r}, \boldsymbol{\Omega})$, containing two components:

$$V_{\text{ext}}(\mathbf{r}, \boldsymbol{\Omega}) = V_{\text{LJ}}(\mathbf{r}) + V_{\text{coul}}(\mathbf{r}, \boldsymbol{\Omega}) \quad (4.6)$$

The external potential term calculates the contribution of V_{ext} :

$$\mathcal{F}_{\text{ext}}[\rho] = \int d\mathbf{r} d\boldsymbol{\Omega} V_{\text{ext}}(\mathbf{r}, \boldsymbol{\Omega}) \rho(\mathbf{r}, \boldsymbol{\Omega}) \quad (4.7)$$

The Lennard-Jones potential is given by

$$V_{\text{LJ}}(\mathbf{r}) = \sum_u \sum_v 4\epsilon_{uv} \left[\left(\frac{\sigma_{uv}}{r_{uv}} \right)^{12} - \left(\frac{\sigma_{uv}}{r_{uv}} \right)^6 \right] \quad (4.8)$$

where u stands for solute, v stands for solvent, $\epsilon_{uv} = \sqrt{\epsilon_u \epsilon_v}$ and $\sigma_{uv} = (\sigma_u + \sigma_v)$ are the geometric and arithmetic average Lennard-Jones parameters between solute and solvent, according to the Lorentz-Berthelot mixing rules. r_{ij} is the norm of relative site-site vector

$$\mathbf{r}_{uv} = \mathbf{r} + \mathbf{R}(\boldsymbol{\Omega}) \mathbf{s}_v - \mathbf{r}_u \quad (4.9)$$

where \mathbf{r}_u and \mathbf{s}_j are the coordinates of solute/solvent molecules in the molecular frame, and $\mathbf{R}(\Omega)$ is the rotation matrix of the Euler angles Ω .

In cases where the solvent site wears only one LJ centre, eq. (4.9) reduces to

$$\mathbf{r}_{uv} = \mathbf{r} - \mathbf{r}_u \quad (4.10)$$

which is actually what we use in the code as the solvent is SPC/E water.

The Coulomb interaction is calculated by [solving the Poisson equation \[44\]](#). The charge density of the solute is projected onto a space grid,

$$\rho_q(\mathbf{r}) = \sum_u q_{ijk} \quad (4.11)$$

where q_{ijk} is the charge on the space grid distributed by its nearby point charge as shown in figure 4.1 (a).

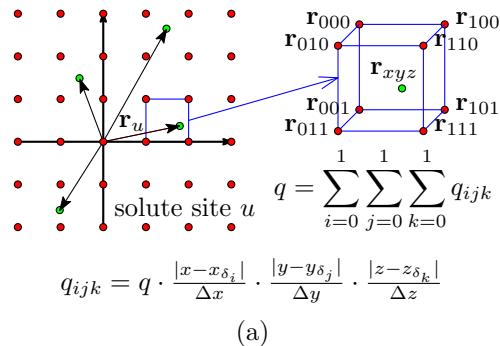


Figure 4.1: Charge density projected onto grids. (a) Solute. (b) Solvent.

$$V_{\text{coul}}(\mathbf{r}, \Omega) = \sum_v q_v V_q(\mathbf{r}_u) \quad (4.12)$$

where $V_q(\mathbf{r})$ is the electrostatic potential created by the charge distribution $\rho_q(\mathbf{r})$. It can be computed using a periodic Poisson Solver. The Poisson equation reads

$$\nabla^2 V_q(\mathbf{r}) = -\frac{\rho_q(\mathbf{r})}{\epsilon_0} \quad (4.13)$$

or in Fourier space

$$\hat{V}_q(\mathbf{k}) = \frac{\hat{\rho}_q(\mathbf{k})}{\varepsilon_0 k^2} \quad (4.14)$$

where $\hat{V}_q(\mathbf{k})$, $\hat{\rho}_q(\mathbf{k})$ are the Fourier transform of $V_q(\mathbf{r})$, $\rho_q(\mathbf{r})$ respectively. $\rho_q(\mathbf{r})$ is transformed forward, $\hat{V}_q(\mathbf{k})$ is computed and transformed backward. $V_q(\mathbf{r}_u)$ is then obtained by interpolation.

4.1.3 The excess term

The two terms $\mathcal{F}_{\text{id}}[\rho]$ and $\mathcal{F}_{\text{ext}}[\rho]$ are physically exact, while the excess term $\mathcal{F}_{\text{exc}}[\rho]$ depends on the exact correlation function, which is a priori unknown. As in the previous section, we invoke here the HRF approximation which amounts to a second-order Taylor expansion around the homogeneous fluid at density ρ_0 :

$$\mathcal{F}_{\text{exc}}[\rho] = \frac{k_B T}{2} \int d\mathbf{r}_1 d\Omega \gamma(\mathbf{r}_1, \Omega) \rho(\mathbf{r}_1, \Omega) \quad (4.15)$$

where γ is the normalized gradient of the excess functional:

$$\gamma(\mathbf{r}_1, \Omega_1) = \frac{\delta \beta F_{\text{exc}}}{\delta \rho} = - \int d\mathbf{r}_2 d\Omega_2 \Delta \rho(\mathbf{r}_2, \Omega_2) c(\mathbf{r}_{12}, \Omega_1, \Omega_2) \quad (4.16)$$

To evaluate the gradient γ for each (\mathbf{r}, Ω) , $N \equiv N_{\mathbf{r}} N_{\Omega}$ function evaluations (FE) are required. The total number of FE is thus $N^2 = O(N^2)$, which, with typically $N_{\mathbf{r}} = 64^3$ and $N_{\Omega} = 50 \sim 100$, is far too costly for current computing technology. For this reason, Fourier transform is used to treat the spatial convolution in eq. (6.2).

A convolution

$$h(x_1) \equiv f(x_2) \otimes g(x_2) \equiv \int_a^b f(x_2) g(x_1 - x_2) dx_2 \quad (4.17)$$

has the property that

$$\mathfrak{F}[h(x_1)] = \mathfrak{F}[f(x_2)]\mathfrak{F}[g(x_2)] \quad (4.18)$$

\mathfrak{F} being the Fourier transform operation. As $\mathbf{r}_{12} = \mathbf{r}_1 - \mathbf{r}_2$, eq. (6.2) is a 3D convolution,

which leads to

$$\hat{\gamma}(\mathbf{k}, \Omega_1) = -\beta^{-1} \int d\Omega_2 \Delta\hat{\rho}(\mathbf{k}, \Omega_2) \hat{c}(\mathbf{k}, \Omega_1, \Omega_2) \quad (4.19)$$

Thus the integral $\int d\mathbf{r}_2$ in eq. (6.2) is transformed into a simple product in eq. (6.3). To get $\hat{\gamma}(\mathbf{k}, \Omega_1)$ with given $\Delta\hat{\rho}(\mathbf{k}, \Omega_2)$, only $N_r N_\Omega^2$ FE are needed. To this computational cost should be added the transform from $\Delta\rho(\mathbf{r}, \Omega)$ to $\Delta\hat{\rho}(\mathbf{k}, \Omega)$ and the backward transform from $\hat{\gamma}(\mathbf{k}, \Omega)$ to $\gamma(\mathbf{r}, \Omega)$ which are both of order $N_\Omega \cdot O(N_r \log_2 N_r)$ due to the properties of Fast Fourier Transforms (FFT). The total number of FE is thus reduced from quadratic complexity $O(N_r^2 N_\Omega^2)$ to $N_r N_\Omega^2 + 2N_\Omega \cdot O(N_r \log_2 N_r) = O(N_r \log_2 N_r N_\Omega^2)$. As the total number of spatial grid N_r is of magnitude $10^5 \sim 10^6$, this procedure, which is mathematically equivalent to the direct evaluation (6.2), offers a great advantage in terms of computational efficiency (figure A.1 in section 1).

Once $\gamma(\mathbf{r}, \Omega)$ is obtained by inverse Fourier transform of $\hat{\gamma}(\mathbf{k}, \Omega)$, the excess functional can be calculated as eq. (6.1).

The input $\hat{c}(\mathbf{k}, \Omega_1, \Omega_2)$ is an angular-dependent DCF of the homogeneous solvent which depends on the relative position of two molecules r_{12} and their orientations (two in the initial code [68], and three developed in this thesis). This quantity is provided with high precision and in different representations by Luc Belloni, using a mixture of Monte-Carlo simulations and inversion of the angular-dependent MOZ equation within a rotational invariant expansion [54]. A detailed comparison of some DCFs is in appendix [ref].

4.2 ANGULAR DEPENDENT INTEGRAL EQUATION THEORY

To extend the IET formalism to molecular cases, Blum [4, 5] proposed an angular-dependent form of the pair distribution function, which is then used in the IET formalism. Fries & Patey [17] proposed the numerical solution of the full HNC theory using angle-dependent pair potentials. The description below is based on these articles.

4.2.1 Expansion of angular-dependent distribution functions

The angular-dependent PDF is defined as

$$g(\mathbf{X}_1, \mathbf{X}_2) \equiv g(\mathbf{r}_1, \mathbf{r}_2, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) \quad (4.20)$$

which possesses the translational invariance if the fluid is homogeneous:

$$g(\mathbf{X}_1, \mathbf{X}_2) = g(\mathbf{r}_{12}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) \quad (4.21)$$

and the rotational invariance (**attention indices**):

$$g(\mathbf{X}_1, \mathbf{X}_2) = \sum_{m,n,l=0}^{\infty} \sum_{|\mu,\mu'| \leq m, |\nu,\nu'| \leq n, |\lambda| \leq l} g_{\mu,\mu' \nu,\nu' \lambda}^{mn} (\|\mathbf{r}_{12}\|) R_{\mu\mu'}^m(\boldsymbol{\Omega}_1) R_{\nu\nu'}^n(\boldsymbol{\Omega}_2) R_{\lambda 0}^l(\hat{\mathbf{r}}_{12}) \quad (4.22)$$

The suggestion of Wigner gives a set of rotation invariant:

$$\Phi_{\mu'\nu'}^{mn}(\hat{\mathbf{r}}_{12}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) = f^{mn} \sum_{\mu\nu\lambda} \begin{pmatrix} m & n & l \\ \mu & \nu & \lambda \end{pmatrix} R_{\mu\mu'}^m(\boldsymbol{\Omega}_1) R_{\nu\nu'}^n(\boldsymbol{\Omega}_2) R_{\lambda 0}^l(\hat{\mathbf{r}}_{12}) \quad (4.23)$$

such that

$$g(\mathbf{X}_1, \mathbf{X}_2) = \sum_{mnl} \sum_{\mu'\nu'} g_{\mu'\nu'}^{mnl}(\|\mathbf{r}_{12}\|) \Phi_{\mu'\nu'}^{mnl}(\hat{\mathbf{r}}_{12}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) \quad (4.24)$$

and

$$g_{\mu'\nu'}^{mnl} = \sum_{\mu\nu\lambda} \begin{pmatrix} m & n & l \\ \mu & \nu & \lambda \end{pmatrix} g_{\mu,\mu'\nu'\lambda}^{mnl} \quad (4.25)$$

4.2.2 Blum's reduction of molecular OZ equation

The molecular Ornstein-Zernike MOZ equation is defined as

$$h(\mathbf{X}_1, \mathbf{X}_2) - c(\mathbf{X}_1, \mathbf{X}_2) = \frac{\rho}{8\pi^2} \int d\mathbf{X}_3 h(\mathbf{X}_1, \mathbf{X}_3) c(\mathbf{X}_3, \mathbf{X}_2) \quad (4.26)$$

which can be both expanded as

$$h(\mathbf{X}_1, \mathbf{X}_2) = \sum_{mnl} \sum_{\mu'\nu'} h_{\mu'\nu'}^{mnl}(\|\mathbf{r}_{12}\|) \Phi_{\mu'\nu'}^{mnl}(\hat{\mathbf{r}}_{12}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) \quad (4.27)$$

$$c(\mathbf{X}_1, \mathbf{X}_2) = \sum_{mnl} \sum_{\mu'\nu'} c_{\mu'\nu'}^{mnl}(\|\mathbf{r}_{12}\|) \Phi_{\mu'\nu'}^{mnl}(\hat{\mathbf{r}}_{12}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) \quad (4.28)$$

$$\eta(\mathbf{X}_1, \mathbf{X}_2) = h(\mathbf{X}_1, \mathbf{X}_2) - c(\mathbf{X}_1, \mathbf{X}_2) = \sum_{mnl} \sum_{\mu'\nu'} \eta_{\mu'\nu'}^{mnl}(\|\mathbf{r}_{12}\|) \Phi_{\mu'\nu'}^{mnl}(\hat{\mathbf{r}}_{12}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) \quad (4.29)$$

As an analog to the FFT the fast Hankel transform can deal these rotational invariant projections into k -space, such that

$$\hat{c}_{\mu\nu}^{mnl}(k) = 4\pi i^l \int dr r^2 j_l(kr) c_{\mu\nu}^{mnl}(r) \quad (4.30)$$

$$\hat{\eta}_{\mu\nu}^{mnl}(k) = 4\pi i^l \int dr r^2 j_l(kr) \eta_{\mu\nu}^{mnl}(r) \quad (4.31)$$

where $j_l(kr)$ are the spherical Bessel functions.

The χ -transform defined by Blum gives

$$\hat{c}_{\mu\nu,\chi}^{mn}(k) = \sum_{l=|m-n|}^{m+n} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} \hat{c}_{\mu\nu}^{ml}(k) \quad (4.32)$$

$$\hat{\eta}'_{\mu\nu,\chi}^{mn}(k) = \sum_{l=|m-n|}^{m+n} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} \hat{\eta}_{\mu\nu}^{ml}(k) \quad (4.33)$$

If $\Phi_{\mu'\nu'}^{ml}(\mathbf{r}_{12}, \Omega_1, \Omega_2)$ is defined with the factor $f^{ml} = [(2m+1)(2n+1)]^{1/2}$, the MOZ equation can be reduced by Blum's reduction such that

$$\hat{\eta}'_{\mu\nu,\chi}^{mn}(k) = \rho \sum_{n_1} \sum_{\nu_1=-n_1}^{n_1} (-)^{\chi+\nu_1} [\hat{\eta}'_{\mu\nu_1,\chi}^{mn_1}(k) + \hat{c}_{\mu\nu_1,\chi}^{mn_1}(k)] \hat{c}_{\nu_1\nu,\chi}^{n_1n}(k) \quad (4.34)$$

which reduces the calculation of $(\mathbf{X}_1, \mathbf{X}_3)$ for $(\mathbf{X}_3, \mathbf{X}_2)$ to a sum of n_1, ν_1 .

(The part of HNC has non need in this thesis.)

CODE MDFT

The code MDFT upon which all the development during this thesis is based is a Fortran 95 sequential code developed by Maximilien Levesque, Daniel Borgis *et al.* [ref], which implement the MDFT theory. It reads the force field (pair potential) $u(\mathbf{r}, \Omega)$ describing the solute and the solvent as input, as well as necessary parameters like the temperature T , number density of solvent n_0 , etc. It minimizes the functional and gives the equilibrium density $\rho(\mathbf{r}, \Omega)$, then computes output properties. The main structure of the code is shown in figure 5.1.

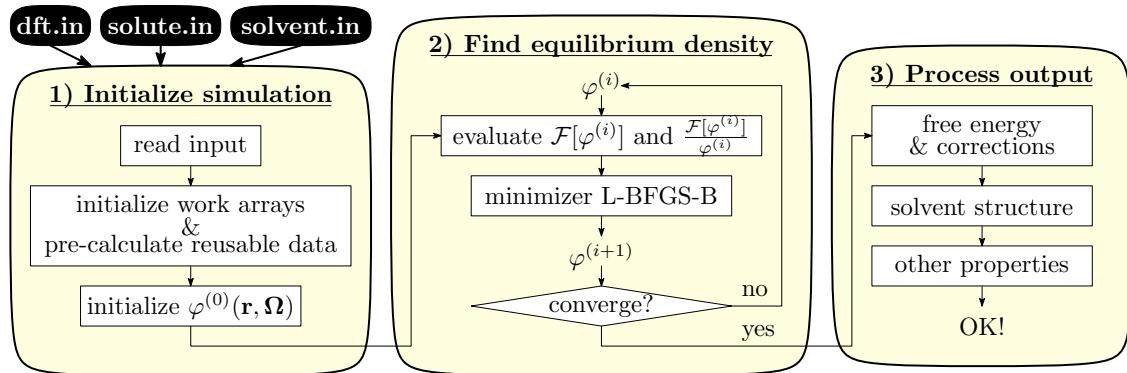


Figure 5.1: Main structure of code MDFT

5.1 SUPERCELL DISCRETIZATION

$L_x \times L_y \times L_z \text{ \AA}^3$ space is discretized on a regular grid of $n_{fft_1} \times n_{fft_2} \times n_{fft_3}$ nodes. Angular grid is discretized with Lebedev (L) or Gauss-Legendre (GL) quadrature for $\Omega \equiv (\Theta, \Phi)$, $\Theta \in [0, \pi]$, $\Phi \in [0, 2\pi]$ and regular quadrature for $\Psi \in [0, \pi]$ as we used the code mainly

for water. The number of each angular dimension is linked to the order of quadrature, m_{\max} , which is discussed mainly in the chapter of theory.

The solute center is at $\left(\frac{L_x}{2}, \frac{L_y}{2}, \frac{L_z}{2}\right)$ of the box, characterized by \mathbf{r}_T . If the internal coordinates of solute \mathbf{r}_M , the solute coordinates in the box $\mathbf{r} = \mathbf{r}_M + \mathbf{r}_T$.

5.2 MINIMIZER L-BFGS-B

The minimizer adopted by MDFT is the L-BFGS-B [6, 69] package version 3.0 written in Fortran 77, implementing the limited-memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm with constraints of the form $l \leq x \leq u$ to the variable x . During the evaluation of the initial code which uses L-BFGS, the constraint function is not used.

The functional $\mathcal{F}[x_i]$ and the gradient of functional $\nabla \mathcal{F}[x_i] = \frac{\delta \mathcal{F}}{\delta x}(x_i)$ are required by L-BFGS to minimize the functional. It saves the variables x_i and gradient of the past m iterations, which is a memory eater.

The functional in MDFT to be minimized is eq. (4.3), and its gradient is

$$\frac{\delta \mathcal{F}[\rho]}{\delta \rho(\mathbf{r}, \boldsymbol{\Omega})} = \beta^{-1} \ln \left(\frac{\rho(\mathbf{r}, \boldsymbol{\Omega})}{\rho_0} \right) + V_{\text{ext}}(\mathbf{r}, \boldsymbol{\Omega}) + V_{\text{exc}}(\mathbf{r}, \boldsymbol{\Omega}) \quad (5.1)$$

where ρ_0 is the angular density of bulk solvent,

$$\rho_0 = \begin{cases} n_0 & \text{if atomic, } \boldsymbol{\Omega} \equiv 1 \\ n_0/4\pi & \text{if linear, } \boldsymbol{\Omega} \equiv (\Theta, \Phi) \\ n_0/8\pi^2 & \text{if non-linear, } \boldsymbol{\Omega} \equiv (\Theta, \Phi, \Psi) \end{cases} \quad (5.2)$$

5.3 TREATMENT TO AVOID UNPHYSICAL DENSITY

During minimization, the density variable $\rho(\mathbf{r}, \Omega)$ can have unphysical negative numbers, which also cause the divergence of the minimization. To avoid this phenomenon, a normalized $\varphi(\mathbf{r}, \Omega)$ is used as the variable during the minimization in place of $\rho(\mathbf{r}, \Omega)$, so that:

$$\rho(\mathbf{r}, \Omega) = \rho_0 \varphi^2(\mathbf{r}, \Omega) \quad (5.3)$$

According to the definition (5.3), we see:

$$\frac{\delta \rho(\mathbf{r}, \Omega)}{\delta \varphi} = 2\rho_0 \varphi(\mathbf{r}, \Omega) \quad (5.4)$$

Therefore the gradient to feed the L-BFGS minimizer is:

$$\frac{\delta \mathcal{F}}{\delta \varphi} = \frac{\delta \mathcal{F}}{\delta \rho} \cdot \frac{\delta \rho}{\delta \varphi} = 2\rho_0 \varphi(\mathbf{r}, \Omega) \cdot [\beta^{-1} \ln \varphi^2 + V_{\text{ext}} + V_{\text{exc}}] \quad (5.5)$$

Chapter II

**THEORY: HRF APPROXIMATION, FOR
MOLECULAR SOLVENT**

This chapter presents a complete theory of the \mathcal{F}_{exc} evaluation under HRF approximation:

$$\mathcal{F}_{\text{exc}} = -\frac{\beta^{-1}}{2} \int d\mathbf{r}_1 d\mathbf{r}_2 d\Omega_1 d\Omega_2 \Delta\rho(\mathbf{r}_1, \Omega_1) \Delta\rho(\mathbf{r}_2, \Omega_2) c(r_{12}, \Omega_1, \Omega_2)$$

which is based on previous work of Zhao et al. [68], where the HRF approximation has been applied to linear molecules ($\Omega \equiv (\Theta, \Phi)$). In this thesis, the method is generalized to molecular solvent, using 3 Euler angles, i. e. $\Omega \equiv (\Theta, \Phi, \Psi)$, where the computing cost of the original algorithm is no longer reasonable. Further approximation is therefore made, where the density variable $\rho(\mathbf{r}, \Omega)$ is expanded on generalized spherical harmonics. Theoretically, this approximation gives little loss of accuracy, but makes a great advantage in computing time and memory requirement. The prove is shown in the implementation part.

Section 6 describes the FFT treatment for the spatial convolution in the gradient γ of the excess functional \mathcal{F}_{exc} , which reduces the algorithm complexity from $O(N^2)$ to $O(N \log_2 N)$. A DCF directly issue of Monte Carlo simulation and solution HNC is used, in both intermolecular and projection form. To use the intermolecular form, the matrix which does the transform from laboratory to intermolecular coordinates system is generalized to molecular case compared to previous work, then interpolation of zero and first order is involved. To use the projection form, DCF is reconstructed with all projections. For order of projections $n_{\text{max}} = 1$, the formula of each projections are written explicitly.

Section 7 presents the treatment of angular convolution. As IEM and MDFT have mathematical equivalence, an algorithm inspired by the work of Fries [17] and Blum [4, 5] for IEM is built for MDFT. In this algorithm, the density variable $\rho(\mathbf{r}, \Omega)$ is expanded on generalized spherical harmonics, then rotated onto intermolecular frame. It is shown that in this form, the OZ equation is largely simplified.

The solvent properties involved in this thesis are presented in the next two chapters. Section 8 presents some thermodynamic quantities, including the solvation free energy and its corrections, ... and section 9 gives some forms of structure that can be preformed such as the radical distribution function

REDUCTION OF SPATIAL CONVOLUTION BY FFT

As presented in section 5, to complete the minimization process of MDFT, we need to evaluate the excess functional as well as its gradient:

$$\mathcal{F}_{\text{exc}} = -\frac{\beta^{-1}}{2} \int d\mathbf{r}_1 d\mathbf{r}_2 d\Omega_1 d\Omega_2 \Delta\rho(\mathbf{r}_1, \Omega_1) \Delta\rho(\mathbf{r}_2, \Omega_2) c(\mathbf{r}_{12}, \Omega_1, \Omega_2) \quad (6.1)$$

$$\gamma(\mathbf{r}_1, \Omega_1) = -\beta^{-1} \int d\mathbf{r}_2 d\Omega_2 \Delta\rho(\mathbf{r}_2, \Omega_2) c(\mathbf{r}_{12}, \Omega_1, \Omega_2) \quad (6.2)$$

As discussed in §4.1.3 eq. (6.2) is a 3D convolution, which leads to

$$\hat{\gamma}(\mathbf{k}, \Omega_1) = -\beta^{-1} \int d\Omega_2 \Delta\hat{\rho}(\mathbf{k}, \Omega_2) \hat{c}(\mathbf{k}, \Omega_1, \Omega_2) \quad (6.3)$$

such that the integral $\int d\mathbf{r}_2$ in eq. (6.2) is transformed into a simple product in eq. (6.3), giving a great advantage in terms of computational efficiency for huge spatial grid.

It should be pointed out that the direct correlation function (DCF), $\hat{c}(\mathbf{k}, \Omega_1, \Omega_2)$, used as an input data in eq. (6.3) is very memory-costly. In the previous work, the DCF has been stocked in the intermolecular form $\hat{c}(k, \omega_1, \omega_2)$ to profit an economy of memory, where $(\omega_1, \omega_2) \equiv (\cos\theta_1, \cos\theta_2, \phi_{12})$, and the correspondence of (Ω_1, Ω_2) to (ω_1, ω_2) is calculated directly in the code. However, in the case of full 3 Euler angles $\Omega \equiv (\Theta, \Phi, \Psi)$, neither the storage of $\hat{c}(\mathbf{k}, \Omega_1, \Omega_2)$ which is definitively impossible, nor the direct calculation of correspondence (Ω_1, Ω_2) to (ω_1, ω_2) due to the increased complexity that makes it too costly, can be regarded as a possible solution. For instance, with

a normal setting with 64^3 spatial grid and a Lebedev quadrature of order 2 (14 angles for Θ and Φ), and 3 Ψ -angles, even if the DCF is stocked in simple precision, it takes $64^3 \times 42^2 \times 4$ bytes = 1.76GB, and for a Lebedev quadrature of order 5 and correspondingly 5 Ψ -angles, it takes $64^3 \times 250^2 \times 4$ bytes = 65.5GB. As a normal PC has only 4 to 16 GB of RAM, it can cause a memory leak.

Two strategies are developed treat this 3 Euler angles $\Omega \equiv (\Theta, \Phi, \Psi)$ case. The first one is a direct extension of the previous work [[gendre_CPL_2009](#), [zhao_JCP_2011](#), [borgis_JPC_2012](#)], which reduces the storage using intermolecular DCF, with the more complicate angle correspondence calculation pre-tabulated in the beginning of the implementation, the other calculates the DCF directly from rotational invariant projections, as shown below.

6.1 USING INTERMOLECULAR DCF

The first strategy aims to work in the so-called intermolecular frame described in fig. 6.1 for which, in \mathbf{r} -space, the z axis is oriented along the vector $\mathbf{r}_{12} = \mathbf{r}_2 - \mathbf{r}_1$, or in \mathbf{k} -space is oriented along the vector \mathbf{k} . In the latter case, an orientation $\Omega \equiv (\Theta, \Phi, \Psi)$ in laboratory frame become $\omega \equiv (\theta, \phi, \psi)$ in intermolecular frame.

To store the DCF in the intermolecular coordinates system, it can be defined as $\hat{c}(k, \omega_1, \omega_2)$, where $(\omega_1, \omega_2) \equiv (\cos \theta_1, \cos \theta_2, \phi, \psi_1, \psi_2)$ and k is always oriented along the z axis (figure 6.1) such that only 6 variables are needed instead of 9 for $\hat{c}(\mathbf{k}, \Omega_1, \Omega_2)$, and the storage is considerably reduced. The transform from $\hat{c}(k, \omega_1, \omega_2)$ to $\hat{c}(\mathbf{k}, \Omega_1, \Omega_2)$ relies on the correspondence $\omega(\mathbf{k}, \Omega) \equiv (\cos \theta, \phi, \psi)$, which is pre-calculated as a table of data.

Finding ω from Ω amounts to defining the correspondence between the rotation matrices of the two coordinate systems.

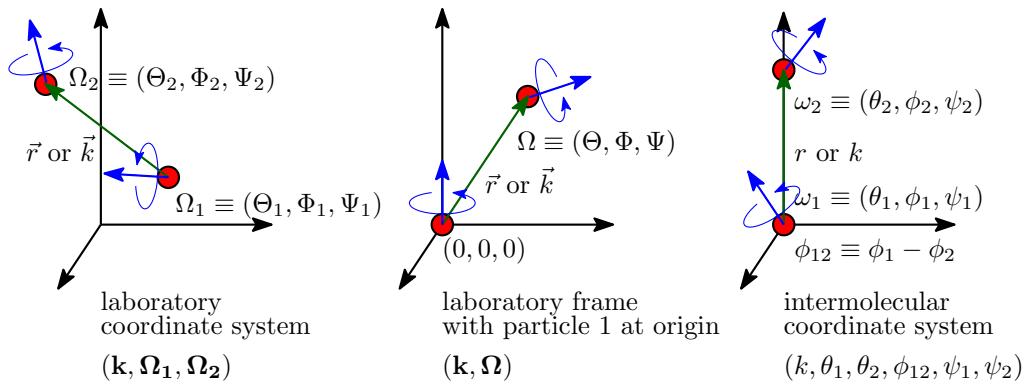


Figure 6.1: Molecules 1 and 2 in different coordinate systems

The rotation matrix $\hat{\mathbf{R}}_\Omega$ rotates the solvent molecule from \mathbf{I} to its orientation $\hat{\mathbf{R}}_\Omega$:

$$\hat{\mathbf{R}}_\Omega \mathbf{I} = \hat{\mathbf{R}}_\Omega \quad (6.4)$$

It can be expressed by 3 rotation operations $\hat{\mathbf{R}}_\Phi$, $\hat{\mathbf{R}}_\Theta$, and $\hat{\mathbf{R}}_\Psi$ which rotate along $z - y - z$ axes (the same convention as defined in Messiah [48] and Gray-Gubbins [22]):

$$\begin{aligned} \hat{\mathbf{R}}_\Omega &= \begin{bmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{bmatrix} & (6.5) \\ &= \begin{bmatrix} \cos \Phi & -\sin \Phi & 0 \\ \sin \Phi & \cos \Phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \Theta & 0 & \sin \Theta \\ 0 & 1 & 0 \\ -\sin \Theta & 0 & \cos \Theta \end{bmatrix} \begin{bmatrix} \cos \Psi & -\sin \Psi & 0 \\ \sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \Phi \cos \Theta \cos \Psi - \sin \Phi \sin \Psi & -\cos \Phi \cos \Theta \sin \Psi - \sin \Phi \cos \Psi & \cos \Phi \sin \Theta \\ \sin \Phi \cos \Theta \cos \Psi + \cos \Phi \sin \Psi & -\sin \Phi \cos \Theta \sin \Psi + \cos \Phi \cos \Psi & \sin \Phi \sin \Theta \\ -\sin \Theta \cos \Psi & \sin \Theta \sin \Psi & \cos \Theta \end{bmatrix} \end{aligned}$$

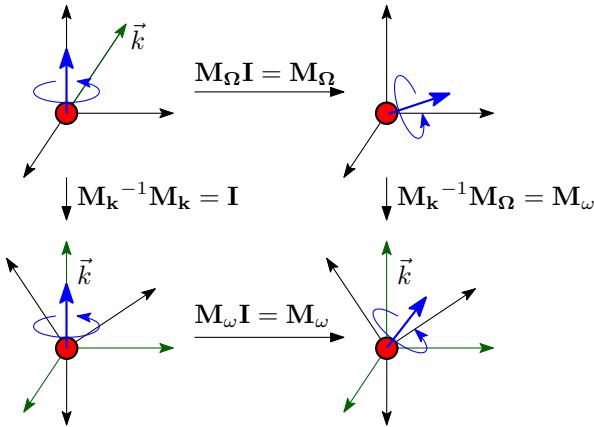


Figure 6.2: Rotation matrices

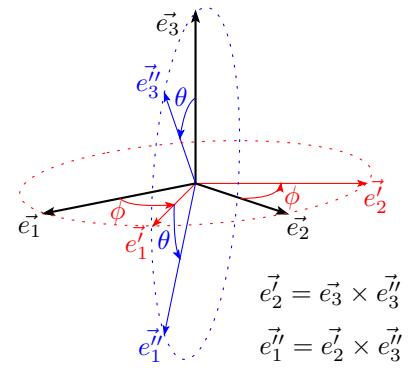


Figure 6.3: Rotation to k-frame

As shown in fig. 6.2, the rotation matrix for transforming the DCF from the intermolecular coordinates to laboratory coordinates $\hat{\mathbf{R}}_\omega$ can be written as:

$$\hat{\mathbf{R}}_\omega = \hat{\mathbf{R}}_{\mathbf{k}}^{-1} \hat{\mathbf{R}}_\Omega \quad (6.6)$$

with the rotation matrix related to \mathbf{k} vector:

$$\hat{\mathbf{R}}_{\mathbf{k}}^{-1} = \begin{bmatrix} \cos \theta_k \cos \phi_k & \cos \theta_k \sin \phi_k & -\sin \theta_k \\ -\sin \phi_k & \cos \phi_k & 0 \\ \sin \theta_k \cos \phi_k & \sin \theta_k \sin \phi_k & \cos \theta_k \end{bmatrix} \quad (6.7)$$

Here we fix $\psi_k = 0$. As θ_k and ϕ_k are calculated from Cartesian coordinates (k_x, k_y, k_z) , in the extreme cases where we cannot define θ_k (for $\|\mathbf{k}\| = 0$) and ϕ_k (for $k_x^2 + k_y^2 = 0$), we can arbitrarily fix those angles to zero.

A faster way to find the rotation matrix of \mathbf{k} , avoiding the evaluation of trigonometric functions, is shown in figure 6.3, where the matrix can be calculated by the cross products of basis vectors from z axis and \mathbf{k} vector ($\mathbf{k} = \mathbf{e}_3''$):

$$\begin{bmatrix} \mathbf{e}_1'' & \mathbf{e}_2' & \mathbf{e}_3'' \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \hat{\mathbf{R}}_{\mathbf{k}} = \hat{\mathbf{R}}_{\mathbf{k}} \quad (6.8)$$

The two ways to calculate \mathbf{k} differ only in the case of $\hat{\mathbf{k}} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$, where one is the inverse of the other. This is due to the different definitions of ϕ_k (0 or π when \vec{k}_z superposes with \vec{k}_z) in the two cases. Tests have shown that it has no influence on the final result of the excess functional evaluation.

Therefore, the elements of $\hat{\mathbf{R}}_\omega$ can be calculated according to eq. (6.6):

$$\begin{aligned} \hat{\mathbf{R}}_\omega &= \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix} \\ &= \begin{bmatrix} \cos \phi \cos \theta \cos \psi - \sin \phi \sin \psi & -\cos \phi \cos \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \sin \theta \\ \sin \phi \cos \theta \cos \psi + \cos \phi \sin \psi & -\sin \phi \cos \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \sin \theta \\ -\sin \theta \cos \psi & \sin \theta \sin \psi & \cos \theta \end{bmatrix} \end{aligned} \quad (6.9)$$

The angles ω are thus found as:

$$\begin{aligned} \cos \theta &= w_z \\ \phi &= \arccos(w_x / (w_x^2 + w_y^2)^{\frac{1}{2}}) \\ \psi &= \arccos(-u_z / (u_z^2 + v_z^2)^{\frac{1}{2}}) \end{aligned} \quad (6.10)$$

The resulting angles are between normal intervals, $\cos \theta \in [-1, 1]$, $\phi \in [0, 2\pi]$. As water possesses C_{2v} symmetry, we take $\psi \in [0, \pi]$.

Here the DCF $c(k, \omega_1, \omega_2) \equiv c(k, \cos \theta_1, \cos \theta_2, \phi_{12}, \psi_1, \psi_2)$ is stored in a discrete set of angles for each value of k (typically $(8, 8, 8, 8, 8)$ in the case of water, which uses the symmetries in §D.5.1 to reduce the number of ϕ and ψ by two) such that the correspondence from (Ω_1, Ω_2) to (ω_1, ω_2) usually falls in between angular grid points of the intermolecular grid. An interpolation can be done at different orders: zeroth order interpolation, which directly takes the nearest point, or linear interpolation.

6.1.1 Zero-order interpolation of DCF

At this order, for each possible value of \mathbf{k} and Ω , the corresponding $\cos \theta$ and ψ which relate to a single solvent molecule are stored as an index (single precision integer), which gives the nearest angle in a pre-defined table:

$$i_{\cos \theta} = \lfloor (\cos \theta + 1)(n_{\cos \theta}/2) \rfloor + 1 \quad (6.11)$$

$$i_\psi = \text{mod}(\lfloor \psi(n_\psi/\pi) \rfloor, n_\psi) + 1$$

where $\lfloor f \rfloor$ is the floor function. For the angle ϕ which relate to two solvent molecules, the operation $\phi = \phi_1 - \phi_2$ introduces a double error when integer indices are used, as shown in figure 6.4.

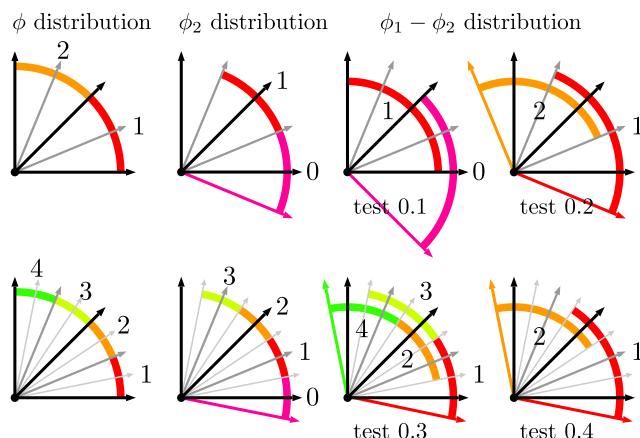


Figure 6.4: $\phi_1 - \phi_2$ distribution: Test 0.1 is the direct subtraction of ϕ established in the same way with θ and ψ , as shown in the top first schema. Test 0.2 tabulates ϕ_2 by taking the nearest point in another manner, as shown in the second schema. In test 0.3-0.4, all ϕ or only ϕ_2 is doubled.

In the actual implementation, as an integer takes 4 bytes and a real takes 8 bytes, there is no profit to tabulate ϕ in integer two times, thus ϕ is stored directly in real.

6.1.2 Linear interpolation of DCF

At this order, $\omega(\mathbf{k}, \Omega)$ is stored in double precision. All angles are stored in real type, and the corresponding DCF is calculated as

$$c(\omega) = w_0 c(\omega_0) + w_1 c(\omega_1) \quad (6.12)$$

where $w_0 = \frac{\omega_1 - \omega}{\omega_1 - \omega_0}$ and $w_1 = \frac{\omega - \omega_0}{\omega_1 - \omega_0}$. Here ω is one of the 5 dimensions in $\tilde{\omega}(\mathbf{k}, \Omega_1, \Omega_2) \equiv (\cos \theta_1, \cos \theta_2, \phi, \psi_1, \psi_2)$, ω_0 and ω_1 are the 2 nearest value points, while other variables are fixed. If we express the weight for each dimension as $w_{n_i}^i$ where $i = 1, 2, 3, 4, 5$ is the i th variable, the total equation with 5 variables is:

$$c(\tilde{\omega}) = \left[\sum_{n_1=0}^1 \sum_{n_2=0}^1 \sum_{n_3=0}^1 \sum_{n_4=0}^1 \sum_{n_5=0}^1 \left(\prod_i^5 w_{n_i}^i c(\tilde{\omega}_{n_1, n_2, n_3, n_4, n_5}) \right) \right] \quad (6.13)$$

These two equations are available for both interpolation and extrapolation, where the latter applies, e.g., for $\cos \theta_1$ and $\cos \theta_2$.

An error evaluation of these two strategies of interpolation is shown in appendix G. Results demonstrate that the linear interpolation scheme is absolutely essential. On the other hand, as seen in eq. (6.13), it is computationally much more expensive than the simple histogram scheme as it requires $2^5 = 32$ times of operations.

6.2 DIRECT CALCULATION OF DCF FROM ROTATIONAL INVARIANT PROJECTIONS

Another strategy to calculate $\hat{c}(\mathbf{k}, \Omega_1, \Omega_2)$ is to use the DCF expressed in terms of rotational invariant projections, which takes far less memory than in the intermolecular form thanks to their angular independence and symmetric properties.

6.2.1 Using projections in form of $\hat{c}_{\mu\nu}^{mnl}(k)$

As described by Blum [4, 5], $\hat{c}(\mathbf{k}, \Omega_1, \Omega_2)$ can be expanded as

$$\hat{c}(\mathbf{k}, \Omega_1, \Omega_2) = \sum_{mnl\mu\nu} \hat{c}_{\mu\nu}^{mnl}(k) \Phi_{\mu\nu}^{mnl}(\hat{\mathbf{k}}, \Omega_1, \Omega_2) \quad (6.14)$$

where $\Phi_{\mu\nu}^{mnl}(\hat{\mathbf{k}}, \Omega_1, \Omega_2)$ are rotational invariants that depends on both the spatial and angular coordinates of the two particles (detailed in appendix D).

For projections of order $n_{\max} = 1$ ($n, m \leq 1$), the DCF can be expressed in very simple form. Only 4 projections $\hat{c}^{mnl}(k)$ are independent: $\hat{c}_S \equiv \hat{c}^{000}$, $\hat{c}_\Delta \equiv \hat{c}^{110}$, $\hat{c}_D \equiv \hat{c}^{112}$ and $\hat{c}_+ \equiv \hat{c}^{011} = -\hat{c}^{101}$, with the corresponding rotational invariants expressed below both in laboratory and intermolecular frames:

$$\Phi^{000} = 1$$

$$\Phi^{011} = i\mathbf{k} \cdot \Omega_1 = i \cos \theta_1$$

$$\Phi^{101} = i\mathbf{k} \cdot \Omega_2 = i \cos \theta_2$$

$$\Phi^{110} = -\sqrt{3}\Omega_1 \cdot \Omega_2 = -\sqrt{3}(\sin \theta_1 \sin \theta_2 \cos \phi_{12} + \cos \theta_1 \cos \theta_2)$$

$$\begin{aligned} \Phi^{112} &= \sqrt{\frac{3}{10}} [3(\mathbf{k} \cdot \Omega_1)(\mathbf{k} \cdot \Omega_2) - \Omega_1 \cdot \Omega_2] \\ &= \sqrt{\frac{3}{10}} (2 \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 \cos \phi_{12}) \end{aligned} \quad (6.15)$$

where the orientations in laboratory frame Ω are here expressed as an orientational vector $\Omega = (\sin \Theta \cos \Phi, \sin \Theta \sin \Phi, \cos \Theta)$ in the cartesian coordinate system. To express the DCF at higher orders, the number of FE needed for $\Phi_{\mu\nu}^{mnl}(\hat{\mathbf{k}}, \Omega_1, \Omega_2)$ becomes huge and the DCF should be calculated in intermolecular frame as indicated above.

6.2.2 Using projections in form of $\hat{c}_{\mu\nu,\chi}^{mn}(k)$

Compared to the expression of $\Phi_{\mu\nu}^{ml}(\hat{\mathbf{k}}, \Omega_1, \Omega_2)$ in laboratory frame (eq. (D.2) in appendix D), its intermolecular form has far fewer terms (eq. (D.18) in appendix D), such that

$$\hat{c}(k, \omega_1, \omega_2) = \frac{1}{2l+1} \sum_{mn\mu\nu\chi} \hat{c}_{\mu\nu,\chi}^{mn}(k) r_{\chi\mu}^m(\theta_1) r_{-\chi\nu}^n(\theta_2) e^{-i\chi(\phi_{12}\equiv\phi_1-\phi_2)} e^{-i\mu\psi_1} e^{-i\nu\psi_2} \quad (6.16)$$

where r is the generalized Legendre polynomial, $m, n \leq n_{\max}$, $|\mu| \leq m$, $|\nu| \leq n$, and $\chi \in [-\min(m, n), \min(m, n)]$.

$r_{\chi\mu}^m(\theta)$, $e^{-i\chi\phi}(\phi)$ and $e^{-i\mu\psi}(\psi)$ can be separately pre-tabulated for each given \mathbf{k} , to avoid double evaluation of each term.

E.q. (6.16) replaces the interpolation of eq. (6.13) by an exact formula and it requires the projections $\hat{c}_{\mu\nu,\chi}^{mn}(k)$ to be stored in memory rather than the full angular representation $\hat{c}(k, \omega_1, \omega_2)$. It also requires the passage from orientations in laboratory frame to orientations in intermolecular frame, i.e. use of the formulas (6.10) for each \mathbf{k} vector.

ANGULAR CONVOLUTION, A BETTER ALGORITHM

In section 6, the spatial convolution is treated by FFT thanks to the transitional invariance that leads to $\mathbf{r}_{12} = \mathbf{r}_1 - \mathbf{r}_2$. However, as the angular grid is not homogeneous, the relative coordinates of two angles cannot be simply represented $\boldsymbol{\Omega}_{12} = \boldsymbol{\Omega}_1 - \boldsymbol{\Omega}_2$, therefore we cannot take advantage of the convolution property shown in eq. (4.17-4.18) for the angles. On the other hand, these two-particle quantities have rotational invariance. As proposed by Blum [4, 5] and used by Fries and Patey [17], a rotational invariant expansion technique reduces the molecular Ornstein-Zernike (MOZ) equation into smaller irreducible matrix equations (§4.2). As there is a mathematical equivalence between IET and MDFT (§3.5), where eq. (6.3) can be regarded as the MOZ equation, this formalism can also be applied to MDFT.

7.1 ANGULAR CONVOLUTION USING BLUM'S REDUCTION

To build a relation between the irreducible form of the MOZ equation deduced by Blum (in the definition of Fries & Patey, detailed in §4.2, with the symmetry $\hat{c}'_{\nu\mu,\underline{\chi}}^{nm}(k) = \hat{c}'_{\mu\nu,\underline{\chi}}^{mn*}(k)$)

$$\hat{\gamma}'_{\lambda\mu,\underline{\chi}}^{lm}(k) = \sum_{n=0}^{n_{\max}} \sum_{\nu=-n}^n (-)^{\chi+\nu} \Delta \hat{\rho}'_{\lambda\underline{\nu},\underline{\chi}}^{ln}(k) \hat{c}'_{\mu\nu,\chi}^{mn*}(k) \quad (7.1)$$

Here the projections $F_{\mu\nu,\chi}^{mn}$ are defined as in [17]. Eq. (7.1) is mathematically identical with [4, 5] but using

$$R_{\mu'\mu}^m = D_{\mu\mu'}^{m*}.$$

The difference

and the MDFT formalism, a generalized spherical harmonic transform (GSHT) treatment is proposed by Luc Belloni, developing the functional gradient $\hat{\gamma}$ and the density $\hat{\rho}$ in eq. (6.3) on Wigner generalized spherical harmonics (GSH):

$$\hat{\gamma}(\mathbf{k}, \Omega_1) = \sum_{m\mu'\mu} f_m \hat{\gamma}_{\mu'\mu}^m(\mathbf{k}) R_{\mu'\mu}^m(\Omega_1) \quad (7.2)$$

$$\Delta\hat{\rho}(\mathbf{k}, \Omega_2) = \sum_{n\nu'\nu} f_n \Delta\hat{\rho}_{\nu'\nu}^n(\mathbf{k}) R_{\nu'\nu}^n(\Omega_2) \quad (7.3)$$

where $0 \leq m, n \leq n_{\max}$, $|\mu'|, |\mu| < m$ and $|\nu'|, |\nu| < n$. $f_m = (2m+1)^{\frac{1}{2}} = \|R_{\mu'\mu}^m\|^{-1}$ is the normalization factor (according to Luc's definition).

The DCF can also be expanded on rotational invariants [17]:

$$\hat{c}(k, \Omega_1, \Omega_2) = \sum_{mnl\mu\nu} f_m f_n \hat{c}_{\mu\nu}^{mnl}(k) \sum_{\mu'\nu'\lambda'} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} R_{\mu'\mu}^m(\Omega_1) R_{\nu'\nu}^n(\Omega_2) R_{\lambda'0}^l(\hat{\mathbf{k}}) \quad (7.4)$$

As GSH possess orthogonality eq. (F.21) and symmetry eq. (F.15), eq. (6.3) can be rewritten by (7.2, 7.3, 7.4), which gives:

$$\hat{\gamma}_{\mu'\mu}^m(\mathbf{k}) = \sum_{nl\nu} \hat{c}_{\mu\nu}^{mnl}(k) \sum_{\nu'\lambda'} (-)^{\nu'+\nu} \Delta\hat{\rho}_{\nu'\nu}^n(\mathbf{k}) \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} R_{\lambda'0}^l(\hat{\mathbf{k}}) \quad (7.5)$$

thus the OZ equation is expanded on GSHs and rotational invariants.

In fact, $\hat{c}_{\mu\nu,\chi}^{mn}(k)$

Note that eq. (7.5) is reducible. Blum's χ -transform [4] defines:

is the conjugate
of the invariant
in eq. (7.1).

$$\hat{c}_{\mu\nu,\chi}^{mn}(k) = \sum_{l=|m-n|}^{m+n} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} \hat{c}_{\mu\nu}^{mnl}(k) \quad (7.6)$$

$$\hat{c}_{\mu\nu}^{mnl}(k) = (2l+1) \sum_{\chi=-\min(m,n)}^{\min(m,n)} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} \hat{c}_{\mu\nu,\chi}^{mn}(k) \quad (7.7)$$

Invariants of form $F_{\mu\nu,\chi}^{mn}(k)$ have a very simple relation with their combined function $F(k, \omega_1, \omega_2)$ in the intermolecular coordinate system (eq. (D.26, D.27)). In MDFT formalism, the projections of $\hat{\gamma}$ and $\hat{\rho}$ in the local frame ($\omega_i = \hat{\mathbf{k}}\Omega_i$) are

$$\hat{\gamma}'(\mathbf{k}, \omega_1) = \sum_{m\chi\mu} f_m \hat{\gamma}_{\chi\mu}^m(\mathbf{k}) R_{\chi\mu}^m(\omega_1) \quad (7.8)$$

$$\Delta\hat{\rho}'(\mathbf{k}, \omega_2) = \sum_{n\chi\nu} f_n \Delta\hat{\rho}_{\chi\nu}^n(\mathbf{k}) R_{\chi\nu}^n(\omega_2) \quad (7.9)$$

and with the rotation formula of GSH (eq. (F.23)), we have

$$\hat{\gamma}_{\chi\mu}^m(\mathbf{k}) = \sum_{\mu'} \hat{\gamma}_{\mu'\mu}^m(\mathbf{k}) R_{\mu'\chi}^m(\hat{\mathbf{k}}) \quad (7.10)$$

$$\Delta\hat{\rho}_{\nu'\underline{\nu}}^n(\mathbf{k}) = \sum_{\chi} \Delta\hat{\rho}_{\chi\underline{\nu}}^n(\mathbf{k}) R_{\underline{\nu}\chi}^{n*}(\hat{\mathbf{k}}) = \sum_{\chi} \Delta\hat{\rho}_{\chi\underline{\nu}}^n(\mathbf{k}) (-)^{\chi+\nu'} R_{\nu'\chi}^n(\hat{\mathbf{k}}) \quad (7.11)$$

Using eq. (7.10), (7.5), eq. (7.11), eq. (7.7) and GSH products relation eq. (F.24) and 3j-symbol orthogonality eq. (F.7), we deduce that:

$$\hat{\gamma}_{\chi\mu}^m(\mathbf{k}) = \sum_{n\nu} (-)^{\chi+\nu} \hat{c}_{\mu\nu,\chi}^{mn*}(k) \Delta\hat{\rho}_{\chi\underline{\nu}}^n(\mathbf{k}) \quad (7.12)$$

If we take the conjugate of $\hat{c}_{\mu\nu,\chi}^{mn*}(k)$, eq. (7.12) is mathematically identical to eq. (7.1), as:

$$\hat{\gamma}_{\chi\mu}^m(\mathbf{k}) = \sum_{l\lambda} \hat{\gamma}_{\lambda\mu,\underline{\chi}}^{lm}(k) R_{\underline{\chi}\lambda}^l(\omega) \quad (7.13)$$

$$\hat{\rho}_{\chi\underline{\nu}}^n(\mathbf{k}) = \sum_{l\lambda} \Delta\hat{\rho}_{\lambda\underline{\nu},\chi}^{ln}(k) R_{\chi\lambda}^{l*}(\omega) \quad (7.14)$$

according to the rotational invariant transform in eq. (D.27).

In this way, the integral of the angular part in eq. (6.3) can be reduced to a sum of a few terms. Table 7.1 shows some parameters linking to computing cost of different algorithms. It shows that the expansion on GSHs (eq. (7.5)) does not give any reduction of FE compared to its 6D function form (eq. (6.3))*; but after the Blum's χ -transform,

* Only if we do not need to calculate

$\hat{c}(\mathbf{k}, \Omega_1, \Omega_2)$.

the OZ function is largely reduced. As the spatial convolution takes advantage of the transitional invariance r_{12} , the χ -transform in fact makes use of the rotational invariance.

m_{\max}	0	1	2	3	4	5
N_{Θ}	1	2	3	4	5	6
N_{ang} (Gauss-Legendre)	1 (1)	18 (6)	75 (45)	196 (84)	405 (225)	726 (330)
N_{ang} (Lebedev $\times \psi$)	1 (1)	18 (6)	70 (42)	182 (78)	342 (190)	550 (250)
N_{proj}	1 (1)	10 (4)	35 (19)	84 (40)	165 (85)	286 (140)
FE for eq. (6.3)	1 (1)	324 (36)	5625 (2025)	38416 (7056)	164025 (50625)	527076 (108900)
FE for eq. (7.5)	1 (1)	262 (34)	4787 (1459)	36588 (8116)	175989 (47221)	633490 (150566)
FE for eq. (7.12)	1 (1)	34 (6)	259 (75)	1092 (252)	3333 (877)	8294 (2002)

Table 7.1: Number of FE needed by OZ equation of different form for arbitrary solvent (outside the parentheses) and solvent possessing C_{2v} symmetry (inside the parentheses)

7.2 FAST GENERALIZED SPHERICAL HARMONIC TRANSFORM

In the angular convolution algorithm above, the OZ equation is reduced to a few function evaluations by taking advantage of the orthogonality and symmetries of GSHs. To use this algorithm as an analogous to the treatment of the convolution with FFT for spatial grids, it *a priori* should be fast. This is exactly possible owing to the exponential components in the definition of GSH, and a detailed discussion will be given later.

The FGSHT provides a forward-backward transform between a general angular function

$F(\Omega) \equiv F(\cos \Theta, \Phi, \Psi)$ and its projections $F_{\mu' \mu}^m$ ($|\mu'|, |\mu| \leq m$):

$$F_{\mu' \mu}^m = \frac{f_m}{8\pi^2} \int d\Omega F(\Omega) R_{\mu' \mu}^{m*}(\Omega) \quad (\text{forward}) \quad (7.15)$$

$$F(\Omega) = \sum_{m, \mu', \mu} f_m F_{\mu' \mu}^m R_{\mu' \mu}^m(\Omega) \quad (\text{backward}) \quad (7.16)$$

where $R_{\mu'\mu}^m(\boldsymbol{\Omega})$ is the Wigner generalized spherical harmonics (Appendix F), which form a complete orthogonal set, being defined as:

$$R_{\mu'\mu}^m(\boldsymbol{\Omega}) = r_{\mu'\mu}^m(\Theta) e^{-i(\mu'\Phi + \mu\Psi)} \quad (7.17)$$

7.2.1 Equivalence of order in angular quadratures and projections

Suppose that $F(\boldsymbol{\Omega})$ is a polynomial of both $\cos\Theta$, $\cos\Phi$ and $\cos\Psi$ of order n , ($n+1$ polynomial terms). To expand completely this function as shown in equation (7.16), at least $m_{\max} = n$ is needed, where m_{\max} is the highest order of projections $F_{\mu'\mu}^m$ in the expansion. Then to evaluate exactly the integration in equation (7.15), at least $n+1$ for $\cos\Theta$ (Gauss-Legendre grid), $2n+1$ for Φ (equal-spaced grid), $2n+1$ for Ψ (equal-spaced grid) points of angular grid are needed (c. f. appendix C). In the case of water which possesses a C_2 symmetry $F(\Psi + \pi) = F(\Psi)$, only projections of even μ are nonzero: (Vérifier les facteurs $2p^*i$ ci-dessous; a priori $1/2*pi$ dans la première + mettre des bornes $0-2*pi$ aux intégrales) (What???)

$$\begin{aligned} F_\mu &= \int d\Psi F(\Psi) e^{i\mu\Psi} = \int d(\Psi + \pi) F(\Psi + \pi) e^{i\mu(\Psi+\pi)} \\ &= e^{i\mu\pi} \int d\Psi F(\Psi) e^{i\mu\Psi} = e^{i\mu\pi} F_\mu \end{aligned} \quad (7.18)$$

$$F_\mu = \begin{cases} 0 & \mu = 2n+1, n \in \mathbb{Z} \\ F_\mu & \mu = 2n, n \in \mathbb{Z} \end{cases} \quad (7.19)$$

Therefore the function

$$F(\Psi) = \sum_\mu F_\mu e^{-i\mu\Psi} \quad (7.20)$$

can be rewritten as:

$$F(\Psi_2/2 \equiv \Psi) = \sum_{\mu_2 \equiv \mu/2} F_{2\mu_2} e^{-i\mu_2 \Psi_2} \quad (7.21)$$

As $|\mu_2| \leq n/2$, $F(\Psi_2/2 \equiv \Psi)$ is a polynomial of $\cos \Psi_2$ of order $\text{floor}(n/2) \equiv [n/2]$,

in the forward transform

$$F_{2\mu_2 \equiv \mu} = \int d\Psi F(\Psi) e^{i\mu\Psi} = \frac{1}{2} \int d\Psi_2 F(\Psi_2/2 \equiv \Psi) e^{i\mu_2 \Psi_2} \quad (7.22)$$

the total degree $\cos \Psi_2$ polynomial in the integrand is $2[n/2]$, then $2[n/2] + 1$ points of Ψ_2 (or Ψ) are needed.

For further implementation, we take these conclusions, but distinguish the order of quadrature m_{\max} (linked to the angular grid) and the order of projection n_{\max} (linked to the GSH transform) for numerical reason.

7.2.2 Integration of Φ, Ψ using FFT

Here we write eq. (7.15, 7.16) in an explicit way:

$$F_{\mu'\mu}^m = \frac{f_m}{8\pi^2} \sum_{i=0}^{m_{\max}} w_i \sum_{j=0}^{2m_{\max}} \sum_{k=0}^{2\lfloor m_{\max}/s \rfloor} F(\Theta_i, \Phi_j, \Psi_k) R_{\mu'\mu}^{m*}(\Theta_i, \Phi_j, \Psi_k) \quad (7.23)$$

$$F(\Theta_i, \Phi_j, \Psi_k) = \sum_{m=0}^{n_{\max}} f_m \sum_{\mu'=-m}^m \sum_{\substack{\mu=-m \\ \text{mod } (\mu, s)=0}}^m F_{\mu'\mu}^m R_{\mu'\mu}^m(\Theta_i, \Phi_j, \Psi_k) \quad (7.24)$$

where w_i is the weight of Gauss-Legendre quadrature ($m_{\max} + 1$ points of Θ_i), normalized to the total angular integration; and $s = 1$ or 2 according to the symmetry C_s of solvent.

To integrate eq. (7.23) in a direct way, $(m_{\max} + 1)(2m_{\max} + 1)(2\lfloor m_{\max}/s \rfloor + 1) = N_\Theta N_{\Phi\Psi} = N$ FE are needed for each $F_{\mu'\mu}^m$, an overall $O(N_{FE}^2)$ process is needed and *vice versa*. Therefore, a faster algorithm proposed by Numerical Recipes [**Numerical_Recipes_3ed**] suggests to reduce this cost to $O(N_\Theta^2 N_{\Phi\Psi} \ln N_{\Phi\Psi} \simeq N^{4/3})$ by FFT.

Following this idea, eq. (7.23) can be rewritten as:

$$F_{\mu'\mu}^m = \frac{f_m}{8\pi^2} \sum_{i=0}^{m_{\max}} w_i r_{\mu'\mu}^m(\Theta_i) F_{\mu'\mu}(\Theta_i) \quad (7.25)$$

where $F_{\mu'\mu}(\Theta_i)$ is the Φ, Ψ integration evaluated using trapezoid (or Gauss-Chebyshev) quadrature:

$$\begin{aligned} F_{\mu'\mu}(\Theta_i) &= \sum_{j=0}^{2m_{\max}} \sum_{k=0}^{2\lfloor m_{\max}/s \rfloor} F(\Theta_i, \Phi_j, \Psi_k) e^{i(\mu' \Phi_j + \mu \Psi_k)} \\ &= \sum_{j=0}^{2m_{\max}} \sum_{k=0}^{2\lfloor m_{\max}/s \rfloor} F(\Theta_i, \Phi_j, \Psi_k) e^{2\pi i \mu' j / (2m_{\max} + 1)} e^{2\pi i \mu k / (2\lfloor m_{\max}/s \rfloor + 1)} \end{aligned} \quad (7.26)$$

that shares the same formula with an FFT-2D process of $(2m_{\max} + 1)(2\lfloor m_{\max}/s \rfloor + 1)$ elements.

Similarly, the backward process (7.16) can be rewritten as:

$$\begin{aligned} F(\Theta_i, \Phi_j, \Psi_k) &= \sum_{m=0}^{n_{\max}} f_m \sum_{\mu'=-m}^m \sum_{\substack{\mu=-m \\ \text{mod } (\mu, s)=0}}^m F_{\mu'\mu}^m R_{\mu'\mu}^m(\Theta_i, \Phi_j, \Psi_k) \\ &= \sum_{\mu'=-n_{\max}}^{n_{\max}} \sum_{\substack{\mu=-n_{\max} \\ \text{mod } (\mu, s)=0}}^{n_{\max}} \sum_{m=\max(|\mu'|, |\mu|)}^{n_{\max}} f_m F_{\mu'\mu}^m R_{\mu'\mu}^m(\Theta_i, \Phi_j, \Psi_k) \\ &= \sum_{\mu'=-n_{\max}}^{n_{\max}} \sum_{\substack{\mu=-n_{\max} \\ \text{mod } (\mu, s)=0}}^{n_{\max}} F_{\mu'\mu}(\Theta_i) e^{2\pi i \mu' j / (2m_{\max} + 1)} e^{2\pi i \mu k / (2\lfloor m_{\max}/s \rfloor + 1)} \end{aligned} \quad (7.27)$$

with

$$F_{\mu'\mu}(\Theta_i) = \sum_{m=\max(|\mu'|, |\mu|)}^{n_{\max}} f_m F_{\mu'\mu}^m r_{\mu'\mu}^m(\Theta_i) \quad (7.28)$$

When $n_{\max} \leq m_{\max}$, the double sum in eq. (7.27) is included in the FFT-2D process of $(2m_{\max} + 1)(2\lfloor m_{\max}/s \rfloor + 1)$ elements. However, if $n_{\max} > m_{\max}$, the FFT-2D process only gives a partial sum of $|\mu'|, |\mu| \leq m_{\max}$, the other terms in eq. (7.27) can only be calculated by a GSHT process as $F_{\mu'\mu}(\Theta_i)$ is not periodic for μ' and μ . There can be

Note that the GSHT is able to treat the case $n_{\max} > m_{\max}$.

further approximations to treat this problem, but for practical usage, we only consider the case of $n_{\max} \leq m_{\max}$.

The FFTW3 library [18] is used for implementation, which performs discrete Fourier Transform (DFT) as defined below:

$$Y_k = \sum_{j=0}^{n-1} X_j e^{-2\pi i j k / n} \quad (\text{forward}) \quad (7.29)$$

$$X_j = \sum_{k=0}^{n-1} Y_k e^{2\pi i j k / n} \quad (\text{backward}) \quad (7.30)$$

Note that after a forward-backward Fourier transform, the original function is multiplied by a normalization factor N_k , which is the total number of nodes k .

For input function Y_k ($k = 0, \dots, n - 1$) in real number, FFTW3 only outputs elements $k = 0, \dots, \lfloor n/2 \rfloor$ ($\lfloor n/2 \rfloor + 1$ complex numbers of X_j are stocked), with the “Hermitian” symmetry

$$Y_k = Y_{n-k}^* \quad (7.31)$$

used to regenerate elements of $k > \lfloor n/2 \rfloor$. The resulting X_j issue from the corresponding backward transform is purely real. As the angular function $F(\Omega)$ is real, and the GSHs possess symmetry of eq. (F.15):

$$R_{-\mu', -\mu}^m(\Omega) = (-1)^{\mu' + \mu} R_{\mu' \mu}^{m*}(\Omega) \quad (7.32)$$

the symmetry relation between the projections are

$$F_{-\mu', -\mu}^m = (-1)^{\mu' + \mu} F_{\mu' \mu}^{m*} \quad (7.33)$$

Therefore only the projections of $\mu \geq 0$ need to be stocked, which can be calculated with only these FFTW3 output elements reduced by the Hermitian symmetry. The full process of FFTW3-2D real to real transform is illustrated in figure 7.1.

dim 1		in_forward / out_backward (real)	in_backward / out_forward (complex)
array index		1 2 3 ... $m'+1$ $m'+2$... $2m'$ $2m'+1$	1 2 3 ... $m'+1$
real index		0 1 2 ... m' $m'+1$... $2m'-1$ $2m'$	0 1 2 ... m'
$k \leftrightarrow \mu$		1 2 3 ... $m'+1$ $m'+2$... $2m'$ $2m'+1$	0 1 2 ... m'
dim 2		in_forward / out_backward (complex)	in_backward / out_forward (complex)
array index		1 2 3 ... $m+1$ $m+2$... $2m$ $2m+1$	1 2 3 ... $m+1$ $m+2$... $2m$ $2m+1$
real index		0 1 2 ... m $m+1$... $2m-1$ $2m$	0 1 2 ... m $m+1$... $2m-1$ $2m$
$j \leftrightarrow \mu'$		1 2 3 ... $m+1$ $m+2$... $2m$ $2m+1$	0 1 2 ... m $-m$ $-m+1$... -1

Figure 7.1: Indices arrangement in a complete forward-backward FFT-2D process of $m' \times m$ elements. The DFT of dim 1 (k to μ) and dim 2 (j to μ') are done sequentially and *vice versa*. Array index is the one used by Fortran array, real index is the one shown in eq. (7.29) and (7.30), k and j indices shown in the left as well as μ and μ' in the right are those in eq. (7.26) and (7.27). Here $m = m_{\max}$ and $m' = \lfloor m_{\max}/s \rfloor$.

As the output array of FFTW3 is periodic,

$$e^{2\pi i \mu k/n} = e^{2\pi i (\mu - n)k/n} e^{2\pi i k} = e^{2\pi i (\mu - n)k/n} \quad (7.34)$$

the indices $\mu = m_{\max} + 1, \dots, 2m_{\max}$ actually correspond to $\mu = -m_{\max}, \dots, -1$. Note that eq. (7.26) and (7.27) do not possess the periodicity of eq. (7.34), only in the domain of definition of μ' and μ some intermediary functions share the same formula with FFT.

Moreover, from eq. (7.26), (7.28) and (7.33), we can verify that

$$F_{\mu'\mu}(\Theta) = F_{-\mu',-\mu}^*(\Theta) \quad (7.35)$$

The latter is used in the code since, according to the definition in eq. (7.29) and (7.30), $F_{-\mu',-\mu}(\Theta)$ is calculated instead of $F_{\mu'\mu}(\Theta)$.

7.3 OPERATIONAL ALGORITHM

As described above, the whole process of γ and \mathcal{F}_{exc} functional evaluation proposed by this algorithm can be concluded as 8 operations:

1. Firstly, the Fourier transform of the density is computed:

$$\Delta\hat{\rho}(\mathbf{k}, \Omega) = \int d\mathbf{r} \Delta\rho(\mathbf{r}, \Omega) e^{-i\mathbf{k}\cdot\mathbf{r}} \quad (7.36)$$

2. Then $\Delta\hat{\rho}(\mathbf{k}, \Omega)$ is expanded on generalized spherical harmonics

$$\Delta\hat{\rho}_{\mu'\mu}^m(\mathbf{k}) = \frac{f_m}{8\pi^2} \int d\Omega \Delta\hat{\rho}(\mathbf{k}, \Omega) R_{\mu'\mu}^{m*}(\Omega) \quad (7.37)$$

Note that these two steps, the same with their backward transform, are commutable, which will be discussed after.

3. Afterwards the projections in k-frame are then rotated into the local coordinate system along the unit vector $\hat{\mathbf{k}}$

$$\Delta\hat{\rho}'_{\chi\mu}^m(\mathbf{k}) = \sum_{\mu'} \Delta\hat{\rho}_{\mu'\mu}^m(\mathbf{k}) R_{\mu'\chi}^m(\hat{\mathbf{k}}) \quad (7.38)$$

where the evaluation of rotation matrix elements by recurrence is detailed in appendix E.

4. Next, computing the OZ equation with Blum's reduction:

$$\hat{\gamma}'_{\chi\mu}^m(\mathbf{k}) = \sum_{n,\nu} (-1)^{\chi+\nu} \hat{c}_{\mu\nu,\chi}^{mn}(\mathbf{k}) \Delta\hat{\rho}'_{\chi\nu}^n(\mathbf{k}) \quad (7.39)$$

5. The γ projections are then transformed back to global coordinates system:

$$\hat{\gamma}_{\mu'\mu}^m(\mathbf{k}) = \sum_{\chi} \hat{\gamma}_{\chi\mu}^m(\mathbf{k}) R_{\mu'\chi}^{m*}(\hat{\mathbf{k}}) \quad (7.40)$$

6. From here the function in angular frame can thus be rebuilt

$$\hat{\gamma}(\mathbf{k}, \boldsymbol{\Omega}) = \sum_{m, \mu', \mu} f_m \hat{\gamma}_{\mu'\mu}^m(\mathbf{k}) R_{\mu'\mu}^m(\boldsymbol{\Omega}) \quad (7.41)$$

7. Then the inverse Fourier transform of these projections is:

$$\gamma(\mathbf{r}, \boldsymbol{\Omega}) = \int d\mathbf{k} \hat{\gamma}(\mathbf{k}, \boldsymbol{\Omega}) e^{i\mathbf{r}\cdot\mathbf{k}} \quad (7.42)$$

8. Finally, the functional \mathcal{F}_{exc} is computed by

$$\mathcal{F}_{\text{exc}} = \frac{1}{2} \int d\mathbf{r} d\boldsymbol{\Omega} \Delta\rho(\mathbf{r}, \boldsymbol{\Omega}) \gamma(\mathbf{r}, \boldsymbol{\Omega}) \quad (7.43)$$

7.3.1 Commutativity between operations

As mentioned in the operational algorithm, three types of operations are being done before and after the OZ equation. They are

1. Fast Fourier Transform for 3-dimensional spatial grid (FFT3D): implemented by package FFTW3 [18], mathematically leading to no accuracy lost;
2. Fast generalized spherical harmonics transform (FGSHT): has real or complex input, is exact if $F(\boldsymbol{\Omega})$ can be given as an expansion of GSHs of order at most m_{\max} ;
3. Rotation between laboratory coordinate system and local system linked to vector \mathbf{k} (RotS): can be done for both function and projections. It introduces a minus error

in accuracy at origin and border of the box, which will be discussed in the next chapter.

Their commutativity is shown in figure 7.2.

(All those operations have commutativity properties that are illustrated in figure 7.2.

Going from $f(\mathbf{r}, \Omega)$ to $f_{\mu'\mu}^m(\mathbf{k})$ can be done either by either spatial 3D-FFT for each orientation and FGSHT transform for each value \mathbf{k} or by transforming first to angular projections $f_{\mu'\mu}^m(\mathbf{r})$ for each \mathbf{r} and then transforming each projection to \mathbf{k} -space. In the same way, the passage from $f(\mathbf{k}, \Omega)$ to in laboratory frame to projections $f'_{\mu'\mu}^m(\mathbf{k})$ in local frame can be obtained by rotation then projections or projections then rotation. Finally, going from $f(\mathbf{r}, \Omega)$ in r-space and laboratory frame to $\hat{f}'(\mathbf{k}, \omega_k)$ in \mathbf{k} -space and local frame can be done by 3D-FFT/rotation or rotation/3D-FFT. These commutation properties are due to the fact that all operations are linear so that the order of sums or integrals can be interchanged. See Appendix xx for details.)

(But coordinate rotation and FFT3D are not commutative. I'd rather prove it here than put it in appendix.)

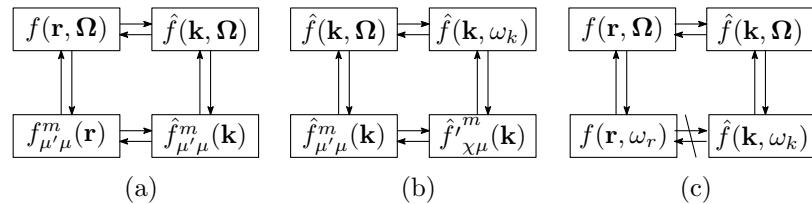


Figure 7.2: Commutativity of operations. (a) FFT3D and FGSHT; (b) RotS and FGSHT; (c) FFT3D and RotS

7.3.1.1 FFT3D and FGSHT

The FFT3D

$$f(\mathbf{r}) = \int \hat{f}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}} \quad (7.44)$$

$$\hat{f}(\mathbf{k}) = \int f(\mathbf{r}) e^{-i\mathbf{k}\cdot\mathbf{r}} \quad (7.45)$$

does not depend on the angular part of the function, and

$$f(\Omega) = \sum_{m\mu'\mu} f_m f_{\mu'\mu}^m R_{\mu'\mu}^m(\Omega) \quad (7.46)$$

$$f_{\mu'\mu}^m = \int f_m f(\Omega) R_{\mu'\mu}^m(\Omega) \quad (7.47)$$

does not depend on the spatial part of the function. The two operations are commutative.

7.3.1.2 FGSHT and coordinate rotation

Function \hat{f}' intermolecular frame can be deduced from the function \hat{f} in laboratory frame

$$\hat{f}(\mathbf{k}, \Omega) = \hat{f}'(\mathbf{k}, \omega_k)$$

where the two can both be expanded on GSH

$$\hat{f}(\mathbf{k}, \Omega) = \sum_{m\mu'\mu} f_m \hat{f}_{\mu'\mu}^m(\mathbf{k}) R_{\mu'\mu}^m(\Omega)$$

$$\hat{f}'(\mathbf{k}, \omega_k) = \sum_{m\chi\mu} f_m \hat{f}_{\chi\mu}^m(\mathbf{k}) R_{\chi\mu}^m(\omega_k)$$

And the relations between projections are simple

$$f'_{\chi\mu}{}^m(\mathbf{k}) = \sum_{\mu'} R_{\mu'\chi}^m(\hat{\mathbf{k}}) f_{\mu'\mu}^m(\mathbf{k})$$

$$f_{\mu'\mu}^m(\mathbf{k}) = \sum_{\chi} R_{\mu'\chi}^{m*}(\hat{\mathbf{k}}) f'_{\chi\mu}{}^m(\mathbf{k})$$

Thus the two operations are commutative.

7.3.1.3 Coordinate rotation and FFT3D

The rotation from $f(\mathbf{r}, \boldsymbol{\Omega})$ to $f(\mathbf{r}, \boldsymbol{\omega})$ depends on the vector \mathbf{r} , of which the information is totally lost after FFT3D. The rotation from $f(\mathbf{k}, \boldsymbol{\Omega})$ to $f(\mathbf{k}, \boldsymbol{\omega})$ can only depend on the vector \mathbf{k} , they are not the same rotation, therefore non-commutative.

7.3.2 Reduction by symmetry

A further reduction of computing cost can be made by performing approximately only half of the operations, thanks to the symmetric relations between the projections.

In eq. (7.37), $\Delta\rho(\mathbf{r}, \boldsymbol{\Omega})$ is real. Thanks to the property of GSH (eq. (F.15)),

$$R_{\mu'\mu}^m(\boldsymbol{\Omega}) = (-)^{\mu'+\mu} R_{-\mu'-\mu}^{m*}(\boldsymbol{\Omega}) \quad (7.48)$$

we find

$$\Delta\hat{\rho}_{\mu'\mu}^m(\mathbf{r}) = (-)^{\mu'+\mu} \Delta\hat{\rho}_{-\mu',-\mu}^{m*}(\mathbf{r}) \quad (7.49)$$

therefore only $\mu' > 0$ or $\mu > 0$ is needed to generate all information.

When $\Delta\hat{\rho}_{\mu'\mu}^m(\mathbf{r})$ is transformed into k -space, replacing

$$\Delta\hat{\rho}_{\mu'\mu}^m(\mathbf{k}) = \int d\mathbf{r} \Delta\hat{\rho}_{\mu'\mu}^m(\mathbf{r}) e^{-i\mathbf{r}\cdot\mathbf{k}} \quad (7.50)$$

with eq. (7.49) gives

$$\Delta\hat{\rho}_{\mu'\mu}^m(\mathbf{k}) = (-)^{\mu'+\mu} \Delta\hat{\rho}_{-\mu',-\mu}^{m*}(-\mathbf{k}) \quad (7.51)$$

Thus only the projections of $\mu' > 0$, $\mu > 0$ or \mathbf{k} where one of the dimensions $k_i > 0$ are independent.

The rotation to local frame is governed by the relation deduced from the symmetries in appendix F, where

$$r_{\mu'\chi}^m(\theta) = (-)^{m+\chi} r_{\underline{\mu}'\chi}^m(\pi - \theta) \quad (7.52)$$

$$R_{\mu'\chi}^m(\phi, \theta, \psi) = (-)^{m+\chi} R_{\underline{\mu}'\chi}^m(-\phi, \pi - \theta, \psi) = \quad (7.53)$$

(prove?):

(proof: should refer to eq (1.6) in Luc's notes. So add this set of equations):

(But I just can't deduce it !!!!!)

$$R_{\mu'\chi}^m(\hat{\mathbf{k}}) = (-)^m R_{\mu',-\chi}^m(-\hat{\mathbf{k}}) = (-)^{m+\mu'+\chi} R_{-\mu',\chi}^m(-\hat{\mathbf{k}}) \quad (7.54)$$

which gives

$$\Delta \hat{\rho}_{\chi\mu}^m(\mathbf{k}) = (-)^{m+\mu+\chi} \Delta \hat{\rho}_{\chi,-\mu}^{m*}(-\mathbf{k}) \quad (7.55)$$

Thanks to the symmetry (eq. (appendix))

$$\hat{c}_{\mu\nu,\chi}^{mn}(k) = (-)^{m+n+\mu+\nu} \hat{c}_{\underline{\mu}\underline{\nu},\chi}^{mn*}(k) \quad (7.56)$$

$$\hat{c}_{\mu\nu,\chi}^{mn}(k) = (-)^{m+n} \hat{c}_{\nu\mu,\chi}^{nm}(k) \quad (7.57)$$

$$\hat{c}_{\mu\nu,\underline{\chi}}^{mn}(k) = \hat{c}_{\underline{\mu}\underline{\nu},\chi}^{mn}(k) = (-)^{m+n} \hat{c}_{\mu\nu,\chi}^{mn*}(k) \quad (7.58)$$

$\hat{\gamma}_{\chi\mu}^m(\mathbf{k})$ possesses the same symmetry. Thus the OZ equation can be reduced by a factor of two. THIS IS NOT EXPLICIT ENOUGH; there are several choices to exploit the symmetry. You should explain at least your choice and how YOU personnaly DID in your code

(In the code I did only the first step 7.51 and the last step)

In the implementation, it should be noted that not exactly the half of points are calculated. As shown in figure 7.3, where the 2D plan corresponds two of the three dimensions in k -space grid, the green points can be generated from the black points by the symmetries of eq. (7.51), but the red points should be all calculated, of which the corresponding points is also a red point or even itself. This ever caused a huge problem in the implementation, as we put $\Delta\hat{\rho}_{\mu'\mu}^m(\mathbf{k})$ and $\hat{\gamma}_{\mu'\mu}^m(\mathbf{k})$ in the same array for reason of memory. It should be assured that these points are calculated only once.

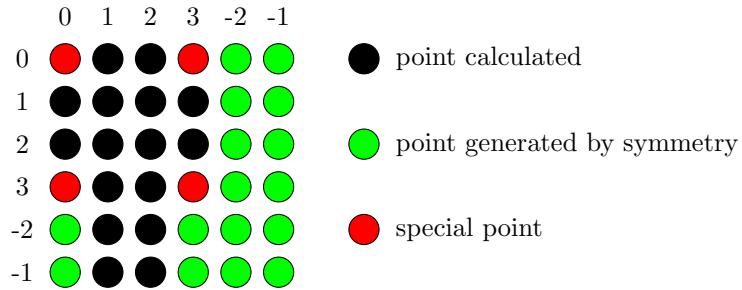


Figure 7.3: Distribution of points to be calculated according to symmetry in a 2D plan

8

FREE ENERGY CORRECTION FOR SINGLE IONS

The solvation free energy is the most important property that we seek; as shown in previous sections, it can be calculated by the minimization of free energy functional $\mathcal{F}[\rho]$.

Here is a discussion about some corrections needed for charged solutes and some related thermodynamic quantities that can be obtained directly from the solvation free energy.

In the calculation of external potential as well as the total solvation free energy, the use of different conventions can lead to a charge-independent offset, which introduces error for charged solutes [31, 38, 39]. This offset is mainly caused by two sources: (1) resulting from the use of a finite system size; in our case, is a system with cubic periodic boundary conditions, which presents artificial interactions between the ion and its own periodic copies, as well as between the solvent and the periodic copies of the ion (Type-B in [39]); (2) resulting from the choice of convention for summing up the contributions of solvent charges to the electrostatic potential in the sample system (Type-C in [39]).

8.1 CORRECTION OF TYPE B

Type B correction should be added for systems with finite size or periodic boundary conditions, accounting for the error in the solvent polarization

Another way to evaluate this error is to make a numerical extrapolation of the inverse of the box size ($1/L$); it

$$\Delta G_B = \frac{1}{8\pi\epsilon_0} \left(1 - \varepsilon^{-1}\right) \frac{q^2}{L} \left[\xi + \frac{4\pi}{3} \left(\frac{R_I}{L}\right)^2 - \frac{16\pi}{45} \left(\frac{R_I}{L}\right)^5 \right] \quad (8.1)$$

where

ε_0 is the vacuum permittivity;

ε is the solvent permittivity (dielectric constant);

q is the solute charge;

L is the box length;

R_I is the ionic radius;

ξ is the energy per particle in a simple cubic lattice, $\xi \simeq -2.837297$ [50].

As R_I is significantly smaller than the size of the computational box, i. e. $R_I \ll L$, its quadratic as well as higher order of (R_I/L) is considered negligible, thus eq. (8.1) becomes

$$\Delta G_B = \frac{\xi}{8\pi\varepsilon_0} \left(1 - \varepsilon^{-1}\right) \frac{q^2}{L} \quad (8.2)$$

It links to Born correction.

8.2 CORRECTION OF TYPE C

Type-C corrections are needed when the systems to be compared use different electrostatic summation schemes: on the basis of point charges within entire solvent molecules (M scheme) or on the basis of individual point charges (P scheme), shown in figure 8.1 (c) and (d), which brings a fixed free energy difference at the boundary.

It can be deduced analytically by considering the solvent as a canonical ensemble under the orientational disorder limit (ODL) [38], which becomes an isotropic quadrupole (IQ)

γ is elsewhere fluid, whose solvent molecule (figure 8.1 (b)) possesses the same quadrupole trace γ referred to as the

$$\text{spheropole moment } [\text{45, 59}], \quad \gamma = \text{tr}(\mathcal{Q}) = \mathcal{Q}_{xx} + \mathcal{Q}_{yy} + \mathcal{Q}_{zz} \quad (8.3)$$

which is the

spherical

component of the

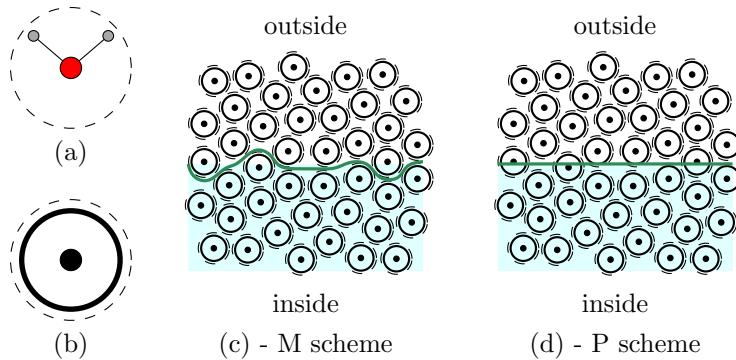


Figure 8.1: IQ model and summation scheme. (a) The solvent molecule. (b) The equivalent isotropic quadrupole (IQ) fluid model. (c) In the M scheme, one evaluates the Coulombic potential generated by the solvent charges belonging to all molecules within the boundary. (d) In the P scheme, one evaluates the Coulombic potential generated by all solvent charges within the boundary.

where the quadrupole moment of the solvent molecule can be calculated by its definition [55]

$$\mathcal{Q}_{ij} = \int_V r_i r_j \rho(\mathbf{r}) dv = \sum_{\alpha=1}^N q^{(\alpha)} r_i^{(\alpha)} r_j^{(\alpha)} \quad (8.4)$$

It can be shown that the charge density of the solvent located within the boundary of the sample system vanishes everywhere, except at the boundary in the M scheme, which results in a uniform normal surface polarization. The correction needed for **M scheme** is

$$\Delta G_C = -q \left(1 - \frac{4\pi R_1^3}{3L^3} \right) \Delta \Phi_{ODL} \quad (8.5)$$

where $\Delta \Phi_{ODL} = (6\epsilon_0)^{-1} \eta \gamma$, η being the solvent number density.

In the same way, when we consider $R_I \ll L$, eq. (8.5) becomes

$$\Delta G_C = - (6\epsilon_0)^{-1} \eta \gamma q \quad (8.6)$$

SOLVATION STRUCTURE

In MDFT, all the information about solvation structure can be deduced from the solvent density $\rho(\mathbf{r}, \Omega)$. This section presents some examples of structure which are used in later chapters.

9.1 RADIAL DISTRIBUTION FUNCTION AND SITE-SITE DISTRIBUTION FUNCTION

When the solvent is homogeneous, the PDF can be reduced to $g(r_{12})$, which is somewhere referred to as the radial distribution function (**RDF!**). However, it is also used as a key character of the structure for inhomogeneous fluid, which can be calculated equivalently as:

$$g(r) = \langle \rho(r, \hat{\mathbf{r}}) \rangle = \frac{\int \rho(r, \hat{\mathbf{r}}) ds_r}{\int ds_r} \quad (9.1)$$

To do this integration, it is required to transform $\rho(\mathbf{r}, \Omega)$ into spherical coordinates. But as $\rho(\mathbf{r}, \Omega)$ in the code is a N -point discrete space grid:

$$\rho(\mathbf{r}) = \int d\Omega \rho(\mathbf{r}, \Omega) = \sum_{i=1}^N \rho_i \delta(\mathbf{r} - \mathbf{r}_i) \quad (9.2)$$

The better way to do the integration is to use a histogram approach.

The grid points is assumed to be homogenous in space, such that the number of point entering in an arbitrary volume v is proportional to this volume. Obviously the grid of $\rho(\mathbf{r}, \Omega)$ satisfies this assumption.

The average value of $g(r)$ between an interval δr is

$$g(r_i) = \langle g(r) \rangle_r^{r+\delta r} = \frac{\int_r^{r+\delta r} g(r) dr}{\delta r} \quad (9.3)$$

Thus

$$g(r_i) = \frac{1}{\delta v_i} \int_r^{r+\delta r} \int_s \rho(r, \hat{\mathbf{r}}) dr ds_r = \frac{1}{\delta v_i} \int_{v_i} \sum_{i=1}^N \rho_i \delta(\mathbf{r} - \mathbf{r}_i) dv_i \quad (9.4)$$

where $\delta v_i = \delta r \cdot s_{r_i} = \int_{v_i} \delta(\mathbf{r} - \mathbf{r}_i) dv_i$ (as the points are homogeneous).

The total function is

$$g(r_i) = \frac{\int_{v_i} \sum_{i=1}^N \rho_i \delta(\mathbf{r} - \mathbf{r}_i) dv_i}{\int_{v_i} \delta(\mathbf{r} - \mathbf{r}_i) dv_i} \quad (9.5)$$

and it becomes to sum up the point values ρ_i in the interval $\delta v = \delta r \cdot S_r$, and divide it by the number of points in this interval.

A site-site distribution function is the same type with the **RDF!**, but the origin for the calculation of \mathbf{r} is no longer at the center of solute, but the site coordinate \mathbf{r}_u , such that the new coordinates are calculate as $\mathbf{r}' = \mathbf{r} - \mathbf{r}_u$. Calculation of solvent site outside the solvent center requires more complicated calculation, invoking the rotation of solvent coordinate to Ω -frame. It has equivant information of the structure than the rotational invariant projections of higher order, here we haven't done the implementation.

9.2 RADIAL POLARIZATION FUNCTIONS

Radial polarization function (RPF) is defined as

$$p(r) = \langle P(r, \hat{\mathbf{r}}) \rangle \quad (9.6)$$

where $P(\mathbf{r})$ is the polarization $P(\mathbf{r}) = \int \boldsymbol{\Omega} \cdot \rho(\mathbf{r}, \boldsymbol{\Omega}) d\boldsymbol{\Omega}$. It can be calculate in the same way as $g(r)$.

9.3 ROTATIONAL INVARIANT EXPANSION

The density can be expanded on rotational invariants:

$$\rho(\mathbf{r}, \boldsymbol{\Omega}) = \sum_{mnl\mu\nu} \rho_{\mu\nu}^{mnl}(r) \Phi_{\mu\nu}^{mnl}(0, \boldsymbol{\Omega}, \hat{\mathbf{r}}) \quad (9.7)$$

$$= \sum_{mnl\mu\nu} \rho_{\mu\nu}^{mnl}(r) f^m f^n \sum_{\eta} \begin{pmatrix} m & n & l \\ \mu & \eta & -\mu - \eta \end{pmatrix} R_{\eta\nu}^n(\boldsymbol{\Omega}) R_{-\mu-\eta, 0}^l(\hat{\mathbf{r}}) \quad (9.8)$$

And the forward transform is as shown below:

$$\rho_{\mu\nu}^{mnl}(r) = f^m f^n \sum_{\eta} \begin{pmatrix} m & n & l \\ \mu & \eta & -\mu - \eta \end{pmatrix} \int d\hat{\mathbf{r}} R_{-\mu-\eta, 0}^{l*}(\hat{\mathbf{r}}) \int d\boldsymbol{\Omega} \rho(r, \hat{\mathbf{r}}, \boldsymbol{\Omega}) R_{\eta, \nu}^{n*}(\boldsymbol{\Omega}) \quad (9.9)$$

A detailed deduction for this generalized formula is in appendix D.

9.4 EQUIVALENCE BETWEEN THE CURVES

It can be proven mathematically the relation between this curves.

Firstly, as

$$\Phi_{00}^{000}(\mathbf{r}, \boldsymbol{\Omega}) = 1 \quad (9.10)$$

there is only one expansion term in eq. (9.7). The projection

$$\rho_{00}^{000}(r) = \rho(\mathbf{r}, \boldsymbol{\Omega}) = g(r) \quad (9.11)$$

Then,

$$\begin{aligned}
 \Phi_{00}^{011}(\mathbf{r}, \boldsymbol{\Omega}) &= \sqrt{3} \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} R_{00}^1(\boldsymbol{\Omega}) R_{00}^1(\hat{\mathbf{r}}) \\
 &+ \sqrt{3} \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix} R_{10}^1(\boldsymbol{\Omega}) R_{-10}^1(\hat{\mathbf{r}}) \\
 &+ \sqrt{3} \begin{pmatrix} 0 & 1 & 1 \\ 0 & -1 & 1 \end{pmatrix} R_{-10}^1(\boldsymbol{\Omega}) R_{10}^1(\hat{\mathbf{r}})
 \end{aligned} \tag{9.12}$$

such that

$$\begin{aligned}
 \Phi_{00}^{011}(\mathbf{r}, \boldsymbol{\Omega}) &= -3R_{00}^1(\boldsymbol{\Omega}) R_{00}^1(\hat{\mathbf{r}}) \\
 &+ 3R_{10}^1(\boldsymbol{\Omega}) R_{-10}^1(\hat{\mathbf{r}}) \\
 &+ 3R_{-10}^1(\boldsymbol{\Omega}) R_{10}^1(\hat{\mathbf{r}}) \tag{9.13} \\
 &= -3\boldsymbol{\Omega} \cdot \hat{\mathbf{r}} \tag{9.14}
 \end{aligned}$$

with $R_{00}^1(\boldsymbol{\Omega}) = \cos \theta$ and $R_{10}^1(\boldsymbol{\Omega}) = -\frac{1}{\sqrt{2}} \sin \theta e^{-i\phi}$.

We can see

$$\rho_{00}^{011}(r) = \frac{\int d\hat{\mathbf{r}} d\boldsymbol{\Omega} \rho(\mathbf{r}, \boldsymbol{\Omega}) \Phi_{00}^{011*}(\mathbf{r}, \boldsymbol{\Omega})}{\int d\hat{\mathbf{r}} d\boldsymbol{\Omega} \|\Phi_{00}^{011}(\mathbf{r}, \boldsymbol{\Omega})\|^2} = -\frac{1}{3} \frac{\int d\hat{\mathbf{r}} P(\mathbf{r}) \cdot \hat{\mathbf{r}}}{\int d\hat{\mathbf{r}} d\boldsymbol{\Omega} \|\boldsymbol{\Omega} \cdot \hat{\mathbf{r}}\|^2} \tag{9.15}$$

... (yes, in the implementation we see they are different curves. No exact equivalence.)

Chapter III

IMPLEMENTATION

The code MDFT developed in this thesis is based on the branch master of Git project MDFT (<https://github.com/maxlevesque/MDFT/>), version [Fri Jun 20 19:05:52 2014 +0200]. All the implementations are run on **POINCARÉ** machines of IDRIS, which involved two kinds of machines:

poincare[001-092]: 2 processors Sandy Bridge E5-2670 (2.60GHz, 8 cores per processor, being 16 cores per node); 32 GB of memory per node.

poincarebig[01-02]: 4 processors AMD Opteron 6282 (2.60GHz, 16 cores per processor, being 64 cores per node); 128 GB of memory per node.

The former is used for regular calculation, whose memory hierarchy is shown in figure 9.1. The later is only used in the evaluation of accuracy, in case of memory leak in the former.

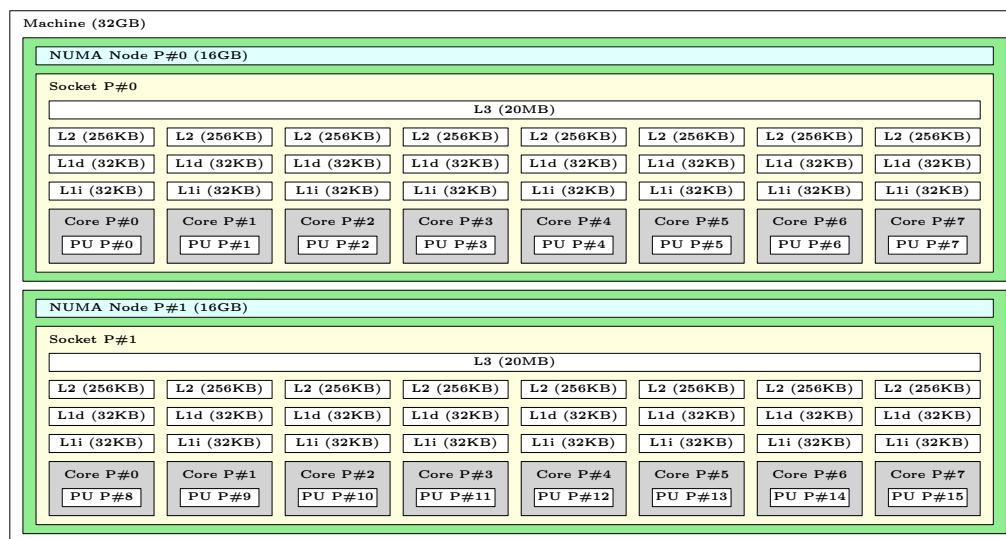


Figure 9.1: Structure of a POINCARÉ node

Section 10, gives a brief introduction of all the implementation algorithms built in this thesis.

Section 11 compares the accuracy of each algorithm, from parts of the code to entire branches.

Section 12 computing performance and memory limits for the sequential code. Due to the time limit, the parallelized version has not been totally built, it is put in the perspectives.

ALGORITHMS AND BRANCHES

According to the commutativity of operations (see §7.3.1), the only possible algorithms to evaluate $\gamma(\mathbf{r}, \Omega)$ from $\Delta\rho(\mathbf{r}, \Omega)$ are shown in the figure 10.1.

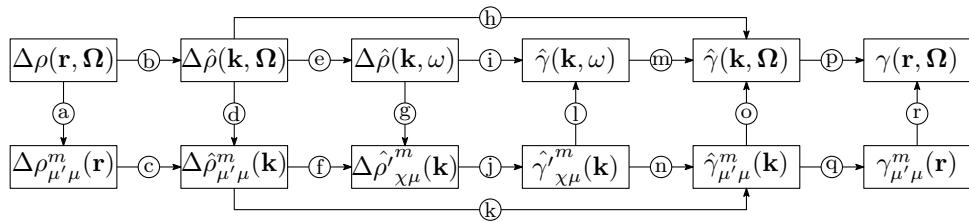


Figure 10.1: Possible algorithms for γ evaluation

Several branches are built to test and compare between algorithms, which are shown below in table 10.1 and will be detailed in the following context.

These branches should give numerically the same result in certain conditions, that will be discussed in later sections.

METHOD	SUB-METHOD	DESCRIPTION	THEORY
reference	dipole	calculate $n(r)$ and $P(r)$ separately	§5 [ref]
naive	standard	use $c_{\mu\nu,\chi}^{mn}(k)$ as input DCF	§6.2.2
	zero-order	use $\hat{c}(k, \omega_1, \omega_2)$ and take the nearest point	§6.1.1
	interpolation	use $\hat{c}(k, \omega_1, \omega_2)$ with linear interpolation	§6.1.2
	dipole	use c_S, c_Δ, c_D issue from [ref]	§6.2.1
	nmax1	use $c_S, c_\Delta, c_D, c_\pm$ issue from [54]	§6.2.1
convolution	standard	algorithm with symmetry reduction	§7.3.2
	asymm	algorithm without symmetry reduction	§7.3.2
	pure_angular	inverse FFT and FGSHT	§10

Table 10.1: Branch option in MDFT

10.1 BRANCHES "NAIVE"

Branches **naive** are the algorithms mentioned in section 6, which go through the path

$$(b) \rightarrow (h) \rightarrow (p)$$

in figure 10.1, calculating directly $\hat{\gamma}(\mathbf{k}, \Omega)$ from $\Delta\hat{\rho}(\mathbf{k}, \Omega)$. The difference between branches is the way to calculate $\hat{c}(\mathbf{k}, \Omega_1, \Omega_2)$. Branch **naive_standard** use $c_{\mu\nu,\chi}^{mn}(k)$ as input DCF. Branch **naive_zero-order** and **naive_interpolation** using $\hat{c}(k, \omega_1, \omega_2)$ with zero-order and linear interpolation, where the former is rejected in the implementation due to a lack of precision (appendix G).

10.2 BRANCHES "CONVOLUTION"

Branches **convolution_asymm** and **convolution_standard** are operational algorithms of angular convolution show in section 7, which go through the path

$$(a) \rightarrow (c) \rightarrow (f) \rightarrow (j) \rightarrow (n) \rightarrow (q) \rightarrow (r)$$

Branches **convolution_asymm** uses the original operational algorithm (§7.3) without symmetry reduction (§7.3.2), and **convolution_standard** with it.

Branch **convolution_pure-angular** goes through the path

$$(b) \rightarrow (d) \rightarrow (f) \rightarrow (j) \rightarrow (n) \rightarrow (o) \rightarrow (p)$$

which inverse the first and last two steps of the two algorithms mentioned above.

10.3 TESTING BRANCHES FOR $n_{\max}=1$

Branches `naive_dipole`, `naive_nmax1` pass by $(b) \rightarrow (h) \rightarrow (p)$, using DCF separately of the references [ref] and [54], whose slight difference is shown in §B.3.1. Branch `reference_dipole` use DCF in [ref], which is the original method in MDFT to calculate \mathcal{F}_{exc} via multipole expansion. In addition with branch `convolution_standard`, which can also use the two DCF mentioned above, a test of validation can be performed, which should at any case exactly numerically the same if the same DCF is used.

10.4 OTHER PATHS

Considering the necessity, other paths such as those passing by (i) and (k) are only built for local test usage (c. f. discussion in following sections).

NUMERICAL AND PHYSICAL ACCURACY

This chapter gives a systematic comparison between algorithms for the evaluation of the excess free energy \mathcal{F}_{exc} and its gradient γ in terms of accuracy. As the theory does not contain any unpredictable random part, the comportment of the code is mathematically predictable. For example, certain algorithms should give the same result at machine precision (10^{-13} to 10^{-15}) in certain conditions. A lost of accuracy comparing to the prediction can be classified as two different kinds. One is the theoretical lost, for example, certain equation is only valid for an infinite order. This kind of lost is unavoidable but should be worked out explicitly. Another source is the unknown lost, contains all kinds of incompatibility in the result that cannot explained mathematically. This kind of lost can mainly due to a bug in the implementation that cannot be find out, or it is a theoretical lost that haven't been worked out. They also deserve a discussion. All those comparisons aim to give a global view of the credibility of the results given by this code.

11.1 GENERALIZED SPHERICAL HARMONICS TRANSFORM

As discussed in §7.2, the function after a forward-backward GSHT process

$$F_{\mu'\mu}^m = \frac{f_m}{8\pi^2} \sum_{i=0}^{m_{\max}} w_i \sum_{j=0}^{2m_{\max}} \sum_{k=0}^{2[m_{\max}/s]} F(\Theta_i, \Phi_j, \Psi_k) R_{\mu'\mu}^{m*}(\Theta_i, \Phi_j, \Psi_k) \quad (11.1)$$

$$F(\Theta_i, \Phi_j, \Psi_k) = \sum_{m=0}^{n_{\max}} f_m \sum_{\mu'=-m}^m \sum_{\substack{\mu=-m \\ \text{mod } (\mu, s)=0}}^m F_{\mu'\mu}^m R_{\mu'\mu}^m(\Theta_i, \Phi_j, \Psi_k) \quad (11.2)$$

only remains the same when it is a polynomial of both $\cos \Theta$, $\cos \Phi$ and $\cos \Psi$ of order n_{\max} , where n_{\max} is the highest order of GSH in the expansion, and $m_{\max} = n_{\max}$ the order of quadrature used. However, as in the reality, the density variable ρ is not a simple polynomial, and the choice of m_{\max} and n_{\max} is tightly linked to the performance, it is important to know how much these choices will affect the results. The FFT process is implemented by package FFTW3 [18], which is verified to be leading to strictly no accuracy lost (at machine precision). That means the FGSHT process will have strictly with the GSHT process. Here we do not need to distinguish the two.

11.1.1 m_{\max} and n_{\max} of projections

The numerical error tests of a forward-backward GSHT process with different order n_{\max} of GSH and m_{\max} of quadrature are shown in table 11.1.

$m \setminus n$	0	1	2	3	4	5
0	0 (0)	9.00 (3.00)	34.00 (18.00)	83.00 (39.00)	164.00 (84.00)	285.00 (139.00)
1	0 (0)	0 (0)	0 (1.67)	4.34 (6.07)	7.06 (13.63)	14.88 (17.30)
2	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	5.65 (2.71)
3	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
4	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
5	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
(a) $f(\Omega) = 1$						
$m \setminus n$	0	1	2	3	4	5
0	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
1	0.96 (0.96)	0 (0)	0 (0)	2.56 (6.99)	10.76 (14.15)	13.83 (21.21)
2	0.46 (0.46)	0 (0)	0 (0)	0 (0)	0 (0)	1.36 (0.50)
3	0.86 (0.86)	0.66 (0.66)	0.66 (0.66)	0 (0)	0 (0)	0.66 (0.66)
4	0.99 (0.99)	0.80 (0.80)	0.80 (0.80)	0 (0)	0 (0)	0 (0)
5	0.83 (0.83)	1.01 (1.01)	1.01 (1.01)	0 (0)	0 (0)	0 (0)
(b) $f(\Omega) = \cos 3\Theta$						
$m \setminus n$	0	1	2	3	4	5
0	0 (0)	9.00 (3.00)	34.00 (18.00)	83.00 (39.00)	164.00 (84.00)	285.00 (139.00)
1	0 (0)	0 (0)	0 (1.67)	4.34 (6.07)	7.06 (13.63)	14.88 (17.30)
2	1.00 (1.00)	1.00 (1.00)	0.50 (0.50)	1.53 (1.53)	1.15 (1.15)	3.65 (0.89)
3	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	0.83 (0.83)	1.10 (1.10)	1.11 (1.11)
4	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	0.90 (0.90)	0.90 (0.90)	0.69 (0.69)
5	1.00 (1.00)	1.00 (1.00)	1.00 (1.00)	0.94 (0.94)	0.94 (0.94)	0.80 (0.80)
(c) $f(\Omega) = \cos 3\Phi$						
$m \setminus n$	0	1	2	3	4	5
0	0 (0)	5.03 (1.68)	19.01 (10.06)	46.40 (21.80)	91.68 (46.96)	- (77.70)
1	0 (0)	0 (0)	0 (0.51)	1.32 (1.85)	2.15 (4.15)	4.53 (5.26)
2	0.56 (0.56)	0.56 (0.56)	0.07 (0.07)	0.55 (0.55)	0.76 (0.76)	2.05 (1.00)
3	0.47 (0.47)	0.47 (0.47)	0.47 (0.47)	0 (0)	0.46 (0.46)	0.46 (0.46)
4	0.56 (0.56)	0.56 (0.56)	0.56 (0.56)	0 (0)	0 (0)	0 (0)
5	0.51 (0.51)	0.51 (0.51)	0.51 (0.51)	0 (0)	0 (0)	0 (0)
(d) $f(\Omega) = R_{30}^3(\Omega)$						

Table 11.1: Maximum absolute error E_a^{\max} introduced by a forward-backward GSHT process of function f (outside the parentheses) and the corresponding transform for function with C_{2v} symmetry (with two times less Ψ points of quadrature, inside the parentheses). Differences should be theoretically null in the table is in bold character.

It conforms the theoretical prediction where should lead to no accuracy lost ($m_{\max} > n_{\max}$) and f the polynomial that can be expanded on GSHs of order at most n_{\max}). It should be noted, that as we do firstly a forward then a backward transform, when $m_{\max} < n_{\max}$, even the input function is of order at most m_{\max} , the output function is of order n_{\max} in the presence of $R_{\mu'\mu}^m$ which is of order n_{\max} , thus the two functions are different. The function in which the order of $\cos \Phi$ and $\cos \Psi$ is greater than $\cos \Theta$ (case of $f(\Omega) = \cos 3\Phi$), the two functions are different too, because it cannot be expanded to a finite number of GSHs.

11.1.2 From ρ to γ

To conclude from above, the error for an arbitrary function like ρ , which is not a combination of GSHs, can be huge. One evidence is the appearance of unphysical density $\rho(\mathbf{r}, \Omega) < 0$ ($\Delta\rho(\mathbf{r}, \Omega)/\rho_0 < -1$) at certain point after a forward-backward GSHT process (figure 11.1). (other better way?)

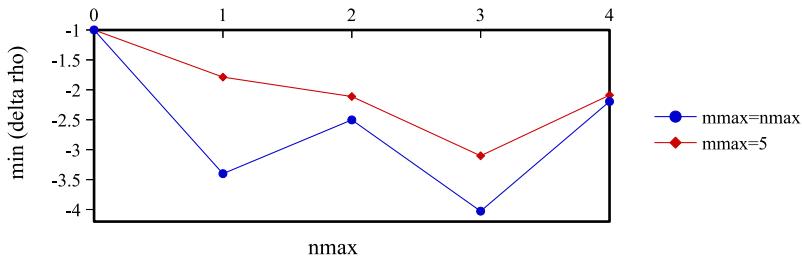


Figure 11.1: The minimum value of $\Delta\rho(\mathbf{r}, \Omega)/\rho_0$ after a forward-backward GSHT process with respect to n_{\max} . Computed for a 45^3 grid ($L = 25$) for a converged density of an artificial charged LJ center $\text{CH}_4^{+0.4}$.

Theoretically, we expect this minimum value approach to zero when increasing m_{\max} or n_{\max} , which is not exactly the case. That means perhaps the order of expansion is still far from to find a tendency. Knowing that $\rho(\mathbf{r}, \Omega)/\rho_0 \rightarrow 0$ at center of the solute and $\rho(\mathbf{r}, \Omega)/\rho_0 \rightarrow 1$ far from the solute, the error can be said oblivious within the computing capacity ($n_{\max} < 5$), that means we cannot expand rightly the density ρ on GSH projections, where. But this have a much less effect on the functional gradient γ that we

evaluate, because in a convolution product, $\Delta\rho(\mathbf{k}, \Omega)$ and the DCF $\hat{c}(k, \Omega_1, \Omega_2)$ can be both expended, and product of higher order terms vanishes more easily. Latter we will show that the profile of γ and the free energy \mathcal{F}_{exc} can already converge within $n_{\text{max}} < 5$.

11.2 COMPARISON BETWEEN BRANCHES

As shown in figure 10.1, if we fixe $\Delta\rho(\mathbf{r}, \Omega)$ to a recombination of GSH projections, all methods using the same DCF should give mathematically identical results. The most direct comparison is the free energy evaluated during 1 iteration. And to be more strict, is also interesting to compare the profile of γ .

11.2.1 *Difference in energy evaluation*

As shown in table 11.2, the methods using the same DCF at the same m_{max} which is mathematically identical in an infinite condition, give nearly the same results.

METHOD	n_{max}	DCF	FREE ENERGY (kJ/mol)
dipole	1	[ref mdft]	13.1915264499904339
naive_dipole	1	[ref mdft]	13.1915269013357985
naive_nmax1	1	[54]*	18.6052247636086854
convolution_standard	1	[54]*	18.6093390102806886
naive_interpolation	2	[54]	26.8444355457069044
naive_standard	2	[54]	26.9897310488084479
convolution_standard	2	[54]	26.9163932581793155
convolution_asymm	2	[54]	26.9163932581793155
convolution_pure-angular	2	[54]	26.9163932581793155

Table 11.2: Free energy calculated during 1 iteration for a 32^3 grid ($L = 20\text{\AA}$) for a fake LJ center $\text{CH}_4^{+0.33}$, using a converged density as input. Here $m_{\text{max}} = n_{\text{max}}$. “*” means ancient DCF, not so much difference.

The light difference between `naive_nmax1` and `convolution_standard` at $m_{\max} = 1$ is due to the artificial decoration at k -border showing in later section, and the difference between `naive_interpolation` and `naive_standard` and `convolution` methods are also acceptable, which is natural to be lightly different due to the interpolation error and the GSH expansion (the ρ recombination of GSH projections will be discussed later). This supports by the way that we do not need the same order of GSH expansion for γ than for $\Delta\rho$.

11.2.2 A single k -kernel

Firstly, we are interested in the local paths from $\Delta\hat{\rho}_{\mu'\mu}^m(\mathbf{k})$ to $\hat{\gamma}_{\mu'\mu}^m(\mathbf{k})$, that can be tested independently for a certain \mathbf{k} . As shown in figure 11.2, four algorithms are available to the purpose.

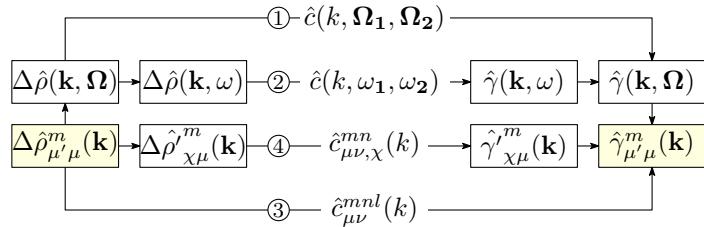


Figure 11.2: Schema of a k -kernel test

The program that compares each element of $\hat{\gamma}_{\mu'\mu}^m(\mathbf{k})$ issued from these 4 algorithms for a given $\Delta\hat{\rho}_{\mu'\mu}^m(\mathbf{k})$ is done by Mr. Luc Belloni, which shows that the $\hat{\gamma}_{\mu'\mu}^m(\mathbf{k})$ for the four algorithms are strictly identical. This means, the final result of energy and structure is independent to the choice of path inside a k -kernel, if $\Delta\hat{\rho}(\mathbf{k}, \Omega)$ can be fully expended on GSHs.

11.2.3 *k-border effect*

Here we test the whole process shown in figure 10.1, with $\Delta\rho(\mathbf{r}, \Omega)$ generated from a recombination of GSH projections. Firstly, we compare the three **convolution** algorithms passing by GSH expansion. For a 64^3 grid, $n_{\max} = 3$, the three algorithms **convolution_standard**, **convolution_asymm**, and **convolution_pure_angular** gives the same free energy, but lightly different result when comparing each element of $\gamma(\mathbf{r}, \Omega)$, and this difference seems to decrease when increase the number of grid points. More, the projections $\gamma_{\mu'\mu}^m(\mathbf{r})$ which should be purely real as explained in §7.3.2, have a light imaginary part. But surprisingly, for a 65^3 grid, it gives numerically the same result for both the three algorithms at machine precision.*

The difference between these methods is found to be a special *k*-border effect linking to even number grids.

As the symmetry

$$\Delta\hat{\rho}_{\chi\mu}^m(\mathbf{k}) = (-)^{m+\mu+\chi}\Delta\hat{\rho}_{\chi,-\mu}^{m*}(-\mathbf{k}) \quad (11.3)$$

* The detailed value of γ which the paragraph of description is based on haven't been noted, as it was regarded as a bug in the code at that time, and

$$\Delta\hat{\rho}_{\mu'\mu}^m(\mathbf{k}) = (-)^{\mu'+\mu}\Delta\hat{\rho}_{-\mu',-\mu}^{m*}(-\mathbf{k}) \quad (11.4)$$

the code had been then modified;

$$R_{\mu'\chi}^m(\hat{k}) = (-)^{m+\mu'+\chi}R_{-\mu',\chi}^m(-\hat{k}) \quad (11.5)$$

and to redo such a process takes a lot of time.

For the points “at border”, it's to say that after the FFT where the point having $\pm k_i = k_i^{\max}$, $i = 1, 2, 3$, for example for k_1 ,

$$\Delta\hat{\rho}_{\mu'\mu}^m(\pm k_1, k_2, k_3) = \Delta\hat{\rho}_{\mu'\mu}^m(k_1^{\max}, k_2, k_3)$$

is naturally put in the same array by FFT for the grids having even number, as shown in For example, for a grid 1D, the FFT having 6 points gives the

figure 11.3.

0	1	\cdots	$k - 1$	k	$-k + 1$	\cdots	-1
				-k			

Figure 11.3: k -border effect

As FFT possesses periodicity, the symmetry 11.7 can always be respected at border.

However, as

$$R_{-\mu',\chi}^m(-\hat{k} \equiv (-k_1, -k_2, -k_3)) \neq R_{\mu',\chi}^m(k_1^{\max}, -k_2, -k_3) \quad (11.6)$$

the symmetries (11.5) and (11.3) are not respected for these points. In the backward process, if we make sense of all the $\gamma_{\mu'\mu}^m(\mathbf{k})$, as

$$\gamma_{\mu'\mu}^m(-\hat{k} \equiv (-k_1, -k_2, -k_3)) \neq \gamma_{\mu'\mu}^m(k_1^{\max}, -k_2, -k_3)$$

the symmetry

$$\gamma_{\mu'\mu}^m(\mathbf{k}) = (-)^{\mu' + \mu} \gamma_{-\mu', -\mu}^{m*}(-\mathbf{k}) \quad (11.7)$$

is not respected totally, and this imposes that $\gamma_{\mu'\mu}^m(\mathbf{r})$ have a imaginary part. This imaginary part has been omitted implicitly in the “real to complex” FFT process of used in for example **convolution_standard**, for FGSHT, or **convolution_pure-angular** for FFT3D process. It is to say, we keep only the part of none-negative \mathbf{k} or none-negative μ , supposing that the part we omit respects the symmetry.

The right way to treat this issue is to artificially impose at the border:

$$R_{\mu',\chi}^m(k_i^{\max}) = \frac{1}{2} [R_{\mu',\chi}^m(k_i) + R_{\mu',\chi}^m(-k_i)] \quad (11.8)$$

where i is the conflict index in figure 11.3. If more than one dimensions are in conflict, this process can be done twice (4 terms for “edges” of the cube) or three times (8 terms for “vertices”). The point $\mathbf{k} = \hat{0}$ is different, as it was define along z axes to avoid

implementation crash, it doesn't respect eq. (11.5) and (11.3) neither. But this point compared to hundreds thousands of total points is negligible.

The energies given by **naive_standard** and the **convolution** algorithms are identical for a 65^3 and $n_{\max} = 3$ grid, but the element of $\gamma(\mathbf{r}, \Omega)$ have a mysterious difference at order of 10^{-2} , seemed to be aleatory. A test redone for a 45^3 grid is shown in figure 11.4.

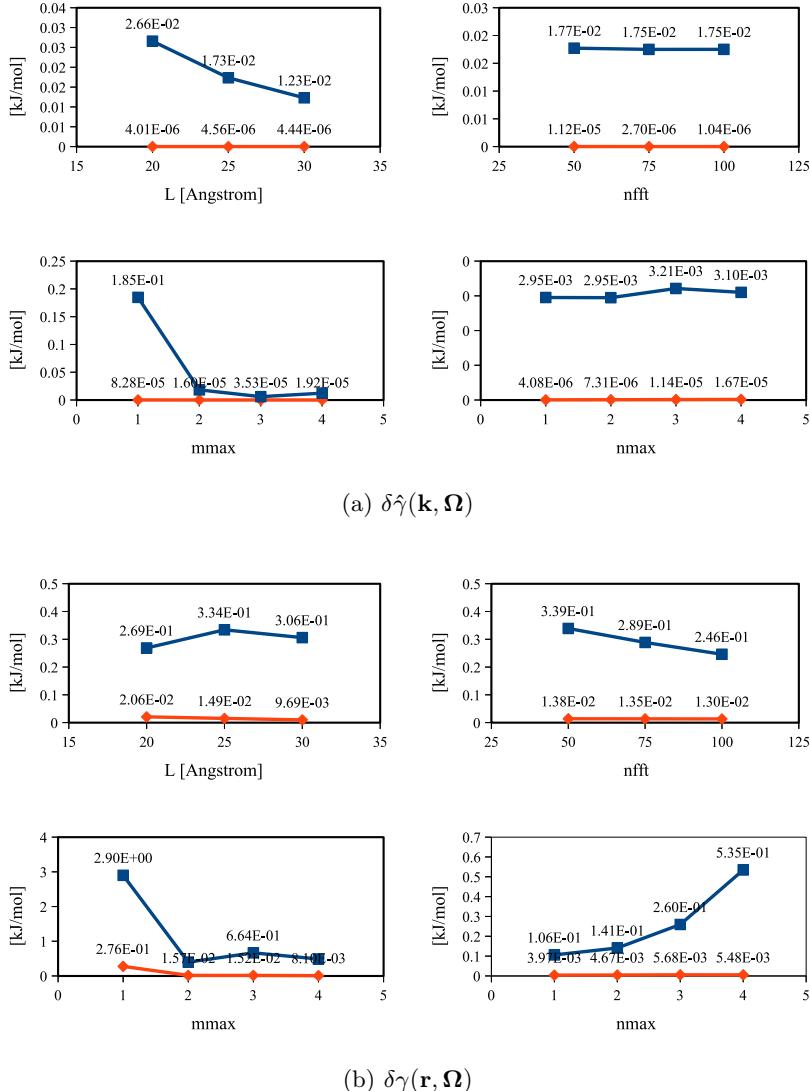


Figure 11.4: Maximum and average difference in $\hat{\gamma}(\mathbf{k}, \Omega)$ and $\gamma(\mathbf{r}, \Omega)$, for tests of different box length L , different number of grid n_{fft} in one dimension, $n_{\max} = 1, 4$ for $m_{\max} = n_{\max}$, and $n_{\max} = 1, 4$ for $m_{\max} = 5$.

We can conclude very rudely that this error depends on the angular quadrature m_{\max} . The dependence is natural, as the difference between algorithms **naive** and **convolution** is the treatment of the angular part. There is also a dependence on L in the k -space, but

after FFT it is mixed. The augmentation of error in the n_{\max} chart is unnatural, implying there is perhaps a still bug in the code.

In a word, this mysterious difference cannot be yet explained, as the **naive** methods does not have the k -border effect linked to symmetry, on the other hand we used a odd grid, we could not yet distinguish that it is a bug in the implementation, in the test or in the theory. The projections $\gamma_{\mu\nu}^{mnk}(r)$ of this two algorithms seems to be identical (figure 11.5), it is to say, the global structure of this two algorithms are almost the same, and the error would not be very decisive.

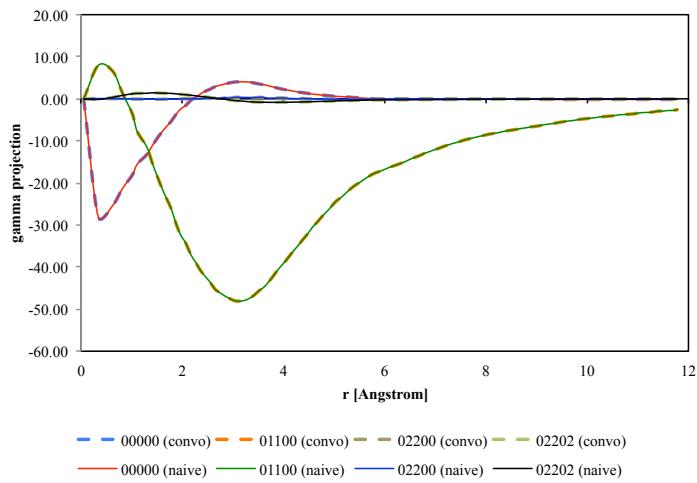


Figure 11.5: A selection of rotational invariant projections $\gamma_{\mu\nu}^{mnk}(r)$ for a 65^3 grid

11.3 INTRINSIC VARIATION OF FREE ENERGY

Before study of free energy dependence on angular algorithms, we are interested in the grid dependance, which can have an influence to the tests later.

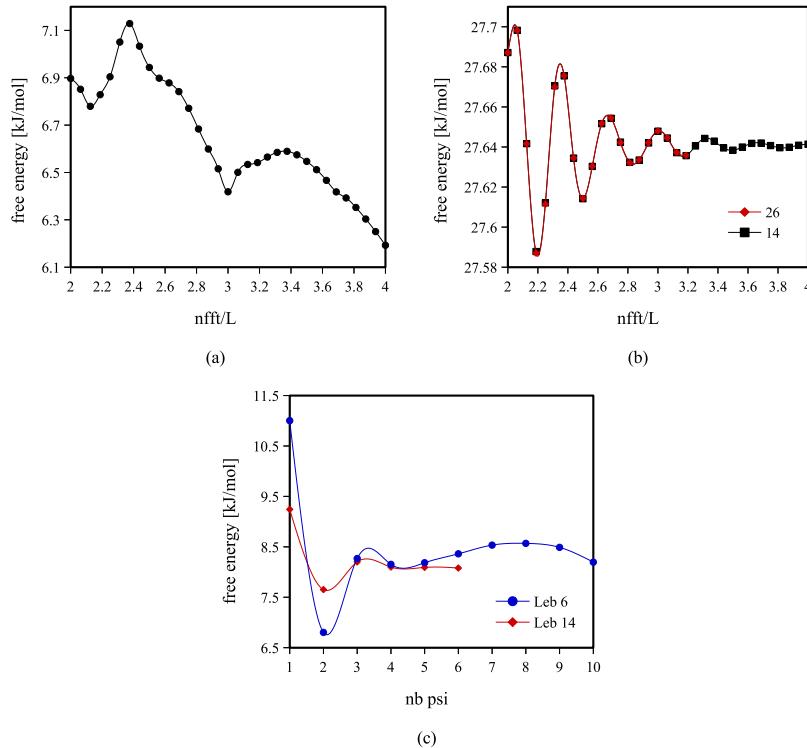


Figure 11.6: Space-grid and Ψ dependence of code MDFT. $L = 32$. (a) $\text{CH}_4^{+0.33}$ using dipole DCF with $m_{\max} = 1$; (b) CH_4 using DCF of $m_{\max} = 5$, Lebedev quadrature of order 1 and 2; (c) acetone using dipole DCF and Lebedev quadrature, varying Ψ , $nfft=128$.

As shown in figure 11.6 (a) and (b), there is a dependence of calculated free energy on the space grid resolution. for a charged solute, the energy has tendency to decrease when increasing the resolution of grid (nfft). And this decrease does not link to the border correction mentioned in §8, as the box length and the charge remain the same for the whole set of test. From (b) we consider that at least 3 points grid in one dimension per Angstrom is needed to reduce the uncertainty due to grid resolution. Figure (c) fixed the Lebedev quadrature for Θ and Φ , but leaved varying the Ψ . We can also see a dependence on Ψ . which does not vanish when increasing the resolution of grid. As during the whole thesis the Ψ is theoretically fixed in the same order with Θ and Φ , this remains an issue for further verification. We can roughly conclude that an error around 1 kJ/mol is common for this code.

11.4 SERIES OF CHARGED LJ CENTRE

To validate the method, we chose a series of LJ centre, which possess the LJ parameters of

For both IET, CH₄ in [ref], and have a various charge from -1.0 to 1.0 (table 11.3).

and DM results,

298K is used

according to

habitude instead

of 303K

recommended in

reference [2]. For

11.4.1 Box length dependance and charge dependance of free energy

MDFT, 300K

and 298K are

As discussed in section 8, for single ions, two types of corrections need to be added on used.

the free energy, which depend on the box length and and charge of the ion. To verify these dependence, we implement a systematic calculation from charge, using 3 different methods, where the parameters are shown in table 11.4. It should be noted that, the **naive_interpolation** only used 14 Lebedev and 3 Ψ angles to converge, which gives exactly the same result with 26 Lebedev and 4 Ψ angles, that means, the **naive** methods do not need an order of quadrature m_{\max} to be greater than the order of DCF n_{\max} . The -1 side has problem of convergence, and all the converged results are presented.

METHOD	nfft/ L	m_{\max}	n_{\max}
naive_nmax1	3	1	1
naive_interpolation	3	2 (Leb)	5
convolution_standard	3	1	1

Table 11.4: Methods and parameters for CH₄ series test. * Leb is Lebedev quadrature, with is mathematically equivalent with Gauss-Legendre quadrature but only $\sim 2/3$ angles.

The direct results collection are shown in figure 11.13, 11.14 and 11.15 at the end of this section. We can see that the dependence of box length for each charge is almost

linear, except the charge between $[-0.2, 0.2]$. This means, the influence of box length is much greater than the intrinsic variation of result that mentioned in 11.3. The charge dependency is traced in figure 11.7, using all the number of slopes with respect to the square of their charge (q^2). A linear regression is done to give the slope $1937.8 \text{ kJ} \cdot \text{mol}^{-1} \cdot \text{\AA}$. This slope correspond to the correction of type-B (normalized to give the right unity):

$$\frac{\xi}{2} \left(1 - \frac{1}{\varepsilon}\right) = 1943.2 \text{ kJ} \cdot \text{mol}^{-1} \cdot \text{\AA} \quad (11.9)$$

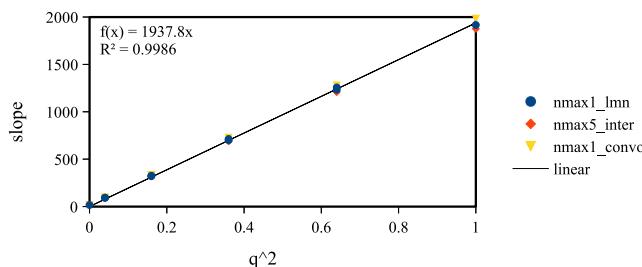


Figure 11.7: Quadratic charge dependence of free energy of CH₄ centre series

The intercept values in each of figure 11.13 to 11.15 correspond to the free energy of an infinite box. The IET results are done with $R_{\max} = 102.4\text{\AA}$, and need a correction of $-2.556k_B T$. (need more details.) The differences in free energy between MDFT and IET are given in figure 11.8. The linear regression is done with all existing points in this figure. The slope $87.653 \text{ kJ} \cdot \text{mol}^{-1}$ corresponds to the correction of type-C (normalized to give the right unity):

$$\eta\gamma = 82.104 \text{ kJ} \cdot \text{mol}^{-1} \quad (11.10)$$

These two numbers are a little different, principally due to a lack of point at the -1 side.

11.4.2 Comparison with IET after corrections

The figure 11.8 after correction with eq. (11.10) gives figure 11.9. It is shown that they are not perfectly agreed with each other. The $n_{\max} = 1$ methods have large difference

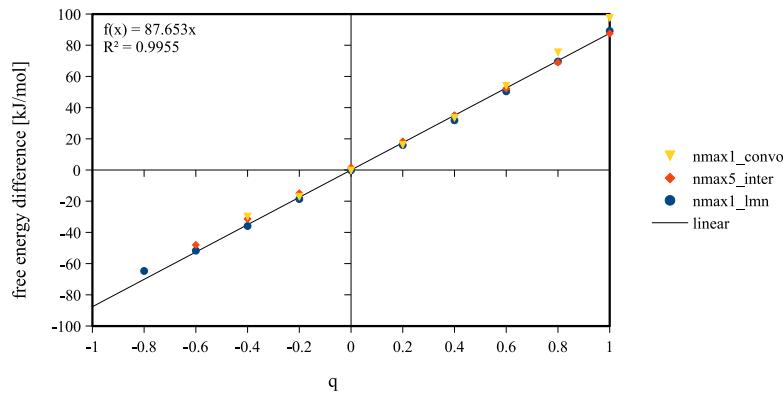


Figure 11.8: Comparison to IET, without P-scheme correction

when the charge increased, especially the **convolution_standard** using GSH expansion. Knowing that during 1 iteration, the **naive_nmax1** and the **convolution_standard** only give a slight difference in free energy (table 11.2). The $n_{\text{max}} = 5$ have an energy shift about $2 \text{ kJ} \cdot \text{mol}^{-1}$, which cannot have an explanation. but overall, to have $2 \text{ kJ} \cdot \text{mol}^{-1}$ per $100 \text{ kJ} \cdot \text{mol}^{-1}$ is already a good result.

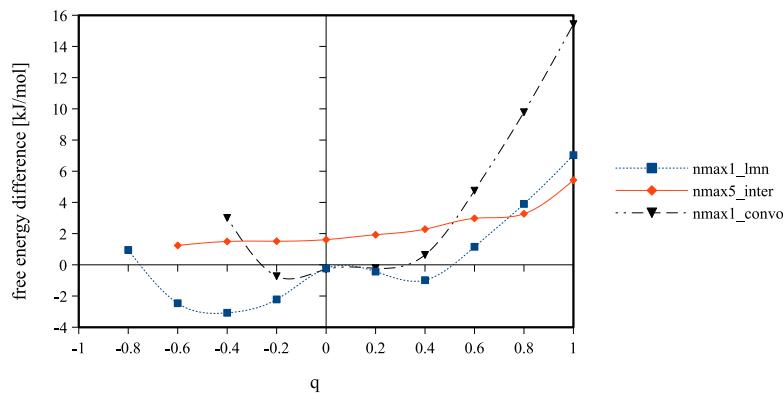


Figure 11.9: Comparison to IET, with P-scheme correction

The case of $m_{\text{max}} = 5$, $n_{\text{max}} = 0, \dots, 5$ with **convolution_standard** is shown in figure 11.10. It is interesting to see how energy evaluate with n_{max} while fixing m_{max} . As we said, γ is more smooth than ρ , that means we can have $n_{\text{max}} < m_{\text{max}}$ to economize computing cost. Results shows that within $n_{\text{max}} \geq 3$ for $n_{\text{max}} = 5$, the error is acceptable. But again, the dependence on q after correction is in incomprehensible. And compared to **naive_interpolation** in figure 11.9, we see that these error for $n_{\text{max}} = 5$ is different.

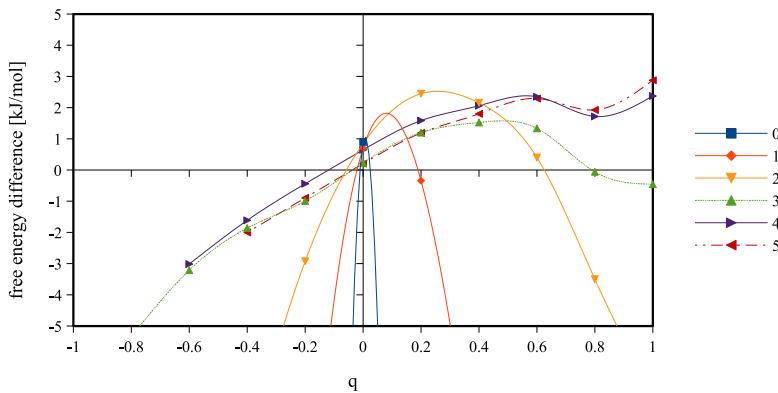
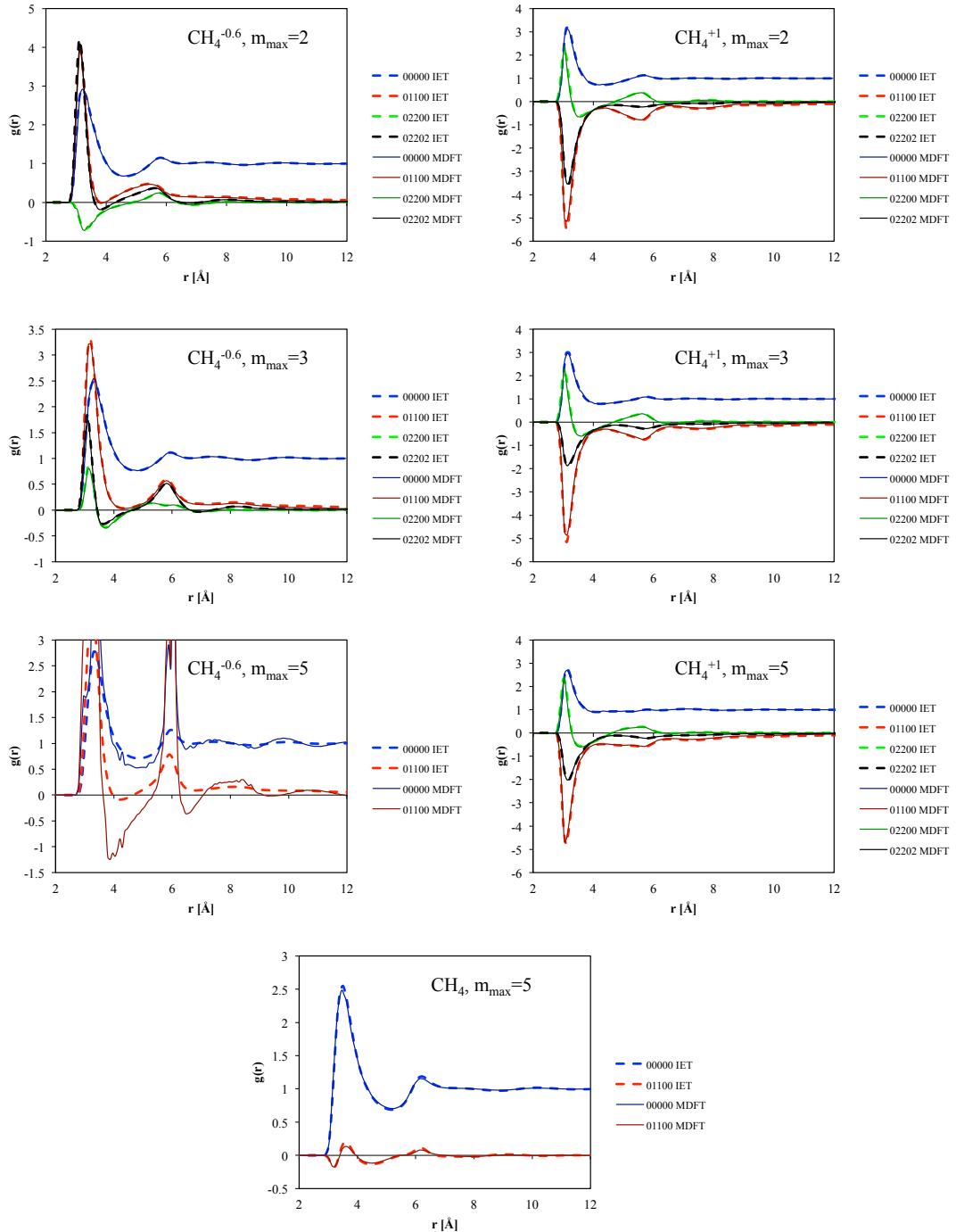


Figure 11.10: Comparison to IET, with P-scheme correction, $m_{\max} = 5$, $n_{\max} = 0, \dots, 5$

The profile of ρ can be expanded on rotational invariants which is discussed in §9. The comparison with IET is done for three charges, 0, -0.6 and +1. Shown in figure 11.11. Watch that the 0 and -1 one of $m_{\max} = 5$ corresponds well the result of IEM, but the -0.6 one have a lot of noise. (In fact, this configuration had difficulty to converge, and the given energy is not good, thus it is deleted in figure 11.10.) The profiles of go much better $m_{\max} = 2, 3, 4$. Normally, more points means more precision. It may mean that with $m_{\max} = 5$, there is perhaps a bug of integer overflow that prevent the convergence for high charges.

Figure 11.11: Comparison to IET. Profile of ρ in rotational invariant projections, $L=24$, $nfft=72$.

11.4.3 Comparison with MD

The comparison of RDF with MD results are shown in figure 11.12. We can see that for negative charges, there are a lot of shifts.

Discussion...

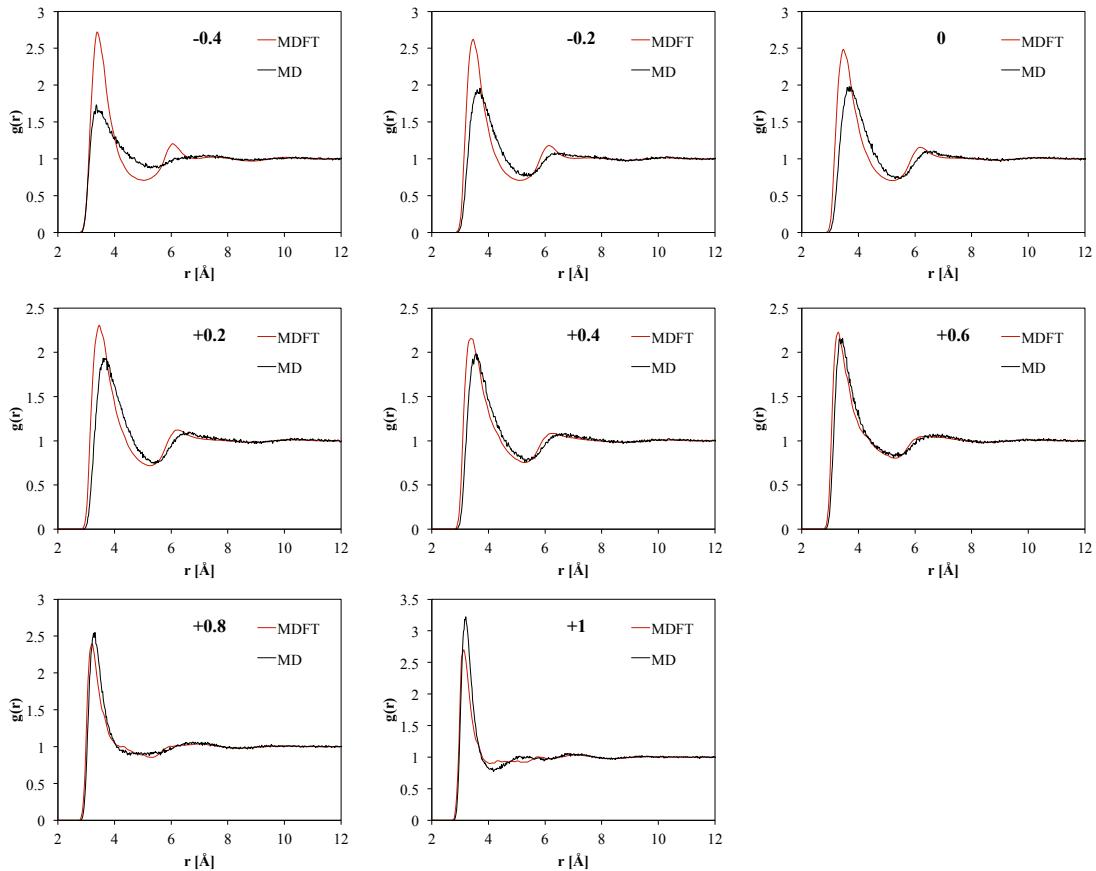


Figure 11.12: Comparison to IET. Profile of ρ in rotational invariant projections.

11.5 PREMIER CONCLUSION

From the results, we see that MDFT ... “capable” to produce the same result with IEM for single ions. (but have more ability to calculate 3D molecules which is not suitable for spherical coordinates...)

naive_interpolation is more stable compared to **convolution** methods and can use less angles for convergence, although in the computing time it cannot compare with **convolution** methods, which will be discussed later.

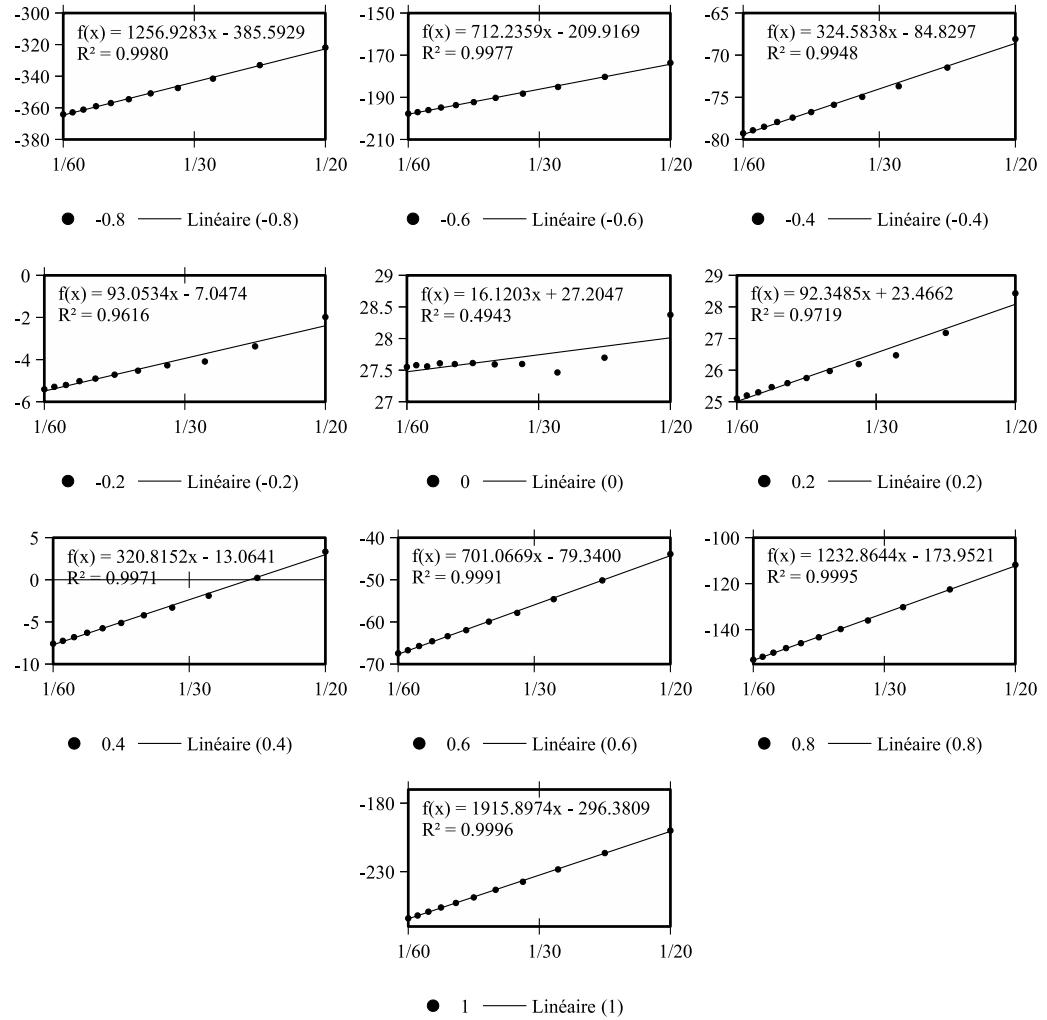


Figure 11.13: Free energy (without correction) of charged CH₄ centre (-1.0 to 1.0) with respect to the box length, for `naive_nmax1` method, with $m_{\max} = n_{\max} = 1$, at 300K.

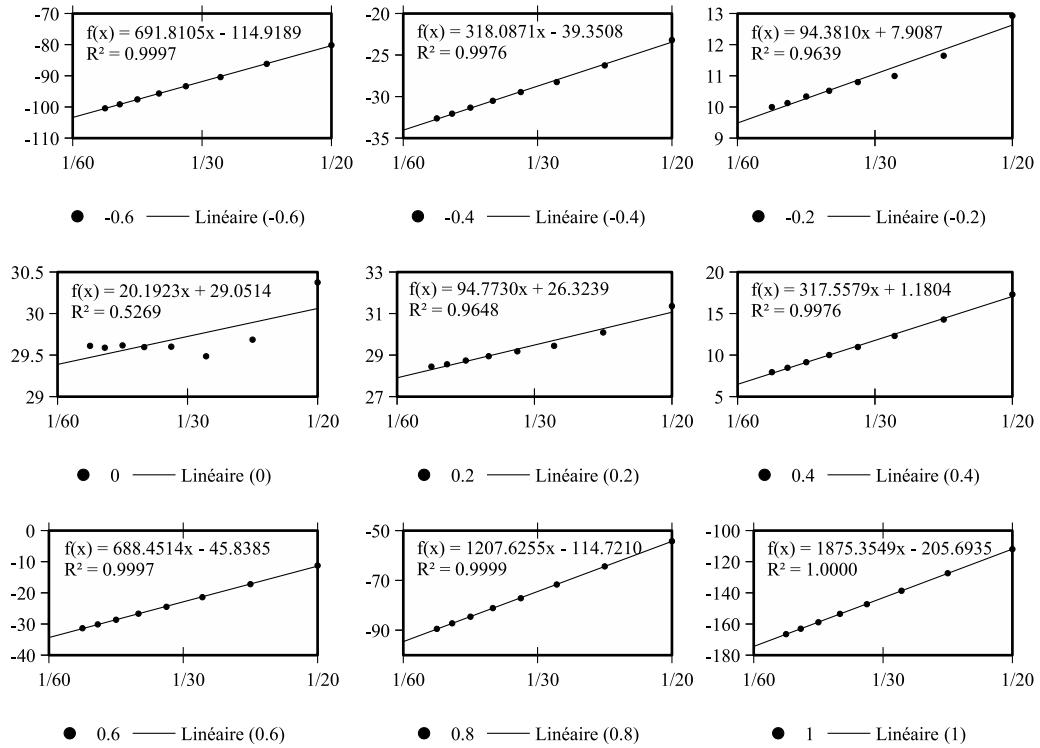


Figure 11.14: Free energy (without correction) of charged CH₄ centre (-1.0 to 1.0) with respect to the box length, for `naive_interpolation` method, with 14 angles of Lebedev quadrature angles for Θ and Φ , 3 for Ψ , DCF of $n_{\max} = 5$, at 300K.

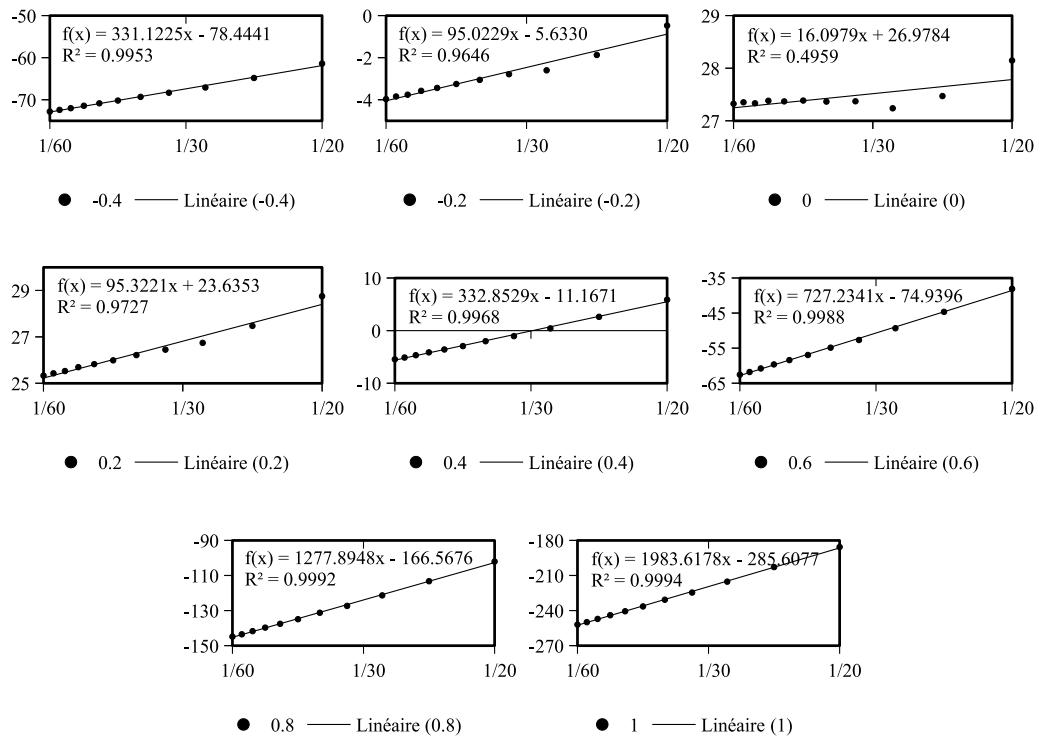


Figure 11.15: Free energy (without correction) of charged CH₄ centre (-1.0 to 1.0) with respect to the box length, for `convolution_standard` method, with $m_{\max} = n_{\max} = 1$, at 298.15K.

COMPUTING PERFORMANCE OF SEQUENTIAL CODE

This section evaluates the computing performance of the code. Our goal is to show that the new algorithm of angular convolution is faster than the old naive one, and the huge amount of simulation has shown that it is absolutely the case. But a raw result, where the implementation goes for an indefinite number of iterations during minimization, cannot give a proper and systematic performance evolution. This gives the propose of this section.

As discussed in appendix A, two main factors give influence to the performance of a sequential code: the algorithm complexity, and the memory delay. To study the algorithm complexity, testing with respect to parameters is done to some simple but important components. The result can match the theoretical algorithm complexity, or completely different due to the overhead of function calling or the inhomogeneity of memory access. Etude of this small parts permits a further understanding of the entire code.

12.1 FFT

The FFT play an important role in the implementation, which is used by the spatial convolution and the FGSHT process.

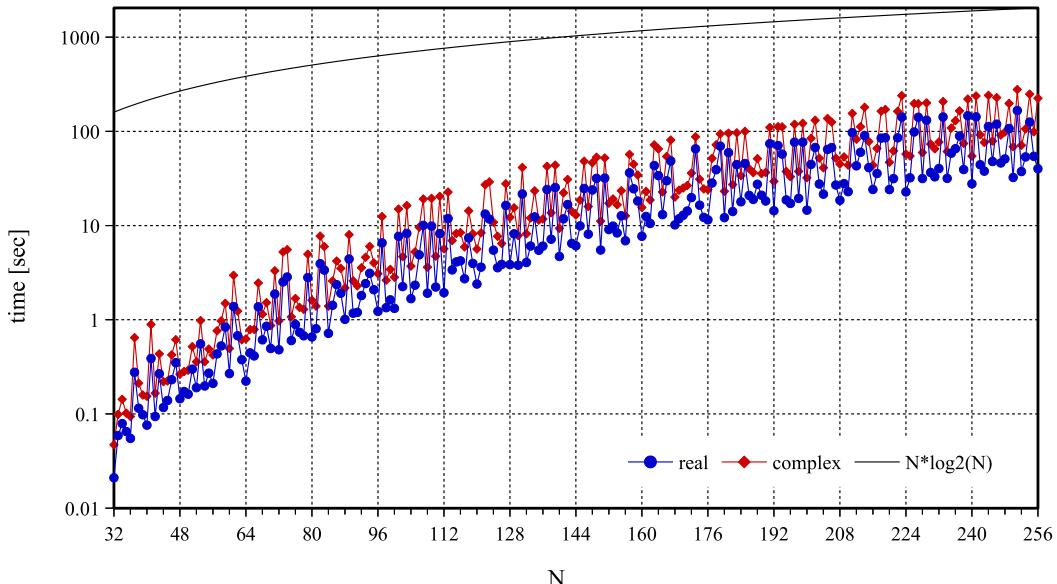


Figure 12.1: timing FFT

As shown in figure 12.1, the dependance on $O(N \log_2 N)$ [Numerical_Recipes_3ed] doesn't totally exist, but of the same form, depending the algorithm of FFT [ref dft]. It should be noted that a grid of prime number is always at the peaks in the figure, that means it can be 2 or more time longer than that of the composite number around. That means we should use a even number grid, where the k -border correction in §11.2.3 is absolutely needed. Apart from this conclusion, to compare between the algorithms for angular part involved in this thesis, we are not really interested in computing performance with respect to the number of spatial grid. However, the ratio of real and complex FFT timing is important, as shown in figure 12.2, which is near the theoretical ratio 0.5. For example, we process n_{angle} real to complex FFT, then $n_{\text{spatial}}/2$ complex to complex FGSHT, or we process n_{spatial} real to complex FGSHT, then $n_{\text{proj}}/2$ complex to complex FFT, should not give a great difference if $n_{\text{angle}} \sim n_{\text{proj}}$ for small n_{max} . If it is not the case, it will have an influence to the choice of algorithm.

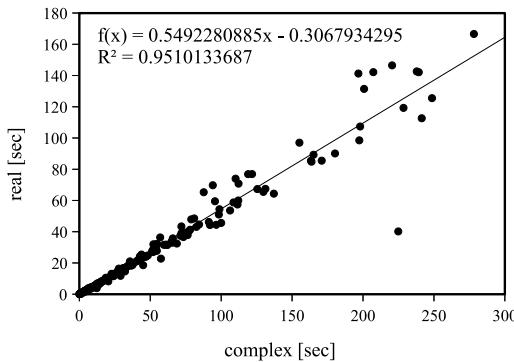
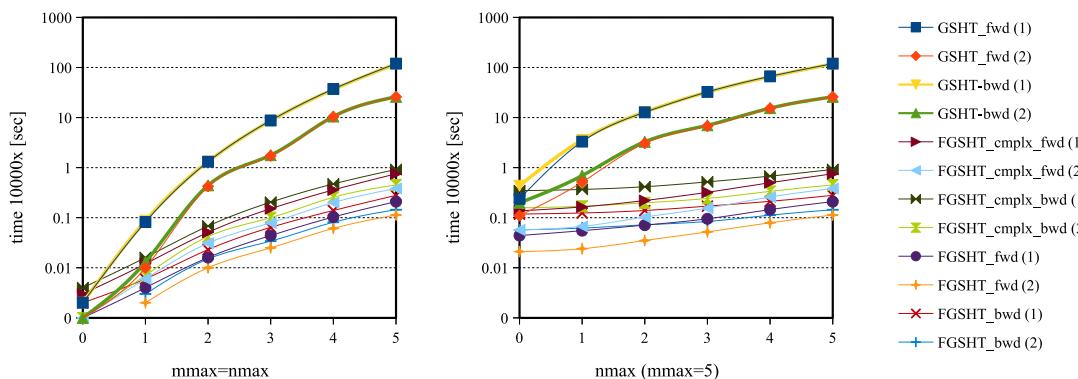


Figure 12.2: FFT real to complex

12.2 FGSHT

The computing times of GSHT and FGSHT are shown in figure 12.3. There is no interest to see in detail how much FFT have accelerate the GSHT process, but clearly FGSHT can be 100 times faster than GSHT, and GSHT for the symmetry of Ψ , $s = 1$ is in average 5 times longer than $s = 2$. As accuracy test shows that GSHT and FGSHT gives exactly the same result, it is free to utilize FGSHT in all the case to have a faster performance.

Figure 12.3: Computing time of GSHT and FGSHT (per 10000 times), between parentheses is the order of symmetry axes s

However, it is important to know the ratio between real and complex FGSHT process for the same reason as FFT. It is shown that this number is 0.3 in all cases, and it does not depend on n_{\max} . The difference between these two is that the real one performs real-to-complex FFT for the Φ, Ψ grid and calculate only a half plus a bit of projections ($\mu \geq 0$)

than the complex one. Normally, the ratio should be greater than 0.5. This means there maybe an extra process in the complex one or it is controlled by the memory. In a word, the final result 0.3 means, that doing n_{spatial} real to complex FGSHT takes only 0.6 times of doing $n_{\text{spatial}}/2$ complex to complex FGSHT, which means in **convolution_standard** we use less time to compute FGSHT than in **convolution_pure-angular**.

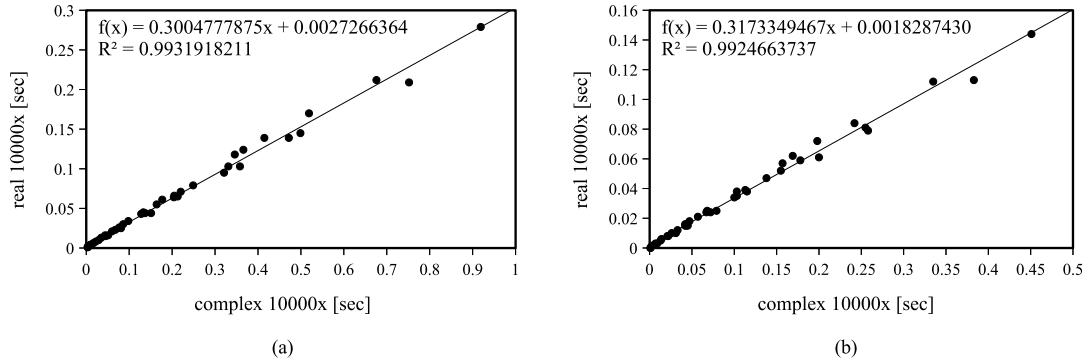


Figure 12.4: FGSHT real to complex

12.3 k -KERNEL

As discussed in the previous section, the final result of energy and structure is independent to the choice of path inside a k -kernel. That means, it is free in precision cost to choose the fast path. As path (1) and (2) in figure 11.2 introduce the transform from $\Delta\hat{\rho}_{\mu'\mu}^m(\mathbf{k})$ to $\Delta\hat{\rho}(\mathbf{k}, \Omega)$ that has no interest in timing, and the entire branches will be compared in later implementation, here we only compare the path (3) and (4), which correspond to eq. (7.5) and (7.12).

The theoretical predictions of the computing time of OZ equation with respect to n_{max} are listed in table 7.1. If the OZ equation is the most time demanding part, the result should have the same proposal. figure 11.2 shows the experimental timing of the whole path (3) and (4).

It is shown that... (There is a problem of code that gives backtrace but obviously (4) is 100 times faster than (3).)

12.4 ENTIRE ITERATION OF \mathcal{F}_{exc} EVALUATION

Apart from all the **naive** methods that will be discussed in §12.4.1, figure 12.5 shows all the comparable **convolution** timing data. We can see **convolution_standard** is the fastest algorithm, and OZ equation is not the longest part in the iteration. All the tests are performed for a $L = 24$, $\text{nfft} = 72$ grid, with 4 series: the three **convolution** methods with $m_{\text{max}} = n_{\text{max}}$, and **convolution standard** with $m_{\text{max}} = 5$, varving n_{max} .

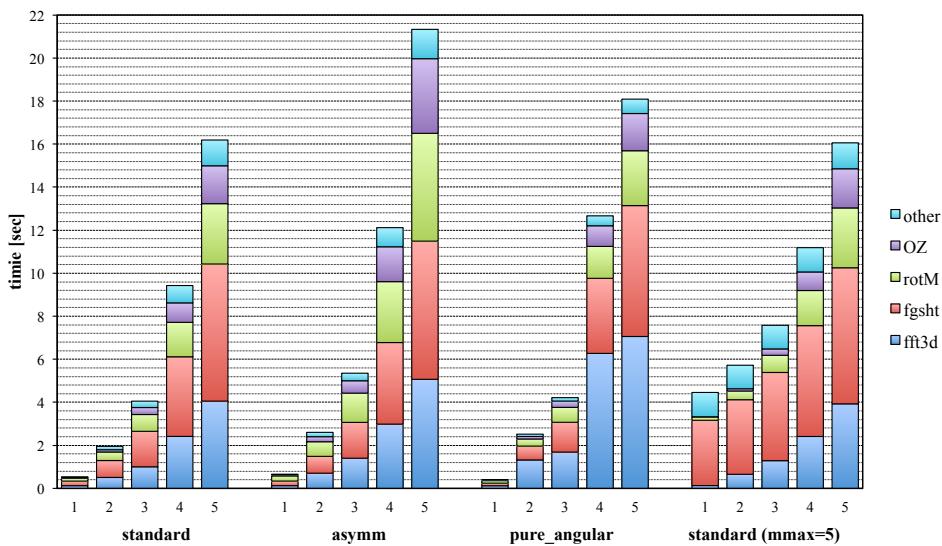


Figure 12.5: Entire iteration of \mathcal{F}_{exc} evaluation: timing overall

12.4.1 Comparison between “naive” methods and “convolution_pure-angular”

The **naive_standard**, **naive_interpolation**, and **convolution_pure-angular** methods share the same processes out of the k -kernel. Table 12.1 shows the timing of loop k of these three methods. It is shown that **convolution_pure-angular** takes far less time than other two methods, of which the loop k takes time in the same order of magnitude than the rest

of iteration. And once $m_{\max} \geq 2$, **naive_interpolation** is faster than **naive_standard**. Note that order 2 of **naive_interpolation** can give already good result for a DCF of $n_{\max} = 5$. So in every case of **naive** methods, **naive_interpolation** should be used. This verified the conclusion of k -kernel test, that the path (4) in figure 11.2 is the fastest.

m_{\max}	naive_standard	naive_interpolation	convolution_pure_angular	OTHER
1	2.34	4.42	0.26	0.15
2	365.95	209.12	1.09	1.43
3	3295.00	752.70	2.37	1.85

Table 12.1: Timing [sec] of loop k of “naive_standard”, “naive_interpolation” and “convolution_pure_angular”, and the rest of iteration

12.4.2 Comparison “convolution_standard” and “convolution_pure_angular”

The comparison of **convolution_standard** and **convolution_pure_angular** is shown in figure 12.6. Their difference is the inversion of FFT and FGSHT. We can see the other parts are almost identical, but the implementation of FFT takes different time. Because in **convolution_standard** the number of FE we need for FFT is the number of projections, and in **convolution_pure_angular** it is the number of angular grid nodes. As there is less projections than angular nodes, **convolution_standard** reasonably takes less time.

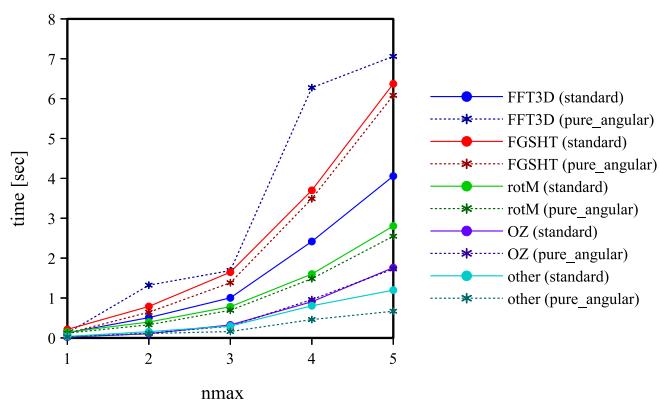


Figure 12.6: comparison of **convolution_standard** and **convolution_pure_angular**

12.4.3 Comparison “convolution_standard” and “convolution_asymm”

The comparison of **convolution_standard** and **convolution_asymm** is shown in figure 12.7. Their difference is that **standard** calculate a half k in the kloop and **asymm** calculate all k in the kloop. They share the same process of FGSHT; for the processes in a k loop (rotM, OZ) **asymm** takes always longer time. As in **asymm** we calculate the FFT for all the projections; and in **standard** we calculate only a half projections with $\mu \geq 0$, it is also different.

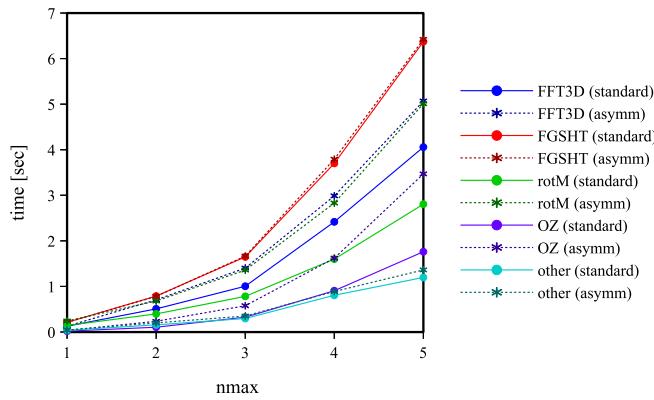


Figure 12.7: comparison of **convolution_standard** and **convolution_asymm**

12.4.4 Comparison mmax and nmax

The comparison of $m_{\text{max}} = n_{\text{max}}$ and $m_{\text{max}} = 5$ for **convolution_standard** is shown in figure 12.8. We see that the choice of quadrature order m_{max} only affect the FGSHT process and the lecture/storage of density variable (other).

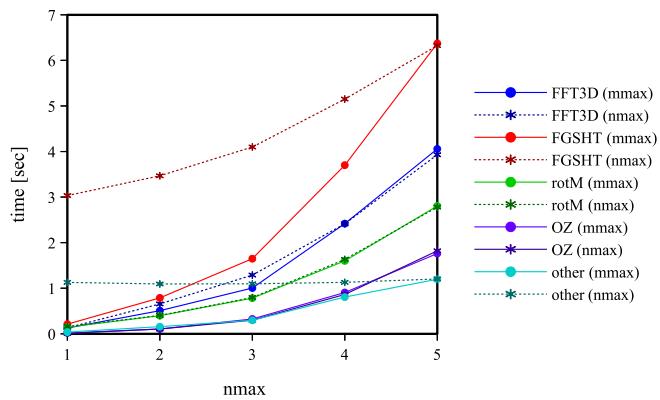


Figure 12.8: comparison of `convolution_standard` for $m_{\text{max}} = n_{\text{max}}$ and $m_{\text{max}} = 5$

12.5 GLOBAL VIEW OF THE SEQUENTIAL CODE PERFORMANCE

We can see that `convolution_standard` is the fastest. The `convolution` methods are of magnitude faster than `naive` methods.

Chapter IV

APPLICATIONS

Only a few applications are made due to the time limit of this thesis, concluding

Section [ref] to show the capability of MDFT to calculate ions and small molecules.

Section [ref] to show a qualitative influence of polar solvent on the reaction including metal-oxo centre.

IONS AND SMALL MOLECULES COMPARED TO MD SIMULATION

13.0.1 *ions*

test solute

#	q	σ	ε	r_x^M	r_y^M	r_z^M	atom name

Table 13.1

[single ions P113]

HYDROGEN TRANSFER REACTION OF MN-OXO: ROLE OF SOLVENT IN REACTION SIMULATION

14.1 REINVESTIGATION OF THE MANGANESE-OXO

This is the case we talked about in the introduction, which is the thesis of university , the motivation of this thesis

Chapter V

CONCLUSION AND PERSPECTIVES

The only one publication during this thesis

CONCLUSION

Here is the publication during this thesis:

PERSPECTIVES

Due to the difficulty of this thesis and my mental state, there are a lot of un finished work and theory. Here lists some to give un idea.

16.1 BUG REPORT

bug in gamma test

rotational invariant projection

16.2 REDUCE MEMORY USE IN MDFT

Computing performance of parallel code

Node-level Parallelization

OpenMP, scalability with respect to the number of thread

Parallelization on Several Nodes

MPI, scalability with respect to the number of node

As shown in chapter III, memory leak can cause divergence and other problem.

16.2.1 *Pass to simple precision*

L-BFGS-B is in double precision

16.2.2 *MPI of the L-BFGS-B minimizer*

As the code is a blackbox, in Fortran 77, it is difficult to parallelize it.

OpenMP, MPI giving the possibility to go beyond the memory limit. Due to the complexity of minimizer L-BFGS, this process is only added on the part of \mathcal{F}_{exc} evaluation. Tests of performance stability with respect to both threads and nodes are made.

16.3 POLARIZABLE SOLUTE

Vext variable

16.4 MDFT VIEWER

This thesis contains a part of visualization

Viewer is an important part of code developing, which provide beautiful visualization and easy analyzing helps to popularize the code. GaussViewer is a good example.

The popular language of visualization is c++, OpenDM, ...

16.5 CLASSICAL SCF METHOD

To be more compatible with Gaussian, [Jensen]

This is only an idea, the mathematical deduction is not fully verified.

16.6 OTHER BRANCHES OF DEVELOPMENT ABOUT MDFT

3-body, polarization,

Chapter VI

APPENDIX

A

BASIC CONCEPTS ABOUT COMPUTING PERFORMANCE

In addition to the theory work, the performance of the code being developed is also an important aspect of this thesis. It is essential to have a fast and accurate method. To evaluate code in a strict and systematic way, some basic concepts of computing performance are listed here.

a.1 ALGORITHM COMPLEXITY

Algorithm complexity is a crucial criteria for sequential code. A definition is given below.

Let f and g be two real (or even complex) functions defined over the natural numbers \mathbb{N} . We write

$$f = O(g) \quad (\text{A.1})$$

if there is a constant $c > 0$ such that from certain number $n > n_0$ we always have $|f(n)| \leq c|g(n)|$. The O is also named as the big-O notation [19], or order of growth. Figure A.1 shows the growth tendency of some frequent functions; from this we can conclude the following:

$$O(1) > O(\log_2 n) > O(n) > O(n \log_2 n) > O(n^2) > O(2^n) > O(n!) \quad (\text{A.2})$$

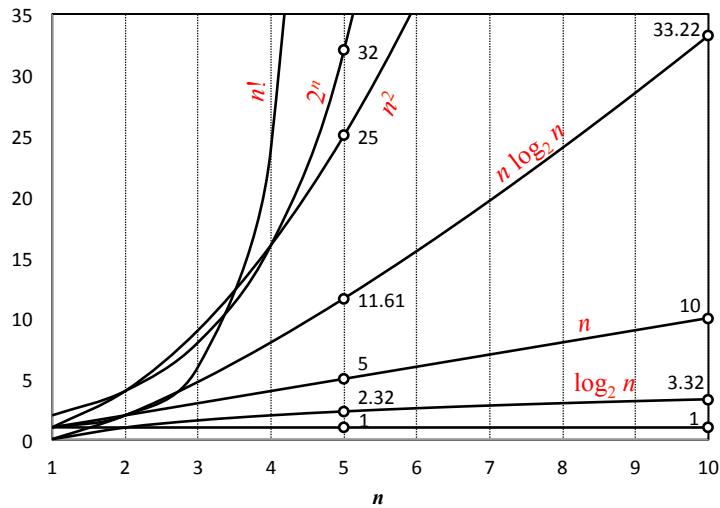


Figure A.1: Function growth

In this thesis, the big-O notation is used to measure algorithm complexity. Other notations can also be used for the same purpose, such as:

- $f = o(g)$ if $f(n)/g(n) \rightarrow 0$, $n \rightarrow \infty$
- The inverse of big-O notation $f = \Omega(g)$ if $g = O(f)$
- The notation $f = \Theta(g)$ means that both $f = O(g)$ and $g = O(f)$ hold, and we can also say they are of the same order.

In a code we always search algorithms with a lower algorithm complexity. Ideally, the implementation of code matches the model and have the same growth tendency as its complexity, but in the practical case, overheads and memory delay can also limit the performance. (part to be modified to adapt implementation results)

a.2 ROOFLINE MODEL AND MEMORY DELAY

The simplest model aiming to distinguish whether a piece of code is limited by the computing power (CPU) or the memory bandwidth (RAM to Caches) is the roofline model [67] for single loop:

$$P = \min(P_{\max}, I \cdot b_S) \quad (\text{A.3})$$

where

P is the applicable peak performance of a loop, assuming that data comes from the level 1 cache, of unity GFlop/s.

I is the computational intensity (“work” per byte transferred) over the slowest data path utilized, of unity Flop/Byte.

b_S is the applicable peak bandwidth of the slowest data path utilized, of unity GByte/s.

As shown in figure A.2, the overall performance is limited by both the peak performance and the memory bandwidth. The computational intensity I depends on the code, while the other two terms in eq. (A.3) depend on hardware. The optimal use of resources occurs at the intersection point.

The roofline model can give an idea of whether the diminuition of algorithm complexity is the most important optimization strategy, because it only counts the number of operations. In most cases, avoiding slow data paths is the key to performance optimization.

As shown in figure A.3, the memory hardware has hierarchical architectures. The fastest ones are the registers included in the microprocessor, which are used for temporary storage of data, instructions and addresses required by the arithmetic logic unit (ALU) and the control unit (CU) in CPU during execution of a program. The lowest is normally the

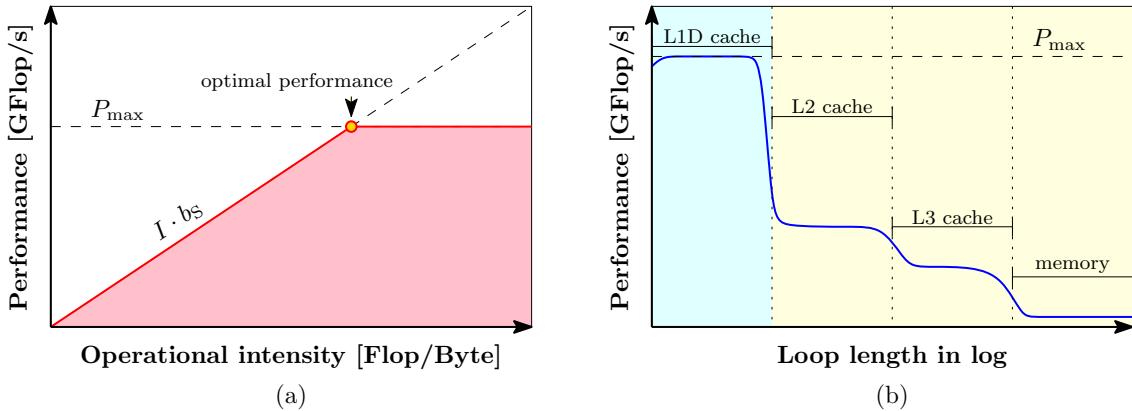


Figure A.2: The roofline model and performance pattern. (a) The roofline model. (b) Performance pattern of a simple loop with respect to the loop length in logarithm. The blue part is limited by operation execution, and the yellow part is limited by memory bottleneck.

input/output (I/O) process. The reading strategy of data (contiguous or not), as well as the size and initialized location of arrays, both play pivotal roles in the overall computing performance.

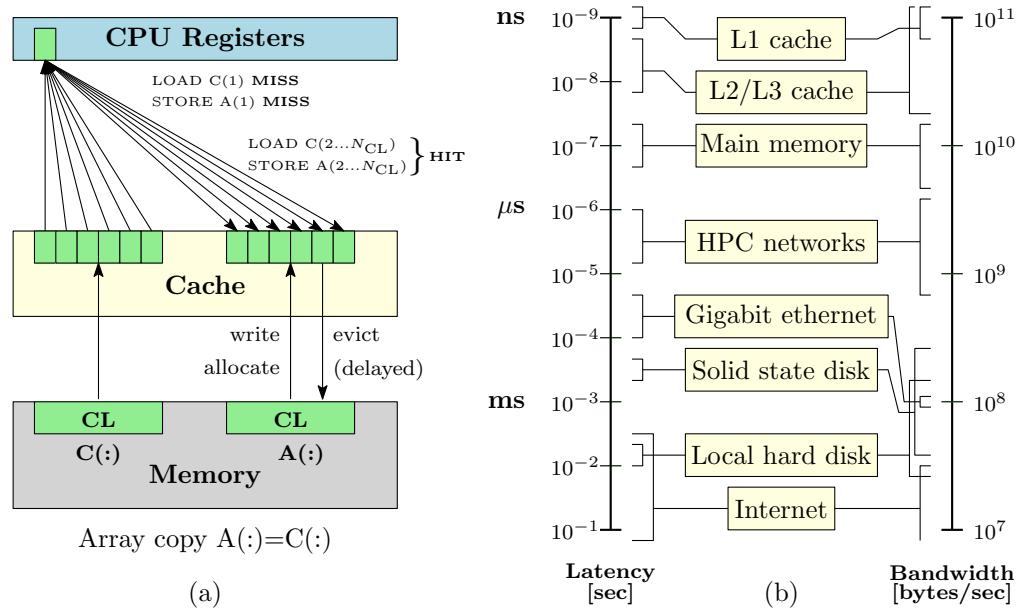


Figure A.3: Memory usage in hardware level [23]. (a) An example of array copy $A(:)=C(:)$. Caches are organized in cache lines (CL), only complete cache lines are transferred between memory hierarchy levels (except registers). HIT/MISS: Load or store instruction does/doesn't find the data in a cache level. (b) Computing latency and memory bandwidth vary by magnitude, from the fastest cache transfers to the lowest processes.

a.3 SCALABILITY OF PARALLELIZED CODE

For parallelized code, scalability is the key issue. Highly scalable codes can take advantage of numerous nodes of HPC centers, so that single core performance no longer matters.

The speed-up is defined as:

$$S(N) = \frac{t(1)}{t(N)} \quad (\text{A.4})$$

And the relative efficiency

$$E(N) = \frac{S(N)}{N} = \frac{t(1)}{Nt(N)} \quad (\text{A.5})$$

$S(N) \sim N$ or $E(N) \sim 100\%$ means the application scales. By contrast, $S(N) < N/2$ or $E(N) < 50\%$ means the application does not scale.

Amdahl's Law gives the theoretical speedup in latency of the execution of a task at fixed workload:

$$S(N) = \frac{1}{\alpha_s + \alpha_p/N} \quad (\text{A.6})$$

where α_s is the serial fraction and α_p the parallel fraction of the source code. Therefore the overall computing speed is limited by the unscalable part:

$$\lim_{N \rightarrow \infty} S(N) = \frac{1}{\alpha_s} \quad (\text{A.7})$$

making it the focus we wish to reduce.

a.4 PROFILING AND TRACING TOOLKITS

There are several types of software and toolkits for performance evaluation. They comprise two categories: profiling and tracing. A trace is a collection of events or timestamps. A profile is a collection of timings. Profiling tools are usually more simple and rapid, but for subroutines that are called a large number of times, the overhead in time measurement is not negligible.

The tool used in this thesis is mainly VTune, where application execution is interrupted every $\sim 100^{-6}$ s and information is stored (call stack, hardware counters, etc.). The execution time overhead is small. ([To be detailed.](#))

B

DIRECT CORRELATION FUNCTION OF WATER

Two source of DCF is used in this thesis: (1) The DCF of bulk water produced by the work of Zhao et al. using MD; (2) Belloni et al. using MC [ref].

b.1 DIPOLE DCF FROM MOLECULAR DYNAMICS SIMULATION

b.1.1 *Principle*

b.2 DCF PROJECTIONS FROM BULK MONTE CARLO SIMULATION

This DCF set is calculated by Belloni *et al.* [**Luc_2012**] and is presented here briefly for the purpose of clarification. g accumulated, solve the inverse OZ equation to find c

First rotational invariant of the Fourier transform of the total correlation function h example:

$$\hat{c}^{000}(k) = \frac{\hat{h}^{000}(k)}{1 + n_0 \hat{h}^{000}(k)}$$

(question of l imaginary in k -space)

b.3 COMPARISON

the notation of Wertheim and Hansen, or in k-space

$$\Phi^{000} = 1$$

$$\Phi^{011} = \hat{\mathbf{k}} \cdot \boldsymbol{\Omega}_1$$

$$\Phi^{101} = \hat{\mathbf{k}} \cdot \boldsymbol{\Omega}_2$$

$$\Phi^{110} = \boldsymbol{\Omega}_1 \cdot \boldsymbol{\Omega}_2 \quad (\text{B.1})$$

$$\Phi^{112} = 3(\hat{\mathbf{k}} \cdot \boldsymbol{\Omega}_1)(\hat{\mathbf{k}} \cdot \boldsymbol{\Omega}_2) - \boldsymbol{\Omega}_1 \cdot \boldsymbol{\Omega}_2$$

Different rotational invariant projections from Luc's c. Luc defines (according to Blum)

$$\Phi^{000} = 1$$

$$\Phi^{011} = i\mathbf{k} \cdot \boldsymbol{\Omega}_1 = i \cos \theta_1$$

$$\Phi^{101} = i\mathbf{k} \cdot \boldsymbol{\Omega}_2 = i \cos \theta_2$$

$$\Phi^{110} = -\sqrt{3}\boldsymbol{\Omega}_1 \cdot \boldsymbol{\Omega}_2 = -\sqrt{3}(\sin \theta_1 \sin \theta_2 \cos \phi_{12} + \cos \theta_1 \cos \theta_2) \quad (\text{B.2})$$

$$\Phi^{112} = \sqrt{\frac{3}{10}} [3(\mathbf{k} \cdot \boldsymbol{\Omega}_1)(\mathbf{k} \cdot \boldsymbol{\Omega}_2) - \boldsymbol{\Omega}_1 \cdot \boldsymbol{\Omega}_2] = \sqrt{\frac{3}{10}} (2 \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 \cos \phi_{12})$$

b.3.1 Comparison with non-coupling dipole DCF in MDFT ($n_{\max} = 1$)

$$c_s^{000}, c_{\Delta}^{110}, c_d^{112}$$

B.3.2 Comparison with respect to n_{\max}

first rotational invariants $m_{\max} = 1$ (4 independent projections $c_s^{000}, c_{\Delta}^{110}, c_d^{112}$ and $c_+^{011} = -c_-^{101}$).

EQUIVALENCE OF QUADRATURE-PROJECTION ORDER

C.1 GAUSSIAN QUADRATURE

Theorem:

Let $P_n(x)$ be a nonzero polynomial of degree n , and $w(x)$ a positive weight function so that

$$\int_a^b x^k P_n(x) w(x) dx = 0, \quad (k = 0, \dots, n-1) \quad (\text{C.1})$$

If $\{x_i\}$ ($i = 1, \dots, n$) are the zeros of $P_n(x)$, then

$$\int_a^b f(x) w(x) dx \simeq \sum_{i=1}^n A_i f(x_i) \quad (\text{C.2})$$

with

$$A_i = \int_a^b l_{i-1}(x) w(x) dx \quad (\text{C.3})$$

is exact for all polynomials $f(x)$ of degree at most $2n - 1$, where $\{l_i\}$ are the usual Lagrange interpolating polynomials.

Proof:

Assume that $f(x)$ is a polynomial of degree at most $2n - 1$. Using long division

$$f(x) = P_n(x)p(x) + r(x) \quad (\text{C.4})$$

$p(x)$ and $r(x)$ are obtained as polynomials of degree at most $n - 1$.

By taking $\{x_i\}$ as the zeros of $P_n(x)$, we can easily find $f(x_i) = r(x_i)$, ($i = 1, \dots, n$), then

$$\begin{aligned} \int_a^b f(x)w(x)dx &= \int_a^b [P_n(x)p(x) + r(x)] w(x)dx \\ &\simeq \underbrace{\sum_{i=1}^n P_n(x_i)p(x_i)w_i}_{=0} + \sum_{i=1}^n A_i r(x_i) \end{aligned} \quad (\text{C.5})$$

is exact for $r(x)$ of degree at most $n - 1$ (c.f. Numerical Recipes [**Numerical_Recipes_3ed**] p.118), and thus exact for $f(x)$ of degree at most $2n - 1$.

C.2 ANGULAR INTEGRATION IN GSHT

To expand a function onto GSHs, as in eq. (7.15), quadrature is needed. Assume that $F(\Omega)$ is a polynomial of $\cos\Theta$, $\cos\Phi$ and $\cos\Psi$ of order n . As $R_{\mu'\mu}^{m*}(\Omega)$ is also a polynomial of order n , the total degree of integrand is $2n$. It should be noted that the surface area element is:

$$d\Omega = \sin\Theta d\Theta d\Phi d\Psi = d\cos\Theta d\Phi d\Psi \quad (\text{C.6})$$

For $\cos \Theta$ integration, considering $w(x) = 1$ and $x = \cos \Theta$, Gauss-Legendre quadrature should be used. Thus $n + 1$ points on x should be taken, with $\{x_i\}$ given by Legendre polynomials $P_{n+1}(x)$.

For Φ and Ψ integration, taking $w(x) = (1 - x^2)^{-\frac{1}{2}}$, the abscissae are given by the $N = n + 1$ roots of the Chebyshev polynomial of the first kind:

$$T_N(x) = \cos(N \cos x) \Rightarrow x_i = \cos\left[\frac{(2i-1)\pi}{2N}\right], \quad i \in 1, \dots, N \quad (\text{C.7})$$

with weight $w_i = \frac{\pi}{N}$, it corresponds to points in $\Phi \in [0, \pi]$ regularly distributed. However, for $\Phi \in [0, 2\pi]$, two times of function evaluation should be calculated:

$$\begin{aligned} & \int_{-1}^1 f(\cos \Phi) \frac{1}{\sqrt{1 - \cos^2 \Phi}} d \cos \Phi \\ &= \begin{cases} \int_{\pi}^0 f(\cos \Phi) d\Phi = - \int_0^{\pi} f(\cos(\Phi)) d\Phi & \Phi \in [0, \pi] \\ \int_{-\pi}^0 f(\cos(\Phi)) d(\Phi) = \int_0^{\pi} f(\cos(-\Phi')) d\Phi' & \Phi' \in [0, \pi] \end{cases} \end{aligned} \quad (\text{C.8})$$

so that

$$\int_0^{2\pi} f(\cos \Phi) d\Phi = \int_{-\pi}^{\pi} f(\cos \Phi) d\Phi = \int_0^{\pi} [f(\cos(-\Phi)) - f(\cos \Phi)] d\Phi \quad (\text{C.9})$$

It corresponds to $2n + 2$ points in $\Phi \in [0, 2\pi]$ regularly distributed. However, it's not the minimal number of points necessary to do the exact integration. Suppose that $\Phi_2 \equiv \Phi/2$,

$$\int_0^{2\pi} f(\cos \Phi) d\Phi = \int_0^{\pi} f(\cos(2\Phi_2)) d\Phi_2 = \int_0^{\pi} [f(2\cos^2 \Phi_2 - 1)] d\Phi_2 \quad (\text{C.10})$$

As $f(2\cos^2 \Phi_2 - 1)$ is a polynomial of Φ of degree $2n$, it's a polynomial of Φ_2 of degree $4n$. Thus only $2n + 1$ points are needed.

ROTATIONAL INVARIANT EXPANSION

If a function $F(\mathbf{X}_1, \mathbf{X}_2)$, $\mathbf{X}_i \equiv (\mathbf{r}_i, \boldsymbol{\Omega}_i)$ has transitional and rotational invariance [5], it can be expanded as

$$F(\mathbf{X}_1, \mathbf{X}_2) = \sum_{mnl\mu\nu} F_{\mu\nu}^{mnl}(\|\mathbf{r}_{12}\|) \Phi_{\mu\nu}^{mnl}(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \hat{\mathbf{r}}_{12}) \quad (\text{D.1})$$

where $\mathbf{r}_{12} \equiv \mathbf{r}_1 - \mathbf{r}_2$ according to the transitional invariance, and

$$\Phi_{\mu\nu}^{mnl}(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \hat{\mathbf{r}}_{12}) = f^{mnl} \sum_{\mu'\nu'\lambda'} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} R_{\mu'\mu}^m(\boldsymbol{\Omega}_1) R_{\nu'\nu}^n(\boldsymbol{\Omega}_2) R_{\lambda'0}^l(\hat{\mathbf{r}}_{12}) \quad (\text{D.2})$$

where $R_{\mu'\mu}^m$ is the Wigner generalized spherical harmonics or Wigner D-symbol defined in the same convention as Messiah [48] (different than Edmonds [13]). f^{mnl} can be any arbitrary non-zero constant [17]. Here we define $f^{mnl} = f^m f^n = \sqrt{2m+1} \sqrt{2n+1}$ according to the definition of Belloni [**Luc_2014**].

Two special cases are adopted in this thesis, these being the laboratory coordinate system with particle 1 at origin (fixed frame) and intermolecular coordinate system (local frame) shown in figure 6.1. Their formalism and symmetry properties are presented later.

d.1 ORTHOGONALITY OF Φ

The rotational invariant Φ in eq. (D.2) is orthogonal, as proven below:

$$\begin{aligned}
 \langle \Phi | \Phi_2 \rangle &= \int d\Omega_1 d\Omega_2 d\hat{r} \Phi_{\mu\nu}^{mn}(\Omega_1, \Omega_2, \hat{r}_{12}) \Phi_{\mu_2\nu_2}^{m_2n_2l_2*}(\Omega_1, \Omega_2, \hat{r}_{12}) \\
 &= f^m f^n f^{m_2} f^{n_2} \sum_{\mu'\nu'\lambda'} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} \sum_{\mu'_2\nu'_2\lambda'_2} \begin{pmatrix} m_2 & n_2 & l_2 \\ \mu'_2 & \nu'_2 & \lambda'_2 \end{pmatrix} \\
 &\quad \times \left\{ \int d\Omega_1 R_{\mu'\mu}^m(\Omega_1) R_{\mu'_2\mu_2}^{m_2*}(\Omega_1) \right. \\
 &\quad \left[\int d\Omega_2 R_{\nu'\nu}^n(\Omega_2) R_{\nu'_2\nu_2}^{n_2*}(\Omega_2) \left(\int d\hat{r} R_{\lambda'0}^l(\hat{r}_{12}) R_{\lambda'_20}^{l_2*}(\hat{r}_{12}) \right) \right] \} \\
 &= f^m f^n f^{m_2} f^{n_2} \sum_{\mu'\nu'\lambda'} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} \sum_{\mu'_2\nu'_2\lambda'_2} \begin{pmatrix} m_2 & n_2 & l_2 \\ \mu'_2 & \nu'_2 & \lambda'_2 \end{pmatrix} \\
 &\quad \times \delta_{m,m_2} \delta_{n,n_2} \delta_{l,l_2} \delta_{\mu,\mu_2} \delta_{\nu,\nu_2} \delta_{\mu',\mu'_2} \delta_{\nu',\nu'_2} \delta_{\lambda',\lambda'_2} \\
 &= (2l+1)^{-1} \sum_{\mu'\nu'\lambda'} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} \tag{D.3}
 \end{aligned}$$

and using the orthogonality of 3j-symbol [13]

$$\sum_{l\lambda'} (2l+1) \begin{pmatrix} m & n & l \\ \mu'_1 & \nu'_1 & \lambda' \end{pmatrix} \begin{pmatrix} m & n & l \\ \mu'_2 & \nu'_2 & \lambda' \end{pmatrix} = \delta_{\mu'_1\mu'_2} \delta_{\nu'_1\nu'_2} \tag{D.4}$$

$$\sum_{\mu'\nu'} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} \begin{pmatrix} m & n & l_2 \\ \mu' & \nu' & \lambda'_2 \end{pmatrix} = (2l+1)^{-1} \delta_{ll_2} \delta_{\lambda'_1\lambda'_2} \tag{D.5}$$

it gives

$$\langle \Phi | \Phi_2 \rangle = (2l+1)^{-1} \tag{D.6}$$

d.2 ROTATIONAL INVARIANCE OF Φ

In any coordinate system, the value of $\Phi_{\mu\nu}^{mnl}$ remains the same. Here is a partial demonstration with the fixed and local frame mentioned above, described in figure 6.1.

Let's use the definition in eq. (D.2):

$$\Phi_{\mu\nu}^{mnl}(\omega_1, \omega_2, 0) = f^{mnl} \sum_{\mu''\nu''\lambda''} \begin{pmatrix} m & n & l \\ \mu'' & \nu'' & \lambda'' \end{pmatrix} R_{\mu''\mu}^m(\omega_1) R_{\nu''\nu}^n(\omega_2) R_{\lambda''0}^l(0) \quad (\text{D.7})$$

$$\Phi_{\mu\nu}^{mnl}(0, \Omega, \hat{\mathbf{r}}) = f^{mnl} \sum_{\mu'\nu'\lambda'} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} R_{\mu'\mu}^m(0) R_{\nu'\nu}^n(\Omega) R_{\lambda'0}^l(\hat{\mathbf{r}}) \quad (\text{D.8})$$

The spherical harmonics have property [13, 48]

$$R_{\mu'\mu}^m(0) = \sum_{\mu''} R_{\mu'\mu''}^m(\hat{\mathbf{r}}) R_{\mu''\mu}^m(\omega_1) \quad (\text{D.9})$$

$$R_{\nu'\nu}^n(\Omega) = \sum_{\nu''} R_{\nu'\nu''}^n(\hat{\mathbf{r}}) R_{\nu''\nu}^n(\omega_2) \quad (\text{D.10})$$

$$R_{\lambda'0}^l(\hat{\mathbf{r}}) = \sum_{\lambda''} R_{\lambda'\lambda''}^l(\hat{\mathbf{r}}) R_{\lambda''0}^l(0) \quad (\text{D.11})$$

so

$$\begin{aligned} \Phi_{\mu\nu}^{mnl}(0, \Omega, \hat{\mathbf{r}}) &= f^{mnl} \sum_{\mu''\nu''\lambda''} R_{\mu''\mu}^m(\omega_1) R_{\nu''\nu}^n(\omega_2) R_{\lambda''0}^l(0) \times \\ &\left[\sum_{\mu'\nu'\lambda'} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} R_{\mu'\mu''}^m(\hat{\mathbf{r}}) R_{\nu'\nu''}^n(\hat{\mathbf{r}}) R_{\lambda'\lambda''}^l(\hat{\mathbf{r}}) \right] \end{aligned} \quad (\text{D.12})$$

According to eq. (4.3.3) in Edmonds [13] or (A.91) in Gray & Gubbins [22]

$$\sum_{\mu' \nu' \lambda'} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} R_{\mu' \mu''}^{m*}(\hat{\mathbf{r}}) R_{\nu' \nu''}^{n*}(\hat{\mathbf{r}}) R_{\lambda' \lambda''}^{l*}(\hat{\mathbf{r}}) = \begin{pmatrix} m & n & l \\ \mu'' & \nu'' & \lambda'' \end{pmatrix} \quad (\text{D.13})$$

where we can also prove

$$\sum_{\mu' \nu' \lambda'} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} R_{\mu' \mu''}^m(\hat{\mathbf{r}}) R_{\nu' \nu''}^n(\hat{\mathbf{r}}) R_{\lambda' \lambda''}^l(\hat{\mathbf{r}}) = \begin{pmatrix} m & n & l \\ \mu'' & \nu'' & \lambda'' \end{pmatrix} \quad (\text{D.14})$$

$\Phi_{\mu\nu}^{mnl}$ remains identical in the two cases

$$\begin{aligned} \Phi_{\mu\nu}^{mnl}(0, \boldsymbol{\Omega}, \hat{\mathbf{r}}) &= f^{mnl} \sum_{\mu'' \nu'' \lambda''} \begin{pmatrix} m & n & l \\ \mu'' & \nu'' & \lambda'' \end{pmatrix} R_{\mu'' \mu}^m(\boldsymbol{\omega}_1) R_{\nu'' \nu}^n(\boldsymbol{\omega}_2) R_{\lambda'' 0}^l(0) \\ &= \Phi_{\mu\nu}^{mnl}(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, 0) \end{aligned} \quad (\text{D.15})$$

Therefore, the projections $F_{\mu\nu}^{mnl}(r)$ also remain rotational invariant in these two coordinate systems.

d.3 TRANSFORM IN LOCAL FRAME

In the intermolecular (local) coordinate system, the 2 molecules are both positioned along the z axis. Using the properties [13, 22, 48] of generalized spherical harmonics:

$$R_{\mu' \mu}^m(\Theta, \Phi, \Psi) = \delta_{\mu' \mu} \quad \text{if } \Theta = \Phi = \Psi = 0 \quad (\text{D.16})$$

and 3j-symbol

$$\begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} \neq 0 \quad \text{only if } \mu' + \nu' + \lambda' = 0 \quad (\text{D.17})$$

$\Phi_{\mu\nu}^{mnl}(\Omega_1, \Omega_2, \mathbf{r}_{12})$ in eq. (D.2) can be simplified to

$$\Phi_{\mu\nu}^{mnl}(\omega_1, \omega_2, 0) = \sum_{\chi} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} f^m f^n R_{\chi\mu}^m(\omega_1) R_{-\chi\nu}^n(\omega_2) \quad (\text{D.18})$$

Thus eq. (D.1) becomes

$$\begin{aligned} F(\omega_1, \omega_2, r) &= \sum_{mnl\mu\nu} F_{\mu\nu}^{mnl}(r) \Phi_{\mu\nu}^{mnl}(\omega_1, \omega_2, 0) \\ &= \sum_{mnl\mu\nu} F_{\mu\nu}^{mnl}(r) f^m f^n \sum_{\chi} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} R_{\chi\mu}^m(\omega_1) R_{-\chi\nu}^n(\omega_2) \end{aligned} \quad (\text{D.19})$$

and the inverse equation

$$\begin{aligned} F_{\mu\nu}^{mnl}(r) &= \int d\omega_1 d\omega_2 F(\omega_1, \omega_2, r) \Phi_{\mu\nu}^{mnl*}(\omega_1, \omega_2, 0) \\ &= f^m f^n \sum_{\chi} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} \times \\ &\quad \int d\omega_1 R_{\chi\mu}^{m*}(\omega_1) \int d\omega_2 R_{-\chi\nu}^{n*}(\omega_2) F(\omega_1, \omega_2, r) \end{aligned} \quad (\text{D.20})$$

The function $F(\omega_1, \omega_2, r)$ and the projections $F_{\mu\nu}^{mnl}(r)$ can be transformed into each other by 2 simple steps.

D.3.1 Transform between $F_{\mu\nu}^{mnl}(r)$ and $F_{\mu\nu,\chi}^{mn}(r)$

Suppose

$$F_{\mu\nu,\chi}^{mn}(r) = \sum_l \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} F_{\mu\nu}^{mnl}(r) \quad (\text{D.21})$$

Using property of 3j-symbol [48]

$$\sum_{\chi} \begin{pmatrix} m & n & l' \\ \chi & -\chi & 0 \end{pmatrix} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} = \frac{\delta_{l'l}}{2l+1} \quad (\text{D.22})$$

we have as the inverse transform

$$F_{\mu\nu}^{mnl}(r) = (2l+1) \sum_{\chi} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} F_{\mu\nu,\chi}^{mn}(r) \quad (\text{D.23})$$

Thus Eq. (D.19) becomes

$$\begin{aligned} F(\omega_1, \omega_2, r) &= \sum_{mnl\mu\nu} (2l+1) \sum_{\chi'} \begin{pmatrix} m & n & l \\ \chi' & -\chi' & 0 \end{pmatrix} F_{\mu\nu,\chi'}^{mn}(r) \times \\ &\quad \sum_{\chi} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} f^m f^n R_{\chi\mu}^m(\omega_1) R_{-\chi\nu}^n(\omega_2) \\ &= \sum_{mn\mu\nu} \sum_{\chi'} \sum_{\chi} F_{\mu\nu,\chi'}^{mn}(r) f^m f^n R_{\chi\mu}^m(\omega_1) R_{-\chi\nu}^n(\omega_2) \times \\ &\quad \sum_l (2l+1) \begin{pmatrix} m & n & l \\ \chi' & -\chi' & 0 \end{pmatrix} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} \end{aligned} \quad (\text{D.24})$$

As

$$\sum_l (2l+1) \begin{pmatrix} m & n & l \\ \chi' & -\chi' & 0 \end{pmatrix} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} = \delta_{\chi'\chi} \quad (\text{D.25})$$

we have

$$F(\omega_1, \omega_2, r) = \sum_{mn\mu\nu\chi} F_{\mu\nu,\chi}^{mn}(r) f^m f^n R_{\chi\mu}^m(\omega_1) R_{-\chi\nu}^n(\omega_2) \quad (\text{D.26})$$

and

$$F_{\mu\nu,\chi}^{mn}(r) = \int d\omega_1 d\omega_2 F(\omega_1, \omega_2, r) f^m f^n R_{\chi\mu}^{m*}(\omega_1) R_{-\chi\nu}^{n*}(\omega_2) \quad (\text{D.27})$$

Thus eq. (D.26, D.27) can be performed either by fast generalized spherical harmonic transform (FGSHT), or being developed into

$$F(\omega_1, \omega_2, r) = \sum_{mn\mu\nu\chi} F_{\mu\nu,\chi}^{mn}(r) f^m f^n r_{\chi\mu}^m(\theta_1) r_{-\chi\nu}^n(\theta_2) e^{-i\chi(\phi_{12}\equiv\phi_1-\phi_2)} e^{-i\mu\psi_1} e^{-i\nu\psi_2} \quad (\text{D.28})$$

and transformed with FFT-3D.

D.3.2 Rotational invariant transform with FFT-3D

Suppose

$$F_{\mu\nu,\chi}^m(r, \theta_2) = \sum_n F_{\mu\nu,\chi}^{mn}(r) f^n r_{-\chi\nu}^n(\theta_2) \quad (\text{D.29})$$

then we have

$$F(\omega_1, \omega_2, r) = \sum_{m\mu\nu\chi} F_{\mu\nu,\chi}^m(r, \theta_2) f^m r_{\chi\mu}^m(\theta_1) e^{-i\chi\phi_{12}} e^{-i\mu\psi_1} e^{-i\nu\psi_2} \quad (\text{D.30})$$

The inverse transform should be

$$F_{\mu\nu,\chi}^{mn}(r) = \frac{1}{2} \int d(\cos\theta_2) F_{\mu\nu,\chi}^m(r, \theta_2) r_{-\chi\nu}^n(\theta_2) \quad (\text{D.31})$$

In the same way, suppose

$$F_{\mu\nu,\chi}(r, \theta_1, \theta_2) = \sum_m F_{\mu\nu,\chi}^m(r, \theta_2) r_{\chi\mu}^m(\theta_1) \quad (\text{D.32})$$

and the inverse transform

$$F_{\mu\nu,\chi}^m(r, \theta_2) = \frac{1}{2} \int d(\cos \theta_1) F_{\mu\nu,\chi}(r, \theta_1, \theta_2) r_{\chi\mu}^m(\theta_1) \quad (\text{D.33})$$

then we have

$$F(r, \omega_1, \omega_2) = \sum_{\mu\nu\chi} F_{\mu\nu,\chi}(r, \theta_1, \theta_2) e^{-i\chi\phi_{12}} e^{-i\mu\psi_1} e^{-i\nu\psi_2} \quad (\text{D.34})$$

which can be treated as a normal FFT of 3 dimensions.

d.4 TRANSFORM IN FIXED FRAME

Similarly, in the laboratory coordinate system

$$\Phi_{\mu\nu}^{mn l}(0, \Omega, \hat{\mathbf{r}}) = \sum_{\eta} \begin{pmatrix} m & n & l \\ \mu & \eta & -\mu - \eta \end{pmatrix} f^m f^n R_{\eta,\nu}^n(\Omega) R_{-\mu-\eta,0}^l(\hat{\mathbf{r}}) \quad (\text{D.35})$$

The rotational invariant does not take advantage of the χ transform as $\mu \neq 0$. The expansion on rotational invariants should be calculated directly.

D.4.1 Expansion of $F(\mathbf{r}, \Omega)$ on rotational invariants

The total equation of the forward transform is as shown below:

$$F_{\mu\nu}^{mn l}(r) = f^m f^n \sum_{\eta} \begin{pmatrix} m & n & l \\ \mu & \eta & -\mu - \eta \end{pmatrix} \int d\hat{\mathbf{r}} R_{-\mu-\eta,0}^{l*}(\hat{\mathbf{r}}) \int d\Omega F(r, \hat{\mathbf{r}}, \Omega) R_{\eta,\nu}^{n*}(\Omega) \quad (\text{D.36})$$

Firstly, the FGSHT is performed:

$$F_{\eta\nu}^n(\mathbf{r}) = \int d\Omega f^n F(\mathbf{r}, \Omega) R_{\eta,\nu}^{n*}(\Omega) \quad (\text{D.37})$$

Then the spherical harmonic transform by histogram should give

$$F_{\eta\nu,\lambda}^{nl}(r) = \int d\hat{\mathbf{r}} R_{\lambda 0}^{l*}(\hat{\mathbf{r}}) F_{\eta\nu}^n(r, \hat{\mathbf{r}}) \quad (\text{D.38})$$

As $F_{\eta\nu}^n(\mathbf{r})$ values are tabulated in the Cartesian grid, we cannot use a quadrature approach without interpolation, so the histogram approach is used.

Histogram for a function f gives:

$$\bar{f}(r) = \int d\theta_r d\phi_r f(x, y, z) \quad (\text{D.39})$$

so if we want to compute

$$\bar{F}(r) = \int d\theta_r d\phi_r R_{\lambda 0}^{l*}(x, y, z) F(x, y, z) \quad (\text{D.40})$$

we just need to propose

$$f(x, y, z) = R_{\lambda 0}^{l*}(x, y, z) F(x, y, z) \quad (\text{D.41})$$

For complex numbers $F_{\eta\nu}^n(\mathbf{r})$, the real and imaginary parts can be calculated separately.

The rotational matrices $R_{\lambda 0}^{l*}(\mathbf{r})$ in Cartesian coordinate system can be pre-generated by recurrence as detailed in appendix E.

Finally, the combination of projections gives:

$$F_{\mu\nu}^{mnl}(r) = f^m \sum_{\eta} \begin{pmatrix} m & n & l \\ \mu & \eta & -\mu - \eta \end{pmatrix} F_{\eta\nu,-\mu-\eta}^{nl}(r) \quad (\text{D.42})$$

It should be noted that $F_{\mu\nu}^{mn l}(r)$ is real?

D.4.2 Rebuilding of $F(\mathbf{r}, \Omega)$ from projections

and the rebuilding of $F(\mathbf{r}, \Omega)$ in a certain orientation is as simple as its definition

$$F(\mathbf{r}, \Omega) = \sum_{mnl\mu\nu} F_{\mu\nu}^{mn l}(r) f^m f^n \sum_{\eta} \begin{pmatrix} m & n & l \\ \mu & \eta & -\mu - \eta \end{pmatrix} R_{\eta\nu}^n(\Omega) R_{-\mu-\eta, 0}^l(\hat{\mathbf{r}}) \quad (\text{D.43})$$

d.5 SYMMETRY

In IEM and MDFT, the rotational invariants are used to describe the solvent. It possess symmetric rules, introduced by the indistinguishability of the two particles, symmetry properties of single particle, and its real number property as a physical quantity. Here we list all the symmetric rules concerning the 2-molecule system.

D.5.1 Symmetric rules of $F(\omega_1, \omega_2)$ in intermolecular form

As shown in figure D.1, function in intermolecular coordinate system $F(\omega_1, \omega_2) \equiv F(\cos \theta_1, \cos \theta_2, \phi, \psi_1, \psi_2)$ possesses symmetry rules:

1. Symmetry of vertical mirror:

$$F(\theta_1, \theta_2, \phi, \psi_1, \psi_2) = F(\theta_1, \theta_2, -\phi, -\psi_1, -\psi_2) \quad (\text{D.44})$$

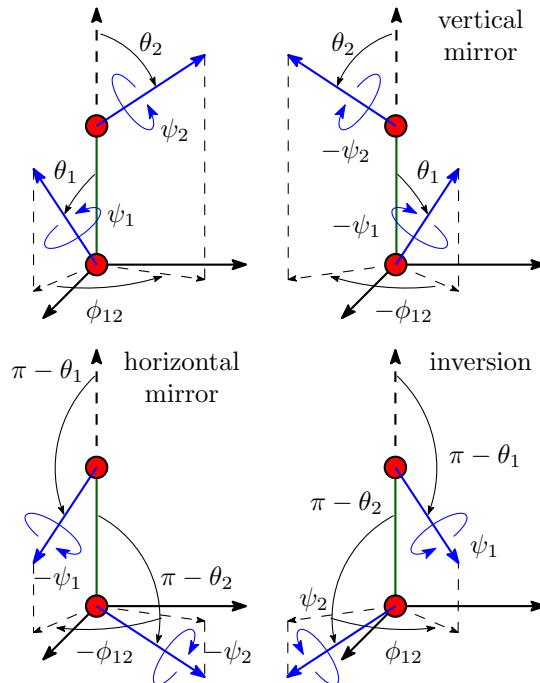


Figure D.1: Symmetry operations of a 2-molecule system

2. Symmetry of inversion:

$$F(\theta_1, \theta_2, \phi, \psi_1, \psi_2) = F(\pi - \theta_2, \pi - \theta_1, \phi, \psi_2, \psi_1) \quad (\text{D.45})$$

And an additive symmetric rule is possessed by particles having

1. [3.]Symmetry axe C_{2n}:

$$F(\theta_1, \theta_2, \phi, \psi_1, \psi_2) = F(\theta_1, \theta_2, \phi, \psi_1 + \pi, \psi_2 + \pi) \quad (\text{D.46})$$

D.5.2 Symmetric rules of rotational invariant projections

The definition of rotational invariant on χ -transform gives

$$c(\theta_1, \theta_2, \phi_{12}, \psi_1, \psi_2) = \frac{1}{2l+1} \sum_{mn\mu\nu\chi} c_{\mu\nu,\chi}^{mn}(r) d_{\chi\mu}^m(\theta_1) d_{\chi\nu}^n(\theta_2) e^{-i\chi\phi_{12}} e^{-i\mu\psi_1} e^{-i\nu\psi_2}$$

Thus

$$\begin{aligned} c(\theta_1, \theta_2, -\phi_{12}, -\psi_1, -\psi_2) &= \frac{1}{2l+1} \sum_{mn\mu\nu\chi} c_{\underline{\mu}\underline{\nu},\underline{\chi}}^{mn}(r) d_{\underline{\chi}\underline{\mu}}^m(\theta_1) d_{\underline{\chi}\underline{\nu}}^n(\theta_2) e^{-i\chi\phi_{12}} e^{-i\mu\psi_1} e^{-i\nu\psi_2} \\ &= \frac{1}{2l+1} \sum_{mn\mu\nu\chi} (-)^{\mu+\nu} c_{\underline{\mu}\underline{\nu},\underline{\chi}}^{mn}(r) d_{\underline{\chi}\mu}^m(\theta_1) d_{\underline{\chi}\nu}^n(\theta_2) e^{-i\chi\phi_{12}} e^{-i\mu\psi_1} e^{-i\nu\psi_2} \end{aligned}$$

$$\begin{aligned} c(\pi - \theta_2, \pi - \theta_1, \phi_{12}, \psi_2, \psi_1) &= \frac{1}{2l+1} \sum_{mn\mu\nu\chi} c_{\mu\nu,\chi}^{mn}(r) d_{\chi\mu}^m(\pi - \theta_2) d_{\chi\nu}^n(\pi - \theta_1) e^{-i\chi\phi_{12}} e^{-i\mu\psi_2} e^{-i\nu\psi_1} \\ &= \frac{1}{2l+1} \sum_{mn\mu\nu\chi} (-)^{m+n+\mu+\nu} c_{\nu\mu,\chi}^{nm}(r) d_{\chi\mu}^m(\theta_1) d_{\chi\nu}^n(\theta_2) e^{-i\chi\phi_{12}} e^{-i\mu\psi_1} e^{-i\nu\psi_2} \end{aligned}$$

and μ, ν are even.

Thus

$$c_{\mu\nu,\chi}^{mn}(r) = c_{\underline{\mu}\underline{\nu},\underline{\chi}}^{mn}(r) = (-)^{m+n} c_{\nu\mu,\chi}^{nm}(r)$$

(Need proof.)

$$\hat{c}_{\mu\nu,\underline{\chi}}^{mn}(\mathbf{k}) = (-)^{m+n} \hat{c}_{\mu\nu,\chi}^{*mn}(\mathbf{k})$$

In MDFT, rotational invariant projections are used to describe DCF. The original $c_{\mu\nu}^{mnl}(\mathbf{r})$ collected by IEM is real.

1. With r and l : As $c_{\mu\nu}^{mnl}(\mathbf{r})$ is real

$$F_{\underline{\mu}\underline{\nu}}^{mnl}(\mathbf{r}) = (-)^{m+n+l} F_{\mu\nu}^{mnl}(\mathbf{r}) \quad (\text{D.47})$$

As the two molecules are interchangeable,

$$F_{\nu\mu}^{nm}(r) = (-)^{m+n} F_{\mu\nu}^{mn}(r) \quad (\text{D.48})$$

2. [2.] With r and χ :

$$F_{\mu\nu,\chi}^{mn}(r) = \sum_{\chi} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} F_{\mu\nu}^{mn}(r) \quad (\text{D.49})$$

$$F_{\underline{\mu}\underline{\nu},\underline{\chi}}^{mn}(r) = F_{\mu\nu,\chi}^{mn}(r) \quad (\text{D.50})$$

$$F_{\nu\mu,\chi}^{nm}(r) = (-)^{m+n} F_{\mu\nu,\chi}^{mn}(r) \quad (\text{D.51})$$

3. [3.] With k and l :

$$\text{TF}(c_{\mu\nu}^{mn}(r)) = \hat{c}_{\mu\nu}^{mn}(\mathbf{k}) \quad (\text{D.52})$$

is real if l is even, and is purely imaginary if l is odd.

4. [4.] With k and χ :

$$F_{\mu\nu,\underline{\chi}}^{mn}(\mathbf{k}) = (-)^{m+n} F_{\mu\nu,\chi}^{*mn}(\mathbf{k}) \quad (\text{D.53})$$

Finally, if the solvent molecule possesses an axe of symmetry C_2 , μ and ν is even.

For spherical solutes, the Ω_1 -dependence vanishes so that $m = \mu = 0$, only the terms $F_{0\nu}^{0nl}(r)$ are not-zero.

CALCULATION OF ROTATION MATRIX ELEMENTS

$R_{\mu\mu'}^m$ BY RECURRENCE

$\mathbf{R}^m(\boldsymbol{\Omega}) \equiv \{R_{\mu'\chi}^m(\boldsymbol{\Omega})\}$ is the rotation matrix of dimension $(2m+1) \times (2m+1)$, defined in Messiah and other books [13, 22, 48].

In MDFT, evaluation of $R_{\mu'\chi}^m(\hat{\mathbf{k}})$ for each m, μ', χ and \mathbf{k} by its definition:

$$R_{\mu'\chi}^m(\hat{\mathbf{k}}) = r_{\mu'\chi}^m(\theta_k) e^{-i\mu' \phi_k} \quad (\text{E.1})$$

is too costly to be done in iterations; on the other hand, to directly stock the value of every element is heavy in terms of memory. An algorithm of $R_{\mu\mu'}^m(\hat{\mathbf{k}})$ evaluation by recurrence described by Choi *et al.* [9] suggests an acceptable cost during the computation, by generating the rotation matrix elements from those of lower order to avoid extra calculation.

e.1 CASE OF $m_{\max} \leq 1$

According to the definition in eq. (E.1), it is easy to find

$$R_{00}^0 = 1 \quad (\text{E.2})$$

For $m = 1$, $\mathbf{R}^1(\hat{\mathbf{k}})$ depends only on the 3×3 orthogonal matrix \mathbf{R} that defines the rotation from the basis vectors of laboratory frame to those of \mathbf{k} -frame:

$$\mathbf{R} = \begin{bmatrix} R_{xx} & R_{yx} & R_{zx} \\ R_{xy} & R_{yy} & R_{zy} \\ R_{xz} & R_{yz} & R_{zz} \end{bmatrix} = \begin{bmatrix} \cos \theta_k \cos \phi_k & -\sin \phi_k & \sin \theta_k \cos \phi_k \\ \cos \theta_k \sin \phi_k & \cos \phi_k & \sin \theta_k \sin \phi_k \\ -\sin \theta_k & 0 & \cos \theta_k \end{bmatrix} \quad (\text{E.3})$$

The matrix \mathbf{R} can be calculated by the cross products of basis vectors as shown in figure 6.3

$$\begin{bmatrix} \mathbf{e}_1'' & \mathbf{e}_2' & \mathbf{e}_3'' \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \mathbf{R} = \mathbf{R} \quad (\text{E.4})$$

The rotation matrix \mathbf{R}^m can be separated into the real \mathbf{F}^m and imaginary \mathbf{G}^m parts, which can be given by the relations

$$R_{\chi\chi'}^m = F_{\chi\chi'}^m + iG_{\chi\chi'}^m \quad (\text{E.5})$$

$$\begin{bmatrix} F_{\underline{1}\underline{1}}^1 & F_{\underline{1}0}^1 & F_{\underline{1}1}^1 \\ F_{0\underline{1}}^1 & F_{00}^1 & F_{01}^1 \\ F_{1\underline{1}}^1 & F_{10}^1 & F_{11}^1 \end{bmatrix} = \begin{bmatrix} (R_{yy} + R_{xx})/2 & R_{xz}/\sqrt{2} & (R_{yy} - R_{xx})/2 \\ R_{zx}/\sqrt{2} & R_{zz} & -R_{zx}/\sqrt{2} \\ (R_{yy} - R_{xx})/2 & -R_{xz}/\sqrt{2} & (R_{yy} + R_{xx})/2 \end{bmatrix} \quad (\text{E.6})$$

$$\begin{bmatrix} G_{\underline{1}\underline{1}}^1 & G_{\underline{1}0}^1 & G_{\underline{1}1}^1 \\ G_{0\underline{1}}^1 & G_{00}^1 & G_{01}^1 \\ G_{1\underline{1}}^1 & G_{10}^1 & G_{11}^1 \end{bmatrix} = \begin{bmatrix} (R_{yx} - R_{xy})/2 & R_{yz}/\sqrt{2} & -(R_{yx} + R_{xy})/2 \\ -R_{zy}/\sqrt{2} & 0 & -R_{zy}/\sqrt{2} \\ (R_{yx} + R_{xy})/2 & R_{yz}/\sqrt{2} & (R_{xy} - R_{yx})/2 \end{bmatrix} \quad (\text{E.7})$$

e.2 CASE OF $m_{\max} > 1$

E.2.1 Recurrence relation for $-m + 1 \leq \chi' \leq m - 1$

The recurrence relation for $-m \leq \chi \leq m$, $-m + 1 \leq \chi' \leq m - 1$ between matrix elements is:

$$R_{\chi\chi'}^m = a_{\chi\chi'}^m R_{00}^1 R_{\chi\chi'}^{m-1} + b_{\chi\chi'}^m R_{10}^1 R_{\chi-1,\chi'}^{m-1} + b_{-\chi,\chi'}^m R_{-1,0}^1 R_{\chi+1,\chi'}^{m-1} \quad (\text{E.8})$$

where

$$\begin{aligned} a_{\chi\chi'}^m &= \left[\frac{(m+\chi)(m-\chi)}{(m+\chi')(m-\chi')} \right]^{\frac{1}{2}} \quad (-m+1 \leq \chi \leq m-1) \\ b_{\chi\chi'}^m &= \left[\frac{(m+\chi)(m+\chi-1)}{2(m+\chi')(m-\chi')} \right]^{\frac{1}{2}} \quad (-m+2 \leq \chi \leq m-2) \end{aligned} \quad (\text{E.9})$$

To separate the real and imaginary parts, suppose

$$H_{\chi\chi'}^m(i, j) = F_{ij}^1 F_{\chi\chi'}^{m-1} - G_{ij}^1 G_{\chi\chi'}^{m-1} \quad (\text{E.10})$$

$$K_{\chi\chi'}^m(i, j) = F_{ij}^1 G_{\chi\chi'}^{m-1} + G_{ij}^1 F_{\chi\chi'}^{m-1} \quad (\text{E.11})$$

therefore

$$F_{\chi\chi'}^m = a_{\chi\chi'}^m H_{\chi\chi'}^m(0, 0) + b_{\chi\chi'}^m H_{\chi-1,\chi'}^m(1, 0) + b_{-\chi,\chi'}^m H_{\chi+1,\chi'}^m(-1, 0) \quad (\text{E.12})$$

$$G_{\chi\chi'}^m = a_{\chi\chi'}^m K_{\chi\chi'}^m(0, 0) + b_{\chi\chi'}^m K_{\chi-1,\chi'}^m(1, 0) + b_{-\chi,\chi'}^m K_{\chi+1,\chi'}^m(-1, 0) \quad (\text{E.13})$$

In the case of $\chi = \pm m$, certain terms in eq. (E.8) are out of definition. They are supposed to be zero. Another way is to suppose that

$$\begin{aligned} a_{\chi\chi'}^m &= 0 \quad \text{for } \chi = \pm m \\ b_{\chi\chi'}^m &= 0 \quad \text{for } \chi = \pm m \text{ and } \chi = \mp(m-1) \end{aligned} \quad (\text{E.14})$$

E.2.2 Recurrence relation for $-m + 2 \leq \chi' \leq m$

For the case $\chi' = \pm m$ which are not covered in eq. (E.8), another recurrence relation supposes that:

$$R_{\chi\chi'}^m = c_{\chi\chi'}^m R_{0,1}^1 R_{\chi,\chi'-1}^{m-1} + d_{\chi\chi'}^m R_{1,1}^1 R_{\chi-1,\chi'-1}^{m-1} + d_{-\chi,\chi'}^m R_{-1,1}^1 R_{\chi+1,\chi'-1}^{m-1} \quad (\text{E.15})$$

$$F_{\chi\chi'}^m = c_{\chi\chi'}^m H_{\chi,\chi'-1}^m(0,1) + d_{\chi\chi'}^m H_{\chi-1,\chi'-1}^m(1,1) + d_{-\chi,\chi'}^m H_{\chi+1,\chi'-1}^m(-1,1) \quad (\text{E.16})$$

$$G_{\chi\chi'}^m = c_{\chi\chi'}^m K_{\chi,\chi'-1}^m(0,1) + d_{\chi\chi'}^m K_{\chi-1,\chi'-1}^m(1,1) + d_{-\chi,\chi'}^m K_{\chi+1,\chi'-1}^m(-1,1) \quad (\text{E.17})$$

with

$$\begin{aligned} c_{\chi\chi'}^m &= \left[\frac{2(m+\chi)(m-\chi)}{(m+\chi')(m+\chi'-1)} \right]^{\frac{1}{2}} \quad (-m+1 \leq \chi \leq m-1) \\ d_{\chi\chi'}^m &= \left[\frac{(m+\chi)(m+\chi-1)}{(m+\chi')(m+\chi'-1)} \right]^{\frac{1}{2}} \quad (-m+2 \leq \chi \leq m-2) \end{aligned} \quad (\text{E.18})$$

and

$$c_{\chi\chi'}^m = 0 \quad \text{for } \chi = \pm m \quad (\text{E.19})$$

$$d_{\chi\chi'}^m = 0 \quad \text{for } \chi = \pm m \text{ and } \chi = \mp(m-1)$$

which is available for $-m + 2 \leq \chi' \leq m$.

E.2.3 Symmetries

The symmetries of $R_{\chi\chi'}^m$ allow us to calculate only a half of the elements:

$$R_{-m,-m'}^l = (-1)^{m+m'} R_{mm'}^{l*} \quad (\text{E.20})$$

which gives

$$F_{-m,-m'}^l = (-1)^{m+m'} F_{mm'}^l \quad (\text{E.21})$$

$$G_{-m,-m'}^l = -(-1)^{m+m'} G_{mm'}^l \quad (\text{E.22})$$

PROPERTIES OF WIGNER 3J-SYMBOL AND GSH

The properties of Wigner 3j-symbol and Wigner generalized spherical harmonics (GSH, Wigner D-symbol) play a huge role in the reduction of molecular Ornstein-Zernike equation as well as finding the relation between rotational invariant projections. Their main properties, presented in Messiah [48], Gray & Gubbins [22] and Edmonds [13], are listed here.

f.1 PROPERTIES OF WIGNER 3J-SYMBOL

Wigner 3j-symbols are equivalent to Clebsch-Gordon (CG) coefficients multiplied by the phase factor:

$$\begin{pmatrix} m & n & l \\ \mu & \nu & -\lambda \end{pmatrix} = \frac{(-)^{m-n+\lambda}}{\sqrt{2l+1}} \langle mn\mu\nu | l\lambda \rangle \quad (\text{F.1})$$

and can be calculated with the Racah formula [48].

Reality

The 3j-symbols are real.

$$\begin{pmatrix} m & n & l \\ \mu & \nu & \lambda \end{pmatrix} = \begin{pmatrix} m & n & l \\ \mu & \nu & \lambda \end{pmatrix}^* \quad (\text{F.2})$$

Selection rules

$$\begin{pmatrix} m & n & l \\ \mu & \nu & \lambda \end{pmatrix} = 0 \text{ if } \begin{cases} \mu + \nu + \lambda = 0 \\ |m - n| < l < m + n \\ (\text{triangular inequalities}) \end{cases} \text{ are not met.} \quad (\text{F.3})$$

Permutation

1. Even permutation

$$\begin{pmatrix} m & n & l \\ \mu & \nu & \lambda \end{pmatrix} = \begin{pmatrix} n & l & m \\ \nu & \lambda & \mu \end{pmatrix} = \begin{pmatrix} l & m & n \\ \lambda & \mu & \nu \end{pmatrix} \quad (\text{F.4})$$

2. Odd permutation

$$\begin{aligned} (-)^{m+n+l} \begin{pmatrix} m & n & l \\ \mu & \nu & \lambda \end{pmatrix} &= \begin{pmatrix} n & m & l \\ \nu & \mu & \lambda \end{pmatrix} \\ &= \begin{pmatrix} m & l & n \\ \mu & \lambda & \nu \end{pmatrix} = \begin{pmatrix} l & n & m \\ \lambda & \nu & \mu \end{pmatrix} \end{aligned} \quad (\text{F.5})$$

3. Simultaneous change of signs of μ , ν and λ

$$\begin{pmatrix} m & n & l \\ \mu & \nu & \lambda \end{pmatrix} = (-)^{m+n+l} \begin{pmatrix} m & n & l \\ -\mu & -\nu & -\lambda \end{pmatrix} \quad (\text{F.6})$$

Orthogonality

$$\sum_{l=|m-n|}^{m+n} \sum_{\lambda=-l}^l (2l+1) \begin{pmatrix} m & n & l \\ \mu & \nu & \lambda \end{pmatrix} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda \end{pmatrix} = \delta_{\mu\mu'} \delta_{\nu\nu'} \quad (\text{F.7})$$

$$\sum_{\mu=-m}^m \sum_{\nu=-n}^n \begin{pmatrix} m & n & l \\ \mu & \nu & \lambda \end{pmatrix} \begin{pmatrix} m & n & l' \\ \mu & \nu & \lambda' \end{pmatrix} = (2l+1)^{-1} \delta_{ll'} \delta_{\lambda\lambda'} \quad (\text{F.8})$$

f.2 PROPERTIES OF GSH

There are many different definitions of GSH given in lectures. Here we adopt the definition in Messiah:

$$R_{\mu'\mu}^m(\phi\theta\psi) = e^{-i\mu'\phi} r_{\mu'\mu}^m(\theta) e^{-i\mu\psi} \quad (\text{F.9})$$

where $r_{\mu\mu'}^m$ is the generalized Legendre polynomial (GLP), which is real, and can be evaluated using the Wigner formula:

$$\begin{aligned} r_{\mu'\mu}^m(\theta) &= [(m+\mu')! (m-\mu')! (m+\mu)! (m-\mu)!]^{\frac{1}{2}} \times \\ &\sum_i \frac{(-)^i (\cos \theta/2)^{2m+\mu'-\mu-2i} (\sin \theta/2)^{2i-\mu'+\mu}}{(m+\mu'-i)! (m-\mu-i)! i! (i-\mu'+\mu)!} \end{aligned} \quad (\text{F.10})$$

Symmetries of $r_{\mu'\mu}^m(\theta)$

$$r_{\mu\mu'}^m(\theta) = (-)^{\mu'-\mu} r_{\mu'\mu}^m(\theta) \quad (\text{F.11})$$

$$r_{\underline{\mu}\mu}^m(\theta) = (-)^{\mu'-\mu} r_{\mu'\underline{\mu}}^m(\theta) \quad (\text{F.12})$$

$$r_{\mu'\mu}^m(\theta) = r_{\mu\mu'}^m(-\theta) \quad (\text{F.13})$$

$$r_{\mu'\mu}^m(\theta + \pi) = (-)^{m+\mu} r_{\mu'\underline{\mu}}^m(\theta) \quad (\text{F.14})$$

where $\underline{\mu} \equiv -\mu$.

Symmetries of $R_{\mu'\mu}^m(\phi\theta\psi)$

$$R_{\mu'\mu}^m(\phi\theta\psi) = (-)^{\mu'-\mu} R_{\underline{\mu}'\underline{\mu}}^{m*}(\phi\theta\psi) \quad (\text{F.15})$$

$$R_{\mu'\mu}^m(\phi\theta\psi) = (-)^{\mu'+\mu} R_{\mu\mu'}^{m*}(\phi\theta\psi) \quad (\text{F.16})$$

$$R_{\mu'\mu}^m(\phi\theta\psi) = (-)^{m+\mu'} R_{\mu'\mu}^m(-\phi, \theta + \pi, \psi) = (-)^{m+\mu} R_{\mu'\mu}^m(\phi, \theta + \pi, -\psi) \quad (\text{F.17})$$

Unitarity and orthogonality

$$\sum_{\mu'} R_{\mu'\mu}^m(\phi\theta\psi) R_{\mu'\mu''}^{m*}(\phi\theta\psi) = \delta_{\mu\mu''} \quad (\text{F.18})$$

$$\sum_{\mu} R_{\mu'\mu}^m(\phi\theta\psi) R_{\mu''\mu}^{m*}(\phi\theta\psi) = \delta_{\mu'\mu''} \quad (\text{F.19})$$

$$\sum_{m\mu'\mu} R_{\mu'\mu}^m(\phi\theta\psi) R_{\mu'\mu'}^{m*}(\phi'\theta'\psi') = \delta_{\phi\phi'}\delta_{\theta\theta'}\delta_{\psi\psi'} \quad (\text{F.20})$$

$$\frac{1}{8\pi^2} \int d\cos\theta d\phi d\psi R_{\mu'\mu}^m(\phi\theta\psi) R_{\nu'\nu}^{n*}(\phi\theta\psi) = \frac{\delta_{mn}\delta_{\mu'\nu'}\delta_{\mu\nu}}{2n+1} \quad (\text{F.21})$$

$r_{\mu'\mu}^m(\theta)$ in terms of $\cos\theta$ and $\sin\theta$

1. If $(-)^{\mu'+\mu} = +1$, $r_{\mu'\mu}^m(\theta)$ is a polynomial of degree m in $\cos\theta$.
2. If $(-)^{\mu'+\mu} = -1$, $r_{\mu'\mu}^m(\theta)/\sin\theta$ is a polynomial of degree $(m-1)$ in $\cos\theta$.

Rotation and product

$$R_{\mu'\mu}^m(\omega) = \sum_{\chi} R_{\mu'\chi}^m(\omega_2) R_{\chi\mu}^m(\omega_1) \quad (\text{F.22})$$

where ω is the result of the successive application of ω_1 and ω_2 in order.

$$R_{\chi\mu}^m(\omega) = \sum_{\mu'} R_{\mu'\chi}^{m*}(\hat{k}) R_{\mu'\mu}^m(\Omega) \quad (\text{F.23})$$

$$R_{\mu'\mu}^m(\Omega) = \sum_{\chi} R_{\chi\mu'}^{m*}(\hat{k}^{-1}) R_{\chi\mu}^m(\omega) = \sum_{\chi} R_{\mu'\chi}^m(\hat{k}) R_{\chi\mu}^m(\omega)$$

Composition relation for GSHs

$$\sum_{\mu'\nu'\lambda'} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} R_{\mu'\mu}^m(\phi\theta\psi) R_{\nu'\nu}^n(\phi\theta\psi) R_{\lambda'\lambda}^l(\phi\theta\psi) = \begin{pmatrix} m & n & l \\ \mu & \nu & \lambda \end{pmatrix} \quad (\text{F.24})$$

f.3 CONVENTION OF GSH

The convention of GSH in books and articles used in this thesis are different. In Messiah [48] and Gray & Gubbins [22], it is defined as in eq. (F.9). In Edmonds [13], it is defined as:

$$D_{\mu'\mu}^m(\phi\theta\psi) = e^{i\mu'\psi} d_{\mu'\mu}^m(\theta) e^{i\mu\phi} \quad (\text{F.25})$$

which can be seen as the inverse rotation matrix of $R_{\mu'\mu}^m$.

In Blum [4, 5], the equation

$$D_{m0}^l(\phi\theta\psi) = (-)^m \left(\frac{4\pi}{2l+1} \right)^{\frac{1}{2}} Y_m^l(\theta\phi) \quad (\text{F.26})$$

is adopted, that means it shares the same definition as Edmonds, where

$$R_{\mu'\mu}^m(\phi\theta\psi) = D_{\mu\mu'}^{m*}(\phi\theta\psi) \quad (\text{F.27})$$

In Fries & Patey [17], the definition of Messiah is used.

ERROR EVALUATION OF INTERPOLATION

STRATEGIES FOR DCF IN LOCAL FRAME

The error introduced by the two interpolation orders for a DCF of order $n_{\max} = 1$ (for which the exact DCF can be computed directly; see details later) is shown in figure G.1.

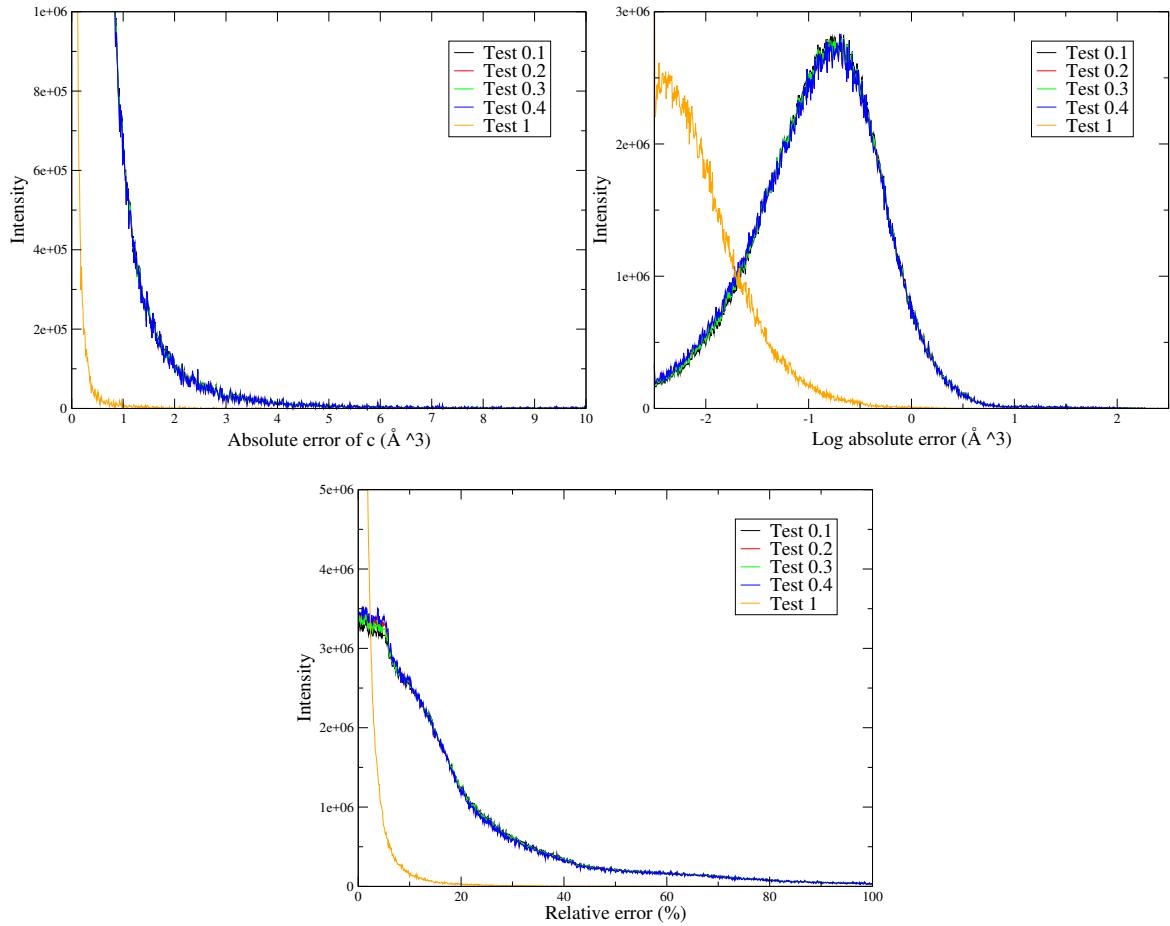


Figure G.1: Error of finding $\hat{c}(\mathbf{k}, \Omega_1, \Omega_2)$ by interpolation compared to direct calculation: Test 0.1-0.4 is zero order interpolation with ϕ tabulated as in figure 6.4. Test 1 is linear interpolation.

Absolute error is the histogram that counts the number of times that the calculated DCF gives the corresponding absolute error E_a^i with a resolution of 0.01, in range of [0, 10]:

$$E_a^i = |c_k^i - c_k| \quad (\text{G.1})$$

where c_k^i is any element of $\hat{c}(\mathbf{k}, \Omega_1, \Omega_2)$ of unity Å³ calculated as described and c_k is the one calculated directly as the reference.

Log absolute error is treated the same way as E_a^i , with E_l^i defined as

$$E_l^i = \log |c_k^i - c_k| \quad (\text{G.2})$$

Relative error is defined as

$$E_r^i = |c_k^i - c_k| / |c_k| \quad (\text{G.3})$$

with resolution of 0.1%, in range of [0, 1].

In all three figures, the 4 curves given by zero order interpolation do not diverge a great deal compared with the linear interpolation one. The result of MDFT also shows that zero order interpolation gives large energy error with a DCF of $n_{\max} = 1$, and has convergency problems in certain cases. We conclude that the linear interpolation scheme is absolutely necessary. On the other hand, as seen in eq. (6.13), it is computationally much more expensive than the simple histogram scheme, as it requires $2^5 = 32$ times of operations.

ORIGINAL DATA OF MDFT IMPLEMENTATION

In this appendix, some original implementation data which is thought redundant to put in the main context are put here. These details may contributes to further researches and gives a global image of the parameter and version sensibility of code MDFT.

h.1 SERIES OF CHARGED CH₄ CENTER

These tests are done with the sequential code.

h.1.1 Branch “naive” result

with Lebedev quadrature

h.1.2 Branch “standard” result

BIBLIOGRAPHY

- [1] URL: https://en.wikipedia.org/wiki/Correlation_function (cit. on p. 28).
- [2] H. J. C. Berendsen, J. R. Grigera, and T. P. Straatsma. “The missing term in effective pair potentials.” In: *The Journal of Physical Chemistry* 91.24 (1987), pp. 6269–6271. DOI: [10.1021/j100308a038](https://doi.org/10.1021/j100308a038) (cit. on pp. 20, 21, 106).
- [3] H.J.C. Berendsen, J.P.M. Postma, W.F. van Gunsteren, and J. Hermans. “Interaction models for water in relation to protein hydration.” In: *Intermolecular Forces*. Ed. by B. Pullman. D. Reidel, 1981, pp. 331–342 (cit. on p. 21).
- [4] L. Blum. “Invariant Expansion. II. The Ornstein Zernike-Equation for Nonspherical Molecules and an Extended Solution to the Mean Spherical Model.” In: *The Journal of Chemical Physics* 57.5 (1972), pp. 1862–1869. DOI: <http://dx.doi.org/10.1063/1.1678503> (cit. on pp. 4, 41, 51, 60, 63, 64, 177).
- [5] L. Blum and A. J. Torruella. “Invariant Expansion for Two-Body Correlations: Thermodynamic Functions, Scattering, and the Ornstein-Zernike Equation.” In: *The Journal of Chemical Physics* 56.1 (1972), pp. 303–310. DOI: <http://dx.doi.org/10.1063/1.1676864> (cit. on pp. 4, 41, 51, 60, 63, 153, 177).
- [6] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. “A Limited Memory Algorithm for Bound Constrained Optimization.” In: *SIAM J. Sci. Comput.* 16.5 (Sept. 1995), pp. 1190–1208. DOI: [10.1137/0916069](https://doi.org/10.1137/0916069) (cit. on p. 46).
- [7] David Chandler and Hans C. Andersen. “Optimized Cluster Expansions for Classical Fluids. II. Theory of Molecular Liquids.” In: *The Journal of Chemical Physics* 57.5

- (1972), pp. 1930–1937. DOI: <http://dx.doi.org/10.1063/1.1678513> (cit. on p. 4).
- [8] Martin Chaplin. *Water Model*. URL: http://www1.lsbu.ac.uk/water/water_models.html (cit. on p. 20).
- [9] Cheol Ho Choi, Joseph Ivanic, Mark S. Gordon, and Klaus Ruedenberg. “Rapid and stable determination of rotation matrices between spherical harmonics by direct recursion.” In: *The Journal of Chemical Physics* 111.19 (1999), pp. 8825–8831. DOI: <http://dx.doi.org/10.1063/1.480229> (cit. on p. 167).
- [10] Snehasis Chowdhuri, Ming-Liang Tan, and Toshiko Ichiye. “Dynamical properties of the soft sticky dipole-quadrupole-octupole water model: A molecular dynamics study.” In: *The Journal of Chemical Physics* 125.14 (2006). DOI: <http://dx.doi.org/10.1063/1.2357117> (cit. on p. 20).
- [11] Christopher J. Cramer and Donald G. Truhlar. “Implicit Solvation Models: Equilibria, Structure, Spectra, and Dynamics.” In: *Chemical Reviews* 99.8 (1999). PMID: 11849023, pp. 2161–2200. DOI: [10.1021/cr960149m](https://doi.org/10.1021/cr960149m) (cit. on pp. 3, 10).
- [12] Liem X. Dang, Julia E. Rice, and Peter A. Kollman. “The effect of water models on the interaction of the sodium–chloride ion pair in water: Molecular dynamics simulations.” In: *The Journal of Chemical Physics* 93.10 (1990), pp. 7528–7529. DOI: <http://dx.doi.org/10.1063/1.459714> (cit. on p. 5).
- [13] A. R. Edmonds. *Angular Momentum in Quantum Mechanics*. 1960 (cit. on pp. 153–156, 167, 173, 177).
- [14] R. Evans. “The nature of the liquid-vapour interface and other topics in the statistical mechanics of non-uniform, classical fluids.” In: *Advances in Physics* 28.2 (1979), pp. 143–200. DOI: [10.1080/00018737900101365](https://doi.org/10.1080/00018737900101365) (cit. on pp. 4, 25).

- [15] R. Evans. “Density Functionals in the Theory of Nonuniform Fluids.” In: *Fundamentals of inhomogeneous fluids*. Ed. by Douglas. Henderson. Marcel Dekker, 1992 (cit. on p. 25).
- [16] Sebastiao Formosinho and Monica Barroso, eds. *Proton-Coupled Electron Transfer. A Carrefour of Chemical Reactivity Traditions*. RSC Catalysis Series. The Royal Society of Chemistry, 2012, P001–157. doi: 10.1039/9781849733168. URL: <http://dx.doi.org/10.1039/9781849733168> (cit. on p. 1).
- [17] P. H. Fries and G. N. Patey. “The solution of the hypernetted-chain approximation for fluids of nonspherical particles. A general method with application to dipolar hard spheres.” In: *The Journal of Chemical Physics* 82.1 (Jan. 1985), pp. 429–440. doi: doi:10.1063/1.448764. (Visited on 01/15/2013) (cit. on pp. 4, 41, 51, 63, 64, 153, 177).
- [18] M. Frigo and S.G. Johnson. “The Design and Implementation of FFTW3.” In: *Proceedings of the IEEE* 93.2 (2005), pp. 216–231. doi: 10.1109/JPROC.2004.840301 (cit. on pp. 70, 73, 96).
- [19] Peter Gács and László Lovász. *Complexity of Algorithms*. Lecture Notes. 1999 (cit. on p. 139).
- [20] Gaussian 09 keyword: SCRF. 2014. URL: http://www.gaussian.com/g_tech/g_ur/k_scrf.htm (cit. on p. 12).
- [21] Lionel Gendre, Rosa Ramirez, and Daniel Borgis. “Classical density functional theory of solvation in molecular solvents: Angular grid implementation.” In: *Chemical Physics Letters* 474.4-6 (June 2009), pp. 366–370. doi: 10.1016/j.cplett.2009.04.077 (cit. on p. 4).
- [22] C.G. Gray and K.E. Gubbins. *Theory of Molecular Fluids: I: Fundamentals*. International Series of Monographs on Chemistry. OUP Oxford, 1984. URL: <http://>

- //books.google.de/books?id=3mz2RcnnMGwC (cit. on pp. 4, 9, 14, 20, 55, 156, 167, 173, 177).
- [23] Georg Hager and Gerhard Wellein. *Formation lecture: Node-Level Performance Engineering*. Dec. 2015. URL: <http://moodle.rrze.uni-erlangen.de/course/view.php?id=274> (cit. on p. 142).
- [24] J. P. Hansen. “Basic Statistical Theory of Liquids.” In: *The Physics and Chemistry of Aqueous Ionic Solutions*. Ed. by M.-C Bellissent-Funel and G. W. Neilson. Dordrecht: Springer Netherlands, 1987, pp. 1–59. DOI: [10.1007/978-94-009-3911-0_1](https://doi.org/10.1007/978-94-009-3911-0_1). URL: http://dx.doi.org/10.1007/978-94-009-3911-0_1 (cit. on p. 4).
- [25] Jean-Pierre Hansen and Ian R. McDonald. “{CHAPTER} 6 - Perturbation Theories.” In: *Theory of Simple Liquids (Second Edition)*. Ed. by Jean-Pierre Hansen and Ian R. McDonald. Second Edition. London: Academic Press, 1986, pp. 145 –192. DOI: <http://dx.doi.org/10.1016/B978-0-08-057101-0.50010-X>. URL: <http://www.sciencedirect.com/science/article/pii/B978008057101050010X> (cit. on p. 25).
- [26] Jean-Pierre Hansen and Ian R. McDonald. *Theory of Simple Liquids*. Ed. by Jean-Pierre Hansen and Ian R. McDonald. Fourth Edition. Oxford: Academic Press, 2013 (cit. on pp. 4, 14, 25).
- [27] Robert B. Hermann. “Theory of hydrophobic bonding. II. Correlation of hydrocarbon solubility in water with solvent cavity surface area.” In: *The Journal of Physical Chemistry* 76.19 (1972), pp. 2754–2759. DOI: [10.1021/j100663a023](https://doi.org/10.1021/j100663a023) (cit. on p. 11).
- [28] Fumio Hirata, ed. *Molecular Theory of Solvation*. en. Vol. 24. Understanding Chemical Reactivity. Dordrecht: Kluwer Academic Publishers, 2004. URL: <http://link.springer.com/10.1007/1-4020-2590-4> (visited on 08/10/2016) (cit. on pp. 4, 35).
- [29] MJ Holst. “The Poisson-Boltzmann Equation.” In: () (cit. on pp. 12, 13).

- [30] Laura D. Hughes, David S. Palmer, Florian Nigsch, and John B. O. Mitchell. “Why are some properties more difficult to predict than others? A study of QSPR models of solubility, melting point, and Log P.” In: *Journal of Chemical Information and Modeling* 48.1 (Jan. 2008), pp. 220–232. DOI: 10.1021/ci700307p (cit. on p. 1).
- [31] Philippe Hunenberger and Maria Reif. “Chapter 2 Fundamental Experimental Problems.” In: *Single-Ion Solvation: Experimental and Theoretical Approaches to Elusive Thermodynamic Quantities*. The Royal Society of Chemistry, 2011, pp. 8–38. DOI: 10.1039/9781849732222-00008. URL: <http://dx.doi.org/10.1039/9781849732222-00008> (cit. on p. 79).
- [32] Guillaume Jeanmairet. “A molecular density functional theory to study solvation in water.” Theses. Université Pierre et Marie Curie - Paris VI, July 2014 (cit. on p. 4).
- [33] Guillaume Jeanmairet, Maximilien Levesque, Rodolphe Vuilleumier, and Daniel Borgis. “Molecular Density Functional Theory of Water.” In: *The Journal of Physical Chemistry Letters* 4 (Jan. 2013), pp. 619–624. DOI: 10.1021/jz301956b (cit. on p. 4).
- [34] Guillaume Jeanmairet, Maximilien Levesque, Volodymyr Sergiievskyi, and Daniel Borgis. “Molecular density functional theory for water with liquid-gas coexistence and correct pressure.” In: *The Journal of Chemical Physics* 142.15 (Apr. 2015), p. 154112. DOI: 10.1063/1.4917485. (Visited on 04/17/2015) (cit. on p. 4).
- [35] Guillaume Jeanmairet, Nicolas Levy, Maximilien Levesque, and Daniel Borgis. “Molecular density functional theory of water including density–polarization coupling.” In: *Journal of Physics: Condensed Matter* 28.24 (June 2016), p. 244005. DOI: 10.1088/0953-8984/28/24/244005 (cit. on p. 4).
- [36] Frank Jensen. *Introduction to Computational Chemistry*. John Wiley & Sons, 2006 (cit. on pp. 3, 10, 12, 14).

- [37] William L. Jorgensen and Julian Tirado-Rives. “Free energies of hydration for organic molecules from Monte Carlo simulations.” In: *Perspectives in Drug Discovery and Design* 3.1 (1995), pp. 123–138. DOI: 10.1007/BF02174470 (cit. on p. 3).
- [38] M. A. Kastenholz and Philippe H. Hünenberger. “Computation of methodology-independent ionic solvation free energies from molecular simulations. I. The electrostatic potential in molecular liquids.” In: *The Journal of Chemical Physics* 124.12, 124106 (2006). DOI: <http://dx.doi.org/10.1063/1.2172593> (cit. on pp. 79, 80).
- [39] Mika A. Kastenholz and Philippe H. Hünenberger. “Computation of methodology-independent ionic solvation free energies from molecular simulations. II. The hydration free energy of the sodium cation.” In: *The Journal of Chemical Physics* 124.22, 224501 (2006). DOI: <http://dx.doi.org/10.1063/1.2201698> (cit. on p. 79).
- [40] Peter G. Kusalik and Igor M. Svishchev. “The Spatial Structure in Liquid Water.” In: *Science* 265.5176 (1994), pp. 1219–1221. DOI: 10.1126/science.265.5176.1219 (cit. on p. 21).
- [41] B. Lee and F.M. Richards. “The interpretation of protein structures: Estimation of static accessibility.” In: *Journal of Molecular Biology* 55.3 (1971), 379 –IN4. DOI: [http://dx.doi.org/10.1016/0022-2836\(71\)90324-X](http://dx.doi.org/10.1016/0022-2836(71)90324-X) (cit. on p. 11).
- [42] Maximilien Levesque, Virginie Marry, Benjamin Rotenberg, Guillaume Jeanmairet, Rodolphe Vuilleumier, and Daniel Borgis. “Solvation of complex surfaces via molecular density functional theory.” In: *The Journal of Chemical Physics* 137.22 (Dec. 2012), pp. 224107–224107–8. DOI: doi:10.1063/1.4769729. (Visited on 12/13/2012) (cit. on p. 4).
- [43] Nicolas Levy, Daniel Borgis, and Massimo Marchi. “A dielectric continuum model of solvation for complex solutes.” In: *Computer Physics Communications* 169.1–3 (2005). Proceedings of the Europhysics Conference on Computational Physics

- 2004CCP 2004Europhysics Conference on Computational Physics 2004, pp. 69–74.
DOI: <http://dx.doi.org/10.1016/j.cpc.2005.03.018> (cit. on p. 13).
- [44] Massimo Marchi, Daniel Borgis, Nicolas Levy, and Pietro Ballone. “A dielectric continuum molecular dynamics method.” In: *The Journal of Chemical Physics* 114.10 (2001), pp. 4377–4385. DOI: <http://dx.doi.org/10.1063/1.1348028> (cit. on pp. 13, 38).
- [45] Lorenzo Maschio, Bernard Kirtman, Roberto Orlando, and Michel Rèrat. “Ab initio analytical infrared intensities for periodic systems through a coupled perturbed Hartree-Fock/Kohn-Sham method.” In: *The Journal of Chemical Physics* 137.20, 204113 (2012). DOI: <http://dx.doi.org/10.1063/1.4767438> (cit. on p. 80).
- [46] A. D. Mcnaught and A. Wilkinson. *IUPAC. Compendium of Chemical Terminology*, 2nd ed. (the "Gold Book"). WileyBlackwell; 2nd Revised edition edition (cit. on p. 2).
- [47] N. David Mermin. “Thermal Properties of the Inhomogeneous Electron Gas.” In: *Physical Review* 137.5A (Mar. 1965), A1441–A1443. DOI: [10.1103/PhysRev.137.A1441](https://doi.org/10.1103/PhysRev.137.A1441). (Visited on 02/10/2014) (cit. on p. 4).
- [48] Albert Messiah. *Quantum Mechanics*. Quantum Mechanics vol. 2. North-Holland, 1981. URL: https://books.google.fr/books?id=VR93vUk8d__C (cit. on pp. 55, 153, 155, 156, 158, 167, 173, 177).
- [49] J. Miyazaki, J. A. Barker, and G. M. Pound. “A new Monte Carlo method for calculating surface tension.” In: *The Journal of Chemical Physics* 64.8 (1976), pp. 3364–3369. DOI: <http://dx.doi.org/10.1063/1.432627> (cit. on p. 16).
- [50] B. R. A. Nijboer and Th. W. Ruijgrok. “On the energy per particle in three- and two-dimensional Wigner lattices.” In: *Journal of Statistical Physics* 53.1 (), pp. 361–382. DOI: [10.1007/BF01011562](https://doi.org/10.1007/BF01011562) (cit. on p. 80).

- [51] Trenton H. Parsell, Meng-Yin Yang, and A. S. Borovik. “C-H Bond Cleavage with Reductants: Re-Investigating the Reactivity of Monomeric MnIII/IV-Oxo Complexes and the Role of Oxo Ligand Basicity.” In: *Journal of the American Chemical Society* 131.8 (2009). PMID: 19196005, pp. 2762–2763. DOI: 10.1021/ja8100825 (cit. on p. 1).
- [52] German L. Perlovich, Tatyana V. Volkova, and Annette Bauer-Brandl. “Towards an understanding of the molecular mechanism of solvation of drug molecules: A thermodynamic approach by crystal lattice energy, sublimation, and solubility exemplified by paracetamol, acetanilide, and phenacetin.” In: *Journal of Pharmaceutical Sciences* 95.10 (), pp. 2158–2169. DOI: 10.1002/jps.20674 (cit. on p. 1).
- [53] GermanL Perlovich and Annette Bauer-Brandl. “Solvation of Drugs as a Key for Understanding Partitioning and Passive Transport Exemplified by NSAIDs.” In: *Chemical Processes with Participation of Biological and Related Compounds*. CRC Press, 2008, pp. 291–325. DOI: doi:10.1201/b12241-9. URL: <http://dx.doi.org/10.1201/b12241-9> (cit. on p. 1).
- [54] Joël Puibasset and Luc Belloni. “Bridge function for the dipolar fluid from simulation.” In: *The Journal of Chemical Physics* 136.15 (Apr. 2012), p. 154503. DOI: doi:10.1063/1.4703899. (Visited on 10/14/2013) (cit. on pp. 40, 91, 93, 99).
- [55] R E Raab and O L De Lange. *Multipole theory in electromagnetism: classical, quantum, and symmetry aspects, with applications*. International series of monographs on physics. Oxford: Clarendon Press, 2005. URL: <https://cds.cern.ch/record/859649> (cit. on p. 81).
- [56] Rosa Ramirez and Daniel Borgis. “Density Functional Theory of Solvation and Its Relation to Implicit Solvent Models.” In: *The Journal of Physical Chemistry B* 109.14 (2005), pp. 6754–6763. DOI: 10.1021/jp045453v. (Visited on 07/05/2010) (cit. on p. 4).

- [57] Rosa Ramirez, Ralph Gebauer, Michel Mareschal, and Daniel Borgis. “Density functional theory of solvation in a polar solvent: Extracting the functional from homogeneous solvent simulations.” In: *Physical Review E* 66.3 (2002), pp. 031206–031206–8. DOI: [10.1103/PhysRevE.66.031206](https://doi.org/10.1103/PhysRevE.66.031206) (cit. on p. 4).
- [58] Benoit Roux and Thomas Simonson. “Implicit solvent models.” In: *Biophysical Chemistry* 78.1–2 (Apr. 1999), pp. 1–20. DOI: [10.1016/S0301-4622\(98\)00226-9](https://doi.org/10.1016/S0301-4622(98)00226-9). (Visited on 04/29/2014) (cit. on p. 13).
- [59] V.R. Saunders, C. Freyria-Fava, R. Dovesi, L. Salasco, and C. Roetti. “On the electrostatic potential in crystalline systems where the charge density is expanded in Gaussian functions.” In: *Molecular Physics* 77.4 (1992), pp. 629–665. DOI: [10.1080/00268979200102671](https://doi.org/10.1080/00268979200102671) (cit. on p. 80).
- [60] Volodymyr P. Sergiievskyi, Guillaume Jeanmairet, Maximilien Levesque, and Daniel Borgis. “Fast Computation of Solvation Free Energies with Molecular Density Functional Theory: Thermodynamic-Ensemble Partial Molar Volume Corrections.” In: *The Journal of Physical Chemistry Letters* 5.11 (2014), pp. 1935–1942. DOI: [10.1021/jz500428s](https://doi.org/10.1021/jz500428s) (cit. on p. 4).
- [61] Pier Luigi Silvestrelli and Michele Parrinello. “Structural, electronic, and bonding properties of liquid water from first principles.” In: *The Journal of Chemical Physics* 111.8 (1999), pp. 3572–3580. DOI: <http://dx.doi.org/10.1063/1.479638> (cit. on p. 21).
- [62] David van der Spoel, Paul J. van Maaren, and Herman J. C. Berendsen. “A systematic study of water models for molecular simulation: Derivation of water models optimized for use with a reaction field.” In: *The Journal of Chemical Physics* 108.24 (1998), pp. 10220–10230. DOI: <http://dx.doi.org/10.1063/1.476482> (cit. on p. 21).

- [63] Michiel Sprik and Michael L. Klein. “A polarizable model for water using distributed charge sites.” In: *The Journal of Chemical Physics* 89.12 (1988), pp. 7556–7560. DOI: <http://dx.doi.org/10.1063/1.455722> (cit. on p. 5).
- [64] P. Tarazona and R. Evans. “A simple density functional theory for inhomogeneous liquids.” In: *Molecular Physics* 52.4 (1984), pp. 847–857. DOI: 10.1080/00268978400101601 (cit. on p. 25).
- [65] Jacopo Tomasi, Benedetta Mennucci, and Roberto Cammi. “Quantum Mechanical Continuum Solvation Models.” In: *Chemical Reviews* 105.8 (2005), pp. 2999–3094. DOI: 10.1021/cr9904009 (cit. on pp. 3, 12).
- [66] Jacopo Tomasi and Maurizio Persico. “Molecular Interactions in Solution: An Overview of Methods Based on Continuous Distributions of the Solvent.” In: *Chemical Reviews* 94.7 (1994), pp. 2027–2094. DOI: 10.1021/cr00031a013 (cit. on pp. 3, 10, 12).
- [67] Samuel Williams, Andrew Waterman, and David Patterson. “Roofline: An Insightful Visual Performance Model for Multicore Architectures.” In: *Commun. ACM* 52.4 (Apr. 2009), pp. 65–76. DOI: 10.1145/1498765.1498785 (cit. on p. 141).
- [68] Shuangliang Zhao, Rosa Ramirez, Rodolphe Vuilleumier, and Daniel Borgis. “Molecular density functional theory of solvation: From polar solvents to water.” In: *The Journal of Chemical Physics* 134.19, 194102 (2011). DOI: <http://dx.doi.org/10.1063/1.3589142> (cit. on pp. 4, 20, 40, 51).
- [69] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. “Algorithm 778: LBFGS-B: Fortran subroutines for large-scale bound-constrained optimization.” In: *ACM Transactions on Mathematical Software* 23.4 (Dec. 1997), pp. 550–560. DOI: 10.1145/279232.279236. (Visited on 06/09/2011) (cit. on p. 46).