

**LUÍS FERNANDO MERCIER FRANCO**

**STUDY ON THE THERMODYNAMICS OF  
BOVINE SERUM ALBUMIN AQUEOUS  
SOLUTIONS: EXPERIMENTS, MODELING  
AND MOLECULAR SIMULATIONS**

Tese apresentada à Escola Politécnica da  
Universidade de São Paulo para obtenção  
do Título de Doutor em Ciências.

São Paulo  
2016

**LUÍS FERNANDO MERCIER FRANCO**

**STUDY ON THE THERMODYNAMICS OF  
BOVINE SERUM ALBUMIN AQUEOUS  
SOLUTIONS: EXPERIMENTS, MODELING  
AND MOLECULAR SIMULATIONS**

Tese apresentada à Escola Politécnica da  
Universidade de São Paulo para obtenção  
do Título de Doutor em Ciências.

Área de Concentração:  
Engenharia Química

Orientador:  
Prof. Dr. Pedro de Alcântara Pessoa  
Filho

São Paulo  
2016

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 12 de fevereiro de 2016.

Assinatura do autor

Assinatura do orientador

### Catálogo-na-publicação

Franco, Luís Fernando Mercier

Study on the thermodynamics of bovine serum albumin aqueous solutions: experiments, modeling and molecular simulations/ L.F. M. Franco. – ed. rev. – São Paulo, 2016.

224 p.

Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Química.

1.Chemical Thermodynamics 2.Statistical Mechanics 3.Molecular Simulation 4.Bovine Serum Albumin I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Química. II.t.

*AD MAJOREM DEI GLORIAM.*

‘τὸ γὰρ τοι θάνατον δεδιέναι, ὧς ἄνδρες, οὐδὲν ἄλλο ἐστὶν ἢ δοκεῖν σοφὸν εἶναι μὴ ὄντα · δοκεῖν γὰρ εἰδέναι ἐστὶν ἃ οὐκ οἶδεν. οἶδε μὲν γὰρ οὐδεὶς τὸν θάνατον οὐδ’ εἰ τυγχάνει τῷ ἀνθρώπῳ πάντων μέγιστον ὃν τῶν ἀγαθῶν, δεδίασι δ’ ὥς εὖ εἰδότες ὅτι μέγιστον τῶν κακῶν ἐστὶ. καὶ τοῦτο πῶς οὐκ ἀμαθία ἐστὶν αὕτη ἢ ἐπονείδιστος, ἢ τοῦ οἴεσθαι εἰδέναι ἃ οὐκ οἶδεν;’

(ΑΠΟΛΟΓΙΑ ΣΩΚΡΑΤΟΥΣ, ΠΛΑΤΟΝ)

“For to fear death, gentlemen, is nothing else than to think one is wise when one is not; for it is thinking one knows what one does not know. For no one knows whether death be not even the greatest of all blessings to man, but they fear it as if they knew that is the greatest of evils. And is not this the most reprehensible form of ignorance, that of thinking one knows what one does not know?”

(The Apology, Plato - translated by Harold North Fowler)

“Porque ter medo da morte, senhores, outra coisa não é senão considerar-se sábio; equivale a imaginar alguém que sabe o que ignora. Ninguém sabe o que seja a morte, e, ignorando até mesmo se porventura não será para os homens o maior dos bens, temem-na como se soubessem com certeza que é o maior dos males. E como poderá deixar de ser censurável semelhante ignorância, isto é, imaginar alguém que sabe o que não sabe?”

(A Apologia de Sócrates, Platão - traduzido por Carlos Alberto Nunes)

# ACNOWLEDGMENTS

(In Portuguese)

São devidos muitos agradecimentos. Em primeiro lugar, ao Prof. Dr. Pedro de Alcântara Pessôa Filho. Agradeço pela sua disponibilidade imensa, pela sua atenção e envolvimento com o trabalho, por ter desde cedo me incentivado a estudar Termodinâmica, Mecânica Estatística e Simulação Molecular e, acima de tudo, pela amizade que prezarei sempre.

Aos amigos e colegas do Grupo de Engenharia de Bioprocessos. Em especial àqueles que contribuíram de forma mais direta com a presente tese: Prof. Dr. Adriano Rodrigues Azzoni, que muito ajudou com questões burocráticas; Paula Cristina Nicioli, que colaborou na parte experimental; assim como as técnicas Orlinda e Andréa sempre dispostas a ajudar. Aos funcionários da secretaria do Departamento de Engenharia Química: Maria Elisete, Graça, Alexandre e Valter.

Ao pessoal do Laboratório de Cristalografia do Instituto de Física da Universidade de São Paulo: Prof. Dr. Cristiano Luis Pinto de Oliveira e Táris Mendes Germano que colaboraram muito com a execução dos experimentos de espalhamento de raio-X a baixos ângulos.

Aos amigos e professores do Departamento de Engenharia Química: Prof. Dr. André Gonçalves Antunha e Prof. Dr. José Luis de Paiva com os quais durante todo o tempo do doutoramento pude discutir a fundo problemas científicos, nem sempre diretamente ligados a tese.

Aos alunos do curso de Engenharia Química ingressantes em 2014. Com eles pude conviver como monitor em três disciplinas: Termodinâmica Química I, Termodinâmica Química II e Probabilidade. Tenho certeza que eu aprendi mais com eles do que o contrário. Que a profissão que eles escolheram traga-lhes tanta felicidade quanto sempre trouxe a mim.

Aos amigos Eng. Diego Lanfredi Cádiz e Prof. Dr. José Luis de Paiva pela revisão de parte do texto desta tese. E, em particular, ao Eng. Rafael Risnik Romeiro pela contribuição ao cálculo do fator de forma no último capítulo desta tese.

À FAPESP, pelas bolsas concedidas para o doutoramento e para o estágio no exterior, e também à CAPES e ao CNPq.

Por fim aos meus familiares. Ao meu avô Antônio Sérgio Pacheco Mercier que sempre me estimulou nos estudos e sem o qual eu jamais teria chegado aqui. À minha avó Zenaide Goulart Pacheco Mercier pelo apoio incondicional. À minha mãe, Priscila Goulart Pacheco Mercier, e ao meu irmão, Henrique Mercier Franco, pela constante solicitude. À Letícia Falasqui Tachinardi Rocha com quem convivi durante todo o período desta tese; sem o seu carinho este caminho teria sido bem mais árido.

(In English)

I would like to thank Prof. Dr. Edward J. Maginn, who kindly accepted me in his group at the University of Notre Dame for eight months. Without his help most of this thesis would have been impossible.

I also would like to thank all my colleagues and friends from Maginn's group: Prof. Dr. Jindal Shah, Akash Sharma (*in memoriam*), Akihito Otani, Andrea Revecó, Brian Yoo, Cassiano Aimoli, Eliseo Marin Rimoldi, Daniela Kerlé, Hao Wu, Katie Maerzke, Marta Batista, Quintin Sheridan, Ramesh Singh, Samir Budhathoki, Sandip Khan, Surya Prakash Tiwari, and Yong Zhang.

I would like to thank the Center for Research Computing (CRC) of the University of Notre Dame for all the computational resources that I have used during this work.

Finally, I am greatly indebted to all the members of the committee: Prof. Dr. Edward J. Maginn, Prof. Dr. Marcelo Castier, Prof. Dr. Charles Rubber de Almeida Abreu, Prof. Dr. Everson Alves Miranda and Prof. Dr. Pedro de Alcântara Pessoa Filho for all the valuable comments and insightful discussions for this revised version of the dissertation.

## RESUMO

Nesta tese apresenta-se uma investigação sobre a interação entre duas proteínas em soluções aquosas salinas. Experimentos, modelagem e simulações moleculares foram realizadas para conseguir um melhor entendimento do fenômeno. Albumina de soro bovino foi usada como proteína modelo. Uma expressão para o fator de estrutura de proteínas globulares em solução aquosa é apresentada neste trabalho. Esta expressão foi obtida considerando-se um potencial intermolecular dado pela soma de um núcleo duro, uma contribuição atrativa tipo van der Waals e uma contribuição de potencial coulômbico blindado. Dados experimentais de espalhamento de raios-X a baixos ângulos para a albumina de soro bovino em soluções aquosas contendo sais de sódio com diferentes concentrações de proteína e valores de pH também são apresentados. A expressão desenvolvida para o fator de estrutura descreve com precisão estes dados experimentais, desde que uma dependência entre o parâmetro atrativo com a concentração de proteína seja estabelecida. Uma expressão para a pressão osmótica foi derivada do fator de estrutura. Com parâmetros atrativos ajustados aos dados de espalhamento de raios-X, a pressão osmótica da albumina de soro bovino em solução aquosa pôde ser predita com grande correlação com os dados experimentais. Uma derivação dos potenciais termodinâmicos usando a nova equação osmótica de estado é apresentada. Aplicando o critério de equilíbrio de fases, foi possível calcular o equilíbrio fluido-fluido para a albumina de soro bovino em solução aquosa. Embora tal separação não tenha sido observada experimentalmente em um pH igual ao ponto isoelétrico, ela foi de fato observada experimentalmente para um valor de pH menor do que o ponto isoelétrico. As predições parecem ser valiosas para discutir como a especificidade iônica afeta o diagrama de fases de proteínas. De modo a avaliar como proteínas interagem umas com as outras usando técnicas de dinâmica molecular, dois novos campos de força *coarse-grained* são propostos. O primeiro, para o sulfato de sódio em solução aquosa, evita a associação não-física que é observada para campos de força atomísticos não-polarizáveis. Este modelo é capaz de prever propriedades dinâmicas e termodinâmicas. O segundo, para a albumina de soro bovino em solução aquosa, é usado como uma nova estratégia para avaliar o fator de forma de espalhamento de proteínas como uma ferramenta de baixa resolução na predição de estruturas proteicas.



# ABSTRACT

The interaction between two proteins into salt aqueous solutions is investigated throughout this thesis. Experiments, modeling and molecular simulations were carried out to get a better understanding of the phenomenon. Bovine serum albumin was used as a model protein. An analytical expression for the structure factor for globular proteins in aqueous solution is presented in this work. This expression was obtained considering an intermolecular potential given by the sum of a hard core, a van der Waals attractive and a screened Coulomb contribution. Experimental data of Small Angle X-Ray Scattering for bovine serum albumin in aqueous solutions containing sodium salts at different protein concentrations and pH values are also presented. The expression developed for the structure factor describes accurately these experimental data provided a dependence of the attractive parameter on protein concentration is established. An expression for the osmotic pressure was derived from the structure factor. With attractive parameters adjusted from X-ray scattering data, the osmotic pressure of bovine serum albumin aqueous solutions could be predicted with very good agreement with experimental data. A derivation of the thermodynamic potentials, such as the chemical potential, using the new osmotic equation of state is presented. Applying the phase equilibrium criterion, the fluid-fluid phase equilibrium for bovine serum albumin in salt aqueous solution was calculated. Although such separation was not experimentally observed at the isoelectric point, it was indeed experimentally observed for a pH value below the isoelectric point. The predictions seem to be valuable to discuss how ion specificity affects the phase diagram of proteins. To apply molecular dynamic techniques to simulate how proteins interact to each other in salt aqueous solutions, two new coarse-grained force fields are proposed. The first one, meant for sodium sulfate aqueous solution, avoids the unphysical association observed for non-polarizable atomistic force fields; and allows the prediction of thermodynamic and dynamic properties. The second one, meant for bovine serum albumin in aqueous solution, is used as a new strategy to evaluate the scattering form factor of proteins as a low resolution technique for protein structure prediction.

# CONTENTS

## List of Figures

## List of Tables

<b>1</b>	<b>Introduction</b>	<b>27</b>
1.1	Funding . . . . .	29
1.2	Additional Information . . . . .	29
<b>2</b>	<b>Prolegomena to integral equation theory</b>	<b>30</b>
2.1	Distribution functions . . . . .	30
2.2	Static structure factor . . . . .	33
2.3	Percus-Yevick Approximation for hard spheres . . . . .	35
2.4	Mean Spherical Approximation for macroions in solution . . . . .	37
<b>3</b>	<b>Basic radiation scattering theory</b>	<b>42</b>
3.1	Driven damped harmonic oscillator . . . . .	43
3.2	Scattering intensity . . . . .	44
3.3	Scattering of a system with many atoms . . . . .	47
3.4	Form factor expressions . . . . .	48
<b>4</b>	<b>Introduction to molecular simulation</b>	<b>51</b>
4.1	Molecular dynamics . . . . .	53

4.1.1	Verlet algorithm . . . . .	54
4.1.2	Leap Frog algorithm . . . . .	55
4.2	Monte Carlo simulations . . . . .	56
4.2.1	Markov chains . . . . .	56
4.2.2	Microscopic reversibility . . . . .	58
4.2.3	Metropolis Algorithm . . . . .	59
4.2.4	Monte Carlo algorithm . . . . .	60
<b>5</b>	<b>From the structure factor to osmotic pressure</b>	<b>61</b>
5.1	Introduction . . . . .	61
5.2	Materials and Methods . . . . .	65
5.2.1	Reagents . . . . .	65
5.2.2	SAXS experiments . . . . .	65
5.2.3	Monte Carlo simulations . . . . .	67
5.3	Theoretical framework . . . . .	68
5.3.1	Isothermal compressibility and the osmotic equation of state	73
5.4	Results . . . . .	74
5.4.1	Validation of the structure factor . . . . .	74
5.4.2	Structure factor of Bovine Serum Albumin (BSA) in aqueous salt solutions . . . . .	75
5.4.3	Osmotic equation of state . . . . .	82
<b>6</b>	<b>From the osmotic pressure to fluid-fluid coexistence curve</b>	<b>87</b>

6.1	Introduction . . . . .	87
6.2	Derivation of thermodynamic potentials . . . . .	89
6.3	Condition of equilibrium . . . . .	90
6.4	Coexistence curve through molecular simulation . . . . .	91
6.5	Fluid-fluid coexistence curve . . . . .	93
<b>7</b>	<b>Other related questions</b>	<b>100</b>
7.1	Introduction . . . . .	100
7.2	Sodium sulfate unphysical association . . . . .	100
7.2.1	Introduction . . . . .	100
7.2.2	Force Field . . . . .	103
7.2.3	Molecular dynamic details . . . . .	105
7.2.3.1	Parameterization . . . . .	105
7.2.3.2	Density . . . . .	106
7.2.3.3	Dynamic properties . . . . .	106
7.2.4	Results and discussions . . . . .	108
7.2.4.1	Parameterization of $\alpha$ . . . . .	108
7.2.4.2	Thermodynamic properties . . . . .	109
7.2.4.3	Dynamic properties . . . . .	113
7.3	Critic review on the reference line for Walden plot . . . . .	118
7.4	A coarse-grained force field for Bovine Serum Albumin . . . . .	125
7.4.1	Introduction . . . . .	125

7.4.2	Coarse-grained force field . . . . .	127
7.4.3	Molecular dynamics and form factor calculation details . .	129
7.4.4	Results and Discussions . . . . .	130
<b>8</b>	<b>Conclusions and suggestions for future work</b>	<b>135</b>
	<b>References</b>	<b>138</b>
	<b>Appendix A – The many body effect</b>	<b>148</b>
	<b>Appendix B – Code for Hayter-Penfold structure factor</b>	<b>149</b>
	<b>Appendix C – Codes for Monte Carlo simulations</b>	<b>157</b>
C.1	Code to generate a FCC crystal . . . . .	159
C.2	Code for NVT ensemble Monte Carlo simulation . . . . .	162
C.3	Code to calculate the structure factor and the radial distribution function . . . . .	169
C.4	Code to calculate local particle density along $z$ -direction . . . . .	175
	<b>Appendix D – Code for Nelder and Mead algorithm</b>	<b>179</b>
	<b>Appendix E – Code for phase envelope calculation</b>	<b>183</b>
	<b>Appendix F – Code for viscosity via Green-Kubo</b>	<b>190</b>
	<b>Appendix G – SAXS intensity data</b>	<b>194</b>
	<b>Appendix H – Fitted SAXS intensity data</b>	<b>209</b>

Appendix I – Figures for molten NaCl simulations	217
Appendix J – Plots for viscosity and electric conductivity	220
Appendix K – Relation between the mean diffusivity coefficient and ionic diffusivities	222

# LIST OF FIGURES

1	Radial distribution function as a function of relative distance for three states of matter: gas, liquid and crystal. These states are represented within the plot by snapshots of Monte Carlo simulations.	32
2	Static structure factor for hard sphere fluid with $\eta = 0.25$ . Continuous line, Percus-Yevick Approximation. Open circles, Monte Carlo simulations. . . . .	37
3	Static structure factor for macroions into solution using Hayter-Penfold equation (HAYTER; PENFOLD, 1981) with $\kappa = 0.04$ and $\gamma = 10.0$ . Continuous line, $\eta = 0.3$ . Dashed-dotted line, $\eta = 0.2$ . Dotted line, $\eta = 0.05$ . . . . .	39
4	Static structure factor for macroions into solution using Hayter-Penfold equation (HAYTER; PENFOLD, 1981) with $\eta = 0.05$ and $\kappa = 0.04$ . Continuous line, $\gamma = 100.0$ . Dashed-dotted line, $\gamma = 10.0$ . Dotted line, $\gamma = 0.0$ . . . . .	40
5	Static structure factor for macroions into solution using Hayter-Penfold equation (HAYTER; PENFOLD, 1981) with $\eta = 0.05$ and $\gamma = 10.0$ . Continuous line, $\kappa = 0.20$ . Dashed-dotted line, $\kappa = 0.10$ . Dotted line, $\kappa = 0.04$ . . . . .	41

6	Representation of the oscillating electron-positive nucleous system. The axes represent the coordinate system. The continuous line represents electric field and the dotted line represents the magnetic field. $\vec{n}$ is the unit vector in the same direction as the scattered wave. $\Gamma$ is the angle between $\vec{x}$ and $\vec{n}$ ; and $\theta$ is the angle between $\vec{z}$ and $\vec{n}$ . . . . .	43
7	Scattering intensity as a function of the relative frequency. . . . .	46
8	Homogeneous sphere X-ray scattering scheme. . . . .	49
9	Form factor as a function of $k = qR$ . Continuous line, Equation (3.18) for a homogeneous sphere. Dotted line, Equation (3.19) for an oblate ellipsoid of revolution with $\epsilon = 0.4$ . . . . .	49
10	NVT Monte Carlo simulations of Lennard-Jones particles. The initial configuration on the left side is a face centered cubic (FCC) crystal. The final configuration on the right side is a representation of the melted system. . . . .	60
11	Structure factor (Equation 5.32) as a function of $\eta$ and $k$ (Equation 5.20) calculated for a system of uncharged particles and attractive parameter $\varepsilon/k_B T = 1.67$ . . . . .	75
12	Structure factor as a function of $k = q\sigma$ and $\varepsilon/k_B T$ for a system of uncharged particles with $\eta = 0.2$ . Open circles, Monte Carlo results (standard deviations are smaller than symbol size). Continuous line, Equations (5.14) and (5.32). . . . .	76



- 13 Form factor for bovine serum albumin (BSA) in aqueous solution. Experimental results for the scattering intensity (in a.u., arbitrary units) of BSA in aqueous solution with  $6.8 \text{ mg}\cdot\text{mL}^{-1}$ , open circles. Curve calculated using CRY SOL (SVERGUN; BARBER-ATO; KOCH, 1995), continuous line. The inset plot shows  $\beta(q)$  (Equation 5.3) considering an oblate ellipsoid of revolution. . . . . 77
  
- 14 X-ray scattering intensity (in a.u., arbitrary units) of BSA in  $1.0 \text{ mol}\cdot\text{L}^{-1} \text{ NaNO}_3$  aqueous solution at  $\text{pH}=4.9$  and  $23^\circ\text{C}$  for different BSA concentrations at very small  $q$  values. Open symbols, experimental data; continuous line, Equation (5.32) with  $\varepsilon$  calculated using Equation (5.38); dotted-dashed line, Equation (5.32) with constant  $\varepsilon$ ; dashed line, Hard-Sphere Percus-Yevick solution. 79
  
- 15 A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in  $1.0 \text{ mol}\cdot\text{L}^{-1} \text{ NaNO}_3$  aqueous solution at  $\text{pH}=4.9$  and  $23^\circ\text{C}$ . Open symbols, experimental data; continuous line, Equation (5.32) with  $\varepsilon$  calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with  $\varepsilon$  calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution. . . . . 84
  
- 16 A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in  $1.0 \text{ mol}\cdot\text{L}^{-1} \text{ NaNO}_3$  aqueous solution at  $\text{pH}=6.3$  and  $23^\circ\text{C}$ . Open symbols, experimental data; continuous line, Equation (5.32) with  $\varepsilon$  calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with  $\varepsilon$  calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution. . . . . 85

17	Osmotic pressure of BSA in $1.0 \text{ mol}\cdot\text{L}^{-1}$ NaCl aqueous solution. Open circles, experimental data from Wu and Prausnitz (WU; PRAUSNITZ, 1999) (pH=4.5); continuous line, Equation (5.42) with parameters calculated at pH=4.9; dashed line, Hard-Sphere Percus-Yevick equation of state. . . . .	86
18	Illustrative example of the calculation of equilibrium densities using a single canonical Monte Carlo simulations with 2048 Lennard-Jones particles. The first image represents the initial state with a dense face centered cubic (FCC) configuration on the very right side and vacuum in the rest of the parallelepiped. The second image represents the final configuration exhibiting two phases in equilibrium. The graph at the bottom is a plot of the variation of local particle density as function of direction $z$ . . . . .	94
19	Liquid-liquid coexistence curve for bovine serum albumin aqueous solution at pH=4.9 and $23^\circ\text{C}$ using Equations (5.36) and (6.6) with parameters given at Table 1. Continuous line, $1.0 \text{ mol}\cdot\text{kg}^{-1}$ $\text{NaNO}_3$ . Dotted line, $1.0 \text{ mol}\cdot\text{kg}^{-1}$ $\text{Na}_2\text{SO}_4$ . Dashed-dotted line, $1.0 \text{ mol}\cdot\text{kg}^{-1}$ NaCl. . . . .	95

20	Liquid-liquid coexistence curve for BSA aqueous solution at pH=4.9 and 23°C using Equations (5.36) and (6.6) with parameters given at Table 1. Continuous line, 1.0 mol·kg <sup>-1</sup> NaNO <sub>3</sub> . Dotted line, 1.0 mol·kg <sup>-1</sup> Na <sub>2</sub> SO <sub>4</sub> . Dashed-dotted line, 1.0 mol·kg <sup>-1</sup> NaCl. Open circles, Lennard-Jones liquid-vapor equilibrium calculated in this work. Closed circles, Lennard-Jones liquid-vapor equilibrium calculated using grand-canonical transition-matrix Monte Carlo and histogram re-weighting obtained from NIST (SHEN; SIDERIUS; KRELKELBERG, 2015). Closed squares, Square-well liquid-vapor equilibrium calculated for $\lambda = 1.375$ (LÓPEZ-RENDÓN; REYES; OREA, 2006). . . . .	97
21	Guinier plot for small-angle X-ray scattering of BSA 1.0 mol·kg <sup>-1</sup> NaCl aqueous solution at pH=3.4 and 23°C. Open circles, experimental data for BSA concentration of 25.9 mg·mL <sup>-1</sup> . Open squares, experimental data for BSA concentration of 5.6 mg·mL <sup>-1</sup> . Continuous line, Guinier's approximation to BSA concentration of 25.9 mg·mL <sup>-1</sup> . . . . .	98
22	Liquid-liquid coexistence curve for BSA 1.0 mol·kg <sup>-1</sup> NaCl aqueous solution at pH=3.4 and 23°C using Equations (5.36) and (6.6) with $\varepsilon_0/k_B T = 4.03$ and $\varepsilon'/k_B T = 21.19$ . Continuous line, binodal curve. Dashed line, spinodal curve. . . . .	99
23	Snapshot of a 2.0 ns NPT molecular dynamic simulation for 270 CHARMM19 sodium sulfate particles in 10000 SPC/E water particles at 303.15 K. Oxygens are represented in red, sodium ions in blue and sulfur ions in yellow. Water molecules are represented as lines. . . . .	102

24	Representation of the coarse-grained model. $W$ stands for the coarse-grain water model (SHINODA; DEVANE; KLEIN, 2007), a single sphere which accounts for three atomistic water molecules. $NA$ stands for the coarse-grain sodium ion (SHINODA; DEVANE; KLEIN, 2011), a single sphere which accounts for atomistic sodium ion and three atomistic water molecules. $SU$ stands for the coarse-grain sulfate ion, a single sphere which accounts for the atomistic sulfate ion. The diameters of $W$ , $NA$ and $SU$ are set to be equal.	103
25	Deviation of density, $\chi$ , between the simulated and experimental data (ISONO, 1984) for seven different sodium sulfate molalities (0.05; 0.10; 0.50; 0.70; 1.00; 1.20 and 1.50 mol·kg <sup>-1</sup> ) in aqueous solution at 298.15 K as a function the association parameter $\alpha$ . Open circles, true values of $\chi$ . Continuous line, fitting equation (7.12).	109
26	Coordination number for 1.5 mol·kg <sup>-1</sup> sodium sulfate aqueous solution at 303.15 K. Continuous line, our coarse-grained force field. Dotted-dashed line, non-polarizable atomistic force field.	110
27	Snapshot of a molecular dynamic simulation of the proposed coarse-grained force field for 1.5 mol·kg <sup>-1</sup> sodium sulfate aqueous solutions at 303.15 K. Red beads represent water molecules, blue beads sodium ions and yellow beads sulfate ions.	111
28	Density of sodium sulfate aqueous solution as a function of temperature and salt concentration. Open circles, results by molecular dynamic simulations using the proposed coarse-grained force field. Closed circles, experimental data (ISONO, 1984). The standard deviations of the simulated data are lower than the symbol size.	112

29	Thermal expansion coefficient of sodium sulfate aqueous solution as a function salt concentration. Open circles, results by finite difference method of molecular dynamic simulations using the proposed coarse-grained force field. Closed circles, results by finite difference method using the experimental data (ISONO, 1984). . . .	113
30	Viscosity of $0.05 \text{ mol}\cdot\text{kg}^{-1}$ sodium sulfate aqueous solution as a function of temperature. Open circles, results by molecular dynamic simulations using the proposed coarse-grained force field. Closed circles, experimental data (ISONO, 1984). . . . .	115
31	Ionic conductivity of $0.05 \text{ mol}\cdot\text{kg}^{-1}$ sodium sulfate aqueous solution as a function of temperature. Open circles, results by molecular dynamic simulations using the proposed coarse-grained force field. Closed circles, experimental data (ISONO, 1984). . . . .	116
32	Walden plot for $0.05 \text{ mol}\cdot\text{kg}^{-1}$ sodium sulfate aqueous solution at various values of temperature. Open circles, results by molecular dynamic simulations using the proposed coarse-grained force field. Closed circles, experimental data (ISONO, 1984). Open squares, Nernst-Einstein conductivity obtained from diffusivity calculation by molecular dynamic simulation using the proposed coarse-grained force field. Dotted line, KCl ideal line. . . . .	117
33	Schematic representation of the general analysis on Walden plot for ionic liquids. . . . .	121

34	Walden plot for 1 M aqueous KCl solution experimental data. The dashed line is the KCl ideal line. Ionic conductivities were obtained by Wu <i>et al.</i> (WU <i>et al.</i> , 1994). Densities and dynamic viscosities were obtained by Kestin <i>et al.</i> (KESTIN; KHALIFA; CORREIA, 1981a). And mean diffusivity coefficients were obtained by Longworth (LONGSWORTH, 1957). . . . .	123
35	Walden plot for 0.005 M aqueous NaCl solution experimental data. The dashed line is the KCl ideal line. Limiting molar ionic conductivities were obtained by Gunning <i>et al.</i> (GUNNING; GORDON, 1942). Viscosities were obtained by Kestin <i>et al.</i> (KESTIN; KHALIFA; CORREIA, 1981b). And mean diffusivity coefficients were obtained by Fell <i>et al.</i> (FELL; HUTCHISON, 1971). . . . .	124
36	Snapshots of equilibration and production NPT molecular dynamics of BSA coarse-grained force field. MET beads are gray, GLU are red and HIS are blue. . . . .	131
37	Radius of gyration as a function of time step for NPT molecular dynamics of the coarse-grained BSA in aqueous solution free of salt.	132
38	Form factor of bovine serum albumin. Open circles, small-angle X-ray experimental data (FRANCO; OLIVEIRA; FILHO, 2015). Red line, crysalllographic structure predicion with CRY SOL (SVERGUN; BARBERATO; KOCH, 1995). Continuous line, the proposed coarse-grained force field using CRY SOL to calculate the form factor. Dotted line, ellipsoid of revolution. . . . .	133

39	A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L <sup>-1</sup> NaCl aqueous solution at pH=4.9 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with $\varepsilon$ calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with $\varepsilon$ calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution. . . . .	210
40	A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L <sup>-1</sup> NaNO <sub>3</sub> aqueous solution at pH=4.9 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with $\varepsilon$ calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with $\varepsilon$ calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution. . . . .	211
41	A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L <sup>-1</sup> Na <sub>2</sub> SO <sub>4</sub> aqueous solution at pH=4.9 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with $\varepsilon$ calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with $\varepsilon$ calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution. . . . .	212

42	A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L <sup>-1</sup> NaCl aqueous solution at pH=6.3 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with $\varepsilon$ calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with $\varepsilon$ calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution. . . . .	213
43	A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L <sup>-1</sup> NaNO <sub>3</sub> aqueous solution at pH=6.3 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with $\varepsilon$ calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with $\varepsilon$ calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution. . . . .	214
44	A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L <sup>-1</sup> Na <sub>2</sub> SO <sub>4</sub> aqueous solution at pH=6.3 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with $\varepsilon$ calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with $\varepsilon$ calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution. . . . .	215



45	A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L <sup>-1</sup> NaCl aqueous solution at pH=3.4 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with $\varepsilon$ calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with $\varepsilon$ calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution. . . . .	216
46	Initial FCC configuration at the left side and final molten NaCl configuration at the right side. Blue beads represents sodium ions and green beads represents chloride. . . . .	217
47	Viscosity of molten NaCl calculated from equilibrium molecular dynamics simulations using Green-Kubo relations at 1100 K. The inlet plot shows the stree autocorrelation function. . . . .	218
48	Eletric current autocorrelation function of molten NaCl calculated from equilibrium molecular dynamics simulations at 1300 K. The inlet plot shows the eletric conductivity using Green-Kubo relations.	219
49	Viscosity of 0.05 mol·kg <sup>-1</sup> sodium sulfate aqueous solution at 293.15 K. Red lines represent the standard deviation calculated by five different initial configurations. Continuous black line, average viscosity. Continuous green line, fitted viscosity by Equation (7.15). Dotted line, experimental value (ISONO, 1984). . . . .	220

50	Ionic conductivity of $0.05 \text{ mol}\cdot\text{kg}^{-1}$ sodium sulfate aqueous solution at 293.15 K. Red lines represent the standard deviation calculated by five different initial configurations. Continuous black line, average ionic conductivity. Continuous green line, fitted ionic conductivity by Equation (7.16). Dotted line, experimental value (ISONO, 1984). . . . .	221
----	---	-----

# LIST OF TABLES

1	Values of parameters $\varepsilon^o$ and $\varepsilon'$ , Equation (5.38), values of $\chi$ for the fitting procedure, Equation (5.39), and values of the osmotic second virial coefficient $B$ , Equation (5.44), calculated for BSA in aqueous solutions with $1.0 \text{ mol}\cdot\text{L}^{-1}$ sodium salts. . . . .	78
2	Values of Lennard-Jones critical temperature and density. . . . .	93
3	Critical properties for liquid-liquid equilibrium lines for BSA in salt aqueous solutions at the isoelectric point. . . . .	95
4	Values of dielectric constant of pure water as a function of temperature according to an empirical equation (UEMATSU; FRANCK, 1980). . . . .	105
5	Number of particles for each sodium sulfate concentration. . . . .	106
6	Viscosity and ionic conductivity of molten NaCl obtained from equilibrium molecular dynamics simulations using the Born-Huggins-Mayer-Tosi-Fumi force field and the Green-Kubo relations. . . . .	107
7	Nonbonded interaction parameters for each pair. . . . .	129
8	Description of the list of files in the master file for Monte Carlo simulations . . . . .	158
9	Description of each command in “sci” files. . . . .	159
10	Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector, $q$ , for BSA in $1.0 \text{ mol}\cdot\text{L}^{-1}$ $\text{NaNO}_3$ aqueous solutions at $\text{pH} = 4.9$ , at several protein concentrations, $c_p$ , ( $\text{mg}\cdot\text{mL}^{-1}$ ). . . . .	195

11	Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector, $q$ , for BSA in 1.0 mol·L <sup>-1</sup> NaCl aqueous solutions at pH = 4.9, at several protein concentrations, $c_p$ , (mg·mL <sup>-1</sup> ). . . . .	197
12	Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector, $q$ , for BSA in 1.0 mol·L <sup>-1</sup> Na <sub>2</sub> SO <sub>4</sub> aqueous solutions at pH = 4.9, at several protein concentrations, $c_p$ , (mg·mL <sup>-1</sup> ). . . . .	199
13	Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector, $q$ , for BSA in 1.0 mol·L <sup>-1</sup> NaNO <sub>3</sub> aqueous solutions at pH = 6.3, at several protein concentrations, $c_p$ , (mg·mL <sup>-1</sup> ). . . . .	201
14	Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector, $q$ , for BSA in 1.0 mol·L <sup>-1</sup> NaCl aqueous solutions at pH = 6.3, at several protein concentrations, $c_p$ , (mg·mL <sup>-1</sup> ). . . . .	203
15	Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector, $q$ , for BSA in 1.0 mol·L <sup>-1</sup> Na <sub>2</sub> SO <sub>4</sub> aqueous solutions at pH = 6.3, at several protein concentrations, $c_p$ , (mg·mL <sup>-1</sup> ). . . . .	205
16	Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector, $q$ , for BSA in 1.0 mol·L <sup>-1</sup> NaCl aqueous solutions at pH = 3.4, at several protein concentrations, $c_p$ , (mg·mL <sup>-1</sup> ). . . . .	207

17	Stokes radii for monovalent ions obtained in the literature (PAU; BERG; MCMILLAN, 1990) and values of $\delta$ and $\zeta$ for alkali halides according to Equations (K.11) and (K.10) respectively. . . . .	224
----	--	-----

# 1 INTRODUCTION

The epigraph of this thesis was judiciously chosen. Once I was told that a thesis should bring more questions than answers. Well, this is precisely what Socrates was trying to teach in his defense to the Athenians at the agora. After the conclusion of this thesis, the candidate can honestly state that he knows almost nothing on how proteins interact to each other. This thesis is the presentation of an ensemble of advances on this subject, but certainly not a final word on this matter.

Chemical engineering was born as an industrial need and it will remain so as long as the name endures. Therefore, even though one needs to understand a problem in a deep scientific level, one must not forget the main goal of chemical engineering.

In the last century, bioprocesses and biotechnology evolved at the highest speed ever. Nonetheless, most of the equations and models used in the design and operation of such processes are quite old and not always so effective. Biological systems comprise *per se* the most challenging systems to be studied. As in the traditional chemical industry, most of the cost of operation is due to separation processes (usually called downstream processes) in biochemical plants.

The main goal of this thesis is the investigation on how proteins interact with each other in aqueous saline solutions, and how one may treat and model these interactions to predict the phase diagram, which is the most important

information to design and operate a separation process.

The next three chapters are devoted to brief introductions to three important fields of knowledge: integral equation theory, scattering theory, and molecular simulation. It is important to stress that these chapters are rather concise and are not meant to be an exhaustive text on each of the topics. These subjects are classical topics in Physics and there are several canonical textbooks available. Nevertheless, these chapters are presented because these subjects are not so usual in the body of knowledge of chemical engineering.

The fifth chapter is devoted to the development of a new analytic expression for the structure factor of a fluid whose attractive potential is described by a van der Waals potential. Monte Carlo simulations are used to validate such model and small-angle X-ray scattering experiments with bovine serum albumin (BSA) are fitted with this model. A new equation of state is derived from this structure factor.

The sixth chapter is devoted to the prediction of the liquid-liquid coexistence curve using the new equation of state developed on the fifth chapter. This prediction might be useful to discuss ion specificity. The seventh chapter is a compilation of three main developments: a new coarse-grained force field for sodium sulfate aqueous solutions, a critical review on the universal reference line for Walden plot and a new coarse-grained force field for bovine serum albumin in aqueous solution. The idea behind trying to simulate proteins was to calculate the potential of mean force between two proteins in aqueous solutions. Although this task has not been completed, three interesting results emerged.

Given the relative independence amongst the chapters, a section devoted exclusively to the literature review will not be presented. Instead, the literature review will be distributed amongst the various chapters.

## 1.1 Funding

This project has been developed thanks to the Brazilian agencies São Paulo Research Foundation (FAPESP, grants 2011/22070-5 and 2013/18057-2, R\$125,822.36 + US\$16,320.00), CAPES and CNPq.

## 1.2 Additional Information

This thesis has been elaborated only with free and open source softwares. It is written in L<sup>A</sup>T<sub>E</sub>X using VIM as text editor. All the graphs were plotted with Gnuplot (WILLIAMS; KELLEY; many others, 2012). All molecular images were rendered with VMD (HUMPHREY; DALKE; SCHULTEN, 1996). Other free and open source softwares that have been used are properly cited within the text. The codes written during this project are available on Appendices B, C, D, E and F. They have been compiled with gcc or gfortran and run in Ubuntu 14.04 LTS.



## 2 PROLEGOMENA TO INTEGRAL EQUATION THEORY

The title of this chapter is meant to be strictly understood. There is no intention to write in this section more than some prolegomena. The subject of integral equation theory is one of the richest and broadest branches of statistical mechanics. Detailed reviews on this issue may be found in the literature (BARKER; HENDERSON, 1976; CACCAMO, 1996). Therefore, we shall give only a simple preamble to this topic, including solely the theories from which the structure factors have been used to interpret scattering intensities of proteins into aqueous solutions. To introduce the structure factor from integral equation theories, one may begin at the very definition of the structure factor through its parallel in real space, the radial distribution function.

### 2.1 Distribution functions

If there is any interaction amongst the particles of a given system, there will be a nonindependent spatial distribution of such particles. Knowing the correlation function which describes this “nonindependence”, one may calculate all the relevant thermodynamic properties as internal energy, pressure and chemical potential.

For a system containing  $N$  particles, this correlation function may be defined

as (MCQUARRIE, 2000):

$$g^{(n)}(\vec{r}_1, \dots, \vec{r}_n) = \frac{V^n N!}{N^n (N-n)!} \frac{\int \dots \int e^{-U_N/k_B T} d\vec{r}_{n+1} \dots d\vec{r}_N}{\int \dots \int e^{-U_N/k_B T} d\vec{r}_1 \dots d\vec{r}_N} \quad (2.1)$$

where  $\vec{r}_n$  is the position of the  $n$ -th particle,  $V$  is the volume of the system,  $U_N$  is the interparticle potential,  $k_B$  is the Boltzmann constant and  $T$  is the absolute temperature of the system.

Obviously, for the pair distribution function,  $g^{(2)}(\vec{r}_1, \vec{r}_2)$ , one has:

$$g^{(2)}(\vec{r}_1, \vec{r}_2) = \frac{V^2 N(N-1)}{N^2} \frac{\int \dots \int e^{-U_N/k_B T} d\vec{r}_3 \dots d\vec{r}_N}{\int \dots \int e^{-U_N/k_B T} d\vec{r}_1 \dots d\vec{r}_N} \quad (2.2)$$

In a fluid of spherically symmetric particles, the pair distribution function depends only upon the relative distance  $r$  between particles 1 and 2. Therefore, this correlation function is mostly expressed as  $g(r)$ , the radial distribution function. If  $g(r) = 1$  for the entire range of  $r$ , no correlation amongst the particles in the system is found, as for an ideal gas.

Generically, using such function, one may write the pressure,  $p$ , and the chemical potential,  $\mu$ , as:

$$p = \rho k_B T - \frac{2\pi\rho^2}{3} \int_0^{+\infty} \frac{du(r)}{dr} g(r) r^3 dr \quad (2.3)$$

$$\mu = k_B T \ln \rho \Lambda^3 + 4\pi\rho \int_0^1 \int_0^{+\infty} u(r) g(r; \xi) r^3 dr d\xi \quad (2.4)$$

where  $\rho$  is the number density,  $u(r)$  is the interparticle two-body potential,  $\Lambda$  is the de Broglie thermal wavelength and  $\xi$  is a coupling parameter.

Figure 1 shows the typical behavior of the radial distribution function for three states of matter: gas, liquid and a face centered cubic crystal. In a gas there is some correlation near the central particle, but at long ranges, there is practically no correlation. In a crystal, the correlations are expressed as a series of

Dirac delta functions on the equilibrium sites of the particles. This fully ordered behavior is what may be seen in a diffractogram. The liquid is by far the most complicated. There is some order, however, not as well defined as in a crystal.

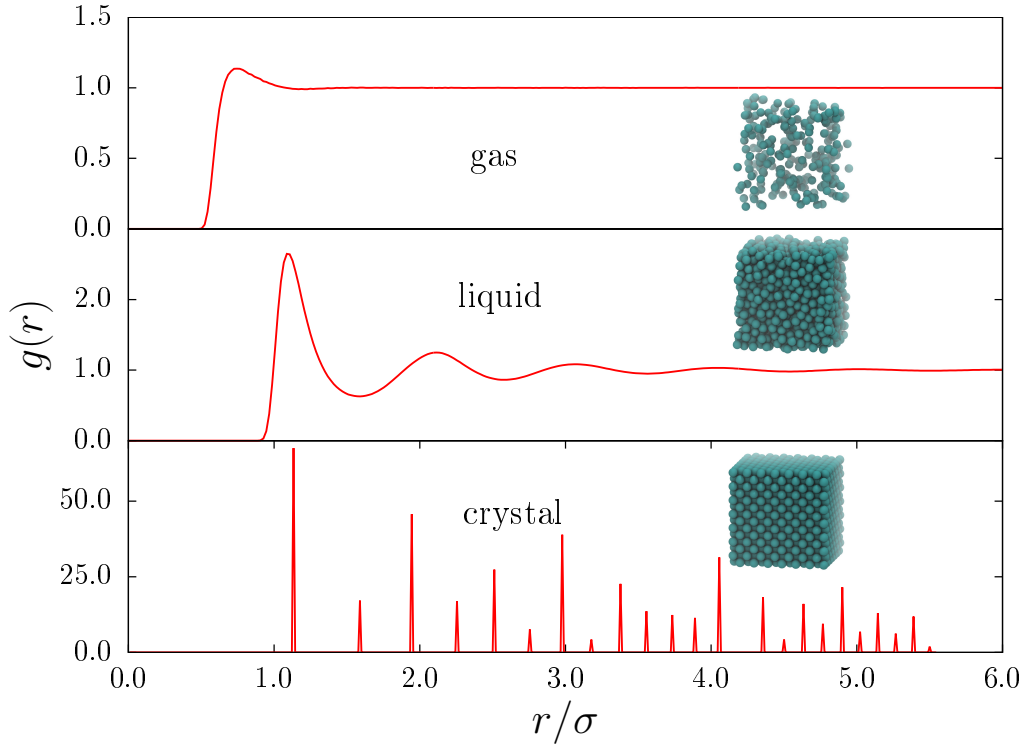


Figure 1: Radial distribution function as a function of relative distance for three states of matter: gas, liquid and crystal. These states are represented within the plot by snapshots of Monte Carlo simulations.

Alternatively, there is another, and equivalent, way to define the radial distribution function in terms of the local particle density (HANSEN; MCDONALD, 1986). The density at a point  $\vec{r}$  is given by:

$$\rho(\vec{r}) = \sum_{j=1}^N \delta(\vec{r} - \vec{r}_j) \quad (2.5)$$

where  $\delta(\vec{r})$  is the Dirac delta function.

Evidently, in a homogeneous system, the ensemble average density at  $\vec{r}$  equals the macroscopic density of the system. This is the very definition of a homogeneous system, *id est*, every point in the system will have the same density and,

therefore, this density is the very density of the system.

In the classical limit, the van Hove space-time correlation function (MCQUARRIE, 2000), which is the density-density autocorrelation function (HANSEN; MCDONALD, 1986), may be written as:

$$G(\vec{r}, t) = \frac{1}{N} \left\langle \sum_{j=1}^N \sum_{k=1}^N \delta(\vec{r} + \vec{r}_j(0) - \vec{r}_k(t)) \right\rangle \quad (2.6)$$

where  $t$  is time.

For  $t = 0$ , one has the static density-density autocorrelation function:

$$G(\vec{r}, 0) = \frac{1}{N} \int \langle \rho(\vec{r}' + \vec{r}) \cdot \rho(\vec{r}') \rangle d\vec{r}' \quad (2.7)$$

Replacing Equation (2.5) on Equation (2.7), one has:

$$G(\vec{r}, 0) = \frac{1}{N} \left\langle \int \sum_{j=1}^N \sum_{k=1, k \neq j}^N \delta(\vec{r}' + \vec{r} - \vec{r}_j) \delta(\vec{r}' - \vec{r}_k) d\vec{r}' \right\rangle + \delta(\vec{r}) \quad (2.8)$$

When  $\vec{r}' = \vec{r}_k$ ,  $\delta(\vec{r}' - \vec{r}_k) = 1$ , otherwise,  $\delta(\vec{r}' - \vec{r}_k) = 0$ . Therefore:

$$G(\vec{r}, 0) = \frac{1}{N} \left\langle \sum_{j=1}^N \sum_{k=1, k \neq j}^N \delta(\vec{r} + \vec{r}_k - \vec{r}_j) \right\rangle + \delta(\vec{r}) \quad (2.9)$$

Thus, the radial distribution function can be written as:

$$g(r) = \frac{V}{N^2} \left\langle \sum_{j=1}^N \sum_{k=1, k \neq j}^N \delta(\vec{r} + \vec{r}_k - \vec{r}_j) \right\rangle \quad (2.10)$$

## 2.2 Static structure factor

If one takes the Fourier transform of the density,  $\rho_q$ , expressed in Equation (2.5), one will have (HANSEN; MCDONALD, 1986):

$$\rho_q(\vec{q}) = \int e^{-i\vec{q} \cdot \vec{r}} \rho(\vec{r}) d\vec{r} \quad (2.11)$$

where  $i$  is the imaginary unit.

Substituting Equation (2.5) on Equation (2.11):

$$\rho_q(\vec{q}) = \int \sum_{j=1}^N e^{-i\vec{q}\cdot\vec{r}} \delta(\vec{r} - \vec{r}_j) d\vec{r} \quad (2.12)$$

which may be simplified using the definition of Dirac delta function as:

$$\rho_q(\vec{q}) = \sum_{j=1}^N e^{-i\vec{q}\cdot\vec{r}_j} \quad (2.13)$$

The autocorrelation function of such Fourier transform of the number density is defined as:

$$S(\vec{q}) = \frac{1}{N} \langle \rho_q(\vec{q}) \cdot \rho_q(-\vec{q}) \rangle \quad (2.14)$$

where  $S(\vec{q})$  is also called the static structure factor, which may also be written as:

$$\begin{aligned} S(\vec{q}) &= \frac{1}{N} \left\langle \sum_{j=1}^N \sum_{k=1}^N e^{-i\vec{q}\cdot\vec{r}_j} e^{i\vec{q}\cdot\vec{r}_k} \right\rangle \\ &= 1 + \frac{1}{N} \left\langle \sum_{j=1}^N \sum_{k=1, k \neq j}^N e^{-i\vec{q}\cdot(\vec{r}_j - \vec{r}_k)} \right\rangle \end{aligned} \quad (2.15)$$

Therefore, for a homogeneous fluid, the static structure factor has an intrinsic relation to the pair distribution function:

$$S(\vec{q}) = 1 + \rho \int e^{-i\vec{q}\cdot(\vec{r} - \vec{r}')} g(\vec{r}) d\vec{r} \quad (2.16)$$

And for an isotropic fluid:

$$S(q) = 1 + 4\pi\rho \int g(r) r^2 \frac{\sin qr}{qr} dr \quad (2.17)$$

This means that the static structure factor contains the same amount of information as the radial distribution function, however not in the real space, but in the Fourier space. The choice of which one is better to be used is just a matter of convenience. For the radiation scattering problem, the static structure factor happens to be much more convenient, because it naturally emerges from

the scattering theory as we shall present in the next chapter. One may recover the radial distribution function by the inverse Fourier transform of the static structure factor. Also, at the limit of low  $\vec{q}$ , the static structure factor gives the ratio between the isothermal compressibilities of the fluid and of the ideal gas.

A brief introduction to the Ornstein-Zernike equation is given in Section 5.3. Loosely, this equation relates the total correlation function,  $h(r)$ , which is the radial distribution function shifted by 1, and the direct correlation function,  $c(r)$ . From the application of the Fourier transform on this equation, it raises a relation between the static structure factor and the Fourier transform of the direct correlation function,  $C(q)$ . Most of the research on this field was made assuming approximations for the  $c(r)$  in terms of the interaction potential to evaluate the static structure factor. Such approximations are called closure relations. Several closure relations are reported in the open literature. For some simple interaction potentials, some approximations are analytically solvable. Here, we shall present two of the most common solutions applied to biological systems: Percus-Yevick Approximation for hard spheres and Mean Spherical Approximation for macroions in solution.

## 2.3 Percus-Yevick Approximation for hard spheres

The Percus-Yevick Approximation is given by (PERCUS; YEVICK, 1958):

$$c(r) = \begin{cases} g(r) [1 - e^{U(r)/k_B T}], & \text{if } r \leq \sigma \\ 0, & \text{if } r > \sigma \end{cases} \quad (2.18)$$

For a hard sphere fluid, the interaction potential is:

$$U(r) = \begin{cases} \infty, & \text{if } r \leq \sigma \\ 0, & \text{if } r > \sigma \end{cases} \quad (2.19)$$

The analytical solution for a hard sphere fluid within the Percus-Yevick approximation was obtained independently, and almost simultaneously, by Wertheim (WERTHEIM, 1963) and Thiele (THIELE, 1963). The static structure factor had its functional form presented by Ashcroft and Lekner (ASHCROFT; LEKNER, 1966):

$$C(k) = -\frac{24\eta}{\rho k^6} \left\{ \alpha k^3 [\sin k - k \cos k] + \beta k^2 [2k \sin k - (k^2 - 2) \cos k - 2] \right. \\ \left. + \gamma [(4k^3 - 24k) \sin k - (k^4 - 12k^2 + 24) \cos k + 24] \right\} \quad (2.20)$$

where  $\eta = \rho\pi\sigma^3/6$  is the packing fraction,  $k = q\sigma$ ,  $\sigma$  is the sphere diameter, and:

$$\alpha = \frac{(1 + 2\eta)^2}{(1 - \eta)^4} \quad (2.21)$$

$$\beta = -\frac{\eta (18 + 20\eta - 12\eta^2 + \eta^4)^2}{3 (1 - \eta)^4} \quad (2.22)$$

$$\gamma = \frac{\eta\alpha}{2} \quad (2.23)$$

Figure 2 shows the remarkable agreement between the results of such approximation and the results from Monte Carlo simulations of a hard sphere fluid. The technique of Monte Carlo as well as molecular dynamics will be explained latter, in Chapter 4.

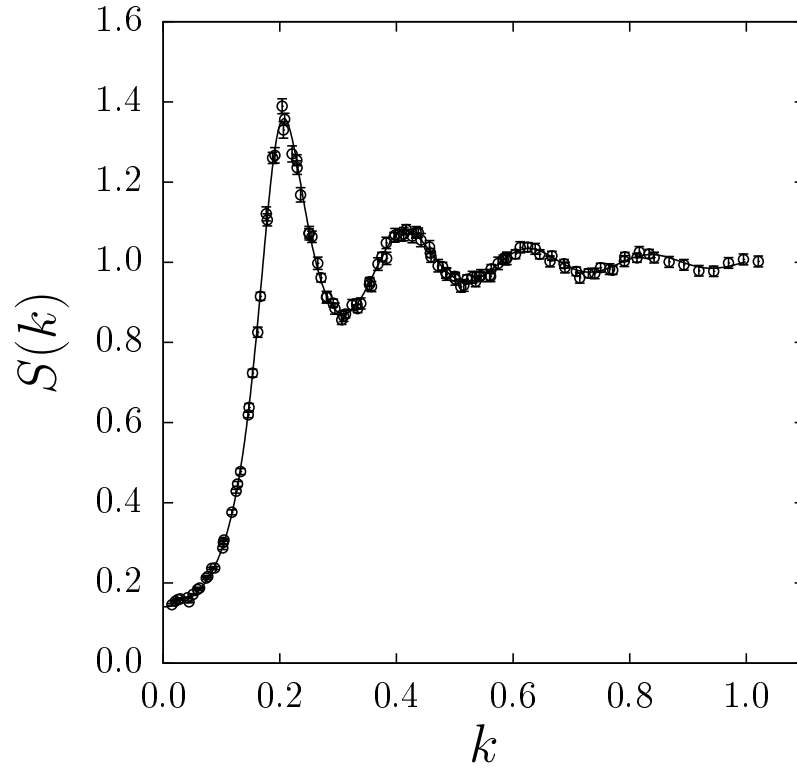


Figure 2: Static structure factor for hard sphere fluid with  $\eta = 0.25$ . Continuous line, Percus-Yevick Approximation. Open circles, Monte Carlo simulations.

## 2.4 Mean Spherical Approximation for macroions in solution

The Mean Spherical Approximation is given by (LEBOWITZ; PERCUS, 1966):

$$c(r) = \begin{cases} g(r) [1 - e^{U(r)/k_B T}], & \text{if } r \leq \sigma \\ -U(r)/k_B T, & \text{if } r > \sigma \end{cases} \quad (2.24)$$

where  $U(r)$  is the interparticle interaction potential,  $k_B$  is the Boltzmann constant and  $T$  is the absolute temperature.



The interaction potential for macroions in solution is given by:

$$U(r) = \begin{cases} +\infty, & \text{if } r \leq \sigma \\ \frac{z_m^2 e^2}{\pi \epsilon_0 \epsilon (2 + \kappa \sigma)^2 r} \exp[-\kappa(r - \sigma)], & \text{if } r > \sigma \end{cases} \quad (2.25)$$

where  $\sigma$  is the particle diameter,  $z_m$  is the electronic charge of the macroions in units of the elementary charge  $e$ ,  $\kappa$  is the inverse of Debye's length,  $\epsilon_0$  is the vacuum permittivity and  $\epsilon$  is the medium dielectric constant.

This potential may be rewritten in a dimensionless way:

$$\frac{U(x)}{k_B T} = \begin{cases} +\infty, & \text{if } x \leq 1 \\ \gamma \frac{\exp(-Kx)}{x} & \text{if } x > 1 \end{cases} \quad (2.26)$$

where  $x = r/\sigma$ ,  $\gamma = z_m^2 e^2 \sigma \exp(\kappa \sigma) / \pi \epsilon_0 \epsilon (2 + \kappa \sigma)^2$  and  $K = \kappa \sigma$ .

There is something subtle here: for this sort of potential, one is considering the solution as a continuous medium and the macroions as discrete particles; this certainly places such approach within the McMillan-Mayer framework (MCMILLAN; MAYER, 1945).

The solution for this particular interaction potential through the Mean Spherical Approximation is due to Hayter and Penfold (HAYTER; PENFOLD, 1981). The resulting functional form of the static structure factor is a huge set of equations, which is available in the original paper. For the sake of conciseness, we shall skip the tedious presentation of such ensemble of equations and the reader to whom these are necessary is referred to the original work.

Although this model is considered an analytical solution, the algorithm to calculate it is somehow tricky. Since the roots of a fourth-order polynomial function must be found within the calculation, one could use a Newton-Raphson algorithm to find the roots coupled with Ruffini's rule to exclude the already found roots writing the polynomial in Horner form. All real roots which obey the restriction

of  $g(r) = 0$  for  $r < \sigma$  are correct. And to evaluate the radial distribution function, a numerical integration method, such as trapezoidal rule, might be used to calculate the inverse Fourier transform of the static structure factor. An *ad hoc* code to calculate this structure factor was written in C language and is presented on Appendix B.

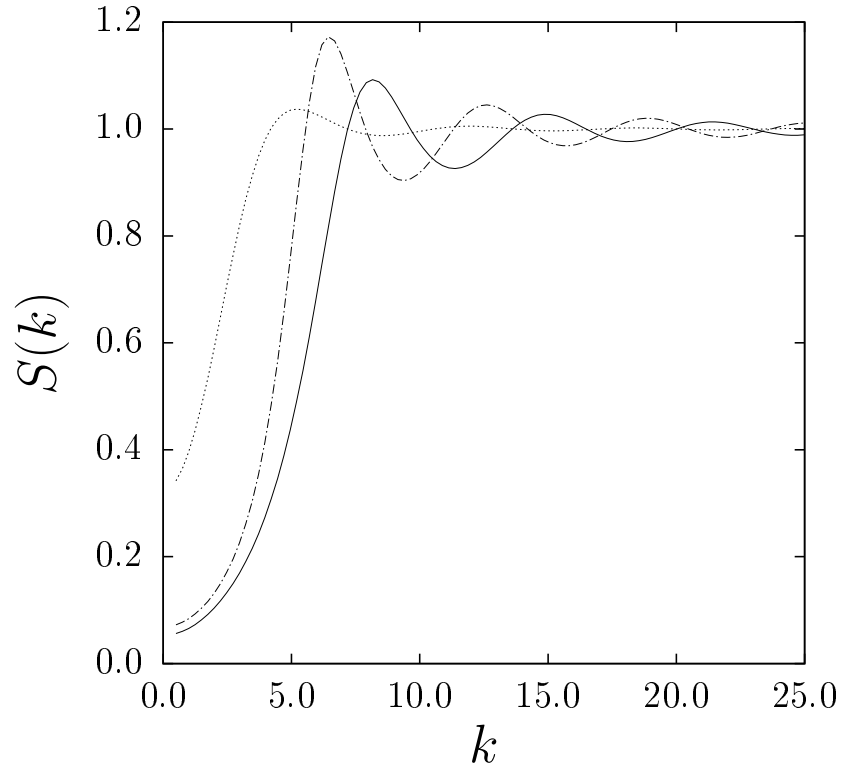


Figure 3: Static structure factor for macroions into solution using Hayter-Penfold equation (HAYTER; PENFOLD, 1981) with  $\kappa = 0.04$  and  $\gamma = 10.0$ . Continuous line,  $\eta = 0.3$ . Dashed-dotted line,  $\eta = 0.2$ . Dotted line,  $\eta = 0.05$ .

Figure 3 presents the static structure factor for macroions into solution using the equation proposed by Hayter and Penfold (HAYTER; PENFOLD, 1981) for three different values of packing fraction, ( $\eta$ ). By decreasing the packing fraction and keeping the inverse of Debye's length constant, the structure factor limit at low  $q$  increases, and therefore, the isothermal compressibility increases and also the attraction between two macroions is favored. This is not obvious and will be discussed later in Chapter 5. Nevertheless, it is not only the structure factor limit at low  $q$  that changes, but also the position of the peaks. Thus, the whole

structure changes and not only the compressibility.

Figure 4 presents the same structure factor, but for three different values of  $\gamma$ , which is proportional to the square charge of the macroion. As  $\gamma$  increases, the isothermal compressibility decreases, indicating that the repulsion among the macroions becomes stronger. Nevertheless, the positions of the peaks remain the same even though the height of the peaks changes.

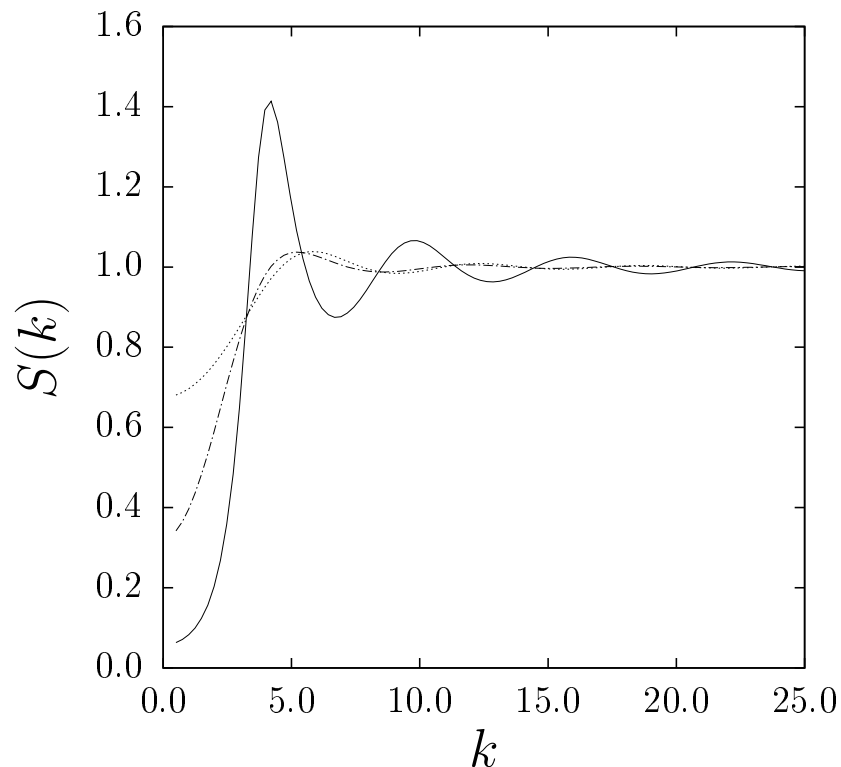


Figure 4: Static structure factor for macroions into solution using Hayter-Penfold equation (HAYTER; PENFOLD, 1981) with  $\eta = 0.05$  and  $\kappa = 0.04$ . Continuous line,  $\gamma = 100.0$ . Dashed-dotted line,  $\gamma = 10.0$ . Dotted line,  $\gamma = 0.0$ .

Figure 5 presents the variation of the structure factor of Hayter and Penfold (HAYTER; PENFOLD, 1981) as the inverse of the Debye's length,  $\kappa$ , changes. The increment of  $\kappa$  increases the isothermal compressibility, which indicates that the larger the value of  $\kappa$ , the more attractive the system becomes. This is in accordance to the linearized Poisson-Boltzmann equation for which the lower the Debye's length, the more screened is the Coulomb potential.

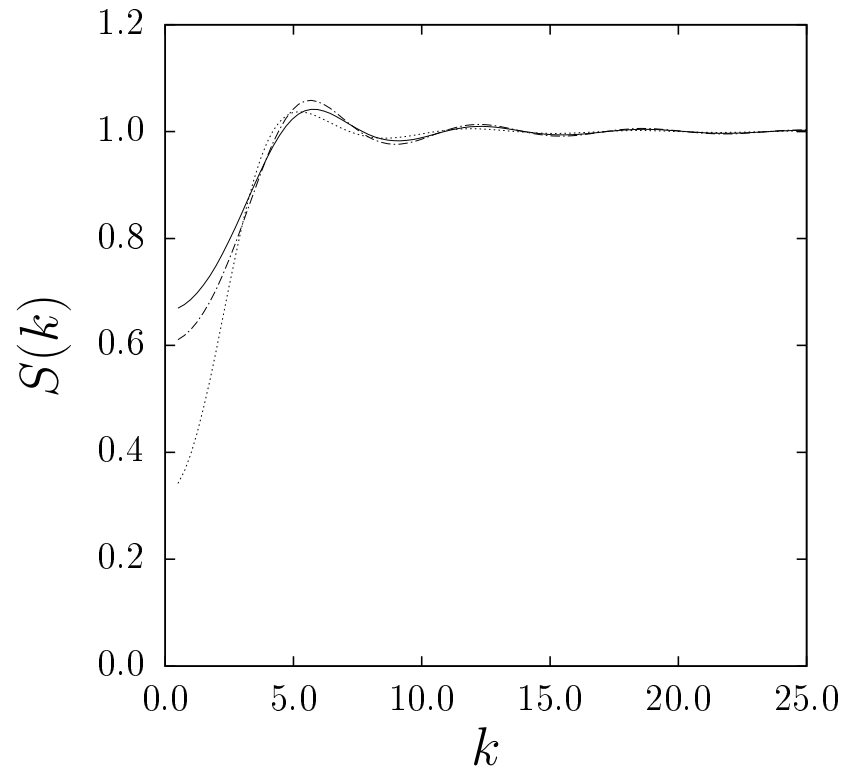


Figure 5: Static structure factor for macroions into solution using Hayter-Penfold equation (HAYTER; PENFOLD, 1981) with  $\eta = 0.05$  and  $\gamma = 10.0$ . Continuous line,  $\kappa = 0.20$ . Dashed-dotted line,  $\kappa = 0.10$ . Dotted line,  $\kappa = 0.04$ .

Even though in this thesis the Hayter-Penfold equation (HAYTER; PENFOLD, 1981) is not explicitly applied, it serves as a basis for comparison and is also as a good way to learn about macroion interaction in solution.

### 3 BASIC RADIATION SCATTERING THEORY

The purpose of this chapter is to provide an introduction to scattering theory to facilitate the appreciation of the small-angle X-ray scattering experiments that have been conducted during this Ph.D. work. The objective, however, is not to give an exhaustive explanation of this subject. Since only the description of X-ray scattering is the main concern, we shall neglect the Compton effect and derive a theory specific to the elastic scatter phenomenon. Even though the elastic scattering violates the momentum conservation, for small-angles this can be easily overcome since the Compton effect depends on the scattering angle, and for small angles, this effect can be neglected.

When a system of electric charges is exposed to electromagnetic waves, these electric charges are set in motion (LANDAU; LIFSHITZ, 1975). Firstly, we shall derive a proper equation of motion for a simple system composed by an oscillating electron with natural frequency,  $\omega_0$ , coupled to a stationary positively charged nucleus, and subjected to a linearly polarized electromagnetic wave. Such equation of motion may be the same as for a driven damped harmonic oscillator. Secondly, we shall evaluate the scattering intensity associated with this interaction. At this point, we shall observe the differences between Rayleigh and Thompson scatterings. Thirdly, we shall extend the framework to include scattering with many atoms with several electrons.

### 3.1 Driven damped harmonic oscillator

Consider an incident monochromatic plane electromagnetic wave linearly polarized with electric field,  $\vec{E}$ , as shown in Figure 6, given by:

$$\vec{E} = E_0 e^{-i\omega t} \vec{x} \quad (3.1)$$

where  $E_0$  is the amplitude,  $i$  is the imaginary unit,  $\omega$  is the frequency,  $t$  is time and  $\vec{x}$  is the unit vector that defines  $x$ -direction.

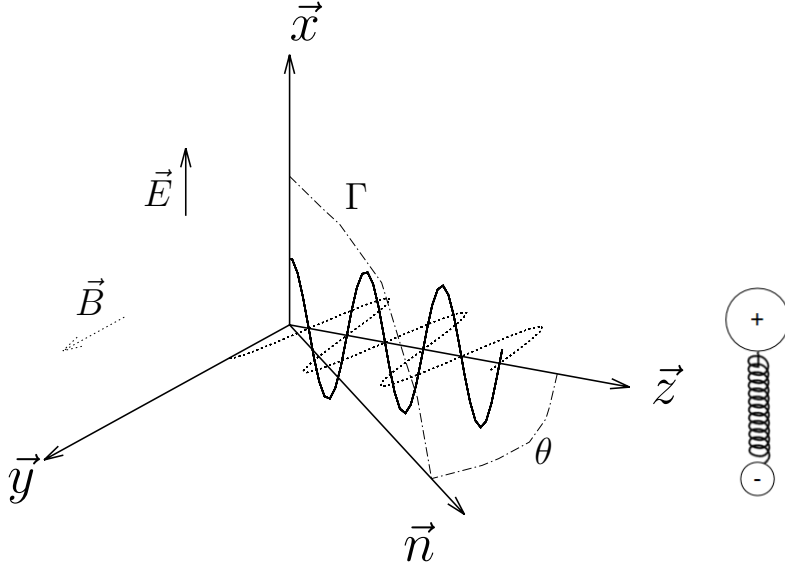


Figure 6: Representation of the oscillating electron-positive nucleus system. The axes represent the coordinate system. The continuous line represents electric field and the dotted line represents the magnetic field.  $\vec{n}$  is the unit vector in the same direction as the scattered wave.  $\Gamma$  is the angle between  $\vec{x}$  and  $\vec{n}$ ; and  $\theta$  is the angle between  $\vec{z}$  and  $\vec{n}$ .

The equation of motion for such driven damped harmonic oscillator is given by the following second-order ordinary differential equation:

$$\frac{d^2 \vec{r}}{dt^2} + 2\zeta\omega_0 \frac{d\vec{r}}{dt} + \omega_0^2 \vec{r} = \frac{e}{m} E_0 e^{-i\omega t} \vec{x} \quad (3.2)$$

where  $\vec{r}$  is the electron position,  $\zeta$  is the damping ratio,  $\omega_0$  is the undamped natural frequency of the electron (defined as the square root of the ratio between the spring elastic constant and the electron mass),  $e$  is the elementary charge and  $m$  is the mass of the electron.

The steady-state solution of this differential equation can be written as:

$$\vec{r}(t) = -\frac{e}{m} \left( \frac{1}{\omega^2 - \omega_0^2 + i2\zeta\omega_0\omega} \right) E_0 e^{-i\omega t} \vec{x} \quad (3.3)$$

Therefore, the acceleration of such electron would be given by:

$$\frac{d^2 \vec{r}}{dt^2} = \frac{e}{m} \left( \frac{\omega^2}{\omega^2 - \omega_0^2 + i2\zeta\omega_0\omega} \right) E_0 e^{-i\omega t} \vec{x} \quad (3.4)$$

## 3.2 Scattering intensity

To obtain the scattered intensity by a single electron exposed to an electromagnetic wave with an electric field given by Equation (3.1), one ought to consider that such time-averaged intensity,  $\langle I \rangle$ , is equal to the time-averaged Poynting vector magnitude,  $\langle \|\vec{S}\| \rangle$  (KOCH; VACHETTE; SVERGUN, 2003).

There is more than one way to define the Poynting vector (KINSLER; FAVARO; MCCALL, 2009), but here the definition given by Landau and Lifschitz (LANDAU; LIFSHITZ, 1969) will be assumed; however, instead of Gaussian units, SI units will be adopted. Thus, the Poynting vector may be defined as:

$$\vec{S} = \frac{1}{\mu_0} \vec{E} \times \vec{B} \quad (3.5)$$

where  $\vec{B}$  is the magnetic field and  $\mu_0$  is the vacuum permeability.

Knowing that  $\vec{E} \cdot \vec{B} = 0$  (*id est*, the vectors are orthogonal) and that  $\|\vec{E}\| = c\|\vec{B}\|$ , where  $c$  is the speed of light, and from the definition of the Poynting vector,

the cross product may be rewritten as:

$$\vec{S} = \frac{c}{\mu_0} \|\vec{B}\|^2 \vec{z} \quad (3.6)$$

where  $\vec{z}$  is the unit vector that defines  $z$ -direction.

Moreover, the magnetic field may be related to the dipole moment which is  $e\vec{r}$ : (LANDAU; LIFSHITZ, 1975):

$$\vec{B} = \frac{\mu_0}{4\pi r c} \left[ \frac{d^2(e\vec{r})}{dt^2} \times \vec{n} \right] \quad (3.7)$$

where  $\vec{n}$  is the unit vector that defines the direction of the scattered wave.

Substituting Equation (3.7) on Equation (3.6), one has:

$$\vec{S} = \left( \frac{1}{4\pi} \right)^2 \frac{e^2}{\epsilon_0 c^3 r^2} \left( \frac{d^2 \vec{r}}{dt^2} \times \vec{n} \right)^2 \vec{z} \quad (3.8)$$

where  $\epsilon_0$  is the vacuum permittivity and  $c^2 = 1/\mu_0\epsilon_0$ .

Rewriting the cross product considering the angle,  $\Gamma$ , between the incident electric field and the scattered wave:

$$\vec{S} = \left( \frac{1}{4\pi} \right)^2 \frac{e^2}{\epsilon_0 c^3 r^2} \left\| \frac{d^2 \vec{r}}{dt^2} \right\|^2 \sin^2 \Gamma \vec{z} \quad (3.9)$$

Considering the steady-state solution for a driven damped harmonic oscillator expressed by Equation (3.4), the Poynting vector will be written as:

$$\vec{S} = \left( \frac{1}{4\pi} \right)^2 \frac{e^4 E_0^2}{m^2 \epsilon_0 c^3 r^2} \left[ \frac{\omega^4}{(\omega^2 - \omega_0^2)^2 + 4\zeta^2 \omega_0^2 \omega^2} \right] \sin^2 \Gamma \vec{z} \quad (3.10)$$

Since (LANDAU; LIFSHITZ, 1969):

$$\langle \sin^2 \Gamma \rangle = \left( \frac{1 + \cos^2 \theta}{2} \right) \quad (3.11)$$



The time-averaged scattering intensity is given by:

$$\langle I \rangle = \left( \frac{1}{4\pi} \right)^2 \frac{e^4 E_0^2}{m^2 \epsilon_0 c^3 r^2} \left[ \frac{\omega^4}{(\omega^2 - \omega_0^2)^2 + 4\zeta^2 \omega_0^2 \omega^2} \right] \left( \frac{1 + \cos^2 \theta}{2} \right) \quad (3.12)$$

Figure 7 presents how the intensity given by Equation (3.12) depends on the relative frequency,  $\omega/\omega_0$ . When the relative frequency is low, *id est*,  $\omega \ll \omega_0$ , one has the so-called Rayleigh scattering, which is responsible for the blue color of the sky. In such scattering, the intensity depends on the frequency to the power of four. When, however, the relative frequency is high, *id est*,  $\omega \gg \omega_0$ , one has the so-called Thomson scattering. In this case, the scattering intensity is constant regardless of the frequency. Because the frequency of X-ray is high, the scattering will be in the region of Thomson scattering.

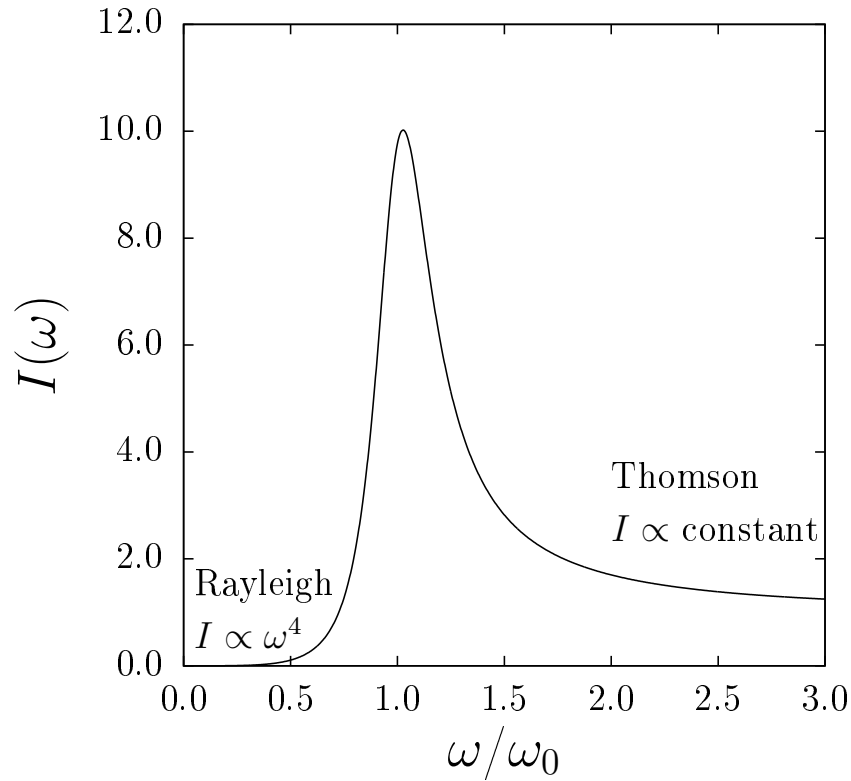


Figure 7: Scattering intensity as a function of the relative frequency.

### 3.3 Scattering of a system with many atoms

By judicious choice, we shall skip several lines of derivation and present the final results, which are the most important aspect for this thesis. The full derivations are presented in classical books (GLATTER; KRATKY, 1982; FEIGIN; SVERGUN, 1987).

Since the Compton effect is neglected, all scattered waves are coherent and, therefore, the scattering amplitudes are added and the intensity is given by the absolute square of the amplitudes. Consider  $N$  identical atoms with scattering amplitude  $F(\vec{q})$ . The intensity will be given by (GLATTER; KRATKY, 1982):

$$I(\vec{q}) = \sum_{j=1}^N \sum_{k=1}^N F(\vec{q}) F^*(\vec{q}) \exp[i\vec{q} \cdot (\vec{r}_j - \vec{r}_k)] \quad (3.13)$$

where  $\vec{q}$  is the wavevector, which is defined as the difference between the scattered and the incident vectors and  $|\vec{q}| = (4\pi/\lambda) \sin \theta$ , being  $\lambda$  the radiation wavelength and  $2\theta$  the scattering angle;  $i$  is the imaginary unit,  $\vec{r}_j$  is the position of the atom  $j$  and  $F^*(\vec{q})$  is the complex conjugate of  $F(\vec{q})$ .

For the sake of simplicity, we shall write the norm of  $\vec{q}$  simply as  $q$ , thus, defining the form factor,  $P(q)$ , as:

$$P(q) = |F(q) F^*(q)|^2 \quad (3.14)$$

and the static structure factor,  $S(q)$ , as:

$$S(q) = \frac{1}{N} \left| \sum_{i=1}^N \exp[i\vec{q} \cdot \vec{r}_i] \right|^2 \quad (3.15)$$

One has that Equation (3.13) may be rewritten as:

$$I(q) = NP(q)S(q) \quad (3.16)$$

This ablation of the scattering intensity allows some important physical inter-

pretations. The form factor is related to the geometry of the scattering particles and the structure factor is related to the interaction potential among the particles. For highly diluted solutions, the particles are so far from each other that one may consider there is no interaction among them, and therefore,  $S(q) = 1$ .

### 3.4 Form factor expressions

Even though the main concern of this thesis is related to the study of the structure factor, it seems quite important to show, at least, some simple analytical models to describe the form factor.

The first one is due to Lord Rayleigh, who evaluated the scattering amplitude of a homogeneous sphere as (PEDERSEN, 1997):

$$F(q, R) = 3 \left[ \frac{\sin(qR) - qR \cos(qR)}{(qR)^3} \right] \quad (3.17)$$

where  $R$  is the radius of the sphere.

According to Equation (3.14), the form factor is given by:

$$P(q, R) = 9 \left[ \frac{\sin(qR) - qR \cos(qR)}{(qR)^3} \right]^2 \quad (3.18)$$

Figure 8 presents a tridimensional plot of a homogeneous sphere scattering following Equation (3.18) for the scattering intensity.

The second one is due to André Guinier and is derived for an ellipsoid of revolution with semi axes  $R$ ,  $R$  and  $\epsilon R$  (PEDERSEN, 1997):

$$P(q, R, \epsilon) = \int_0^{\pi/2} F[q, r(R, \epsilon, \alpha)] \sin \alpha d\alpha \quad (3.19)$$

where  $F[q, r(R, \epsilon, \alpha)]$  is given by Equation (3.17) coupled to:

$$r(R, \epsilon, \alpha) = R \sqrt{(\sin^2 \alpha + \epsilon^2 \cos^2 \alpha)} \quad (3.20)$$

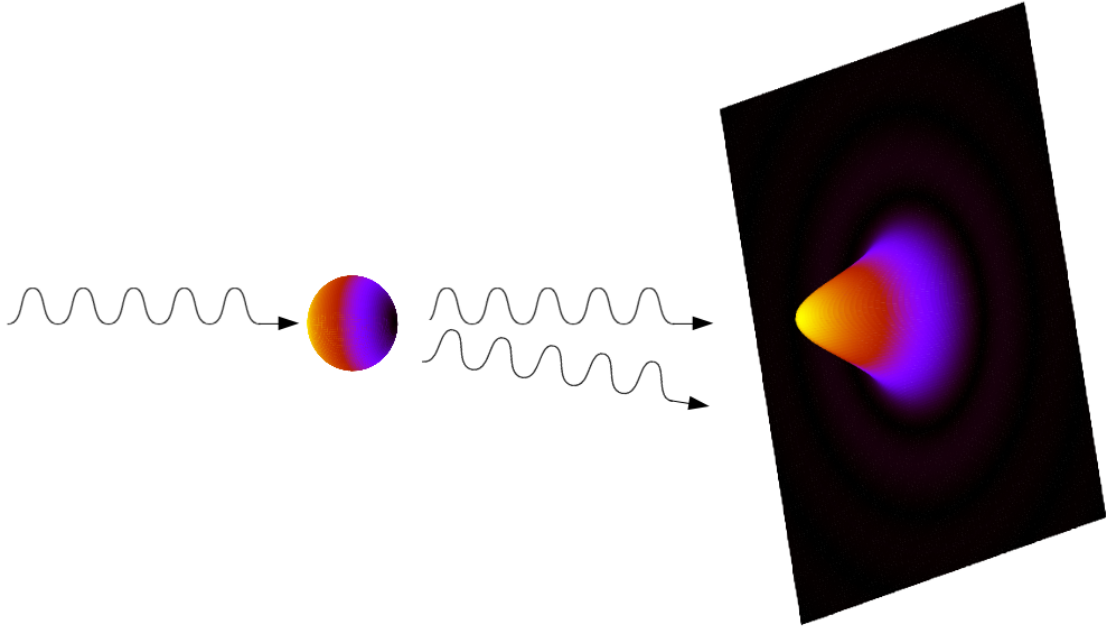


Figure 8: Homogeneous sphere X-ray scattering scheme.

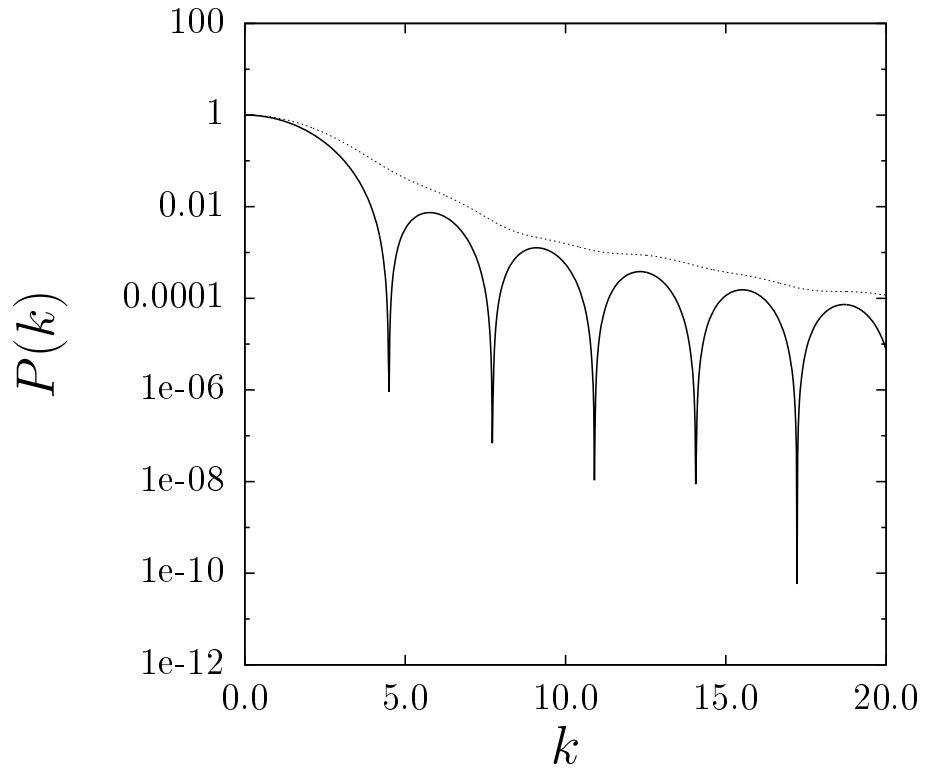


Figure 9: Form factor as a function of  $k = qR$ . Continuous line, Equation (3.18) for a homogeneous sphere. Dotted line, Equation (3.19) for an oblate ellipsoid of revolution with  $\epsilon = 0.4$

Figure 9 shows the comparison between the form factor of a homogeneous sphere and an ellipsoid of revolution. One may observe that the geometry of the particle alters completely the scattering intensity. That is the reason why scattering experiments are so helpful to resolve the structure of complex molecules.

## 4 INTRODUCTION TO MOLECULAR SIMULATION

Statistical mechanics, either on its classical (GIBBS, 1902) or quantum derivation (SCHRÖDINGER, 1989), is based on the Hamiltonian mechanics and not on the most traditional Newtonian framework. Molecular simulations also make use of such approach. In classical terms, the Hamiltonian ( $H$ ) is given by the summation of the kinetic ( $K$ ) and the potential ( $V$ ) energies, and is a function of position ( $\vec{r}$ ) and momentum ( $\vec{p}$ ) coordinates of the particles.

$$H(\vec{r}, \vec{p}) = K(\vec{p}) + V(\vec{r}) \quad (4.1)$$

The kinetic energy is given by the following common expression:

$$K(\vec{p}) = \sum_{j=1}^N \frac{|\vec{p}_j|^2}{2m_j} \quad (4.2)$$

where  $\vec{p}_j$  is the momentum of particle  $j$  and  $m_j$  is the mass of particle  $j$ .

The very first approximation, which is usually assumed (but most of the time remains implicit on classical molecular simulation studies), is the Born-Oppenheimer approximation. In such approximation, one considers the decoupling between nuclear and electronic dynamics due to the nuclei being much heavier than the electrons and, therefore, much slower. In quantum terms, there will be two distinct Schrödinger equations generating two wavefunctions whose

convolution represents the total wavefunction of the system.

$$\Psi_{\text{total}} = \Psi_{\text{nucleus}} \Psi_{\text{electron}} \quad (4.3)$$

This implies that the total Hamiltonian operator will be given by the sum of the nuclear and the electronic Hamiltonian operators (MCQUARRIE, 2000):

$$\hat{\mathcal{H}}_{\text{total}} = \hat{\mathcal{H}}_{\text{nucleus}} + \hat{\mathcal{H}}_{\text{electron}} \quad (4.4)$$

For classical molecular simulations, only the nuclear Hamiltonian is explicitly taken into account. Therefore, the force field (*id est*, the potential) takes into account only the interaction among the nuclei.

A second approximation, which is largely considered, is the pair-wise approximation. According to it, the potential energy,  $V$ , might be described as the sum of two-body potentials,  $V(r_{ij})$ , (HILL, 1987):

$$V(\vec{r}) = \sum_{1 \leq i < j \leq N}^N V(r_{ij}) \quad (4.5)$$

This force field (the interaction potential is usually called force field), however, must encompass intermolecular as well as intramolecular potentials. There are many functional forms for force fields. If the polarization phenomenon is neglected, one has the Lennard-Jones potential, which basically comprises a contact repulsive and a dipolar attractive terms, and a Coulomb potential to describe the electrostatic interactions as the most common form. Such potential could be written, considering only two-body interactions, as (ALLEN; TILDESLEY, 1987):

$$V_{ij}^{\text{inter}}(r_{ij}) = 4\varepsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}} \quad (4.6)$$

where  $V_{ij}$  is the interaction potential between two particles  $i$  and  $j$ ,  $r_{ij}$  is the distance between these two particles,  $\varepsilon_{ij}$  is the energy well depth,  $\sigma_{ij}$  is the distance between the particles  $i$  and  $j$  for which the Lennard-Jones potential is

null,  $q_i$  is the electric charge of the particle and  $\varepsilon_0$  is the vacuum permittivity.

The parameters  $\varepsilon$  and  $\sigma$  are given by each particular nucleus, and the cross terms are evaluated by combining rules, being the Lorentz-Berthelot the most used (ALLEN; TILDESLEY, 1987):

$$\sigma_{ij} = \frac{\sigma_{ii} + \sigma_{jj}}{2} \quad (4.7)$$

$$\varepsilon_{ij} = \sqrt{\varepsilon_{ii}\varepsilon_{jj}} \quad (4.8)$$

For the intramolecular potential, it is necessary to consider the bond potential between two adjacent nuclei, the angular potential among three adjacent nuclei and also the dihedral potential among four adjacent nuclei. For each one of these types of potential there is at least one model.

Beyond the proposition of theories and experiments, computational simulations have been playing a valuable role in providing essentially precise numerical results for problems on statistical mechanics which otherwise would only have approximated solutions (ALLEN; TILDESLEY, 1987). There are two families of molecular simulation methods: molecular dynamics and Monte Carlo. Whilst the former is based on the integration of classical equations of motion for each system in the ensemble, given an interaction potential between the systems, the latter is a stochastic sampling of coordinates. Short introductions on both methods are presented in the next sections.

## 4.1 Molecular dynamics

In molecular dynamics, given an interaction potential between the particles, their initial space coordinates and respective initial momenta, one may advance in time taking into account the force that each particle exerts on the other at



this simulation step. This means that if it is possible to know the interaction potential between two particles,  $V(\vec{r}_{ij})$ , the force,  $\vec{F}_{ij}$ , is given by

$$\vec{F}_{ij} = -\vec{\nabla}V(\vec{r}_{ij}) \quad (4.9)$$

Following Newton's second law, the acceleration,  $\vec{a}$ , of a particle  $i$  is:

$$\vec{a}_i = \frac{1}{m_i} \sum_{j \neq i} \vec{F}_{ij} \quad (4.10)$$

where  $m_i$  is the mass of particle  $i$ .

Or in terms of position:

$$\frac{d^2 \vec{r}_i}{dt^2} = \frac{1}{m_i} \sum_{j \neq i} \vec{F}_{ij} \quad (4.11)$$

where  $\vec{r}_i$  is the position of particle  $i$ .

Therefore, if one is able to discretize the equation of motion in terms of position, one may evaluate the next position of a particle  $i$ . There are several traditional algorithms employed to evaluate this time evolution such as Verlet, Velocity Verlet and Leap Frog (ALLEN; TILDESLEY, 1987). One important thing in the integration of the equations of motion is the energy conservation. That is precisely the reason why some algorithms for solving differential equations may not be useful for molecular dynamics. Moreover, Newton's equations of motion are time reversible, and hence, the integrated solution must also be so (FRENKEL; SMIT, 2002).

#### 4.1.1 Verlet algorithm

The Verlet algorithm is based on the Taylor expansion about  $\vec{r}(t)$  truncated after the acceleration term (ALLEN; TILDESLEY, 1987):

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \vec{v}(t)\Delta t + \frac{1}{2}\vec{a}(t)(\Delta t)^2 \quad (4.12)$$

$$\vec{r}(t - \Delta t) = \vec{r}(t) - \vec{v}(t)\Delta t + \frac{1}{2}\vec{a}(t)(\Delta t)^2 \quad (4.13)$$

If one uses only the first equation, one obtains Euler algorithm, which is not time reversible and fails to keep energy conservation (FRENKEL; SMIT, 2002). Subtracting both equations, one arrives at Verlet algorithm for position:

$$\vec{r}(t + \Delta t) = 2\vec{r}(t) - \vec{r}(t - \Delta t) + \vec{a}(t)(\Delta t)^2 \quad (4.14)$$

With such equation, one is able to calculate the whole trajectory of the system. If one is interested in the calculation of the kinetic energy, one must calculate the velocities, which are given by:

$$\vec{v}(t) = \frac{\vec{r}(t + \Delta t) - \vec{r}(t - \Delta t)}{2\Delta t} \quad (4.15)$$

One may see that the symmetry of equations to  $\vec{r}(t + \Delta t)$  and  $\vec{r}(t - \Delta t)$  makes the solution of Verlet algorithm essentially time reversible (excluding numerical errors).

### 4.1.2 Leap Frog algorithm

To derive the Leap Frog algorithm, one may start with the velocities at half-integer time (FRENKEL; SMIT, 2002):

$$\vec{v}(t + \Delta t/2) = \frac{\vec{r}(t + \Delta t) - \vec{r}(t)}{\Delta t} \quad (4.16)$$

$$\vec{v}(t - \Delta t/2) = \frac{\vec{r}(t) - \vec{r}(t - \Delta t)}{\Delta t} \quad (4.17)$$

The first equation gives the evolution of position:

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \vec{v}(t + \Delta t/2)\Delta t \quad (4.18)$$

The velocities are given by:

$$\vec{v}(t + \Delta t/2) = \vec{v}(t - \Delta t/2) + \vec{a}(t)\Delta t \quad (4.19)$$

In this thesis, all molecular dynamics simulations were carried out using the Leap Frog algorithm, because this is the natural integrator in the package used in this thesis.

If there is no other restraint, the number of particles, the box volume and the system energy are kept constant given a conservative trajectory. This sort of simulations entails the so-called microcanonical ensemble. To simulate other ensembles, one must impose other constraints such as thermostats and barostats.

## 4.2 Monte Carlo simulations

### 4.2.1 Markov chains

To define a Markov chain, let us first recall Kolmogorov's axioms of probability (KOLMOGOROV, 1956). Let  $S$  be a collection of elements  $A_1, \dots, A_n$ , which are random events, then the probability is a function that must obey the following rules:

- i.  $P(A_k) \geq 0$  for all values of  $k$ .
- ii. If  $A_1, \dots, A_n$  are disjoint, *id est*, mutually exclusive, then:

$$P\left(\bigcup_{k=1}^n A_k\right) = \sum_{k=1}^n P(A_k)$$

- iii.  $P(S) = 1$

In such framework, one may define the conditional probability of event  $A_j$  under the condition  $A_k$  as (ROZANOV, 1977):

$$P(A_j|A_k) = \frac{P(A_j \cap A_k)}{P(A_k)} \quad (4.20)$$

Therefore, for a sequence of events  $A_1, \dots, A_n$ , one has:

$$P\left(\bigcap_{k=1}^n A_k\right) = P(A_1) \prod_{k=1}^{n-1} P\left(A_{k+1} \mid \bigcap_{j=1}^k A_j\right) \quad (4.21)$$

Let us now define a stochastic variable  $x_t$ , which assumes only integer values and  $t = 0, 1, 2, 3 \dots$  is time. A stochastic process is completely defined at instant  $l$  by the joint probability distribution (TOMÉ; OLIVEIRA, 2001):

$$P_l\left(\bigcap_{t=0}^l x_t = n_t\right) \quad (4.22)$$

where  $x_t$  assumes the value  $n_t$  at instant  $t$ .

Then, one may define a Markovian process as a stochastic process for which:

$$P_{l+1}\left(x_{l+1} = n_{l+1} \mid \bigcap_{t=0}^l x_t = n_t\right) = P_{l+1}(x_{l+1} = n_{l+1} \mid x_l = n_l) \quad (4.23)$$

In other terms, a Markovian process is such process in which the conditional probability of  $x_t$  be a certain value depends exclusively on the immediately previous value.

Substituting the definition of a Markov chain given by Equation (4.23) on the chain rule expressed by Equation (4.21), one has:

$$P_l\left(\bigcap_{t=0}^l x_t = n_t\right) = P(x_0 = n_0) \prod_{t=0}^{l-1} P(x_{t+1} = n_{t+1} \mid x_t = n_t) \quad (4.24)$$

Nevertheless, the probability that the variable  $x_t$  assumes the value  $n_l$  at instant  $t = l$  is given by:

$$P_l(x_l = n_l) = \sum_{n_{l-1}} P_l \left( \bigcap_{t=0}^l x_t = n_t \right) \quad (4.25)$$

Replacing Equation (4.24) onto Equation (4.26), one has the following law of recurrence:

$$P_l(x_l = n_l) = \sum_{n_{l-1}} P_l(x_l = n_l | x_{l-1} = n_{l-1}) P_{l-1}(x_{l-1} = n_{l-1}) \quad (4.26)$$

The conditional probability in the last equation may be viewed as a transition probability from the state  $x_{l-1} = n_{l-1}$  to state  $x_l = n_l$ . In principle, this transition probability can change at each instant  $t$ . We shall consider here only transition probabilities that remain the same regardless of  $t$ . To simplify the notation, we shall write this transition probability as  $\pi(n_l, n_{l-1})$ . Considering the transition of a state  $n$  to a state  $m$ , one may write:

$$P_l(n) = \sum_m \pi(n, m) P_{l-1}(m) \quad (4.27)$$

### 4.2.2 Microscopic reversibility

Consider the trajectory of a Markov chain as the sequence of states  $n_0 \rightarrow \dots \rightarrow n_l$ . The reverse trajectory is going to be the inverse sequence  $n_l \rightarrow \dots \rightarrow n_0$ . The joint probability distribution for such reverse trajectory is:

$$P_l \left( \bigcap_{t=l}^0 x_t = n_t \right) = P(x_l = n_l) \prod_{t=0}^{l-1} P(x_{t+1} = n_{t+1} | x_t = n_t) \quad (4.28)$$

Therefore, the reverse transition probability,  $\pi'(n, m)$ , is:

$$\pi'(n, m) = \pi(m, n) \frac{P(n)}{P(m)} \quad (4.29)$$

A Markovian process has microscopic reversibility if (ALLEN; TILDESLEY, 1987):

$$\pi'(n, m) = \pi(n, m) \quad (4.30)$$

Replacing the condition of microscopic reversibility onto Equation (4.27), one has:

$$\sum_m [\pi(n, m)P(m) - \pi(m, n)P(n)] = 0 \quad (4.31)$$

which is called detailed balance if each term within the summation is null.

### 4.2.3 Metropolis Algorithm

The transition probability can be written in terms of a matrix  $\alpha$  and the probability of accepting a trial move (FRENKEL; SMIT, 2002):

$$\pi(n, m) = \alpha(n, m)acc(n, m) \quad (4.32)$$

In Metropolis Algorithm, the matrix  $\alpha$  is considered to be a symmetric matrix, *id est*,  $\alpha(n, m) = \alpha(m, n)$ . Thus, the ratio of acceptance can be written as:

$$\frac{acc(n, m)}{acc(m, n)} = \frac{P(n)}{P(m)} \quad (4.33)$$

Since  $acc(n, m)$  is a probability, it cannot be greater than 1. Therefore:

$$acc(n, m) = \begin{cases} P(n)/P(m), & \text{if } P(n) < P(m) \\ 1, & \text{if } P(n) \geq P(m) \end{cases} \quad (4.34)$$

For NVT ensemble, the probability of a state is given by Boltzmann distribution of energy. Thus:

$$acc(n, m) = \min\{1, \exp [-(U_n - U_m)/k_B T]\} \quad (4.35)$$

#### 4.2.4 Monte Carlo algorithm

In Monte Carlo simulations, one particle is randomly selected, the energy of such configuration is evaluated, this particle is randomly displaced and the energy of this new configuration is also evaluated. This move may be accepted or not. If accepted, the iterative process goes on, otherwise, a new displacement of the selected particle is tried. This is a quite rough and general scheme of the so-called Metropolis algorithm, which is usually applied on Monte Carlo simulations (FRENKEL; SMIT, 2002). The Monte Carlo technique, being a stochastic sampling, does not represent a physical trajectory, however, for a thermodynamic equilibrium this is irrelevant provided the microscopic reversibility is guaranteed.

Figure 10 presents an initial solid face centered cubic (FCC) configuration of Lennard-Jones particles and a final liquid configuration from a melting process using Monte Carlo simulations.

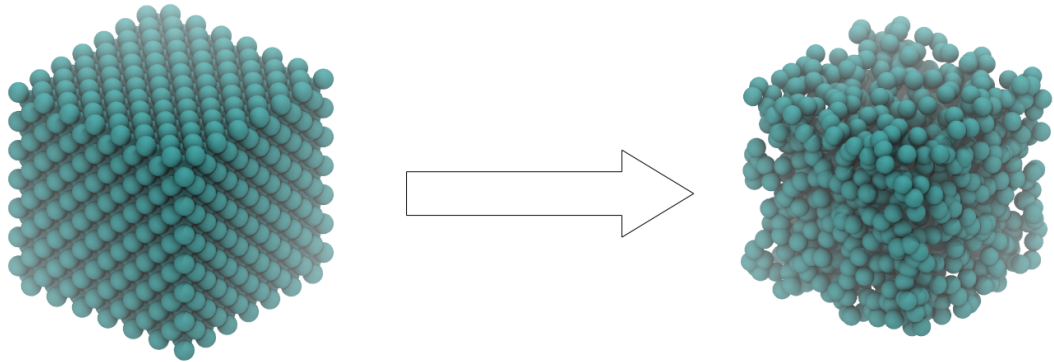


Figure 10: NVT Monte Carlo simulations of Lennard-Jones particles. The initial configuration on the left side is a face centered cubic (FCC) crystal. The final configuration on the right side is a representation of the melted system.

## 5 FROM THE STRUCTURE FACTOR TO OSMOTIC PRESSURE

### 5.1 Introduction<sup>1</sup>

Despite the increasing importance of bioprocesses in the chemical industry, the description of proteins in aqueous solutions, aiming at the development of models suitable to correlate and predict phase diagrams, still poses challenges. Phase diagrams of systems constituted by proteins in aqueous solutions may be very different from those of systems that do not contain biomolecules (PRAUSNITZ, 2008). Moreover, the influence of system conditions (such as temperature, pH and concentration) on the phase diagram is often counterintuitive.

Systems constituted by solutes and solvent, such as those constituted by a protein in aqueous solutions, are usually studied through a modified Hamiltonian. This modification considers only the solutes as individual particles and treats the solvent as a continuum medium characterized by properties such as permittivity, viscosity and density. A rigorous formal treatment to this approach was presented by McMillan and Mayer (MCMILLAN; MAYER, 1945). The McMillan-Mayer approach can be understood as an expanded ensemble in which the number of all component molecules is split in two independent variables: the solvent chemical potential and the number of solute molecules. The McMillan-Mayer framework can be converted in either Gibbs or Helmholtz frameworks (MOLLERUP; BREIL,

---

<sup>1</sup>Most of the content of this chapter is already published (Franco, L.F.M.; Oliveira, C.L.P.; Pessoa Filho, P. de A., Thermodynamics of protein aqueous solutions: from the structure factor to the osmotic pressure, *AIChE J.*, 61, 2871-2880, 2015.)



2009a; MOLLERUP; BREIL, 2009b). The advantage of using the McMillan-Mayer approach is the possibility of describing the behavior of solute molecules in a solvent by using the same equations used for describing the behavior of gas molecules in vacuum (HILL, 1986). This advantage has resulted in its extensive application on the description of colloidal systems.

The basis of the McMillan-Mayer framework is to consider that the interaction between two solute molecules is given by the potential of mean force, *i.e.*, the interaction potential averaged over all solvent molecule configurations (PRAUSNITZ, 2003). Therefore, the key aspect for any calculation within this approach is to determine the potential of mean force between two solute molecules at certain solvent condition. This can be done for protein solutions either through experiments or through molecular simulation. McMillan and Mayer (MCMILLAN; MAYER, 1945) considered the expansion of the grand-partition function in a power series in fugacities in their derivation. Therefore, their framework is usually employed through the virial expansion for the osmotic pressure (MCMILLAN; MAYER, 1945), for which the osmotic second virial coefficient arises as an important parameter. There is great interest in the experimental determination of this coefficient due to its relationship with the outcome of precipitation operations. George and Wilson (GEORGE; WILSON, 1994) proposed that protein crystallization would occur if the second virial coefficient lay in a definite range. Since their work, many investigations have been conducted to get deeper insight on this relation (TESSIER et al., 2003).

Despite recent advances in computational research, much of the experimental research in solution Thermodynamics still uses analytical solutions to interpret experimental data. This is due to intrinsic characteristics inherent to analytical solutions. The most important of these characteristics is the fact that a small set of parameters may provide insightful understanding of the raw data. Particularly

on the field of light, neutron and X-ray scattering, analytical solutions for the structure factor, which accounts for interparticle interaction, are commonly used to reproduce the experimental scattering intensity.

The development of analytical expressions for the structure factor usually considers the Ornstein and Zernike equation (ORNSTEIN; ZERNIKE, 1914), which defines the direct correlation function. However, this equation can only be solved by using approximations known as closure relations. A review on this subject can be found in Caccamo (CACCAMO, 1996). One of these closure relations, which results in analytical solutions for simple interparticle potentials, was formulated by Percus and Yevick (PERCUS; YEVICK, 1958). Such relation was solved independently by Wertheim (WERTHEIM, 1963) and Thiele (THIELE, 1963) for a hard sphere fluid. The resulting functional form for the structure factor was presented by Ashcroft and Lekner (ASHCROFT; LEKNER, 1966). Baxter (BAXTER, 1968) proposed an analytical solution, also through the Percus-Yevick closure relation, for the adhesive hard sphere fluid. Sharma and Sharma (SHARMA; SHARMA, 1976) introduced a modification in the Ashcroft-Lekner equation in order to be consistent with the Carnahan and Starling (CARNAHAN; STARLING, 1969) equation of state for hard sphere fluids.

Lebowitz and Percus (LEBOWITZ; PERCUS, 1966) proposed the Mean Spherical Approximation, formulated as a generalization of the spherical model for Ising spin systems to classical fluids (PALMER; WEEKS, 1973). Such approximation can be considered a perturbation of the Percus-Yevick closure relation (SHARMA; SHARMA, 1977). Expressions for the structure factor for a square well fluid (SHARMA; SHARMA, 1977), for charged particles in a neutralizing background (PALMER; WEEKS, 1973) and for macro-ions interacting via a screened Coulomb potential (HAYTER; PENFOLD, 1981; HANSEN; HAYTER, 1982) were proposed through this closure relation.

For protein solutions, structure factors given by an attractive potential with a hard core have been reported. For  $\gamma$ -crystallin and lysozyme solutions Malfois *et al.* (MALFOIS *et al.*, 1996) and Tardieu *et al.* (TARDIEU *et al.*, 1999) used a Yukawa potential within the Hyper-Netted Chain approximation, which can only be solved numerically. Barbosa *et al.* (BARBOSA *et al.*, 2010) described the structure factor of bovine serum albumin through a similar approach with the extension introduced by Narayanan and Liu (NARAYANAN; LIU, 2003). However, in Zhang *et al.* (ZHANG *et al.*, 2007) and in Barbosa *et al.* (BARBOSA *et al.*, 2010), the attractive parameter (either the depth from the square well potential or the Yukawa parameter) increases when the protein concentration decreases, even though the salt concentration remains constant.

Therefore, a more realistic attractive potential that allows a better understanding of the behavior of protein solutions is still needed. Here, we present a possible solution to the structure factor considering that the potential comprises a repulsive hard core, an attractive potential proportional to  $1/r^6$  - a van der Waals potential - and a repulsive screened Coulomb potential. Small-Angle X-ray Scattering (SAXS) experiments were conducted with bovine serum albumin (BSA) in concentrated salt solutions, which is the most important condition for industrial purposes, and these results were used to obtain the parameters of the intermolecular potential. The expression for the structure factor is used to generate an expression for the osmotic equation of state. With this expression and the parameters obtained from regressing SAXS scattering data, the osmotic pressure curve of BSA as a function of protein concentration in aqueous solution with sodium chloride is predicted with excellent agreement with the experimental data.

## 5.2 Materials and Methods

### 5.2.1 Reagents

The reagents used in the experiments were bovine serum albumin (Sigma-Aldrich, A3059,  $\geq 99.0\%$ ), acetic acid (Sigma-Aldrich, 320099,  $\geq 99.7\%$ ), trihydrated sodium acetate (Sigma-Aldrich, 32318,  $\geq 99.5\%$ ), sodium chloride (Sigma-Aldrich, S7653,  $\geq 99.5\%$ ), sodium sulfate (Sigma-Aldrich, 239313,  $\geq 99.0\%$ ), sodium nitrate (Sigma-Aldrich, S5506,  $\geq 99.0\%$ ) and deionized water (Milli-Q<sup>®</sup>). Protein stock-solutions of  $100.0 \text{ mg}\cdot\text{mL}^{-1}$  were prepared in buffer of acetic acid and trihydrated sodium acetate. pH values were measured in a pH-meter Digimed<sup>®</sup>.

### 5.2.2 SAXS experiments

Small-Angle X-ray Scattering (SAXS) experiments were run at room temperature of  $23 \pm 1^\circ\text{C}$ , using a Bruker's<sup>®</sup> NANOSTAR<sup>®</sup> equipment, with wavelength  $\lambda = 1.5418 \text{ \AA}$  of  $\text{CuK}_\alpha$  radiation and sample-detector distance of 67.0 mm. Scattering intensity data are presented as a function of the wavevector modulus  $q = (4\pi/\lambda) \sin \theta$ , in which  $\lambda$  is the radiation wavelength and  $2\theta$  is the scattering angle. The momentum transfer range was 0.013 to  $0.33 \text{ \AA}^{-1}$ . Background intensities were obtained from the scattering of buffer solutions measured inside the same capillaries. The scattering data were obtained with 300 to 1200 s of exposition, depending on the protein concentration, and were analyzed with the software SUPERSAXS (OLIVEIRA, 2009). In these experiments, the sample was continuously flown inside the vacuum chamber using an external peristaltic pump. By diluting the original stock solution with buffer solution, this mechanism allowed to obtain curves for several protein concentrations. The initial protein concentration was  $100.0 \text{ mg}\cdot\text{mL}^{-1}$ , and the salt concentration of both the stock

protein solution and the buffer solution was  $1.0 \text{ mol}\cdot\text{L}^{-1}$ . The scattering of BSA in sodium chloride, sodium sulfate and sodium nitrate solutions was investigated.

The scattering intensity was used to calculate the form and structure factors through the following analysis. Assuming that interactions are independent of the orientation, the scattering intensity  $I(q)$  at the wavevector modulus  $q$ , for a monodisperse ensemble of anisotropic particles, is given by (KOTLARCHYK; CHEN, 1983):

$$I(q) = KP(q) [1 + \beta(q) (S(q) - 1)] \quad (5.1)$$

where  $K$  is a proportionality constant that depends on the number of scattering particles,  $S(q)$  is the structure factor and  $P(q)$  is the form factor, defined by:

$$P(q) = \langle A^2(q) \rangle \quad (5.2)$$

where  $A(q)$  is the amplitude of scattering at  $q$  and  $\beta(q)$  is the ratio between the square of the average amplitude and the form factor:

$$\beta(q) = \frac{\langle A(q) \rangle^2}{\langle A^2(q) \rangle} \quad (5.3)$$

For spherical particles,  $\beta(q) = 1$ .

The BSA form factor was calculated using CRY SOL (SVERGUN; BARBER-ATO; KOCH, 1995) with the crystallographic structure of monomer A of 4F5S file from Protein Data Bank (BUJACZ, 2012). The amplitude was adjusted using the Nelder and Mead algorithm (NELDER; MEAD, 1965) (a code for this algorithm is presented on Appendix D) for an oblate ellipsoid of revolution ( $41.4\text{\AA} \times 41.4\text{\AA} \times 18.6\text{\AA}$ ), following Guinier's expression (PEDERSEN, 1997) with the volume equivalent to a sphere with diameter  $63.4 \text{ \AA}$ .

### 5.2.3 Monte Carlo simulations

Each Monte Carlo simulation was conducted with 2048 spherical particles of diameter  $\sigma$ . These particles were allocated within a cubic box with edge length of  $17.5 \sigma$ , such that the packing fraction was 0.2. The initial configuration was that correspondent to a face centered cubic crystal. Each translational trial was attempted following the Metropolis method in canonical ensemble and respecting periodic boundary conditions (FRENKEL; SMIT, 2002); the acceptance ratio was set to 0.5. Interaction potentials were truncated at  $3.0 \sigma$ . For equilibration,  $2.048 \times 10^7$  steps were performed. For the production stage, the same amount of cycles was used. Space coordinates were stored each 2048 steps. The interaction potential used in the simulations was:

$$\frac{U(x)}{\varepsilon} = \begin{cases} \infty, & \text{if } x \leq 1 \\ -\left(\frac{1}{x}\right)^6, & \text{if } x > 1 \end{cases} \quad (5.4)$$

where  $\varepsilon$  is the absolute value of the attractive potential at the contact distance,  $x = r/\sigma$  and  $\sigma$  is the particle diameter. Lennard-Jones units were used in the simulations.

The calculation of the structure factor was done with a post-processing code. The structure factor,  $S(q)$ , is related to the fluctuations of the spatial Fourier transform of the number density,  $\rho$ , through (ALLEN; TILDESLEY, 1987):

$$S(q) = \frac{1}{N} \langle \rho(q) \rho(-q) \rangle \quad (5.5)$$

$$\rho(q) = \sum_{j=1}^N e^{i\vec{q} \cdot \vec{r}_j} \quad (5.6)$$

In a cubic box,  $\vec{q} = (2\pi/L)(q_x, q_y, q_z)$  where  $L$  is the box edge length and  $q_x$ ,  $q_y$  and  $q_z$  are integers. Thus, following the approach of Frenkel *et al.* (FRENKEL

et al., 1986) and Cannavacciuolo *et al.* (CANNAVACCIUOLO et al., 2000), the structure factor was calculated at the reciprocal lattice points of the box. A set of 13 directions  $(h, k, l)$  generated by 001, 110 and 111 and their equivalents directions was considered (OLIVEIRA, 2005):

$$S(q_p) = \frac{1}{N} \left[ \sum_{j=1}^N e^{-ip2\pi(hx_j + ky_j + lz_j)/L} \right]^2 \quad (5.7)$$

$$q_p = \frac{2\pi}{L} p \sqrt{(h^2 + k^2 + l^2)}, p = 1, 2, \dots \quad (5.8)$$

Structure factors from different directions were averaged. The variance was calculated according to:

$$\sigma^2 = \frac{1}{M} \sum_{i=1}^M [S_i(q) - \overline{S(q)}] \quad (5.9)$$

where  $M$  is the total number of samples.

### 5.3 Theoretical framework

Using the concept of structure factor is not common in chemical thermodynamics. Therefore, its relationship to other quantities must be established beforehand. The structure factor is simply the Fourier transform of the radial distribution function (CACCAMO, 1996):

$$S(q) = 1 + \rho \int_0^{+\infty} g(r) e^{-iqr} dr \quad (5.10)$$

Defining the total correlation function  $h(r)$  as:

$$h(r) = g(r) - 1 \quad (5.11)$$

The Ornstein-Zernike equation relates the total correlation function,  $h(r)$ , and

the direct correlation function,  $c(r)$ , through (ORNSTEIN; ZERNIKE, 1914):

$$h(r) = c(r) + \rho \int h(|r' - r|)c(r')dr' \quad (5.12)$$

Applying the Fourier transform on this equation results in:

$$H(q) = C(q) + \rho H(q)C(q) \quad (5.13)$$

where  $H(q)$  is the Fourier transform of  $h(r)$ , and  $C(q)$  is the Fourier transform of  $c(r)$ . Coupling equations (5.10) and (5.13):

$$S(q) = \frac{1}{1 - \rho C(q)} \quad (5.14)$$

Therefore, to obtain  $S(q)$  we need an expression for  $C(q)$ . To relate  $C(q)$  with the interparticle potential, the mean spherical approximation (LEBOWITZ; PERCUS, 1966) may be considered:

$$c(r) = \begin{cases} g(r) [1 - e^{U(r)/k_B T}], & \text{if } r \leq \sigma \\ -\frac{U(r)}{k_B T}, & \text{if } r > \sigma \end{cases} \quad (5.15)$$

where  $U(r)$  is the interparticle interaction potential,  $k_B$  is the Boltzmann constant and  $T$  is the absolute temperature. Here we adopted the procedure developed by Sharma and Sharma (SHARMA; SHARMA, 1977). The direct correlation function is equal to a modified Percus-Yevick solution (SHARMA; SHARMA, 1976) for  $r \leq \sigma$ , and is proportional to the interaction potential (likewise the random phase approximation) for  $r > \sigma$ . The interaction potential is considered to be the sum of an attractive contribution and a screened Coulomb contribution:

$$U(r) = \begin{cases} \infty, & \text{if } r \leq \sigma \\ -\varepsilon \left(\frac{\sigma}{r}\right)^6 + \frac{(ze)^2 \exp[-\kappa(r-\sigma)]}{4\pi\epsilon_0\epsilon_r r(1+\kappa\sigma/2)^2}, & \text{if } r > \sigma \end{cases} \quad (5.16)$$

where  $\varepsilon$  is the attractive potential depth,  $\sigma$  is the particle diameter,  $z$  is the



charge of the particle in units of the elementary charge  $e$ ,  $\kappa$  is the inverse of Debye's length,  $\epsilon_0$  is the vacuum permittivity and  $\epsilon_r$  is the medium dielectric constant.

Following the approach of Sharma and Sharma (SHARMA; SHARMA, 1977), the direct correlation function  $c(r)$  can be written as a sum of the direct correlation functions generated by a hard sphere potential,  $c^{hs}(r)$ , by the attractive potential,  $c^{vdW}(r)$ , and by the screened Coulomb potential,  $c^{Coul}(r)$ :

$$c(r) = c^{hs}(r) + c^{vdW}(r) + c^{Coul}(r) \quad (5.17)$$

The Fourier transform of Equation (5.17) is:

$$C(q) = C^{hs}(q) + C^{vdW}(q) + C^{Coul}(q) \quad (5.18)$$

For the hard sphere contribution, we will consider the solution obtained by Ashcroft and Lekner (ASHCROFT; LEKNER, 1966) with the modification introduced by Sharma and Sharma (SHARMA; SHARMA, 1976):

$$\begin{aligned} C^{hs}(k) = & -\frac{24\eta}{\rho k^6} \{ \alpha k^3 [\sin k - k \cos k] + \beta k^2 [2k \sin k - (k^2 - 2) \cos k - 2] \\ & + \gamma [(4k^3 - 24k) \sin k - (k^4 - 12k^2 + 24) \cos k + 24] \} \end{aligned} \quad (5.19)$$

where  $\eta = \rho\pi\sigma^3/6$  is the packing fraction, and:

$$k = q\sigma \quad (5.20)$$

$$\alpha = \frac{[(1 + 2\eta)^2 + \eta^3(\eta - 4)]}{(1 - \eta)^4} \quad (5.21)$$

$$\beta = -\frac{\eta (18 + 20\eta - 12\eta^2 + \eta^4)^2}{3(1 - \eta)^4} \quad (5.22)$$

$$\gamma = \frac{\eta\alpha}{2} \quad (5.23)$$

This expression is consistent with the Carnahan-Starling equation of state (SHARMA; SHARMA, 1976). In this case, the Fourier transform depends on the protein concentration, implicit in the packing fraction  $\eta$ ; however, we left  $C^{hs}$  written simply as function of  $k$ , as this parameter is related to  $q$  and  $r$  (through inverse Fourier transform).

The attractive contribution of the direct correlation function is obtained from the functional form of the interparticle interaction in Equation (5.16):

$$C^{rdW}(q) = 4\pi \frac{\sigma^6}{T^*} \int_{\sigma}^{\infty} \frac{1}{r^6} \frac{\sin(qr)}{qr} r^2 dr \quad (5.24)$$

where  $T^*$  is the reduced temperature, defined by  $T^* = k_B T / \varepsilon$ . This expression results in:

$$C^{rdW}(k) = \frac{\pi\sigma^3}{6T^*k} \left[ k^4 \left( \frac{\pi}{2} - Si(k) \right) + (6 - k^2) \sin(k) + k(2 - k^2) \cos(k) \right] \quad (5.25)$$

where  $k$  is defined by Equation (5.20) and  $Si(x)$  is the integral defined as:

$$Si(x) = \int_0^x \frac{\sin(t)}{t} dt \quad (5.26)$$

From the functional form of the interparticle interaction in Equation (5.16),

the direct correlation function of the electrostatic contribution is:

$$C^{Coul}(q) = -\frac{z^2 e^2 \exp(\kappa\sigma)}{k_B T \epsilon_0 \epsilon_r (1 + \kappa\sigma/2)^2} \int_{\sigma}^{\infty} \frac{\exp(-\kappa r)}{r} \frac{\sin(qr)}{qr} r^2 dr \quad (5.27)$$

where  $\kappa$  is the inverse of Debye's length, defined as:

$$\kappa = \sqrt{\left( \frac{2N_A e^2 I \rho_s}{\epsilon_0 \epsilon_r k_B T} \right)} \quad (5.28)$$

$N_A$  is Avogadro's number,  $\rho_s$  is the solvent density and  $I$  is the ionic strength, defined as:

$$I = \frac{1}{2} \sum_{i=1}^N m_i z_i^2 \quad (5.29)$$

where  $m_i$  is the molality (in mol per kilograms of solvent) of particle  $i$ . The influence of protein on the ionic strength is neglected<sup>2</sup>.

From Equation (5.27):

$$C^{Coul}(k) = -z^2 \frac{\phi}{q\kappa} \left( \frac{\sin k + Q \cos k}{1 + Q^2} \right) \quad (5.30)$$

$$\phi = \frac{e^2}{k_B T \epsilon_0 \epsilon_r (1 + K)^2} \quad (5.31)$$

where  $k = q\sigma$ ,  $Q = q/\kappa$  and  $K = \kappa\sigma/2$ .

The Debye-Hückel point ion limit can be recovered from Equation (5.30), letting  $\sigma \rightarrow 0$  and considering the low density first term expansion.

Coupling Equation (5.18) with Equations (5.19), (5.25) and (5.30), the final

---

<sup>2</sup>As it has been pointed out by Prof. Dr. Marcelo Castier, the influence of the protein on the ionic strength calculation is not neglectable. Nevertheless, the real contribution of the protein to the ionic strength is still under debate, because the very definition of the ionic strength is developed for punctual electric charges. One possible way to minimize this effect is to use a similar approach that has been used for common polyelectrolytes (PESSÔA-FILHO; MAURER, 2008). Using such approach, it is possible to neglect this effect here.

expression for the direct correlation function  $C(k)$  is:

$$\begin{aligned}
C(k) = & -\frac{24\eta}{\rho k^6} \{ \alpha k^3 [\sin k - k \cos k] + \beta k^2 [2k \sin k - (k^2 - 2) \cos k - 2] \\
& + \gamma [(4k^3 - 24k) \sin k - (k^4 - 12k^2 + 24) \cos k + 24] \\
& - \frac{k^5 \varepsilon}{24k_B T} \left[ k^4 \left( \frac{\pi}{2} - Si(k) \right) + (6 - k^2) \sin(k) + k (2 - k^2) \cos(k) \right] \\
& + \frac{k^6 \phi z^2}{4\pi \sigma^3 q \kappa} \left( \frac{\sin k + Q \cos k}{1 + Q^2} \right) \}
\end{aligned} \tag{5.32}$$

### 5.3.1 Isothermal compressibility and the osmotic equation of state

The isothermal compressibility is related to the limit of  $q = 0$  in the structure factor so that:

$$\lim_{q \rightarrow 0} S(q) = \rho k_B T \kappa_T = k_B T \left( \frac{\partial \rho}{\partial \Pi} \right)_T \tag{5.33}$$

where  $\Pi$  is the osmotic pressure and  $\kappa_T$  is the isothermal compressibility. The structure factor presented in Equation (5.32) gives, for an uncharged system (*exempli gratia*, at the isoelectric point), the following isothermal compressibility:

$$\kappa_T = \frac{\pi \sigma^3}{6\eta (\alpha k_B T - 8\varepsilon\eta)} \tag{5.34}$$

where  $\alpha$  is given by Equation (5.21). The osmotic pressure can be calculated through:

$$\Pi = \int_0^\eta \frac{d\eta}{\eta \kappa_T} \tag{5.35}$$

which results in:

$$Z = \frac{\Pi}{\rho k_B T} = \frac{1 + \eta + \eta^2 - \eta^3}{(1 - \eta)^3} - \frac{8}{\eta k_B T} \int_0^\eta \varepsilon(\eta) \eta d\eta \tag{5.36}$$

We left intentionally  $\varepsilon$  as a function of  $\eta$  due to the experimental evidence that  $\varepsilon$  is not constant for proteins in aqueous solutions (ZHANG et al., 2007;

BARBOSA et al., 2010). As  $Z$  must approach 1 as  $\eta$  approaches 0, the expression for  $\varepsilon$  must follow the limit:

$$\lim_{\eta \rightarrow 0} \frac{1}{\eta} \int_0^\eta \varepsilon(\eta) \eta d\eta = 0 \quad (5.37)$$

Equation (5.36) reduces to the Carnahan-Starling-van der Waals equation of state if  $\varepsilon$  is constant.

## 5.4 Results

### 5.4.1 Validation of the structure factor

An example of the structure factor generated by Equation (5.32) is presented in Figure 11. This Figure was generated using a highly attractive parameter ( $\varepsilon/k_B T = 1.67$ ). For high values of  $k$ , the structure factor goes to 1 as expected. For small values of  $\eta$ , the structure factor goes to 1 regardless of  $k$ , which means that dilute systems approach the ideal behavior. However, the first peak enlarges when  $\eta$  increases, and so does the isothermal compressibility (given by the limit of  $k \rightarrow 0$ ) for small values of  $\eta$ . After certain  $\eta$  value, the isothermal compressibility decreases when  $\eta$  increases.

Figure 11 shows that the expected limits are respected, but it does not allow an assessment of whether the approximations assumed in the derivation of Equation (5.32) are adequate. Therefore, comparison with Monte Carlo simulations was carried out. Figure 12 shows the results of the structure factor obtained from Monte Carlo simulations (using the interparticle potential from Equation 5.4) and from Equation (5.32) for an uncharged particle, calculated for different values of the attractive parameter and  $\eta = 0.2$ .

The agreement between the results from Equation (5.32) and from Monte Carlo simulations is remarkable. However, for high values of the attractive pa-

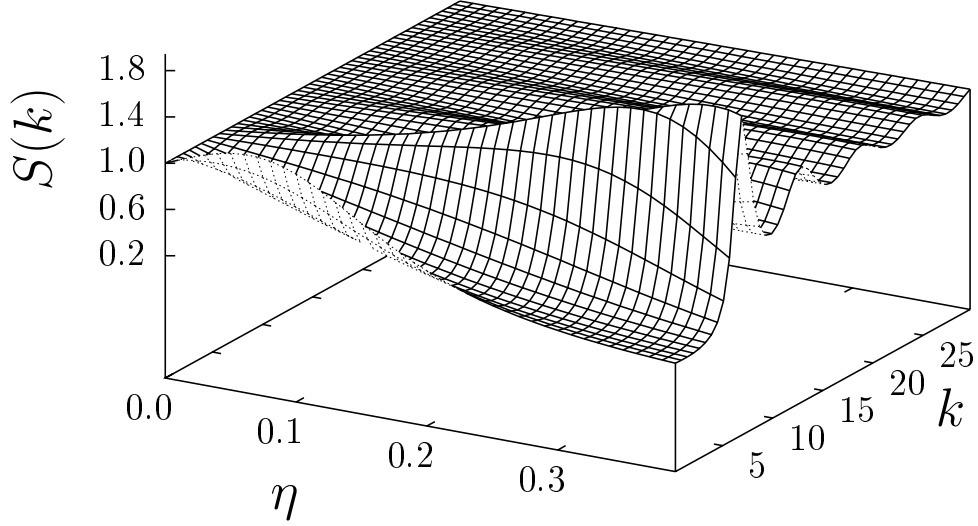


Figure 11: Structure factor (Equation 5.32) as a function of  $\eta$  and  $k$  (Equation 5.20) calculated for a system of uncharged particles and attractive parameter  $\varepsilon/k_B T = 1.67$ .

parameter a non-negligible difference between the isothermal compressibility calculated via Monte Carlo and Equation (5.32) is observed. This means that Equation (5.32) fails to predict the large increment on the isothermal compressibility for highly attractive systems. This failure is related to the hypotheses behind the random phase approximation, which considers the attractive potential as a perturbation of the hard sphere reference system.

#### 5.4.2 Structure factor of Bovine Serum Albumin (BSA) in aqueous salt solutions

Figure 13 shows the form factor for BSA in aqueous solution. We considered that the form factor is equal to the scattering intensity in very dilute solutions. In this case, the experimental data was obtained with a protein concentration of  $6.8 \text{ mg}\cdot\text{mL}^{-1}$ . The agreement between the calculated curve and the experimental

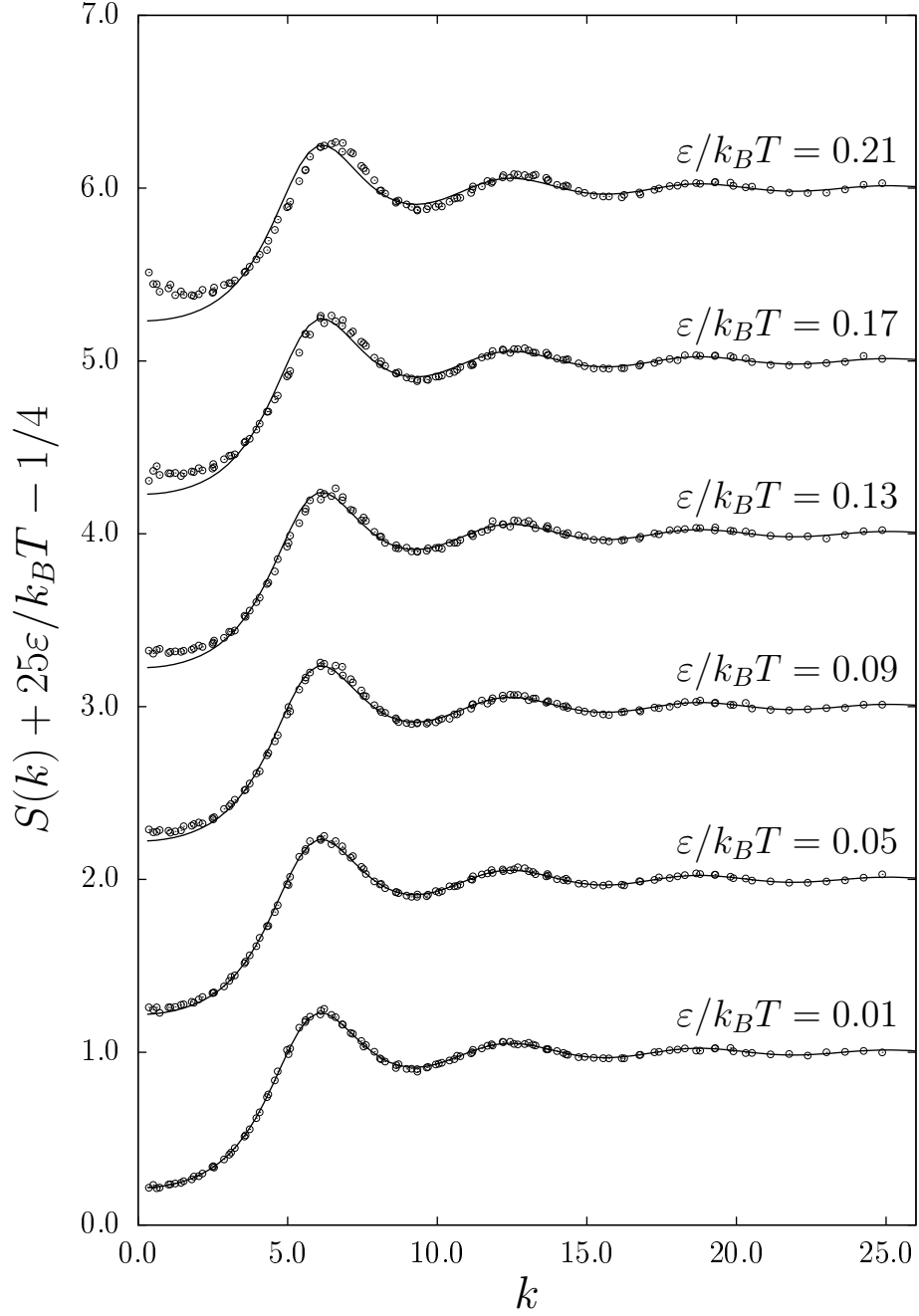


Figure 12: Structure factor as a function of  $k = q\sigma$  and  $\varepsilon/k_B T$  for a system of uncharged particles with  $\eta = 0.2$ . Open circles, Monte Carlo results (standard deviations are smaller than symbol size). Continuous line, Equations (5.14) and (5.32).

data is high. The value of  $\beta(q)$  is close to 1 for small scattering angles, which indicates that at small angles BSA can be modeled as an oblate ellipsoid of revolution.

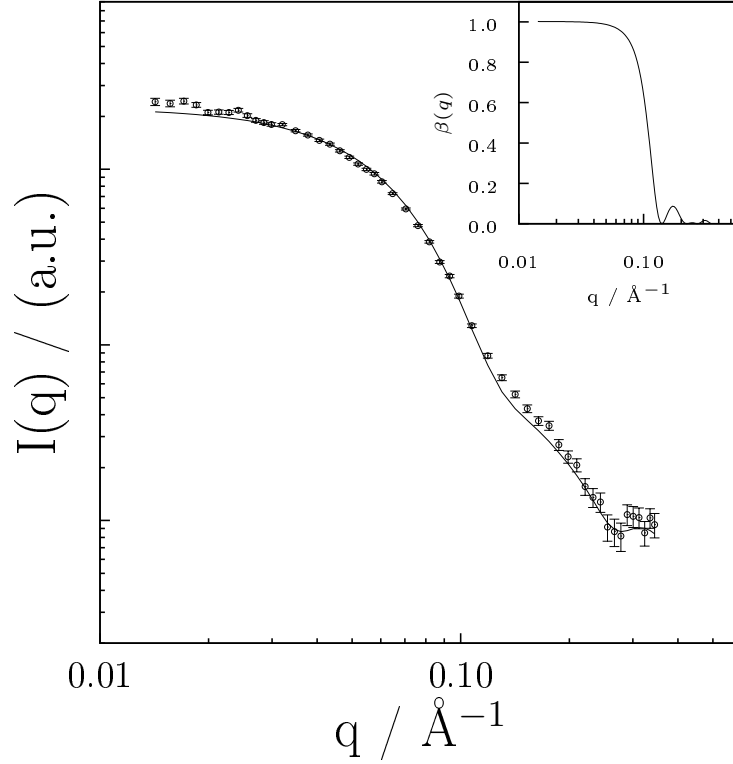


Figure 13: Form factor for bovine serum albumin (BSA) in aqueous solution. Experimental results for the scattering intensity (in a.u., arbitrary units) of BSA in aqueous solution with  $6.8 \text{ mg}\cdot\text{mL}^{-1}$ , open circles. Curve calculated using CRY SOL (SVERGUN; BARBERATO; KOCH, 1995), continuous line. The inset plot shows  $\beta(q)$  (Equation 5.3) considering an oblate ellipsoid of revolution.

The experimental data of scattering intensity as a function of protein concentration and wavevector modulus are presented on Appendix G.

Previous works (BARBOSA et al., 2010; ZHANG et al., 2007) showed that the attractive parameter of proteins in solution may depend on protein concentration. The simplest relation between  $\varepsilon$  and  $\eta$  that follows the thermodynamic restriction imposed by Equation (5.37) is:

$$\varepsilon = \varepsilon^o - \varepsilon' \eta \quad (5.38)$$

where  $\varepsilon^o$  is the value of the attractive parameter at infinite dilution. The parameters of Equation (5.38) were obtained by minimizing the following objective



function:

$$\chi_r^2 = \frac{1}{N_c N - M} \sum_{j=1}^{N_c} \sum_{i=1}^N \left( \frac{I_j^{exp}(q_i) - I_j^{mod}(q_i)}{c_{p,j} \sigma_{ji}} \right)^2 \quad (5.39)$$

where  $N_c$  is the number of different sets of experimental data,  $N$  is the number of experimental intensities for a single protein concentration,  $M$  is the number of parameters used in the fitting procedure,  $I_j^{exp}(q_i)$  is the experimental scattering intensity,  $I_j^{mod}(q_i)$  is the calculated scattering intensity and  $\sigma_{ji}$  is the standard deviation of the experimental scattering intensity at  $q_i$  and protein concentration  $c_{p,j}$ . Table 1 presents the values of parameters  $\varepsilon^o$  and  $\varepsilon'$  adjusted to the experimental data using Equation (5.38), as well as the values of  $\chi$ .

Table 1: Values of parameters  $\varepsilon^o$  and  $\varepsilon'$ , Equation (5.38), values of  $\chi$  for the fitting procedure, Equation (5.39), and values of the osmotic second virial coefficient  $B$ , Equation (5.44), calculated for BSA in aqueous solutions with 1.0 mol·L<sup>-1</sup> sodium salts.

pH	Anion	$\varepsilon^o/k_B T$	$\varepsilon'/k_B T$	$\chi$	$B \times 10^4 / \text{cm}^3 \cdot \text{mol} \cdot \text{g}^{-2}$
4.9	SO <sub>4</sub> <sup>-2</sup>	2.00	8.24	2.52	-2.18
	Cl <sup>-</sup>	2.32	11.10	11.35	-2.65
	NO <sub>3</sub> <sup>-</sup>	2.43	16.95	6.26	-2.81
6.3	SO <sub>4</sub> <sup>-2</sup>	1.44	5.55	1.96	-1.37
	Cl <sup>-</sup>	2.39	14.77	8.05	-2.75
	NO <sub>3</sub> <sup>-</sup>	1.92	11.17	6.18	-2.06

A first question that arises from the analysis of experimental data is whether  $\varepsilon$  actually depends on  $\eta$ . Figure 14 shows the comparison, for two protein solutions, among the results obtained for the Hard Sphere Percus Yevick equation, for Equation (5.32) and constant  $\varepsilon$ , and for Equation (5.32) coupled with Equation (5.38). For concentrated protein solutions, electrostatic interactions are screened by ions in solution, and even the hard sphere potential is sufficient to predict

the scattering intensity. However, as the protein concentration decreases, the experimental scattering deviates from the Percus Yevick prediction. The use of a single value of  $\varepsilon$  does not allow a good correlation for all protein concentrations, and considering that  $\varepsilon$  depends on the protein concentration is necessary.

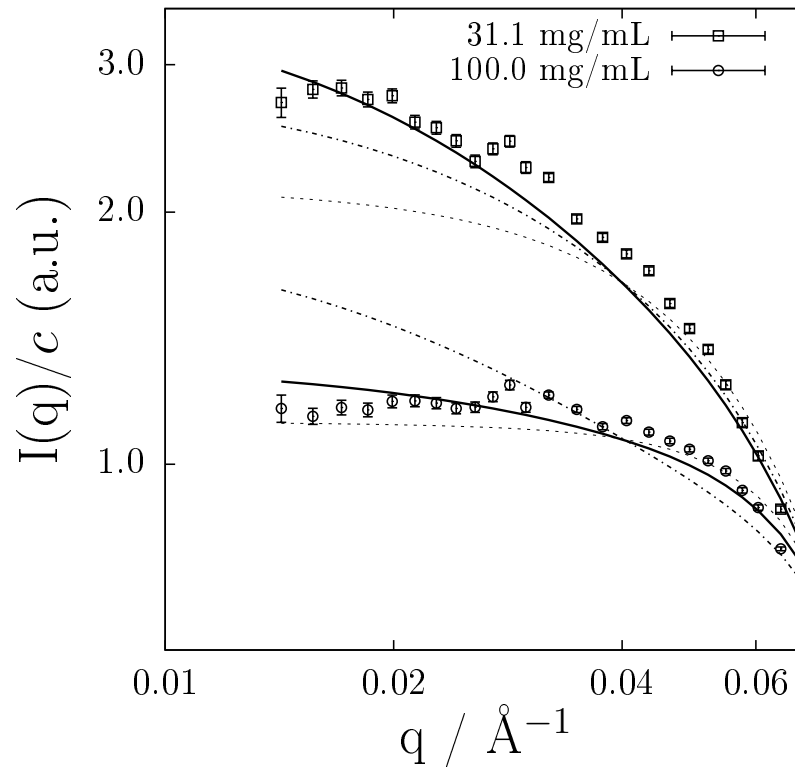


Figure 14: X-ray scattering intensity (in a.u., arbitrary units) of BSA in  $1.0 \text{ mol}\cdot\text{L}^{-1}$   $\text{NaNO}_3$  aqueous solution at  $\text{pH}=4.9$  and  $23^\circ\text{C}$  for different BSA concentrations at very small  $q$  values. Open symbols, experimental data; continuous line, Equation (5.32) with  $\varepsilon$  calculated using Equation (5.38); dotted-dashed line, Equation (5.32) with constant  $\varepsilon$ ; dashed line, Hard-Sphere Percus-Yevick solution.

Figure 15 shows the scattering intensity of BSA in aqueous solution with  $\text{NaNO}_3$  ( $1.0 \text{ mol}\cdot\text{L}^{-1}$ ) at the isoelectric point (*i.e.* the pH value for which the protein net charge is null,  $\text{pH} = 4.9$  for BSA) for three protein concentrations, and Figure 16 presents the scattering intensity of BSA in aqueous solution with  $\text{NaNO}_3$  above the isoelectric point ( $\text{pH} = 6.3$ ). The agreement between the scattering intensities calculated by Equation (5.32) and the experimental data is very good. In both Figures, the calculated structure factor with the proposed model,

Equation (5.32), is also presented. Figures comprising all systems studied are presented on Appendix H.

For the isoelectric point, the net charge is null and the Coulomb contribution to the structure factor vanishes. However, for a pH above the isoelectric point, the protein molecule bears a negative charge that must be considered in Equation (5.27). To obtain the net charge profile of BSA as a function of pH, the  $pK_A$  values obtained by PROPKA (LI; ROBERTSON; JENSEN, 2005) with the crystallographic structure of BSA (BUJACZ, 2012) were considered, and the resulting profile was adjusted so that the isoelectric point was placed at  $pH = 4.9$ .

The results for parameters  $\varepsilon^o$  and  $\varepsilon'$  shown in Table 1 present two interesting features. The first one is that they depend on the pH. Even considering that this dependence is weak, it shows that the decoupling of the attractive part of  $c(r)$  into  $c^{vdW}(r)$  and  $c^{Coul}(r)$ , Equation (5.17), is imperfect - otherwise,  $c^{vdW}(r)$  should not be influenced by the protein net charge. This may be due either to the expressions considered for  $C^{vdW}(q)$  and  $C^{Coul}(q)$ , which involve approximations such as the calculation of the protein net charge, or to the very hypothesis that  $c(r)$  can be decoupled in this way. The decoupling of the interaction potential into short-range and long-range terms is certainly an approximation, but is a useful and widely employed one. On the other hand, the expressions used for  $C^{vdW}(q)$  and  $C^{Coul}(q)$  also entail approximations. The experimental data are not sufficient to decide which one of these aspects is critical. However, these findings warn against using this model uncritically.

The second feature is that the fitting of Equation (5.38) leads to the unexpected conclusion that the more diluted is the protein in solution, the more attractive is its interaction potential. The very fact that the attractive parameter may depend on the solute concentration is not obvious. If the analogy with the

non-ideal gas behavior, hypothesized by the McMillan and Mayer (MCMILLAN; MAYER, 1945) approach, was complete, then this potential should be independent of the solute concentration. However, as previously shown in Figure 14, considering this dependence is necessary even at the isoelectric point, at which the average charge of protein molecules is null; in this case, the decoupling of the attractive part of  $C(q)$  and the expression used for  $C^{Coul}(q)$  play no role in the calculations.

One possible explanation for this contradiction is that the many-body contribution to the interaction potential, which is more relevant for more concentrated protein solutions, would lead to an apparent decrease (in absolute value) of the value of the interaction parameter. However, this contribution always leads to an increase in the attraction between molecules, which would result in an increase in the absolute value of the interaction parameter. A brief general demonstration of this fact is presented in the Appendix A.

Another possible explanation for this fact lies in the charge-dipole interactions. According to Striolo et al. (STRIOLO et al., 2002; STRIOLO et al., 2003) and Bratko et al. (BRATKO et al., 2002), accounting for charge-dipole interactions improves the description of protein-protein interactions. In this case, the following term should be added to the interaction potential (ISRAELACHVILI, 2011):

$$U^{c-d}(r) = -\zeta \left(\frac{\sigma}{r}\right)^4, \text{ if } r > \sigma \quad (5.40)$$

where  $\zeta$  is a parameter proportional to the square of the product between the protein charge,  $z$ , and the protein dipole moment,  $\mu$ .

This would mean that the following term should be added to the direct correlation function, Equation (5.32):

$$C^{c-d}(k) = \frac{2\pi\sigma^3\zeta}{k_B T} \left[ \frac{\sin k}{k} - \cos k + k^2 \left( 2Si(k) - \frac{\pi}{2} \right) \right] \quad (5.41)$$

However, this term does not provide any insight in why  $\varepsilon$  changes with  $\eta$ , as it does not depend on protein concentration and would have the same effect on  $S(q)$  for all protein concentrations. Particularly at the isoelectric point, this contribution is null, since the protein has no net charge and  $\zeta$  is zero.

Finally, the other possible explanation for this finding is that the implicit solvent hypothesis behind the McMillan-Mayer framework is not adequate to describe the behavior of protein solutions. The fact that different salts have different effects on the protein molecules has long been known (ZHANG; CREMER, 2006). The conclusion of this work is subtler: the effect of the same salt at the same concentration may not be the same, and may depend on the concentration of the protein - or likewise on the proportion between salt ions to protein molecules. In this sense, the inclusion of the salt ions as solutes in the study of protein solutions, instead of a co-solvent, seems to be unavoidable even within the McMillan and Mayer framework.

### 5.4.3 Osmotic equation of state

By coupling Equation (5.38) and Equation (5.36) the following expression for  $Z$  is obtained for uncharged systems:

$$Z = \frac{1 + \eta + \eta^2 - \eta^3}{(1 - \eta)^3} - \frac{4\varepsilon^o\eta}{k_B T} + \frac{8\varepsilon'\eta^2}{3k_B T} \quad (5.42)$$

The osmotic second virial coefficient,  $B$ , can be calculated through:

$$B = \lim_{\rho \rightarrow 0} \frac{Z - 1}{\rho} \quad (5.43)$$

From Equation (5.42):

$$B = \frac{2\pi\sigma^3}{3} \left( 1 - \frac{2\varepsilon^o}{k_B T} \right) \quad (5.44)$$

Table 1 presents the value of the osmotic second virial coefficient calculated for BSA in these salt solutions. All values are negative, which indicates the predominance of attractive forces over repulsive ones. The osmotic second virial coefficient is known to depend on the pH, due to the dependence of the average net charge on the pH, which shifts the electrostatic repulsion (NEAL; ASTHAGIRI; LENHOFF, 1998). For BSA, the calculated values of  $B$  at pH 6.3 are lower (in absolute value) than the calculated values of  $B$  at the isoelectric point, except for sodium chloride. The calculated values of  $B$  are similar to those previously reported for BSA in aqueous salt solutions. For instance, the experimental data by Wu and Prausnitz (WU; PRAUSNITZ, 1999) suggests a value of  $B = -1.7 \times 10^{-4} \text{ cm}^3 \cdot \text{mol} \cdot \text{g}^{-2}$  for BSA in aqueous solution with  $1.0 \text{ mol} \cdot \text{L}^{-1}$  sodium chloride at pH = 4.5. For BSA in aqueous solution with  $1.0 \text{ mol} \cdot \text{L}^{-1}$  ammonium sulfate at pH = 4.8, a value of  $B = -0.77 \times 10^{-4} \text{ cm}^3 \cdot \text{mol} \cdot \text{g}^{-2}$  was reported (LU et al., 2009).

Further analysis can be done by considering experimental data on osmotic pressures. Figure 17 shows the results for BSA osmotic pressure as a function of protein concentration in  $1.0 \text{ mol} \cdot \text{L}^{-1}$  NaCl aqueous solution. The continuous line was drawn using Equation (5.42) with parameters presented in Table 1. The developed equation can reproduce with good agreement the experimental data, despite the slight difference in pH values. The comparison with the Hard-Sphere Percus-Yevick equation shows that accounting for the attractive intermolecular interactions is important when calculating the osmotic pressure even at low protein concentrations. The agreement between the proposed equation for  $\Pi$  and the experimental results shows that the simplified hypotheses adopted in the development do not result in a loss of accuracy for the description of macroscopic quantities.

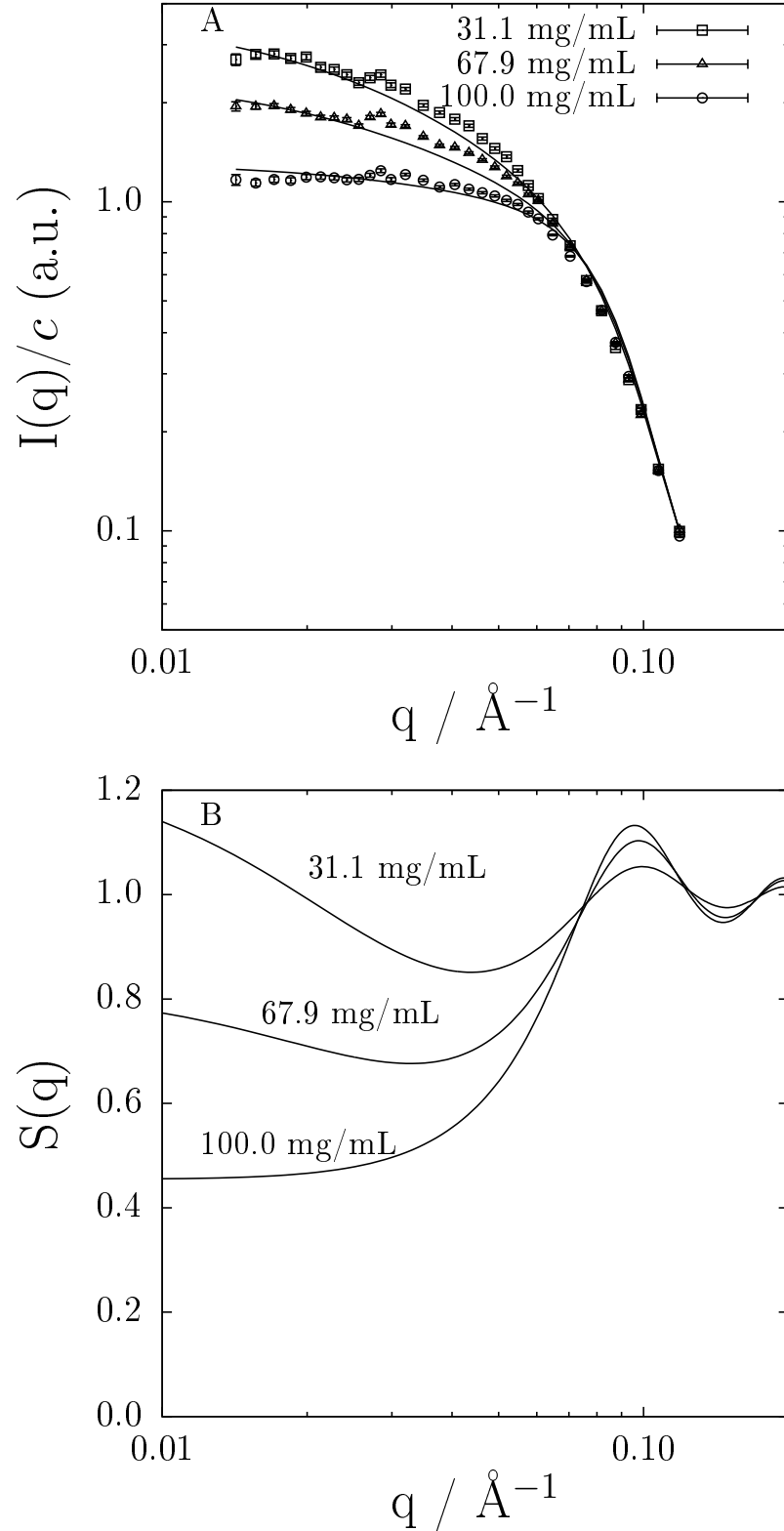


Figure 15: A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L<sup>-1</sup> NaNO<sub>3</sub> aqueous solution at pH=4.9 and 23 °C. Open symbols, experimental data; continuous line, Equation (5.32) with  $\varepsilon$  calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with  $\varepsilon$  calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution.

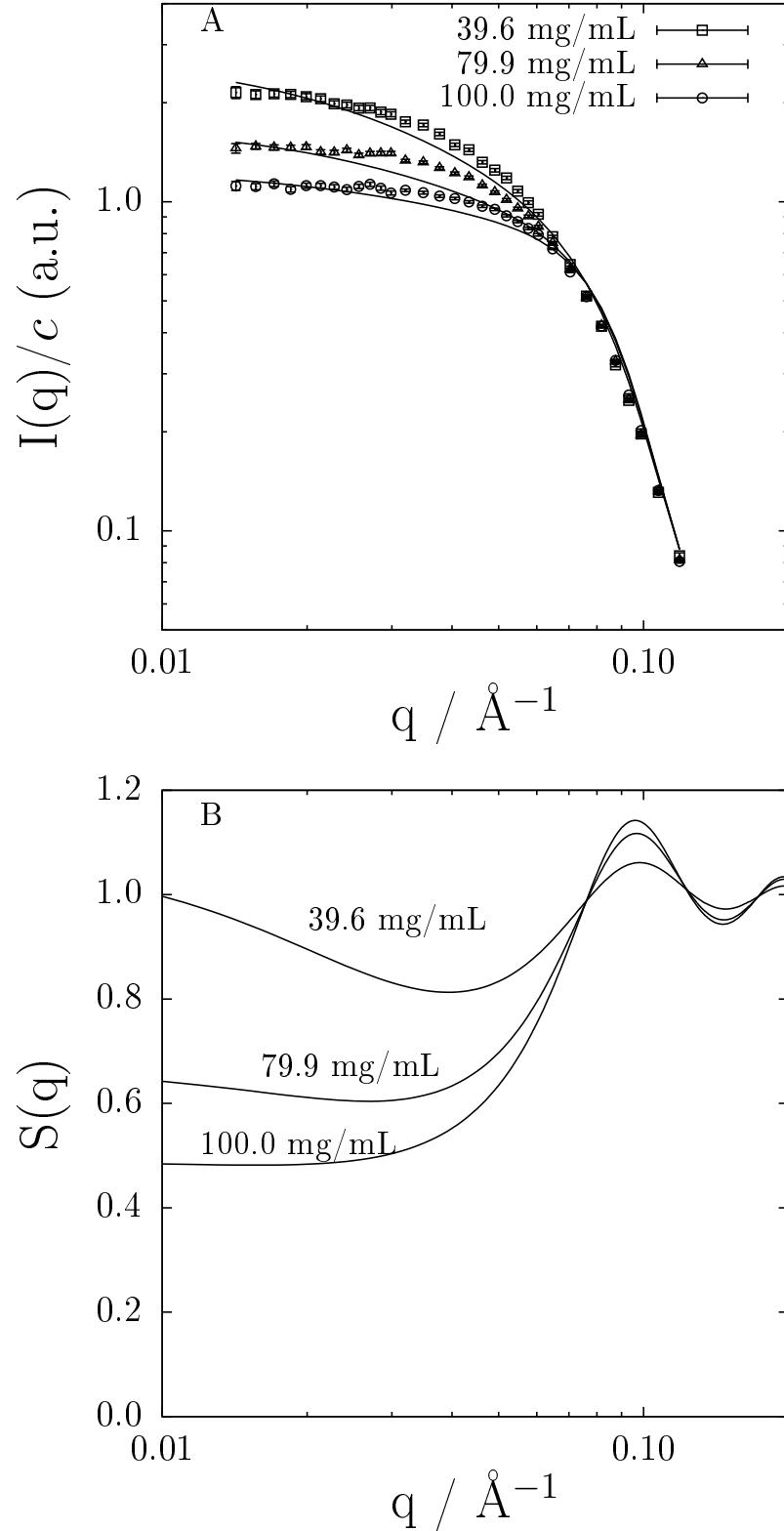


Figure 16: A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L<sup>-1</sup> NaNO<sub>3</sub> aqueous solution at pH=6.3 and 23 °C. Open symbols, experimental data; continuous line, Equation (5.32) with  $\varepsilon$  calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with  $\varepsilon$  calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution.



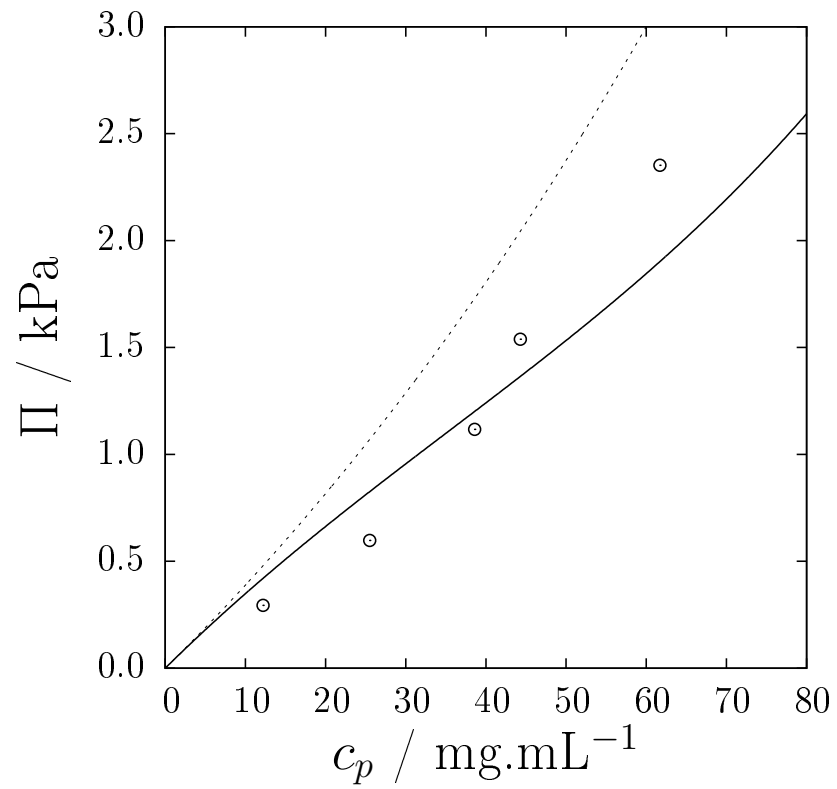


Figure 17: Osmotic pressure of BSA in  $1.0 \text{ mol}\cdot\text{L}^{-1}$  NaCl aqueous solution. Open circles, experimental data from Wu and Prausnitz (WU; PRAUSNITZ, 1999) (pH=4.5); continuous line, Equation (5.42) with parameters calculated at pH=4.9; dashed line, Hard-Sphere Percus-Yevick equation of state.

## 6 FROM THE OSMOTIC PRESSURE TO FLUID-FLUID COEXISTENCE CURVE

### 6.1 Introduction

One of the most important problems related to the thermodynamics of protein systems is the ability to predict phase separation. From the perspective of a chemical engineer, the possibility of drawing a phase diagram is the key point to design, operate and optimize a separation process, *exempli gratia*, precipitation, crystallization, or even chromatography. From the medical point of view, the phase diagram of protein systems is the key to understand the mechanism of some diseases such as cataract, sickle cell disease, Alzheimer's disease and multiple myeloma (BENEDEK, 1997); diseases called "protein condensation diseases".

To plot a phase diagram for protein systems, one needs to know the interaction potential between two proteins in solution. The interaction potential is the Achilles heel of such problem, because with all the thermodynamic relations the knowledge of such potential is the necessary and sufficient condition to predict a phase separation.

There are at least two types of protein system phase separations which encounter resonance in engineering and in medical applications: solid-liquid and liquid-liquid. Solid-liquid phase separation means that two phases result from the phase split: one solid phase containing a solid protein (which may be a crystal or not) and a liquid phase containing mostly the solvent and co-solvents and the

protein. Liquid-liquid phase separation means that two phases also result from the phase split: one with high protein concentration and another with low protein concentration.

The solid-liquid equilibrium has been extensively studied both experimentally and theoretically by means of the solubility. There is a considerably large amount of solubility data for a variety of proteins and conditions such as temperature, salt type, salt concentration and pH. The liquid-liquid equilibrium has also been studied, however, the available experimental coexistence line is restricted to proteins as lysozyme (KATSONIS; BRANDON; VEKILOV, 2006) and  $\gamma$ -crystallin (BROIDE et al., 1991). Most of the studies are concerned to the theoretical prediction of the liquid-liquid coexistence line for different interaction potential models (TAVARES; PRAUSNITZ, 2004; TAVARES; SANDLER, 1997; KASTELIC et al., 2015).

There is a general belief, although there are some controversies, that the short-ranged attractive potential is responsible for the liquid-liquid separation of protein systems (DUDA, 2009).

In the previous chapter, we derived a new equation of state which attractive parameters were fitted to small-angle X-ray scattering intensity data for bovine serum albumin. The main purpose of the present chapter is to calculate the fluid-fluid coexistence curve using this equation of state and the fitted parameters. No fluid-fluid coexistence was observed for this protein at the isoelectric point or at a higher pH value. Nevertheless, such transition was experimentally observed at lower pH values.

## 6.2 Derivation of thermodynamic potentials

Following the McMillan-Mayer ensemble, one has to consider that the protein is viewed as one single component in a continuous medium. Therefore, all thermodynamic potentials are functions of the number of protein molecules and the solvent chemical potential. For the sake of simplicity, the latter dependence will not be explicitly presented, although it must be kept in mind.

Once one has presented an equation of state as Equation (5.42), one is able to derive several thermodynamic potentials as the residual Helmholtz free energy and the residual chemical potential. The residual molar Helmholtz free energy,  $a^R$ , which is the difference between the molar Helmholtz free energy of the system,  $a$  and the molar ideal solution Helmholtz free energy,  $a^{IS}$ , is calculated through the following expression (TESTER; MODELL, 1997):

$$a^R(T, v) = a(T, v) - a^{IS}(T, v^{IS}) = - \int_{+\infty}^v \left( \Pi - \frac{RT}{v} \right) \frac{dv}{v} + RT \ln \left( \frac{v^{IS}}{v} \right) \quad (6.1)$$

where  $T$  is the absolute temperature,  $v$  is the molar volume,  $\Pi$  is the osmotic pressure,  $R$  is the gas constant (which is basically the multiplication of Avogadro's number by the Boltzmann constant) and  $v^{IS}$  is the molar volume of an ideal solution at the same pressure and temperature.

In terms of the packing fraction,  $\eta$ , one would have:

$$\frac{a^R(T, \eta)}{RT} = \int_0^\eta (Z - 1) \frac{d\eta}{\eta} + \ln \left( \frac{\eta}{\eta^{IS}} \right) \quad (6.2)$$

Replacing Equation (5.42) on Equation (6.2), one has:

$$\frac{a^R(T, \eta)}{RT} = \ln \left( \frac{\eta}{\eta^{IG}} \right) + \frac{4\eta - 3\eta^2}{(1 - \eta)^2} - 4 \frac{\varepsilon^0}{k_B T} \eta + \frac{4}{3} \frac{\varepsilon'}{k_B T} \eta^2 \quad (6.3)$$

Since the molar Gibbs free energy,  $g$ , is given by the following Legendre

transform of the molar Helmholtz free energy,  $a$ :

$$g^R = a^R + \Pi v^R \quad (6.4)$$

And that for a single component the molar chemical potential,  $\mu$ , equals the molar Gibbs free energy. Therefore, one has that the residual molar chemical potential, which is related to the natural logarithm of the fugacity coefficient,  $\phi$ , may be calculated as:

$$\frac{\mu^R(T, \eta)}{RT} = \ln \phi(T, \eta) = \frac{a^R(T, \eta)}{RT} + Z - 1 \quad (6.5)$$

Substituting Equations (5.36) and (6.3) on Equation (6.5), one has that the natural logarithm of the fugacity coefficient is:

$$\ln \phi(T, \eta) = \ln \left( \frac{\eta}{\eta^{IS}} \right) + \frac{1 + \eta + \eta^2 - \eta^3}{(1 - \eta)^3} + \frac{4\eta - 3\eta^2}{(1 - \eta)^2} - 1 - 8 \frac{\varepsilon^0}{k_B T} \eta + 4 \frac{\varepsilon'}{k_B T} \eta^2 \quad (6.6)$$

Obviously, if  $\varepsilon' = 0$ , van der Waals-Carnahan-Starling fugacity coefficient is recovered from Equation (6.6).

### 6.3 Condition of equilibrium

The condition of liquid-liquid equilibrium of a single component is (PRAUSNITZ; LICHTENTHALER; AZEVEDO, 1999):

$$T^I = T^{II} \quad (6.7)$$

$$\Pi^I = \Pi^{II} \quad (6.8)$$

$$\mu^I = \mu^{II} \quad (6.9)$$

where  $T^I$  is the absolute temperature of phase  $I$ ,  $\Pi^I$  is the osmotic pressure in phase  $I$  and  $\mu^I$  is the chemical potential of phase  $I$ . One may replace the equality of chemical potentials by the equality of fugacities.

To evaluate the densities of each phase at a particular temperature, the equality of pressures was calculated using the Newton-Raphson algorithm and the equality of chemical potentials using the Maxwell area rule, *id est*:

$$\Pi_e = \frac{1}{v^{II} - v^I} \int_{v^I}^{v^{II}} \Pi(v) dv \quad (6.10)$$

where  $\Pi_e$  is the equilibrium osmotic pressure,  $v^I$  is the specific volume of phase  $I$  and  $\Pi(v)$  is the osmotic pressure given by the equation of state (5.42). Nevertheless, the chemical potential is the core of the algorithm to evaluate the new guess for pressure in the iterative process. The code to do this calculation, written in FORTRAN77, is presented on Appendix E.

## 6.4 Coexistence curve through molecular simulation

One way to check whether the calculations are consistent or not is to compare with molecular simulation predictions. For simple potentials this has been already done, however, for Lennard-Jones potential the densities close to the critical temperature are rather scarce. Thus, Monte Carlo simulation has been applied to get points nearer to the critical point.

There are at least two main approaches to calculate a phase envelope using molecular simulation. One is based on the Gibbs ensemble with two boxes (FRENKEL; SMIT, 2002) and its variants and the other is based on the canonical ensemble with a single box (CHAPELA et al., 1977).

Canonical Monte Carlo simulations were carried out to predict the vapor-liquid coexistence line for Lennard-Jones fluid:

$$U(r) = \begin{cases} +\infty & \text{if } r \leq \sigma \\ 4\epsilon \left[ \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] & \text{if } r > \sigma \end{cases} \quad (6.11)$$

where  $U(r)$  is the interaction potential,  $\sigma$  is the particle diameter and  $\varepsilon$  is the potential depth.

Simulations with 2048 particles were performed in a parallelepiped cell of dimensions  $L_x = L_y = 12\sigma$  and  $L_z = 40\sigma$ . Initially the particles were placed in a face centered cubic (FCC) arrangement at the right side of the box and the rest of the box was in vacuum. Figure 18 illustrates this initial configuration. Each translational trial was attempted following the Metropolis method in canonical ensemble and respecting periodic boundary conditions (FRENKEL; SMIT, 2002); the acceptance ratio was set to 0.45. The interaction potential was truncated at  $3.0\sigma$ . For equilibration,  $8.192 \times 10^7$  steps were performed. For the production stage, the same amount of cycles was used. Space coordinates were stored every 2048 steps. The codes to generate the initial configuration and to run the Monte Carlo simulation are presented in Appendix C. Evidently, this calculation could have been done using Gibbs Ensemble Monte Carlo simulations.

The local particle density along  $z$ -direction is the average of the number of particles in a slab of volume  $\delta V = L_x L_y \delta z$ :

$$\rho(z) = \frac{\langle N(z) \rangle}{L_x L_y \delta z} \quad (6.12)$$

The code to calculate the local particle density along  $z$ -direction is presented in Appendix C; the value of  $\delta z$  was set to be  $0.05\sigma$ . The densities from each production step were averaged and, then, the average density was fitted to the following tangent hyperbolic function (CHAPELA et al., 1977):

$$\rho(z) = \frac{1}{2}(\rho_V + \rho_L) - \frac{1}{2}(\rho_V - \rho_L) \tanh \left[ \frac{2(z - z_0)}{d} \right] \quad (6.13)$$

where  $\rho_V$  is the vapor density,  $\rho_L$  is the liquid density,  $z_0$  is the position of the Gibbs dividing surface and  $d$  is a measure of the thickness of the interface (CHAPELA et al., 1977; OREA; DUDA; ALEJANDRE, 2003). The position of

the liquid and vapor densities are exchangeable on this equation depending on which side the dense phase is placed.

Since it is quite complicated to simulate the system at the critical point, the rectilinear diameter law was used to estimate the critical properties (KATSONIS; BRANDON; VEKILOV, 2006):

$$\frac{\rho_L + \rho_V}{2} = \rho_c + A(T_c - T) \quad (6.14)$$

$$\rho_L - \rho_V = B(T_c - T)^{0.325} \quad (6.15)$$

where  $A$  and  $B$  are fitting parameters,  $\rho_c$  is the critical density and  $T_c$  is the critical temperature. Table 2 presents the comparison between the calculated results and the values reported in the literature. Although the critical temperature agrees well to the reported values in the literature, the critical density is a little bit lower. This fact is probably due to the fitting of the local particle density.

Table 2: Values of Lennard-Jones critical temperature and density.

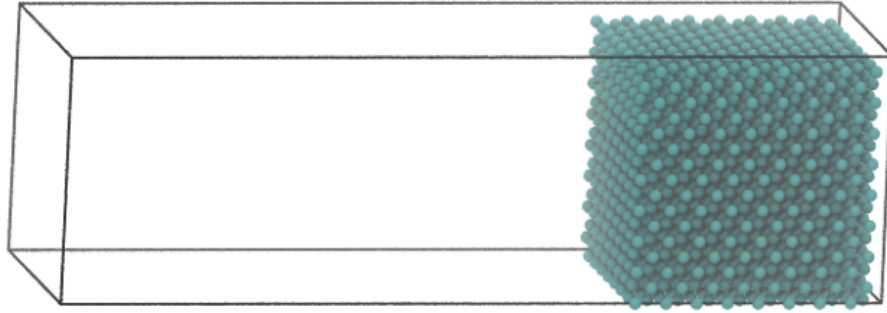
Reference	$T_c^*$	$\rho_c^*$
This work	$1.291 \pm 0.002$	$0.256 \pm 0.009$
(CAILLOL, 1998)	$1.326 \pm 0.002$	$0.316 \pm 0.002$
NIST (SHEN; SIDERIUS; KRELKELBERG, 2015)	1.291	0.317

## 6.5 Fluid-fluid coexistence curve

Figure 19 presents the calculated liquid-liquid coexistence curves for bovine serum albumin in salt aqueous solutions at the isoelectric point and 23°C with three different salts (sodium chloride, sodium nitrate and sodium sulfate) predicted with Equations (5.36) and (6.6) using parameters from Table 1. The criti-



Initial vacuum plus FCC configuration



After  $1.6 \times 10^8$  NVT MC steps of LJ particles

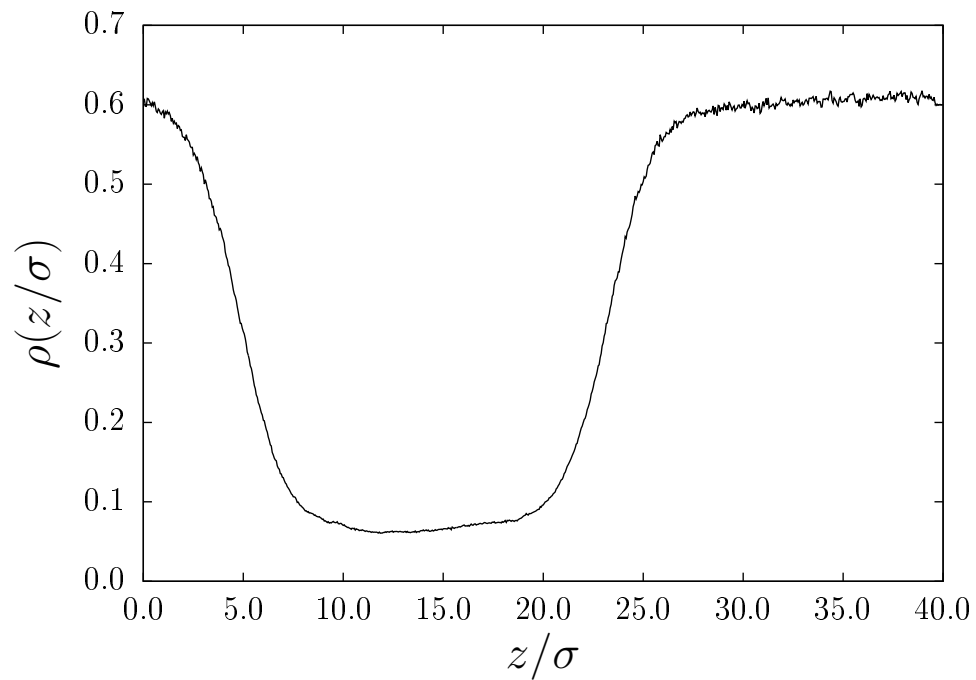
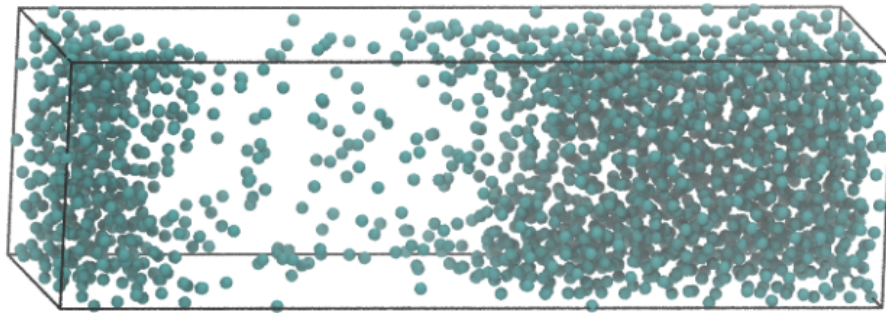


Figure 18: Illustrative example of the calculation of equilibrium densities using a single canonical Monte Carlo simulations with 2048 Lennard-Jones particles. The first image represents the initial state with a dense face centered cubic (FCC) configuration on the very right side and vacuum in the rest of the parallelepiped. The second image represents the final configuration exhibiting two phases in equilibrium. The graph at the bottom is a plot of the variation of local particle density as function of direction  $z$ .

cal temperature is much lower than the freezing point of water at the atmospheric pressure. Therefore, these results are in agreement with which is experimentally observed: there is no liquid-liquid separation for this protein in such solutions at the pH value equal to the isoelectric point. Thus, the predictions of the model are essentially unphysical; however, they may provide interesting information that may be extrapolated to other conditions in which there is a phase separation.

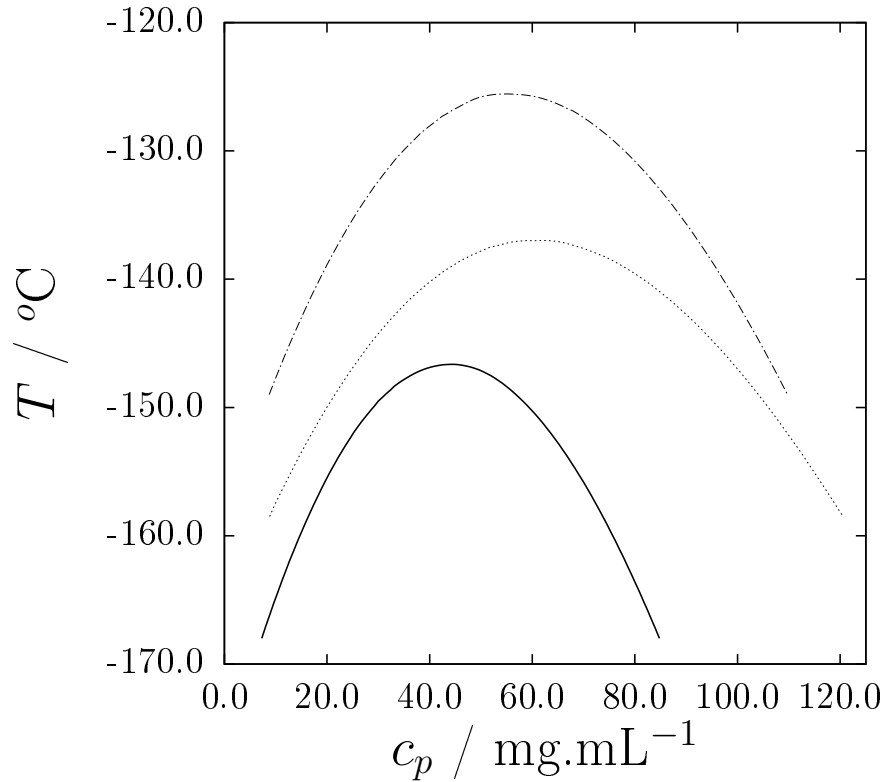


Figure 19: Liquid-liquid coexistence curve for bovine serum albumin aqueous solution at pH=4.9 and 23°C using Equations (5.36) and (6.6) with parameters given at Table 1. Continuous line, 1.0 mol.kg<sup>-1</sup> NaNO<sub>3</sub>. Dotted line, 1.0 mol.kg<sup>-1</sup> Na<sub>2</sub>SO<sub>4</sub>. Dashed-dotted line, 1.0 mol.kg<sup>-1</sup> NaCl.

Table 3 shows the values of critical points for these coexistence lines.

Table 3: Critical properties for liquid-liquid equilibrium lines for BSA in salt aqueous solutions at the isoelectric point.

Salt	$T_c / \text{K}$	$v_c / \text{m}^3 \cdot \text{mol}^{-1}$	$p_c / \text{Pa}$	$Z_c$
NaCl	147.58	1.192	345.73	0.336
NaNO <sub>3</sub>	126.51	1.517	231.97	0.334
Na <sub>2</sub> SO <sub>4</sub>	136.15	1.098	347.28	0.337

The critical properties seem to be salt dependent. This is certainly an important observation in terms of the effect of the ion specificity. Nevertheless, the critical compressibility factors are very similar, and therefore, it is to be expected that, besides the ion specificity, the law of correspondence states is roughly obeyed.

If one plots the reduced temperature as a function of the reduced density, one may see that all the curves are overlapped. Figure 20 presents such plot. Near the critical point, the law of correspondence states is strictly obeyed, however, at low temperature, one may observe a small divergence due to the ion specificity. At this same plot, one interesting aspect rises. Plotting the results of molecular simulation, one may note that for bovine serum albumin the coexistence line is better represented by a not so short-range attractive potential as it was observed for  $\gamma$ -crystallin and lysozyme (DUDA, 2009). This might imply that the range of attractive interaction may depend upon the protein being considered.

The whole discussion seems to be “nonsense”, since this phase separation is unphysical, and no phase separation is observed at the isoelectric point at all. Nevertheless, for pH values below the isoelectric point, a phase separation is indeed observed.

It is quite hard to find a way to show this separation. One way to see it is from scattering data. Using Guinier’s approximation, one may observe the formation of aggregates. At the limit of  $q \rightarrow 0$ , Guinier’s approximation states

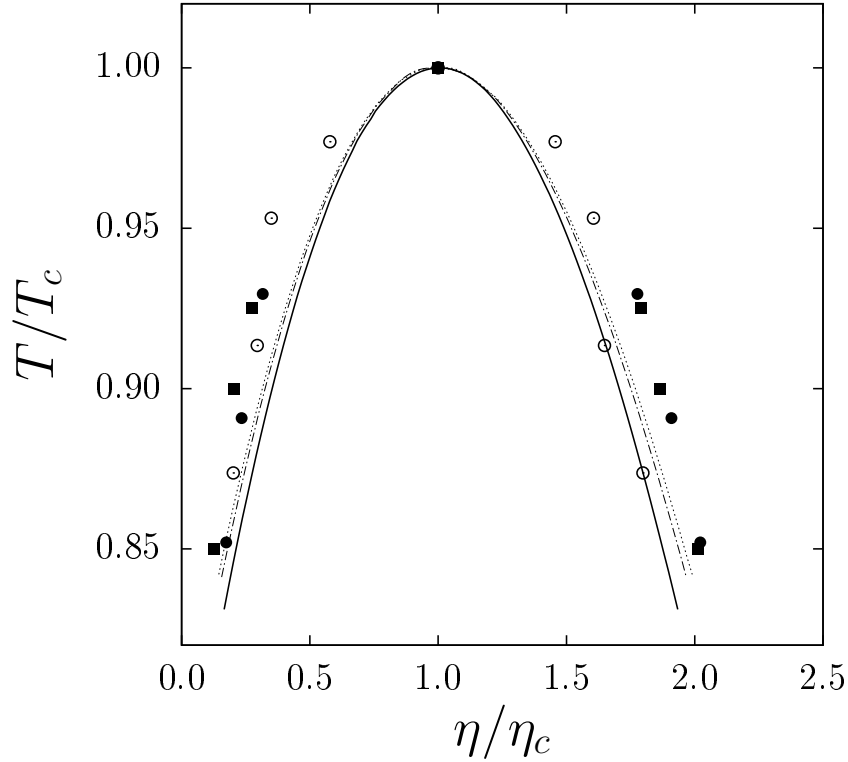


Figure 20: Liquid-liquid coexistence curve for BSA aqueous solution at pH=4.9 and 23°C using Equations (5.36) and (6.6) with parameters given at Table 1. Continuous line, 1.0 mol·kg<sup>-1</sup> NaNO<sub>3</sub>. Dotted line, 1.0 mol·kg<sup>-1</sup> Na<sub>2</sub>SO<sub>4</sub>. Dashed-dotted line, 1.0 mol·kg<sup>-1</sup> NaCl. Open circles, Lennard-Jones liquid-vapor equilibrium calculated in this work. Closed circles, Lennard-Jones liquid-vapor equilibrium calculated using grand-canonical transition-matrix Monte Carlo and histogram re-weighting obtained from NIST (SHEN; SIDERIUS; KRELKELBERG, 2015). Closed squares, Square-well liquid-vapor equilibrium calculated for  $\lambda = 1.375$  (LÓPEZ-RENDÓN; REYES; OREA, 2006).

(GLATTER; KRATKY, 1982; GUINIER, 1994):

$$I(q) = I_0 \exp\left(\frac{-q^2 R_g^2}{3}\right) \quad (6.16)$$

where  $I(q)$  is the scattering intensity,  $I(0)$  is the scattering intensity at  $q = 0$ ,  $q$  is the wavevector modulus and  $R_g$  is the radius of gyration.

If a system follows Guinier's approximation, the plot of the natural logarithm of the scattering intensity against the square modulus of the wavevector must be a straight line. Nevertheless, if there is a non-linear increasing in the scattering intensity, this means there are some aggregates in the sample, which causes the

augment of the radius of gyration. Figure 21 presents the scattering intensities of bovine serum albumin (BSA) 1.0 mol·kg<sup>-1</sup> NaCl aqueous solution at pH = 3.4 and 23°C for two different BSA concentrations<sup>1</sup>. One may visualize that for highly diluted BSA system, there is a non-negligible deviation from the Guinier's approximation, which implicates that there is a formation of aggregates in the sample. And indeed a macroscopic phase separation was observed during the experiment.

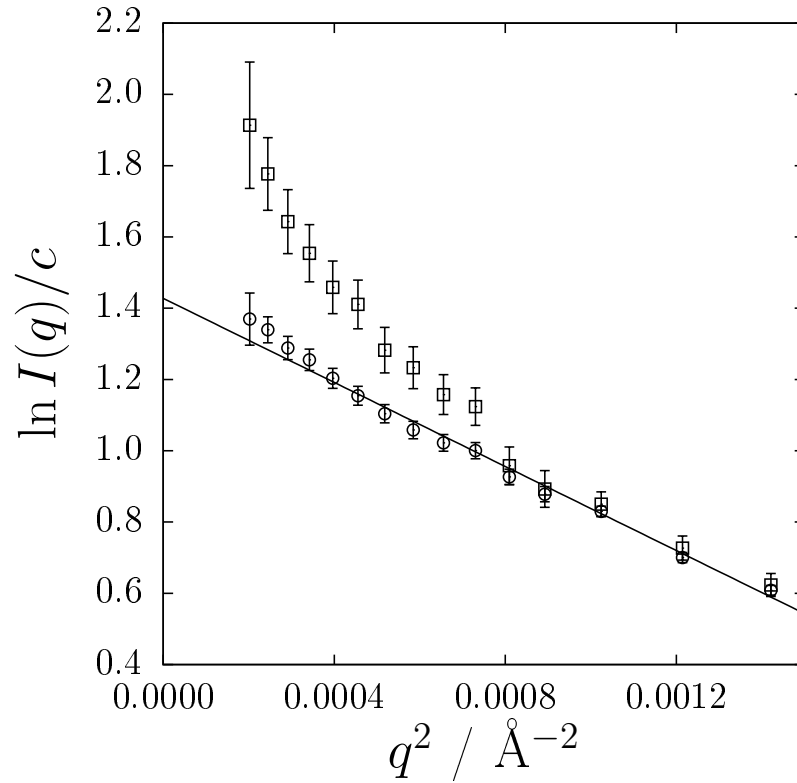


Figure 21: Guinier plot for small-angle X-ray scattering of BSA 1.0 mol·kg<sup>-1</sup> NaCl aqueous solution at pH=3.4 and 23°C. Open circles, experimental data for BSA concentration of 25.9 mg·mL<sup>-1</sup>. Open squares, experimental data for BSA concentration of 5.6 mg·mL<sup>-1</sup>. Continuous line, Guinier's approximation to BSA concentration of 25.9 mg·mL<sup>-1</sup>.

Figure 22 shows the liquid-liquid coexistence line predicted with our model for bovine serum albumin 1.0 mol·kg<sup>-1</sup> NaCl aqueous solution at pH = 3.4 as well as the spinodal curve. The critical temperature is still below the freezing temper-

<sup>1</sup>The experimental procedures used in these experiments were the same as described in Section 5.2

ature of water, however, the estimated critical temperature is much higher than those found at the isoelectric point. Moreover, the range of protein concentration in which there is a phase separation seems to be different comparing Figures 21 and 22. This means that our model is only able to qualitatively predict the phase separation. What is remarkable though is the effect of pH on the protein phase diagram. This could be partially explained by protein changes in conformation. At the isoelectric point, bovine serum albumin finds itself in a configuration called “Normal”, however, at  $\text{pH} = 4.3$ , this protein changes its configuration to one called “Fast” and at  $\text{pH} = 2.7$  to one called “Extended” (PETERS, 1995). These two acid conformations are much more unfolded than the “Normal”. This could be responsible for exposing residues that in the “Normal” conformation are hidden and this exposure changes completely the interactions among the proteins in solution.

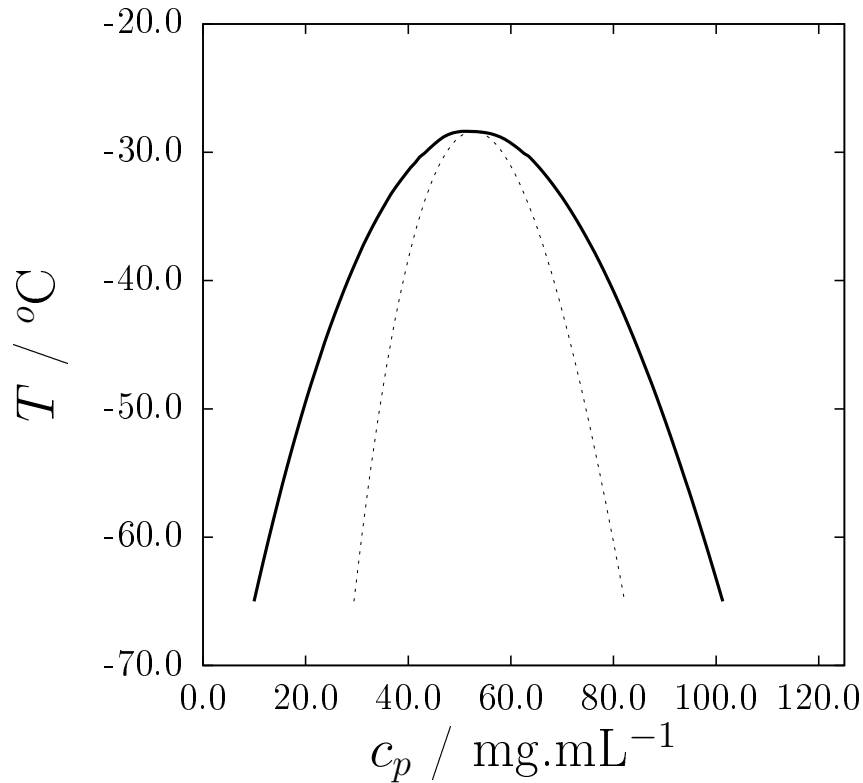


Figure 22: Liquid-liquid coexistence curve for BSA  $1.0 \text{ mol}\cdot\text{kg}^{-1}$  NaCl aqueous solution at  $\text{pH}=3.4$  and  $23^\circ\text{C}$  using Equations (5.36) and (6.6) with  $\varepsilon_0/k_B T = 4.03$  and  $\varepsilon'/k_B T = 21.19$ . Continuous line, binodal curve. Dashed line, spinodal curve.

## 7 OTHER RELATED QUESTIONS

### 7.1 Introduction

The three developments that are shown in this chapter may appear to be a huge detour from the objective of this thesis as explained in the first chapter. The first one is due to the formulation of a coarse-grained model for sodium sulfate aqueous solution for which there is no aggregation of the salt in a concentration below the experimental solubility. The second one is a critic review on the idea of an universal reference line for Walden plot. This review came from the investigation of dynamic properties with the coarse-grained model for sodium sulfate aqueous solution. And finally, the third result is a coarse-grained force field for bovine serum albumin in aqueous solution. This last development is presented here as a new strategy to evaluate the scattering form factor of proteins.

### 7.2 Sodium sulfate unphysical association

#### 7.2.1 Introduction

The relevance of electrolyte solutions for mankind and for nature itself is truly remarkable. Nevertheless, even though ionic solutions are of great importance in physical chemistry as well as in biology systems, approaches to treat such systems are still under development. This is probably one of the most difficult problems in Thermodynamics: the theory of electrolytes. It involves all the subtle questions of

nonelectrolyte systems and, additionally, all the issues related to the electrostatic interactions.

From the point of view of molecular simulation, ionic systems also pose challenges. The first one is due to the long-range behavior of the Coulomb interaction potential, which requires specific strategies of calculation (such as Ewald summation) in a finite box (ALLEN; TILDESLEY, 1987). Other complications include polarizable force fields, which are still very scarcely applied when compared to non-polarizable force fields due to large required computational effort.

The unphysical association among sodium and sulfate ions in aqueous solutions of sodium sulfate is predicted by non-polarizable atomistic force fields and has already been observed in the literature. If polarizable force fields are used, mainly for water, one can get more consistent structures for such solutions (WERNESSEN; JUNGWIRTH, 2010). Although this solves the issue for many cases, the large computational cost of polarizable force fields makes this approach impractical for modeling systems containing large proteins in aqueous salt solutions. Therefore, there is still a need for a simple and reliable model to simulate salt solutions.

Figure 23 shows a snapshot of a 2.0 ns NPT molecular dynamic simulation for an atomistic sodium sulfate force field CHARMM19 (REIHER-III, 1985) with 270 sulfate ions and 540 sodium ions in 10000 SPC/E water (BERENDSEN; GRIGERA; STRAATSMA, 1987) (which corresponds to  $1.5 \text{ mol}\cdot\text{kg}^{-1}$  of sodium sulfate) at 303.15 K. Considering periodic boundary conditions, one may see, at least qualitatively, that there is a clear association of the sodium sulfate salt. One must bear in mind that the experimental solubility of sodium sulfate in water at this same temperature is  $2.03 \text{ mol}\cdot\text{kg}^{-1}$  (OKORAFOR, 1999). Thus, this atomistic force field exhibits an aggregation of the salt with less than 74% of its experimental solubility.



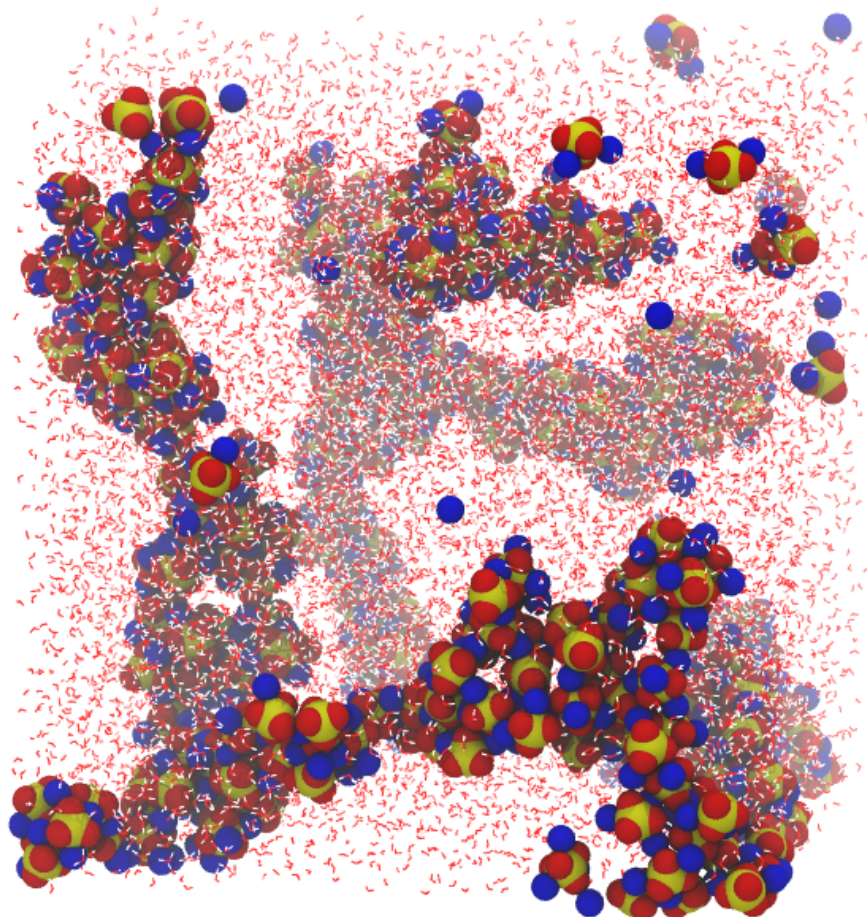


Figure 23: Snapshot of a 2.0 ns NPT molecular dynamic simulation for 270 CHARMM19 sodium sulfate particles in 10000 SPC/E water particles at 303.15 K. Oxygens are represented in red, sodium ions in blue and sulfur ions in yellow. Water molecules are represented as lines.

From a perspective of bioprocesses, this is a problem, since for most of the industrial applications large amounts of salts are used to precipitate proteins. Therefore, a model which avoids this unphysical association is desired.

### 7.2.2 Force Field

Recently, Shinoda *et al.* (SHINODA; DEVANE; KLEIN, 2007) proposed a new coarse-grained force field for water. Each bead represents three water molecules and has no charge. This model is being successfully applied to a variety of systems which requires large computational effort. The force field developed here is inspired by Shinoda *et al.* (SHINODA; DEVANE; KLEIN, 2007) coarse-grained water model. To generate a similar force field for the salt, the coarse-grained sodium ion was considered as a single bead that accounts for one sodium ion and three water molecules as Shinoda *et al.* (SHINODA; DEVANE; KLEIN, 2011) had done before. For the sulfate ion, only a single bead was considered to account for the whole ion with no water molecules. Figure 24 shows such coarse-grained model.

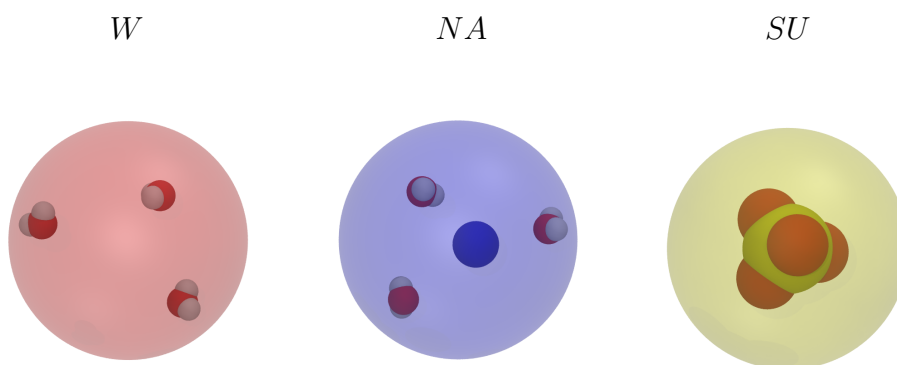


Figure 24: Representation of the coarse-grained model.  $W$  stands for the coarse-grain water model (SHINODA; DEVANE; KLEIN, 2007), a single sphere which accounts for three atomistic water molecules.  $NA$  stands for the coarse-grain sodium ion (SHINODA; DEVANE; KLEIN, 2011), a single sphere which accounts for atomistic sodium ion and three atomistic water molecules.  $SU$  stands for the coarse-grain sulfate ion, a single sphere which accounts for the atomistic sulfate ion. The diameters of  $W$ ,  $NA$  and  $SU$  are set to be equal.

The interparticle potential is considered as to have one electrostatic contribution given by a Coulomb potential and a contribution given by Mie potential.

Following the water model developed by Shinoda *et al.* (SHINODA; DEVANE; KLEIN, 2007), a Mie12-4 potential, instead of the traditional Lennard-Jones (Mie12-6), has been adopted to be keep the consistency. Thus, the interaction potential,  $V_{ij}$ , could be written as:

$$V_{ij} = \frac{q_i q_j}{4\pi\epsilon_0\epsilon_r r_{ij}} + \frac{3\sqrt{3}}{2}\epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^4 \right] \quad (7.1)$$

where  $q_i$  is the charge of particle  $i$ ,  $\epsilon_0$  is the vacuum permittivity,  $\epsilon_r$  is the dielectric constant of the medium,  $\epsilon_{ij}$  is the attractive potential depth between particles  $i$  and  $j$ ,  $\sigma_{ij}$  is the distance between particles  $i$  and  $j$  for which the Mie contribution is null, and the cross terms are given by the following rule:

$$\sigma_{W,W} = \sigma_{S,S} = \sigma_{N,N} = 0.4371 \text{ nm} \quad (7.2)$$

$$\epsilon_{W,W} = \epsilon_{S,S} = \epsilon_{N,N} = 3.7447 \text{ kJ}\cdot\text{mol}^{-1} \quad (7.3)$$

$$\epsilon_{W,S} = \frac{1}{\alpha} \sqrt{\epsilon_{W,W} \epsilon_{S,S}} \quad (7.4)$$

$$\epsilon_{S,N} = \alpha \sqrt{\epsilon_{S,S} \epsilon_{N,N}} \quad (7.5)$$

where  $\alpha$  is an association parameter. If  $\alpha < 1$ , the attractive interaction between sulfate and sodium is decreased whereas the hydration of sulfate is favored.

Since Shinoda *et al.* water model (SHINODA; DEVANE; KLEIN, 2007) has no partial charge, there are no electric dipoles, *ergo* one needs to consider the dielectric constant of the medium in the electrostatic contribution. For this purpose, an empirical equation (UEMATSU; FRANCK, 1980) to determine the pure water dielectric constant at different temperatures has been used. Table 4 presents the values of the dielectric constant of pure water as function of the temperature.

Table 4: Values of dielectric constant of pure water as a function of temperature according to an empirical equation (UEMATSU; FRANCK, 1980).

Temperature / K	Dielectric constant
288.15	81.30
293.15	79.84
298.15	78.41
303.15	77.01
308.15	75.63
318.15	72.95
328.15	70.38

### 7.2.3 Molecular dynamic details

All simulations were performed using GROMACS 4.5.5 software package (BERENDSEN; van-der-SPOEL; van-DRUNEN, 1995). The Leap Frog algorithm was used to integrate the equations of motion with periodic boundary conditions and 2 fs timestep. The Nosé-Hoover thermostat (NOSÉ, 1984; HOOVER, 1985) was used. For NPT calculations, the isotropic Parrinello-Rahman barostat (PARRINELLO; RAHMAN, 1981) was used to keep the pressure at 1 bar (isothermal compressibility of  $4.5 \times 10^{-5} \text{ bar}^{-1}$ ). Long-range as well as short-range interactions were both treated with cutoff radii of 1.5 nm. Positions and velocities were stored at each 1 ps. Electrostatic potential was calculated using Particle Mesh Ewald method (DARDEN; YORK; PEDERSEN, 1993).

#### 7.2.3.1 Parameterization

For the parameterization of the association parameter,  $\alpha$ , NPT ensemble was used with  $T = 298.15 \text{ K}$  and  $p = 1 \text{ bar}$ . Runs of 2 ns were carried out. The first 0.5 ns was used for equilibration and the last 1.5 ns for production. Table 5

sumarizes the number of particles for each sodium sulfate concentration.

Table 5: Number of particles for each sodium sulfate concentration.

Molality of Na <sub>2</sub> SO <sub>4</sub> (mol/kg)	<i>W</i>	<i>NA</i>	<i>SU</i>
0.050	4972	28	14
0.100	4946	54	27
0.500	4730	270	135
0.700	4622	378	189
1.000	4460	540	270
1.200	4352	648	324
1.500	4190	810	405

### 7.2.3.2 Density

To evaluate densities, we carried out 3 ns NPT simulations at seven different temperatures as shown on Table 4 for each sodium sulfate established on Table 5. The first 0.5 ns was used for equilibration and the last 2.5 ns for production.

### 7.2.3.3 Dynamic properties

Dynamic properties such as viscosity and ionic conductivity were calculated within Green-Kubo framework (KUBO, 1957), *i. e.*, one dynamic property,  $\lambda_\Phi$ , may be calculated by the autocorrelation function of the fluxes,  $\vec{j}_\Phi$ , of the entity  $\Phi$ :

$$\lambda_\Phi = \xi \int_0^{+\infty} \langle \vec{j}_\Phi(t+t_0) \cdot \vec{j}_\Phi(t_0) \rangle dt \quad (7.6)$$

where  $\xi$  is a constant.

Thus, if  $\Phi$  is momentum,  $\lambda_\Phi$  is viscosity,  $\vec{j}_\Phi$  is the pressure tensor:

$$\eta = \frac{V}{k_B T} \int_0^{+\infty} \langle P_{\alpha\beta}(t_0+t) \cdot P_{\alpha\beta}(t_0) \rangle dt \quad (7.7)$$

where  $V$  is the volume,  $k_B$  is the Boltzmann constant,  $T$  is the absolute temperature,  $P_{\alpha\beta}$  is the pressure tensor. To improve the statistics, the average over the six independent terms of the pressure tensor were used:  $P_{xy}$ ,  $P_{yz}$ ,  $P_{zx}$ ,  $0.5(P_{xx} - P_{yy})$ ,  $0.5(P_{yy} - P_{zz})$  and  $0.5(P_{xx} - P_{zz})$  (REY-CASTRO; VEGA, 2006). Each element of the pressure tensor was calculated through the following equation:

$$P_{\alpha\beta} = \frac{1}{V} \left( \sum_j m_j v_{\alpha j} v_{\beta j} + \frac{1}{2} \sum_{i \neq j} r_{\alpha ij} f_{\beta ij} \right) \quad (7.8)$$

where  $m_j$  is the mass of particle  $j$ ,  $v_{\alpha j}$  is the  $\alpha$  component of the velocity of particle  $j$ ,  $r_{\alpha ij}$  is the  $\alpha$  component of the relative position between particles  $i$  and  $j$ , and  $f_{\beta ij}$  is the  $\beta$  component of the force acting between  $i$  and  $j$ .

For the ionic conductivity, one has the following Green-Kubo relation:

$$\sigma = \frac{1}{k_B T V} \int_0^{+\infty} \langle \vec{J}(t_0 + t) \cdot \vec{J}(t_0) \rangle dt \quad (7.9)$$

where:

$$\vec{J} = \sum_j q_j \vec{v}_j \quad (7.10)$$

where  $q_j$  is the electric charge of particle  $j$  and  $\vec{v}_j$  is the velocity of particle  $j$ .

Appendix F presents a code to evaluate the viscosity within Green-Kubo framework from the pressure tensor simulated data. To check the codes, simulations to a well known potential for molten sodium chloride: Born-Huggins-Mayer-Tosi-Fumi (MAYER, 1933; TOSI; FUMI, 1964) have been run.

Figure 46, in Appendix I, illustrates the initial FCC arrange and the final molten NaCl from equilibrium molecular dynamics simulations. Figures 47 and 48, in Appendix I, presents the viscosity (and the stress autocorrelation function) and the electric current autocorrelation function (and the ionic conductivity) for molten NaCl respectively. Table 6 presents the agreement between the calculated values and the values found in the literature.

Table 6: Viscosity and ionic conductivity of molten NaCl obtained from equilibrium molecular dynamics simulations using the Born-Huggins-Mayer-Tosi-Fumi force field and the Green-Kubo relations.

Reference	$T$ / K	$\eta$ / mPa.s	$T$ / K	$\sigma$ / S.cm <sup>-1</sup>
This work	1100	$1.025 \pm 0.004$	1300	$3.74 \pm 0.07$
(REY-CASTRO; VEGA, 2006)	1126	$1.11 \pm 0.01$	1305	$3.6 \pm 0.1$

## 7.2.4 Results and discussions

### 7.2.4.1 Parameterization of $\alpha$

To find the optimum value of  $\alpha$ , several simulations were run with different values of  $\alpha$  and the deviation,  $\chi$ , of density between the simulated and the experimental data (ISONO, 1984) for seven different sodium sulfate molalities (0.05; 0.10; 0.50; 0.70; 1.00; 1.20 and 1.50 mol.kg<sup>-1</sup>) in aqueous solutions at 298.15 K was calculated as:

$$\chi = \sqrt{\frac{1}{N} \sum_{j=1}^N (\rho_j^{EXP} - \rho_j^{SIM})^2} \quad (7.11)$$

where  $N$  is the number of experimental points,  $\rho_j^{EXP}$  is the experimental solution density at the  $j$  sodium sulfate molality and  $\rho_j^{SIM}$  is the simulated solution density at the  $j$  sodium sulfate molality.

As one can see in Figure 25, there is a value of  $\alpha$  which minimizes  $\chi$ . Each point on this plot represents the  $\chi$  value for a set of seven different sodium sulfate molalities. To find the optimum value for  $\alpha$ , the values of  $\chi$  were fitted by the following equation:

$$\chi(\alpha) = A_1 \cos(\omega_1 \alpha + \delta) e^{-\alpha/\tau} + A_2 \cos(\omega_2 \alpha) \quad (7.12)$$

where  $A_1$ ,  $\omega_1$ ,  $\delta$ ,  $\tau$ ,  $A_2$  and  $\omega_2$  are fitting parameters.

The optimum value of  $\alpha$  was found to be  $\alpha = 0.459$  using Newton-Raphson algorithm to find the  $\alpha$  for which  $\partial\chi(\alpha)/\partial\alpha = 0$ .

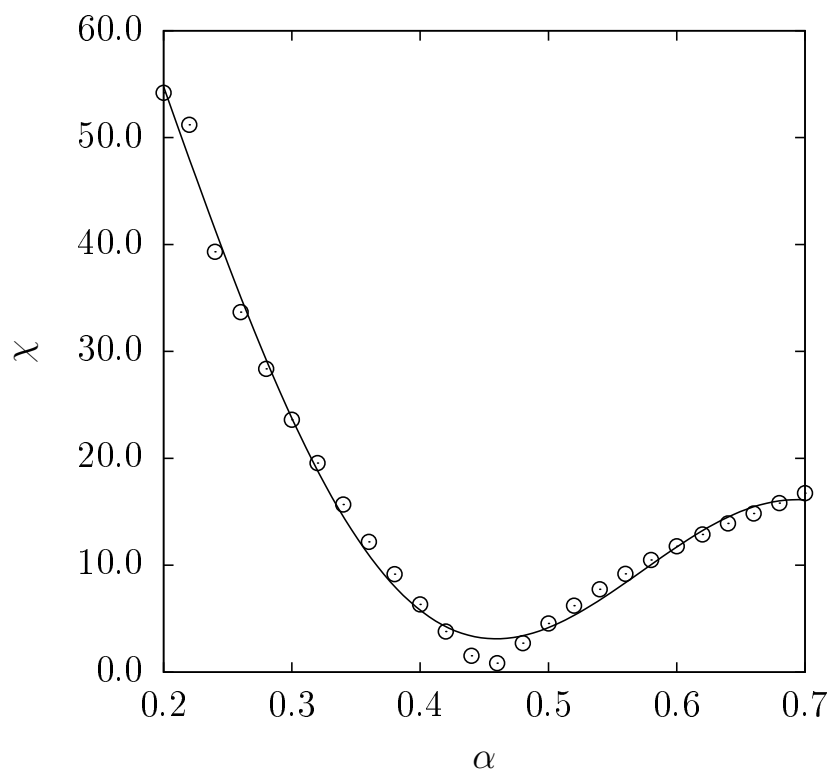


Figure 25: Deviation of density,  $\chi$ , between the simulated and experimental data (ISONO, 1984) for seven different sodium sulfate molalities (0.05; 0.10; 0.50; 0.70; 1.00; 1.20 and 1.50 mol·kg<sup>-1</sup>) in aqueous solution at 298.15 K as a function the association parameter  $\alpha$ . Open circles, true values of  $\chi$ . Continuous line, fitting equation (7.12).

#### 7.2.4.2 Thermodynamic properties

The first check for consistency is to consider the coordination number between two sulfates. For a system in which there is some sort of aggregation, it is to be expected a high peak of coordination number near the central molecule. The coordination number may be defined in terms of the radial distribution function as:

$$\frac{N(r)}{N_{random}} = \frac{3}{r^3} \int_0^r g(r') r'^2 dr' \quad (7.13)$$

where  $N(r)$  is the coordination number,  $N_{random}$  is the coordination number for a uniformly distributed set of particles ( $g(r) = 1$ ), and  $g(r)$  is the radial distribution function.



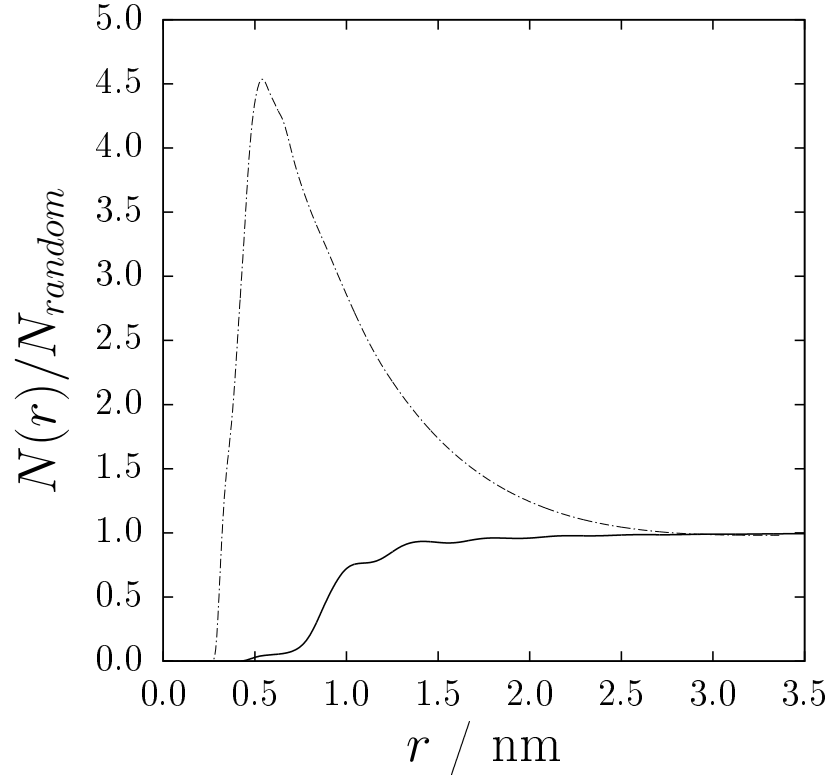


Figure 26: Coordination number for  $1.5 \text{ mol}\cdot\text{kg}^{-1}$  sodium sulfate aqueous solution at 303.15 K. Continuous line, our coarse-grained force field. Dotted-dashed line, non-polarizable atomistic force field.

Figure 26 presents the coordination number of two sulfates for our coarse-grained force field and for the atomistic CHARMM19 force field for sulfate coupled to SPC/E water with  $1.5 \text{ mol}\cdot\text{kg}^{-1}$  sodium sulfate aqueous solution. Even at this reasonably high temperature of 303.15 K, the atomistic force field predicts a quite high coordination number near the central sulfate ion. One must notice that the experimental solubility for sodium sulfate at this same temperature is  $2.03 \text{ mol}\cdot\text{kg}^{-1}$  (OKORAFOR, 1999). Therefore, the atomistic non-polarizable force field fails to predict a correct structure of sodium sulfate aqueous solutions. This is the very motivation underneath this development.

Figure 27 shows a snapshot of a molecular dynamic simulation of our coarse-grained model for  $1.5 \text{ mol}\cdot\text{kg}^{-1}$  sodium sulfate aqueous solution at 303.15 K. Differently from Figure 23, the sulfate ions, represented by yellow beads, are not

clustered at all.

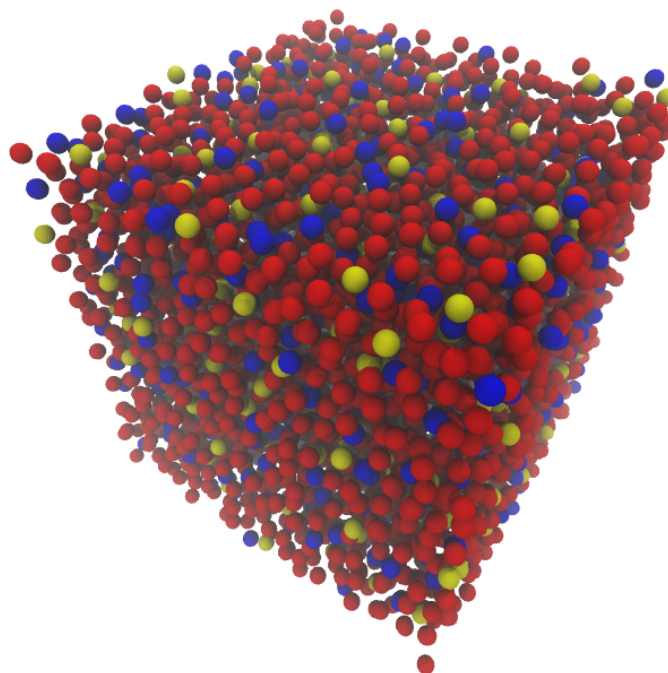


Figure 27: Snapshot of a molecular dynamic simulation of the proposed coarse-grained force field for  $1.5 \text{ mol}\cdot\text{kg}^{-1}$  sodium sulfate aqueous solutions at 303.15 K. Red beads represent water molecules, blue beads sodium ions and yellow beads sulfate ions.

Figure 28 presents the density results for sodium sulfate aqueous solution at various temperatures and salt concentrations. The coarse-grained force field proposed here is able to correlate very well the experimental data for density. The best correlation is found at 298.15 K, but this is not surprising since these data were used in the parameterization of the force field. Also, one may notice that, even though the results are quite good, the slope of density in relation to the temperature is always a little bit higher for the model than for the experimental data. This means the thermal expansion coefficient not well correlated.

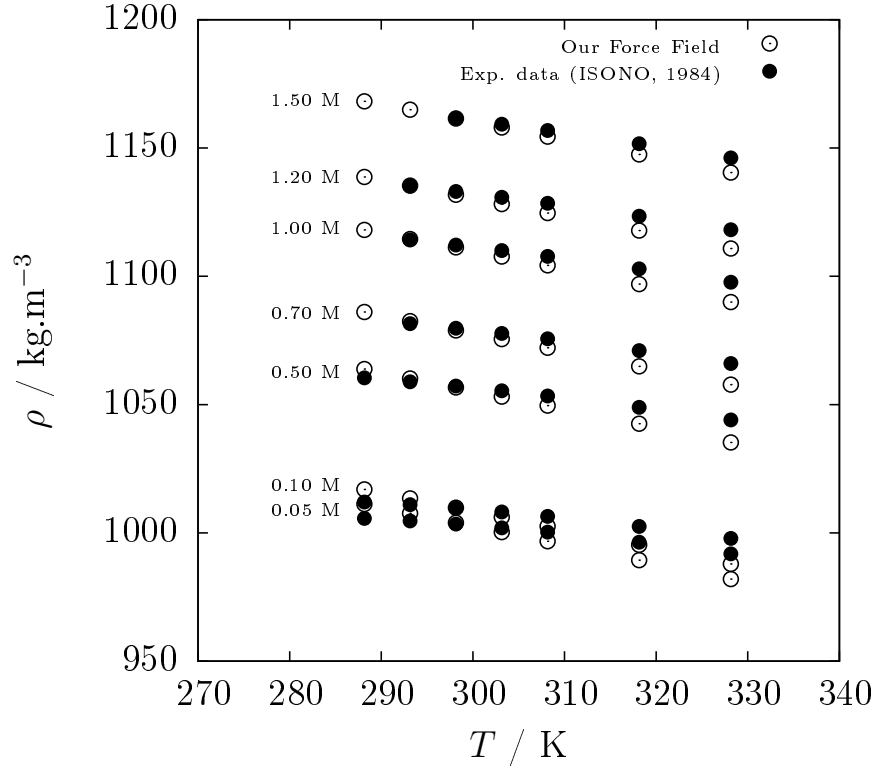


Figure 28: Density of sodium sulfate aqueous solution as a function of temperature and salt concentration. Open circles, results by molecular dynamic simulations using the proposed coarse-grained force field. Closed circles, experimental data (ISONO, 1984). The standard deviations of the simulated data are lower than the symbol size.

The thermal expansion coefficient,  $\alpha_v$ , is defined as:

$$\alpha_v = -\frac{1}{\rho} \left( \frac{\partial \rho}{\partial T} \right)_p \quad (7.14)$$

where  $\rho$  is density,  $T$  is the absolute temperature and  $p$  is the pressure.

One simple way to calculate this property is using the finite difference method with density data, although it is possible to do this calculation through fluctuation theory. Figure 29 presents the results for thermal expansion coefficient of sodium sulfate aqueous solutions using the finite difference method. Despite the order of magnitude is somehow in agreement, the coarse-grained force field fails to predict with accuracy the experimental data. It seems, however, impossible to escape from this disagreement since at the limit of salt free solutions there is a

quite significant difference between the simulated and the experimental thermal expansion coefficient. As the model for pure water fails to predict this property, any model for a salt based on such model for water fails as well.

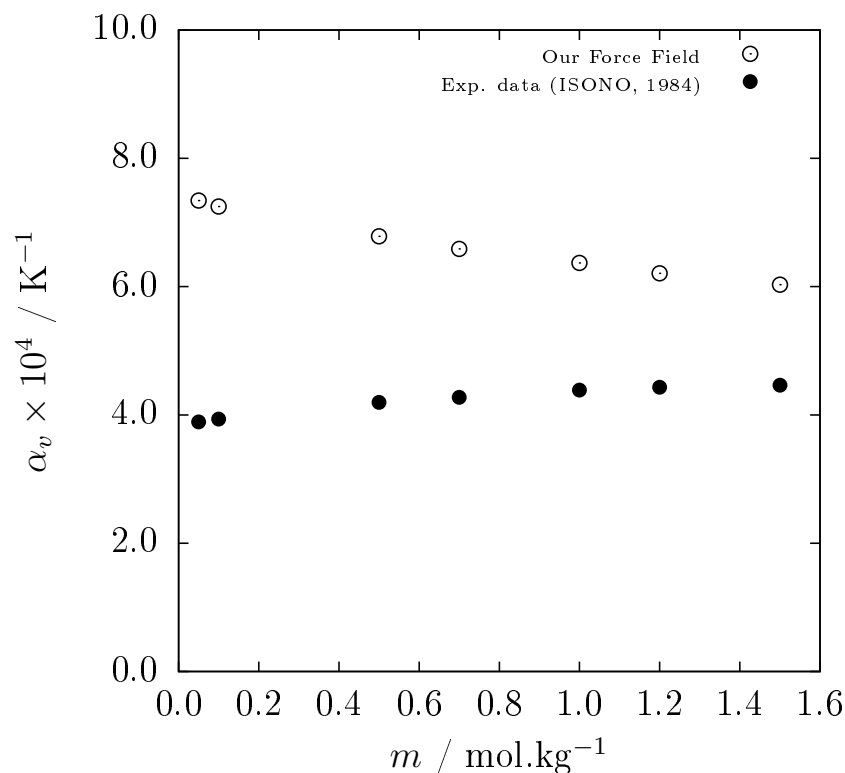


Figure 29: Thermal expansion coefficient of sodium sulfate aqueous solution as a function salt concentration. Open circles, results by finite difference method of molecular dynamic simulations using the proposed coarse-grained force field. Closed circles, results by finite difference method using the experimental data (ISONO, 1984).

### 7.2.4.3 Dynamic properties

Even though we can get no aggregation at all with our force field, this does not mean we are able to predict the correct structure of the real sodium sulfate aqueous solution. To investigate deeper this problem, dynamic properties such as viscosity and ionic conductivity were calculated.

Since there is a lot of issues concerning the convergence and the fluctuations in the autocorrelation function integrations, one ought to fit the results with a

proper empirical equation. For viscosity, the following equation was used (REY-CASTRO; VEGA, 2006):

$$\eta(t) = A\alpha\tau_1 (1 - e^{-t/\tau_1}) + A(1 - \alpha)\tau_2 (1 - e^{-t/\tau_2}) \quad (7.15)$$

where  $A$ ,  $\alpha$ ,  $\tau_1$  and  $\tau_2$  are fitting parameters.

For ionic conductivity, the used fitting equation was that proposed by Schröder and Steinhauser (SCHRÖDER; STEINHAUSER, 2009):

$$\sigma(t) = \sum_{k=1}^n \frac{A_k \tau_k}{1 + \tau_k^2 \omega_k^2} (e^{-t/\tau_k} [\tau_k \omega_k \sin(\omega_k t + \delta_k) - \cos(\omega_k t + \delta_k)] - [\tau_k \omega_k \sin(\delta_k) - \cos(\delta_k)]) \quad (7.16)$$

Examples of the divergence along simulation time in the calculation of dynamic properties are shown on Appendix J. The standard deviation of five different initial configurations simulated under the same conditions becomes very high as the time advances. Since the properties are defined in the limit of infinity time, the criterion to evaluate the property is somehow subjective.

Figure 30 presents the results for viscosity and Figure 31 for ionic conductivity for a diluted 0.05 mol·kg<sup>-1</sup> sodium sulfate aqueous solution at various temperatures. When compared to experimental data, the trend of such properties with temperature appears to be the same. At low temperatures, however, the model underestimates the viscosity and overestimates the conductivity. Nevertheless, even though the viscosity is well predicted, mainly at higher temperatures, the ionic conductivity is consistently higher than the experimental value. One possible explanation is that, avoiding any form of association, the model represents an ion solution in which the ionic mobility is higher than the real one, and therefore, the ionic conductivity is higher. It is remarkable though that such difference appears even at this quite diluted 0.05 mol·kg<sup>-1</sup> sodium sulfate aqueous solution.

The relation between the ionic conductivity and the viscosity may be stud-

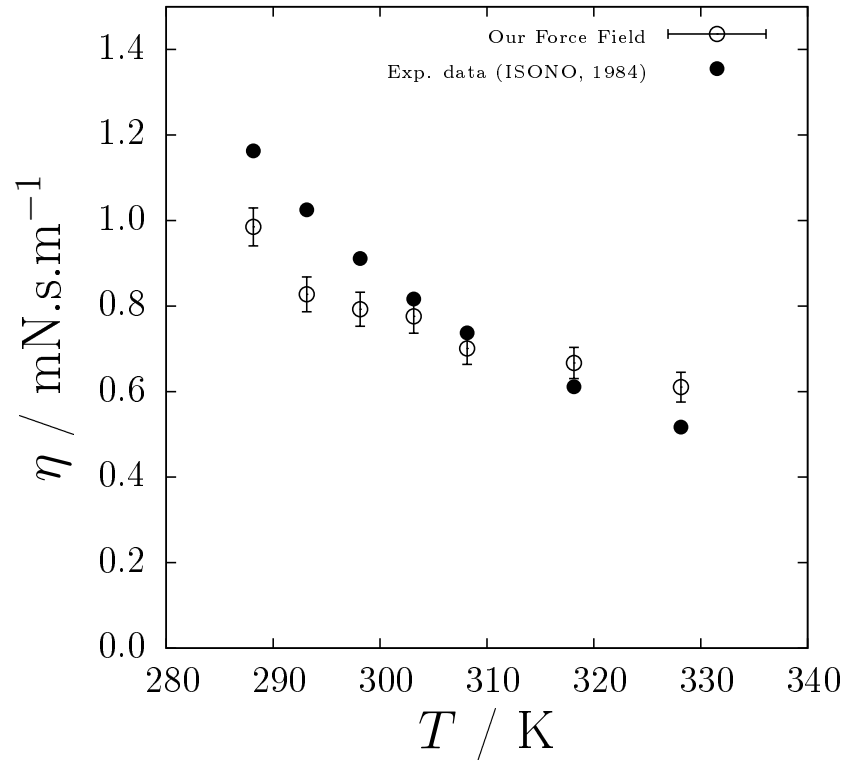


Figure 30: Viscosity of 0.05 mol·kg<sup>-1</sup> sodium sulfate aqueous solution as a function of temperature. Open circles, results by molecular dynamic simulations using the proposed coarse-grained force field. Closed circles, experimental data (ISONO, 1984).

ied using the so-called Walden plot. On the next section there will be a more consistent review on the basis of the Walden plot. By now it is only important to consider that for this plot there is a reference line. If the data lie below this line, it is believed that there is a sort of an ion pair formation or even some aggregation on the system. If the data lie above this line, it is believed that there is a decoupling between the structural relaxation and the dynamic properties (YAMAGUCHI et al., 2013).

Figure 32 presents the Walden plot for 0.05 mol·kg<sup>-1</sup> sodium sulfate aqueous solution. The comparison between our results and the experimental data would mean that avoiding any sort of association among the ions one gets a large deviation on Walden plot, even though the Walden rule is still obeyed.

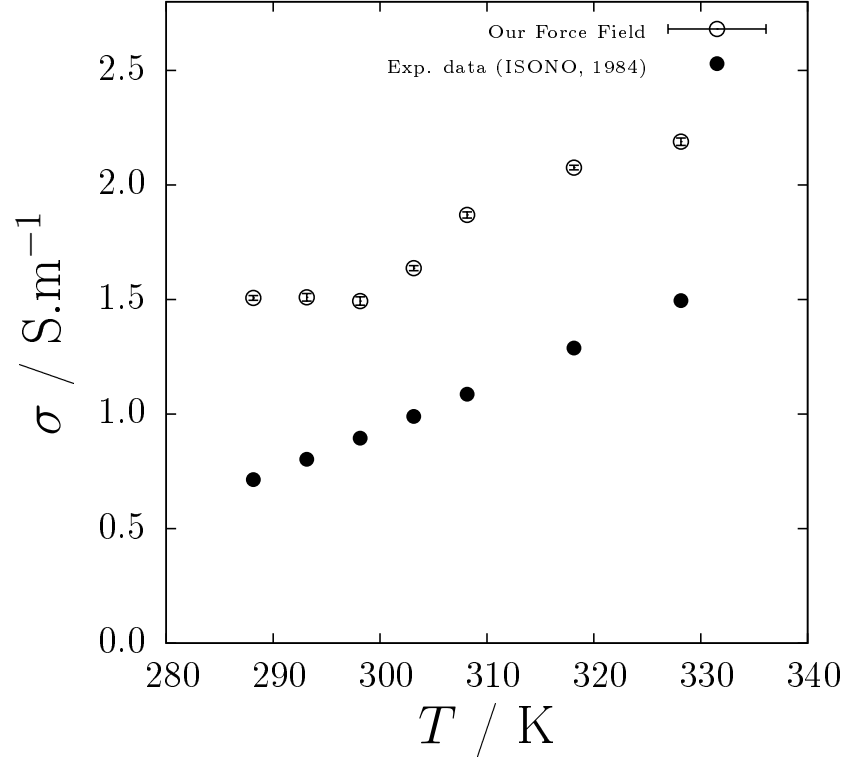


Figure 31: Ionic conductivity of 0.05 mol·kg<sup>-1</sup> sodium sulfate aqueous solution as a function of temperature. Open circles, results by molecular dynamic simulations using the proposed coarse-grained force field. Closed circles, experimental data (ISONO, 1984).

The fact that our results are placed above the ideal line and the experimental data are placed below the ideal line called our attention to the real meaning of this reference line. A critic review on the idea of an universal reference line for the Walden plot will be given in the next section.

In Figure 32 the Nernst-Einstein conductivities are also presented. They were obtained using the self-diffusion coefficients that were calculated by Einstein relation through the mean square displacement (FRENKEL; SMIT, 2002):

$$D = \frac{1}{6} \lim_{t \rightarrow +\infty} \frac{\partial \langle \vec{r}^2(t) \rangle}{\partial t} \quad (7.17)$$

The high agreement between Nernst-Einstein conductivity and the true conductivity data in Walden plot means that our coarse-grained model presents no

degree of correlation among the ions, and this is in agreement with the very formulation of the force field to avoid any association among the ions.

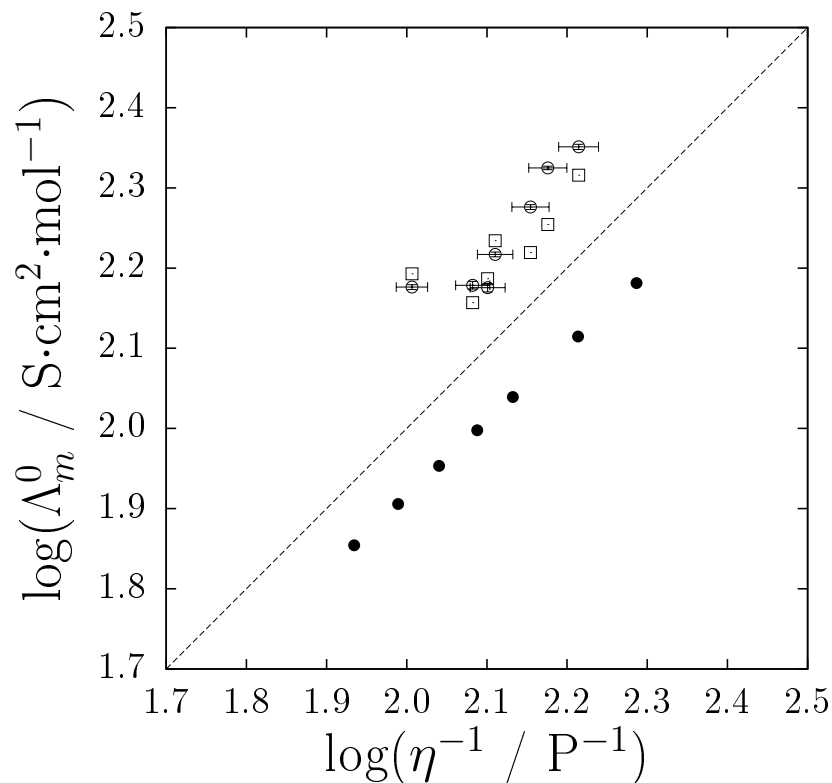


Figure 32: Walden plot for 0.05 mol·kg<sup>-1</sup> sodium sulfate aqueous solution at various values of temperature. Open circles, results by molecular dynamic simulations using the proposed coarse-grained force field. Closed circles, experimental data (ISONO, 1984). Open squares, Nernst-Einstein conductivity obtained from diffusivity calculation by molecular dynamic simulation using the proposed coarse-grained force field. Dotted line, KCl ideal line.

Even though the proposed coarse-grained model fails to quantitatively predict ionic conductivity, it seems to be a good model to be used in simulations involving coarse-grained proteins. The density of the solution and the viscosity are good enough for this kind of calculation. Nevertheless, one must pay attention to calculations in which the ionic conductivity plays an important role.



### 7.3 Critic review on the reference line for Walden plot

When we are celebrating the one-hundred-and-ten-year anniversary of the Walden's paper (WALDEN, 1906) on the relation between the limiting molar ionic conductivity and the viscosity for electrolytes, it urges to revisit the general belief in an universal reference for his rule. In 1906, Walden published a paper in which he presented experimental data on the limiting molar ionic conductivity and the viscosity of infinitely diluted  $\text{N}(\text{C}_2\text{H}_5)_4\text{I}$  in almost thirty different solvents. His empirical observations led him to the conclusion that the product of these two dynamic properties would be constant regardless of the system temperature or the used solvent.

The Walden's rule can be derived considering the Stokes-Einstein relation for the ionic diffusivity,  $D_i$ , and the solution viscosity,  $\eta$ :

$$D_i = \frac{k_B T}{6\pi\eta r_i} \quad (7.18)$$

where  $k_B$  is the Boltzmann constant,  $T$  is the absolute temperature and  $r_i$  is the Stokes radius of ion  $i$ . And also considering the Nernst-Einstein relation (URAHATA; RIBEIRO, 2006) for the ionic conductivity,  $\sigma^{NE}$ :

$$\sigma^{NE} = \frac{e^2}{k_B T} \sum_{i=1}^N \rho_i z_i^2 D_i \quad (7.19)$$

where  $e$  is the elementary charge,  $\rho_i$  is the number density of ion  $i$  and  $z_i$  is the valence of ion  $i$ .

Since the limiting molar ionic conductivity,  $\Lambda$ , is given by:

$$\Lambda = \frac{\sigma}{n_e c} \quad (7.20)$$

where  $n_e$  is a charge normalization factor ( $n_e$  is one for 1:1 electrolytes) and  $c$  is the electrolyte molar concentration; therefore, for a 1:1 electrolyte, the Walden's

rule can be written as:

$$\Lambda^{NE}\eta = \frac{e^2 N_A}{6\pi} \left[ \frac{1}{r^+} + \frac{1}{r^-} \right] \quad (7.21)$$

Nonetheless, the true ionic conductivity does not always equal the Nernst-Einstein ionic conductivity, for implicitly in Nernst-Einstein equation the cross terms among the ions are neglected. Hence, whenever one is intended to reproduce the experimentally determined ionic conductivity, one ought to consider the following expression:

$$\sigma = \sigma^{NE} (1 - \Delta) \quad (7.22)$$

where  $\Delta$  is a measure of the degree of correlation among the ions. Thus, one can rewrite Walden's rule in a more comprehensive form, in terms of the Stokes-Nernst-Einstein framework for a 1:1 electrolyte, as:

$$\Lambda\eta = \frac{e^2 N_A (1 - \Delta)}{6\pi} \left[ \frac{1}{r^+} + \frac{1}{r^-} \right] \quad (7.23)$$

where  $N_A$  is the Avogadro number,  $r^+$  and  $r^-$  are the cation and the anion Stokes radii respectively.

The so-called Walden plot is a log-log graph in which the limiting molar ionic conductivity is displayed as ordinate and the inverse of viscosity as abscissa. If a certain system follows the Walden's rule, a line with unity slope may be drawn on Walden plot; however, unless  $\Delta$ ,  $r^+$  and  $r^-$  are known, there is no way to be sure where such a line would lie on Walden plot.

Notwithstanding the fact that Walden's paper was based upon infinite diluted salt in non-aqueous solvents, to define a reference to which one could discuss one's data, it has been proposed that such a reference might be placed considering aqueous KCl solution; since it is believed that this electrolyte is fully dissociated in water (XU; COOPER; ANGELL, 2003). As pointed out by Harris (HARRIS, 2010), it is not even clear, for those who use this line, whether one should use

0.01 M or 1 M aqueous KCl as a reference point (MACFARLANE et al., 2009; XU; COOPER; ANGELL, 2003). Schreiner *et. al.* (SCHREINER et al., 2010), in order to clarify this problem, have shown that the true experimental data for 0.01 M aqueous KCl lie above the so-called KCl “ideal” line and that the experimental datum for 1 M aqueous KCl at 25 °C gives only a “nice” datum ( $y = x$ ) in which this ideal line of unity slope would pass through.

Despite this clearly arbitrary criterion to establish it, this KCl ideal line has not only been used extensively (YOSHIKAWA; XU; ANGELL, 2003; XU; COOPER; ANGELL, 2003; MACFARLANE et al., 2009; LEE; UENO; ANGELL, 2012; YAMAGUCHI et al., 2013), but also it seems to hide a deeper question: is there any possible universal line that can be drawn in Walden plot to which all systems can be compared?

Data of ionic liquids that lie exactly on KCl ideal line are believed to be fully dissociated and a coupling between the ionic transport and the structural relaxation can be considered. Such ionic liquids are called “good” ionic liquids. A “poor” ionic liquid though has its data placed below below this ideal line and is believed to have an incomplete ionic dissociation. A “superionic” liquid has its data placed above this ideal line, and is believed for such system that the ionic transport and the structural relaxation are decoupled (YAMAGUCHI et al., 2013). In Figure 33, it is shown a schematic representation of this sort of qualitative analysis above mentioned.

In spite of the fact that such analysis has been vastly applied, to the best of our knowledge, a rigorous check of the real aqueous KCl system is still needed.

The experimental determination of the Nernst-Einstein ionic conductivity is based upon the determination of the ionic diffusivities, however, such data are usually reported in terms of a mean diffusivity coefficient for the salt and not for each ionic species. To write a relation between this mean diffusivity coefficient and

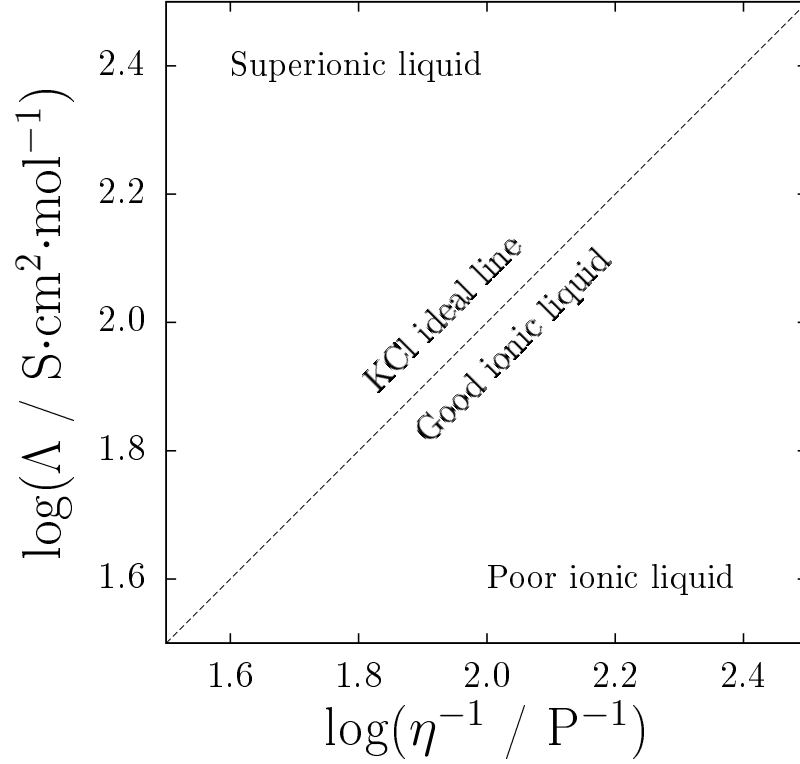


Figure 33: Schematic representation of the general analysis on Walden plot for ionic liquids.

the diffusivities for the cation and the anion, one must consider that the mean diffusivity coefficient,  $D$ , may be expressed as the harmonic mean of the ionic diffusivities weighted by the stoichiometric coefficients (PROBSTEIN, 1994):

$$D = \frac{\nu_+ + \nu_-}{\frac{\nu_+}{D_+} + \frac{\nu_-}{D_-}} \quad (7.24)$$

where  $\nu_+$  is the stoichiometric coefficient of the cation,  $\nu_-$  is the stoichiometric coefficient of the anion,  $D_+$  is the cation diffusivity and  $D_-$  is the anion diffusivity.

This relation is necessary but not sufficient to completely rewrite Nernst-Einstein equation in terms of the mean diffusivity coefficient; one more relation is still needed. Although for monovalent molten salts the equality  $D_+ = D_- = D$  has been assigned (HANSEN; MCDONALD, 1986), for salts diluted in water this is not the case. One possible relation though between the cation and the anion diffusivities is possible to be found for such solutions using the Stokes-Einstein

law. Thus, one may write the ratio of the ionic diffusivities as the inverse ratio of the Stokes radii of the ions:

$$\frac{D_+}{D_-} = \frac{r_-}{r_+} \quad (7.25)$$

For simple ions that form alkali halides, these data are available in the literature (PAU; BERG; MCMILLAN, 1990). The coupling of Equations (7.24) and (7.25) is presented on Appendix K.

Therefore, one could rewrite the Nernst-Einstein equation for a 1:1 electrolyte in terms of the mean diffusivity coefficient as:

$$\Lambda^{NE} = \frac{(1 + \delta)^2 D e^2 N_A}{2 \delta k_B T} \quad (7.26)$$

where  $\delta = r_-/r_+$  as shown on Appendix K.

Figure 34 shows the striking result for the actual aqueous 1.0 mol·L<sup>-1</sup> KCl experimental data that this system contains a non-negligible degree of correlation among the ions. This positive  $\Delta$  tells us that there is a significative correlation. As it has been shown elsewhere (HERTZ, 1977; PADRÓ; TRULLÀS; SESÉ, 1991; HARRIS, 2010), within Green-Kubo framework, this  $\Delta$  is related to the cross terms in the integral of the velocity correlation functions. This result implies that the system which has provided the ideal line is not an ideal system in terms of full ionic dissociation.

Now that we have surmounted the basic idea behind this common qualitative analysis, it remains the necessity to overcome the question on a possible universality for the Walden plot. Should one show another system for which the Nernst-Einstein line is not the same as for aqueous KCl, one will be able to state the impossibility of an universal line on Walden plot.

Figure 35 presents the experimental data for a highly diluted aqueous NaCl.

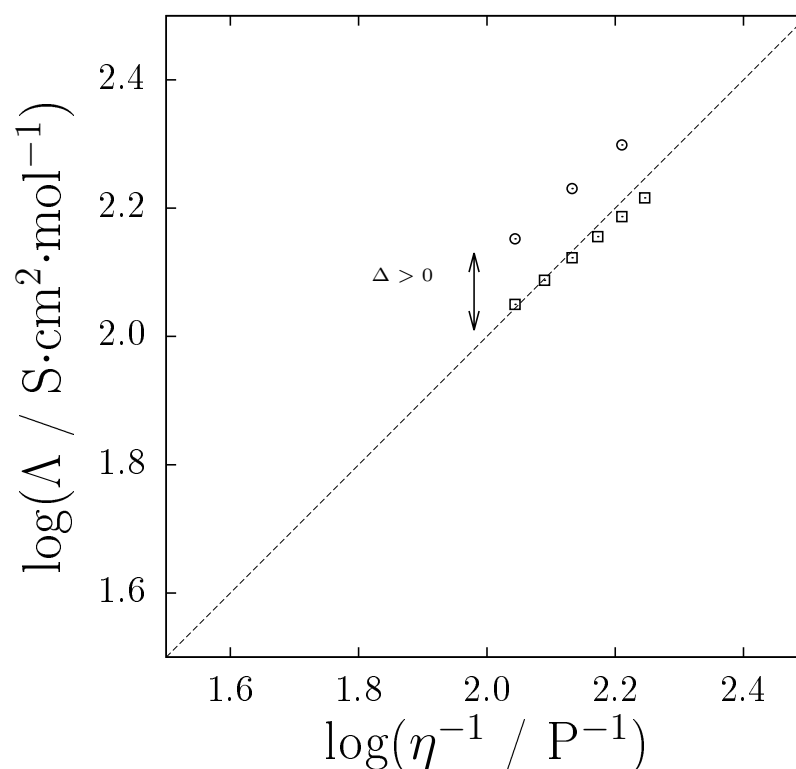


Figure 34: Walden plot for 1 M aqueous KCl solution experimental data. The dashed line is the KCl ideal line. Ionic conductivities were obtained by Wu *et al.* (WU *et al.*, 1994). Densities and dynamic viscosities were obtained by Kestin *et al.* (KESTIN; KHALIFA; CORREIA, 1981a). And mean diffusivity coefficients were obtained by Longworth (LONGSWORTH, 1957).

For this system,  $\Delta$  can be considered practically null, as it would be expected for such high dilution. However, even though the experimental data for this system coincidentally match the KCl ideal line, it is clear that the Nernst-Einstein line is different, *ergo* there is no such a thing as an universal line for the classical Walden plot.

Had one plotted only the true ionic conductivity on a Walden plot for both systems ( $1.0 \text{ mol}\cdot\text{L}^{-1}$  KCl and  $0.005 \text{ mol}\cdot\text{L}^{-1}$  NaCl aqueous solutions), one would have erroneously concluded that these systems share the same degree of correlation among their ions. This misinterpretation may only be avoided by plotting also the Nernst-Einstein ionic conductivity. Therefore no universal line could be drawn on a classical Walden plot to which all systems could be compared

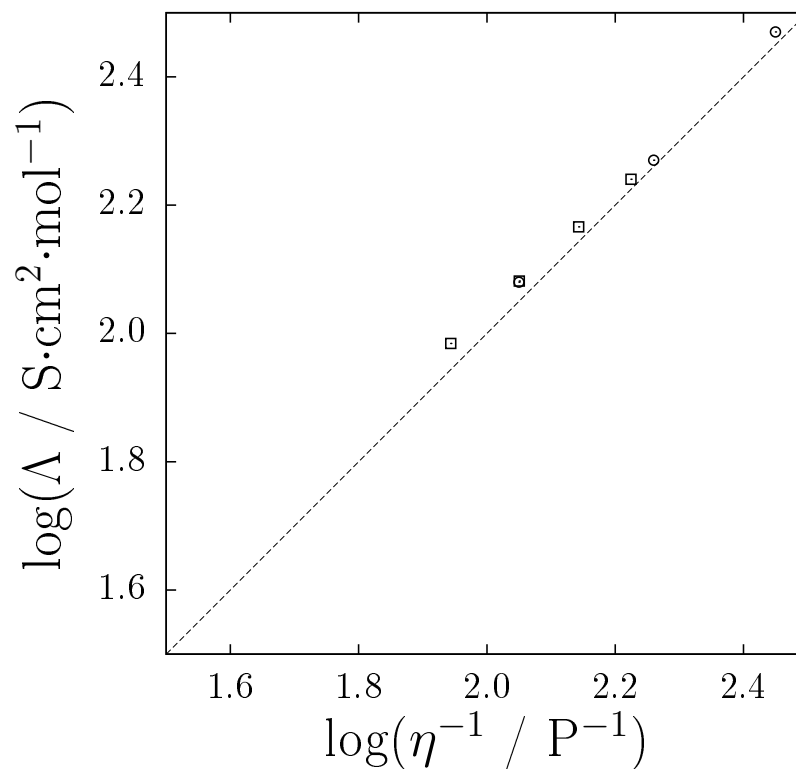


Figure 35: Walden plot for 0.005 M aqueous NaCl solution experimental data. The dashed line is the KCl ideal line. Limiting molar ionic conductivities were obtained by Gunning *et al.* (GUNNING; GORDON, 1942). Viscosities were obtained by Kestin *et al.* (KESTIN; KHALIFA; CORREIA, 1981b). And mean diffusivity coefficients were obtained by Fell *et al.* (FELL; HUTCHISON, 1971).

regardless of their nature.

## 7.4 A coarse-grained force field for Bovine Serum Albumin

### 7.4.1 Introduction

The possibility to simulate proteins with atomistic force fields is already a reality. But for some particular applications and proteins, a coarse-grained force field is not only a helpful model, but the only feasible one depending on the length or time scales of interest (BLANCO et al., 2013).

Bovine Serum Albumin (BSA) is certainly not a small protein. Its primary structure is composed by more than 580 amino acids (PETERS, 1995). Even though the protein itself has less than 10000 atoms, what could be consider not an extremely large system to be simulated with an atomistic force field, a potential of mean force calculation using only two proteins in aqueous solution would require a large number of water molecules. This makes such strategy impractical.

There are many ways to build coarse-grained force fields, as there are many levels of coarse graining. The investigation of all possibilities is not part of the scope of this thesis. Several reviews on this subject are available in the literature (NIELSEN et al., 2004; TOZZINI, 2005; TAKADA, 2012; SAUNDERS; VOTH, 2013). Recently a coarse-grained force field for BSA has been proposed in a study of protein adsorption on silver nanoparticles (VOICESCU; IONESCU; ANGELESCU, 2012). This force field is based on a previous coarse-grained model for polyelectrolytes (ANGELESCU; STENHAMMAR; LINSE, 2007).

One particular application to which a coarse-grained model of proteins would be interesting is as an auxiliary tool for protein structure prediction. There are many proteins for which no crystallographic structure is available. This was the case for BSA until 2012 (BUJACZ, 2012). Since Small-Angle X-ray Scattering (SAXS) for highly diluted protein solutions can provide the form factor (discussed



on Chapter 5), one would use only the primary structure information and molecular simulation to reproduce the form factor obtained from SAXS experiments. This is the main objective of this section.

Using molecular dynamics to predict the form factor is not a novelty itself. Nevertheless, the usual approach is to consider the crystallographic structure of the protein as an initial input for an atomistic simulation (KNIGHT; HUB, 2015). The main purpose here is to use molecular dynamics as a tool to predict the form factor of proteins for which the crystallographic structure is not available, even though the need of the atomic coordinates of proteins seems to be a necessary (but not sufficient) condition for evaluating the scattering intensity (NGUYEN et al., 2014).

Small-angle X-ray scattering is a low resolution method that can provide the shape and the size of the scattering particle (CHACÓN et al., 1998; CHACÓN et al., 2000), and these informations are helpful to predict the structure of such particle. The knowledge of the protein structure seems to be essential to the understanding of protein function and dynamics (AHN et al., 2000), even though intrinsically disordered protein segments, playing important roles in protein function, challenge the once believed strict relationship between structure and function (DUNKER et al., 2001).

Usually there are two types of problems related to scattering intensity data: one is the evaluation of the scattering intensity from the knowledge of the atomic coordinates, and the other is the inverse problem (REIS; APARICIO; ZHANG, 2011): using the scattering intensity to obtain the structure. Since small-angle X-ray scattering is a low resolution technique, no unique atomistic structure can be found in the second problem.

There are at least three *ab initio* methods for shape determination in small-angle scattering: envelop function (SVERGUN et al., 1996), bead model

(CHACÓN *et al.*, 1998) and dummy residues model (SVERGUN; PETOUKHOV; KOCH, 2001). The idea underneath these methods is to retrieve a three-dimensional model using one-dimensional scattering data. Although the techniques employed in this reconstruction have some physical basis (simulated annealing and genetic algorithm), the models do not have any, since they are based on spherical subunits or only on the envelop itself.

The main goal of this section is to provide a more physical model (using coarse-grained models) for evaluating the scattering intensity of a highly diluted protein in aqueous solution knowing only its primary structure, *id est*, the sequence of amino acids, and the radius of gyration.

The coarse-grained model developed here is inspired on Shinoda *et al.* (SHINODA; DEVANE; KLEIN, 2007) coarse-grained water force field. The reason to develop a coarse-grained force field which is consistent with a model for water is the possibility to try other applications in the future, such as the calculation of the potential of mean force of the protein in aqueous solution. This approach is certainly an improvement to implicit solvent models such as that used on Chapter 5.

### 7.4.2 Coarse-grained force field

Each amino acid is represented by a single bead. The beads (in a total of 583 beads) are connected in a single chain. Each bead has a molar mass equal to 114.065 g·mol<sup>-1</sup>. There are three types of beads according to their net charge: MET, GLU and HIS. MET (named after methionine) is an uncharged bead and represents in the chain all residues that have no charge at the BSA isoelectric point (pH=4.9) according to their isolated pK<sub>A</sub> values (including the terminal aspartic acid). GLU (named after glutamic acid) is a negatively charged bead and represents all glutamic acids, all aspartic acids and the terminal alanine residue

(to represent the terminal-COOH). HIS (named after histidine) is a positively charged bead and represents all histidines, lysines and arginines in the chain. All cysteines are considered to be uncharged, and therefore, are represented as MET. No disulfide bonds or hydrogen bonds were considered. The parameters were fitted to give an average radius of gyration similar to the experimental radius of gyration (*circa* 2.9 nm).

The force field is a sum of three contributions:

$$V = \sum_{i=1}^{N_b} V_{nonbond,ij} + V_{bond} + V_{angle} \quad (7.27)$$

where:

$$V_{nonbond,ij} = \frac{q_i q_j}{4\pi\epsilon_0\epsilon_r r_{ij}} + \frac{3\sqrt{3}}{2} \epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^4 \right] \quad (7.28)$$

where  $N_b$  is the number of beads,  $q_i$  is the charge of particle  $i$ ,  $\epsilon_0$  is the vacuum permittivity,  $\epsilon_r$  is the dielectric constant of the medium,  $\epsilon_{ij}$  is the attractive potential depth between particles  $i$  and  $j$ ,  $\sigma_{ij}$  is the distance between particles  $i$  and  $j$  for which the Mie contribution is null. The cross terms are calculated using Lorentz-Berthelot mixing rules; an exception is the interaction between water and MET for which the potential depth is calculated as half as the one given by the mixing rule to account for hydrophobic interactions. Table 7 presents the values for nonbonded interaction parameters for each pair.

The bond interaction is given by:

$$V_{bond} = \sum_{i=1}^{N_b-1} \frac{k_b}{2} (r_{i,i+1} - r_0)^2 \quad (7.29)$$

where  $r_{i,i+1}$  is the distance between two adjacent beads at the same chain,  $r_0$  is the equilibrium bond distance ( $r_0 = 8 \text{ \AA}$ ), and  $k_b$  is the bond constant ( $k_b = 1 \times 10^5 \text{ kJ.mol}^{-1}.\text{nm}^{-2}$ ).

The angle contribution is given by:

$$V_{angle} = \sum_{i=2}^{N_b-1} \frac{k_\theta}{2} (\theta_i - \theta_0)^2 \quad (7.30)$$

where  $\theta_i$  is the angle between three beads in a sequence in the chain,  $\theta_0$  is the equilibrium angle ( $\theta_0 = 180^\circ$ ) and  $k_\theta$  is the angle constant ( $k_\theta = 100.0 \text{ kJ.mol}^{-1}.\text{rad}^{-2}$ ).

Table 7: Nonbonded interaction parameters for each pair.

Bead $i$	Bead $j$	$\sigma_{ij} / \text{\AA}$	$\varepsilon_{ij} / \text{kJ.mol}^{-1}$	$q_i q_j / e^2$
W	W	4.371	3.745	0.0
HIS	HIS	5.000	2.479	+1.0
GLU	GLU	5.000	2.479	+1.0
MET	MET	5.000	2.479	0.0
W	HIS	4.686	3.047	0.0
W	GLU	4.686	3.047	0.0
W	MET	4.686	1.523	0.0
HIS	GLU	5.000	2.479	-1.0
HIS	MET	5.000	2.479	0.0
GLU	MET	5.000	2.479	0.0

### 7.4.3 Molecular dynamics and form factor calculation details

A NPT molecular dynamic simulation was performed using GROMACS 4.5.5 software package (BERENDSEN; van-der-SPOEL; van-DRUNEN, 1995). The initial configuration was set as an helicoidal conformation of the coarse-grained BSA in 8000 coarse-grained water molecules (SHINODA; DEVANE; KLEIN, 2007). The Leap Frog algorithm was used to integrate the equations of motion with periodic boundary conditions and 2 fs timestep. The Berendsen thermostat (BERENDSEN et al., 1984) was used to keep the temperature at 298.15 K. The

isotropic Berendsen barostat was used to keep the pressure at 1 bar (isothermal compressibility of  $4.5 \times 10^{-5} \text{ bar}^{-1}$ ). Long-range as well as short-range interactions were both treated with cutoff radii of 1.5 nm. Positions and velocities were stored at each 1 ps. Electrostatic potential was calculated using Particle Mesh Ewald method (DARDEN; YORK; PEDERSEN, 1993). The dielectric constant of the water was considered to be 78.41 (UEMATSU; FRANCK, 1980). For equilibration, 1.5 ns were used and the rest 0.5 ns were considered as production stage.

The form factor was calculated as the average over the instantaneous form factor of each one of the 500 configurations on the production stage using CRY SOL (SVERGUN; BARBERATO; KOCH, 1995).

#### 7.4.4 Results and Discussions

Figure 36 illustrates the conformations of the coarse-grained BSA during the simulation and also the crystallographic structure (BUJACZ, 2012). One may observe that the initial helicoidal conformation is rapidly folded. And the final configuration has some similarities with the crystallographic structure.

One way to quantitatively measure the folding during the simulation is by the calculation of the radius of gyration:

$$R_g = \sqrt{\frac{1}{N_b} \sum_{i=1}^{N_b} |\vec{r}_i - \vec{r}_{com}|^2} \quad (7.31)$$

where  $\vec{r}_i$  is the position vector of bead  $i$  and  $\vec{r}_{com}$  is the position vector of the center of mass of the BSA.

Figure 37 presents the evolution of the radius of gyration for our coarse-grained model with the simulation time.

It is noteworthy that the average radius of gyration is similar to the radius

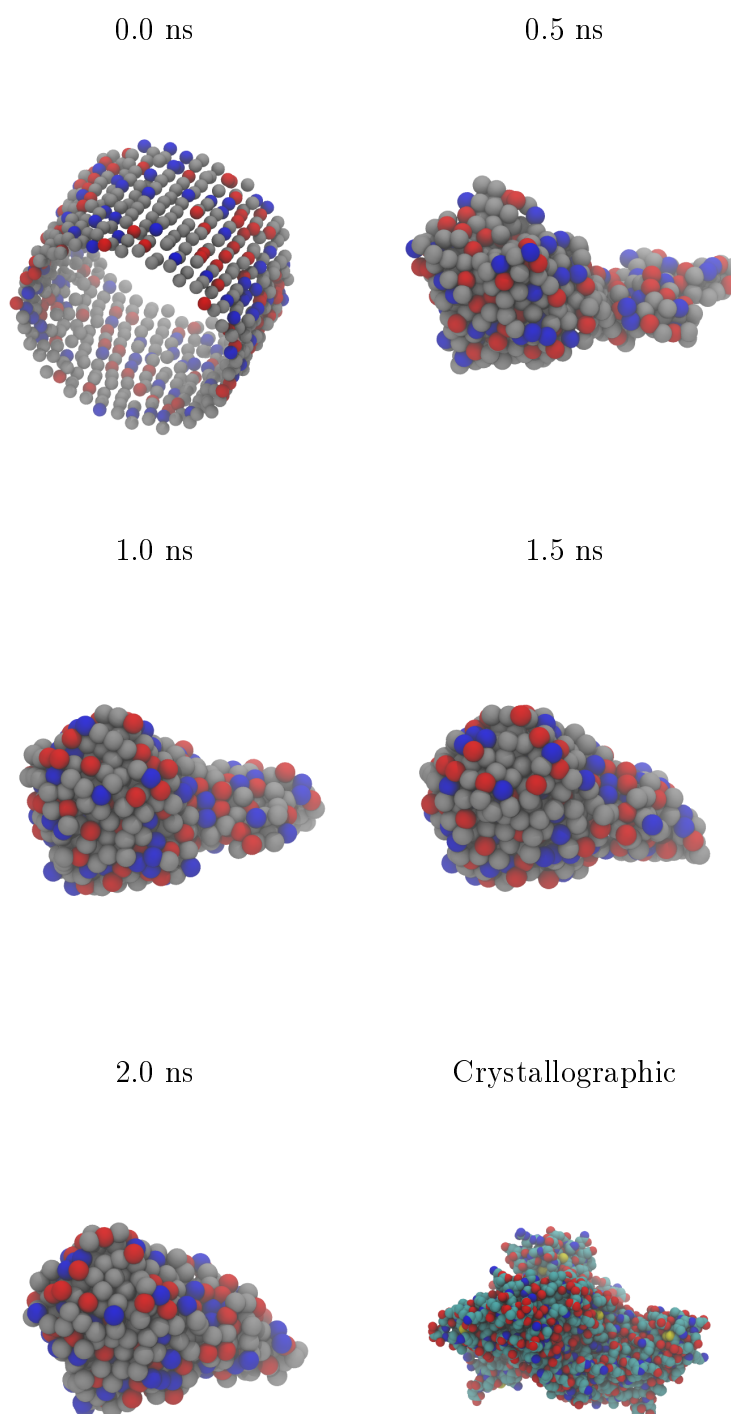


Figure 36: Snapshots of equilibration and production NPT molecular dynamics of BSA coarse-grained force field. MET beads are gray, GLU are red and HIS are blue.

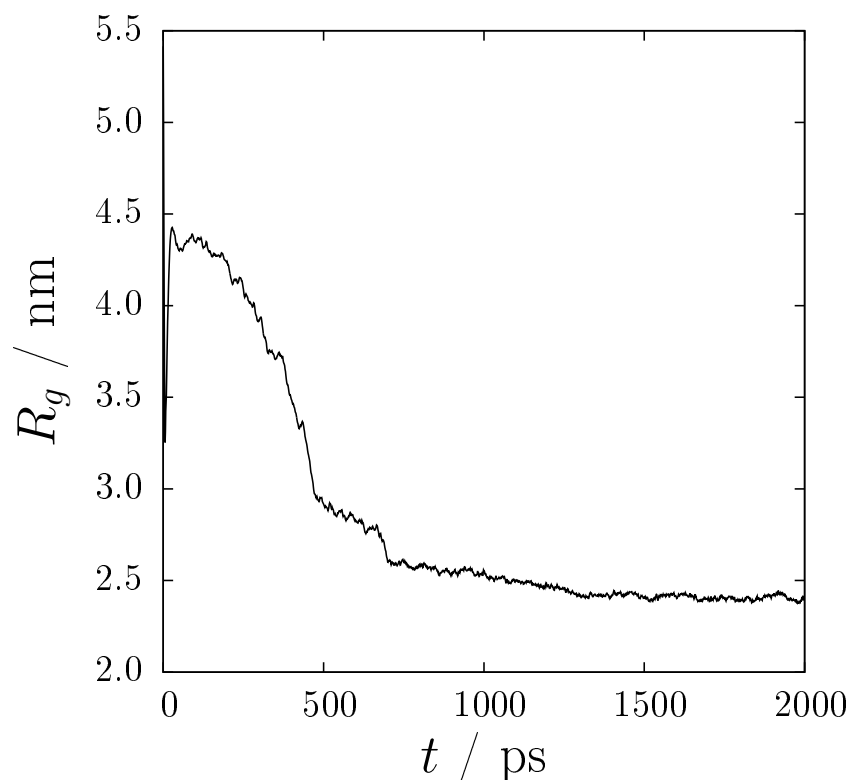


Figure 37: Radius of gyration as a function of time step for NPT molecular dynamics of the coarse-grained BSA in aqueous solution free of salt.

of gyration for the crystallographic structure (*circa* 2.9 nm).

One may observe in Figure 38 that the proposed force field is able to predict the form factor of BSA with reasonable accuracy when compared to the experimental data and also to the predictions using the crystallographic structure. And the form factor calculated using this coarse-grained force field is certainly better than the simple ellipsoid of revolution fitting.

The proposed coarse-grained model neglects important structure informations. The fact that no disulfide bonds or hydrogen bonds are explicitly considered is certainly a problem to accurate protein structure prediction.

Disulfide bonds are relevant not only for the protein structure, but also for the protein activity (HOGG, 2003). Hydrogen bonds are extremely important for the conformation of the secondary structure, *exempli gratia*,  $\alpha$ -helix and

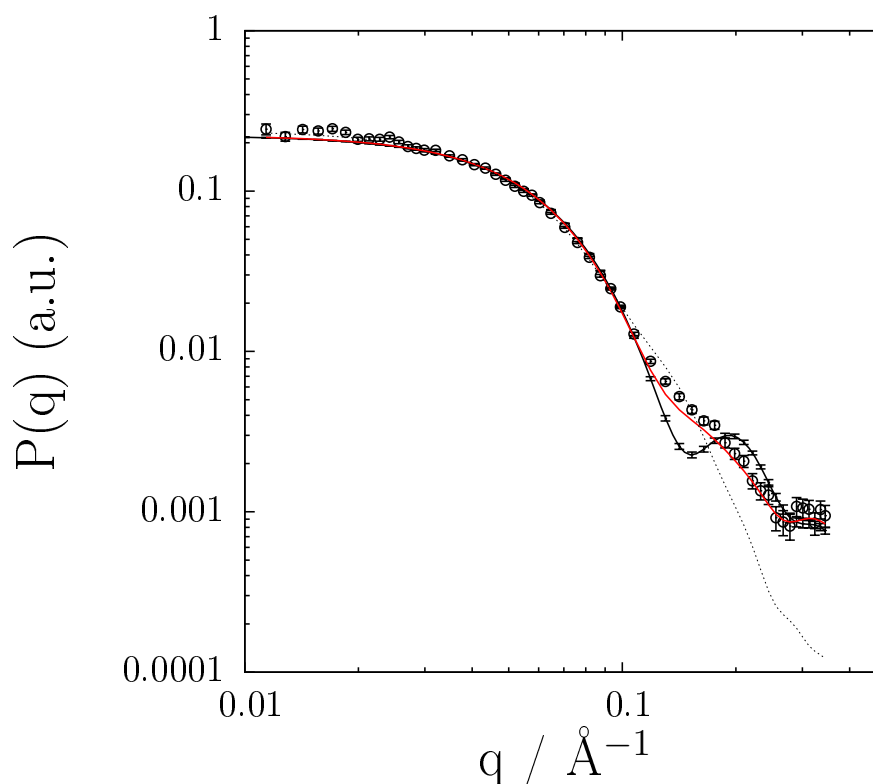


Figure 38: Form factor of bovine serum albumin. Open circles, small-angle X-ray experimental data (FRANCO; OLIVEIRA; FILHO, 2015). Red line, crystallographic structure prediction with CRY SOL (SVERGUN; BARBERATO; KOCH, 1995). Continuous line, the proposed coarse-grained force field using CRY SOL to calculate the form factor. Dotted line, ellipsoid of revolution.

$\beta$ -sheet (PAULING; COREY; BRANSON, 1951; BAKER; HUBBARD, 1984). Models for hydrogen bonding have been proposed for coarse-grained force fields (TAKADA, 2012; MAUPETIT; TUFFERY; DERREUMAUX, 2007).

Small-angle X-ray scattering experiments constitute a low resolution technique for protein structure prediction. Therefore, low resolution models may be able to describe the scattering intensity of a highly diluted protein solution. This is certainly an issue, since many different models may result in a very well description of the scattering intensity, and hence no unique solution may be found.

It turns out that the proposed coarse-grained model, even neglecting important structure features as the secondary structure conformation, is able to reproduce the scattering intensity of BSA in aqueous solution. What is being



claimed here is the possibility to use this very rough approach in situations when the crystallographic structure is not available. Moreover, the ability to predict the form factor using molecular dynamics sounds to have a consistent physical basis.

Unlike the common *ab initio* methods, the model proposed here is based on physical informations, such as the protein primary structure and the radius of gyration. Therefore, using this method, one is able to retrieve the protein envelop with a more physically consistent model.

Thus the coupling between coarse-grained force field molecular dynamics and small-angle X-ray scattering seems to be a promising strategy to provide helpful information on the structure of proteins at a low resolution level.

Since this force field was developed based on a previous water model, the calculation of the potential of mean force of two proteins in aqueous solution using this force field can be done. Also the effect of salts could be studied using models as that developed for sodium sulfate in the last section.

## 8 CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

The general conclusion of this thesis is that there is a subtle limit in the treatment of the ions as co-solvent or co-solute on implicit solvent models. The interaction potential amongst proteins depends on the pH, on the temperature and seems to be also dependent of the ratio between the number of salt ions and the number of protein molecules in solution. This reveals a great challenge, since the interaction potential would be dependent of the protein concentration. The range of attractiveness seems to be protein dependent (very short-ranged for proteins as lysozyme and  $\gamma$ -crystallin and long-ranged for proteins as bovine serum albumin). And coarse-grained force fields for molecular simulations seems to be a very promising way to avoid implicit solvent models in the calculation of the interaction potential between two protein molecules in salt aqueous solutions.

An analytical expression for the structure factor of a fluid described by a van-der-Waals-like attractive potential was developed. The expression predicts quite well the structure of a van der Waals fluid generated through Monte Carlo simulations. Introducing an empirical relation to account for the dependence of the attractive parameter on the protein concentration allowed to correlate accurately experimental X-ray scattering intensities of BSA aqueous solution at relatively high salt concentration. The results posed the question of how far some hypotheses usually assumed in implicit solvent approaches can be taken for granted. The expression for the structure factor was also used to develop an

osmotic equation of state, which was used to predict experimental data of osmotic pressure of BSA solutions with good agreement.

A virtual liquid-liquid coexistence curve for BSA in aqueous solution at the isoelectric point was predicted using the previous developed osmotic equation of state. For a pH value below the isoelectric point, a phase separation was indeed observed and our model was able to qualitatively predict such separation. A comparison with molecular simulation results showed that the state correspondence law is obeyed near the critical point, but the ion specificity became larger to low temperatures. Also, the predicted lines showed that the range of the attractive potential is not so short as it is usually believed.

A new coarse-grained force field for sodium sulfate aqueous solution that avoids unphysical association among the ions was developed. The new model was able to reproduce the experimental data for density, although it fails to predict the thermal expansion coefficient accurately. Dynamic properties such as viscosity and ionic conductivity were calculated within Green-Kubo framework, and the model was able to correlate viscosity data reasonably well, but it overestimates the ionic conductivity. A discussion on the structure and the dynamics was done based on the Walden plot.

A critical review on the idea of an universal reference line for Walden plot was proposed. Showing the experimental data of KCl and NaCl available in the literature, it was proved that no universal reference line to which all systems may be compared can be drawn on Walden plot.

A new coarse-grained model for BSA in aqueous solution was developed. The model was used as a new strategy to evaluate the scattering form factor of proteins as a tool in protein structure prediction. The coarse-grained model was able to reproduce the experimental form factor with quite high accuracy as well as the radius of gyration calculated with the crystallographic structure. This model is

certainly an improvement to the assumption that the protein may be represented as an ellipsoid of revolution.

As stated in the introduction, this thesis raised more questions than brought answers. Therefore, there are few suggestions to continue the work that has begun here. The first one is the extension of the small-angle X-ray experiments changing the salt concentration to understand the effect of the proportion of ions and protein molecules on the attractive potential. The second one is the measure of BSA aqueous solution liquid-liquid coexistence line for pH values below the isoelectric point. This would be most valuable to corroborate the predictions that were made here. The third one is the application of Isothermal Titration Calorimetry to study how the ion specificity changes the phase diagram of proteins. The fourth suggestion is the application of the coarse-grained models proposed in this thesis to evaluate the potential of mean force between two BSA molecules in sodium sulfate aqueous solutions. These simulations would be extremely helpful if they were compared to the small-angle X-ray experiments.

## REFERENCES

- AHN, S. et al. Protein tertiary structural changes visualized by time-resolved X-ray solution scattering. *J. Mol. Biol.*, v. 299, p. 1289–1302, 2000.
- ALLEN, M.; TILDESLEY, D. *Computer Simulation of Liquids*. New York: Oxford Science Publications, 1987.
- ANGELESCU, D. G.; STENHAMMAR, J.; LINSE, P. Packaging of a flexible polyelectrolyte inside a viral capsid: effect of salt concentration and salt valence. *J. Phys. Chem. B*, v. 111, p. 8477–8485, 2007.
- ASHCROFT, N.; LEKNER, J. Structure and resistivity of liquid metals. *Phys. Rev.*, v. 145, p. 84–90, 1966.
- BAKER, E. N.; HUBBARD, R. E. Hydrogen bonding in globular proteins. *Prog. Biophys. Molec. Biol.*, v. 44, p. 97–179, 1984.
- BARBOSA, L. et al. The importance of protein-protein interactions on the pH-induced conformational changes of bovine serum albumin: A small-angle X-ray scattering study. *Biophys. J.*, v. 98, p. 147–157, 2010.
- BARKER, J. A.; HENDERSON, D. What is “liquid”? Understanding the states of matter. *Rev. Mod. Phys.*, v. 48, p. 587–671, 1976.
- BAXTER, R. Percus-Yevick equation for hard spheres with adhesion. *J. Chem. Phys.*, v. 49, p. 2770–2774, 1968.
- BENEDEK, G. B. Cataract as a protein condensation disease. *Invest. Ophthalm. Vis. Sci.*, v. 38, p. 1911–1921, 1997.
- BERENDSEN, H.; GRIGERA, J.; STRAATSMA, T. The missing term in effective pair potentials. *J. Phys. Chem.*, v. 91, p. 6269–6271, 1987.
- BERENDSEN, H. J. C. et al. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.*, v. 81, p. 3684–3690, 1984.
- BERENDSEN, H. J. C.; van-der-SPOEL, D.; van-DRUNEN, R. GROMACS: a message-passing parallel molecular dynamics implementation. *Comput. Phys. Commun.*, v. 91, p. 43–56, 1995.
- BLANCO, M. A. et al. Coarse-grained model for colloidal protein interactions,  $b_{22}$ , and protein cluster formation. *J. Phys. Chem. B*, v. 117, p. 16013–16028, 2013.

- BRATKO, D. et al. Orientation-averaged pair potentials between dipolar proteins or colloids. *J. Phys. Chem. B*, v. 106, p. 2714–2720, 2002.
- BROIDE, M. L. et al. Binary-liquid phase separation of lens protein solutions. *Proc. Natl. Acad. Sci. USA*, v. 88, p. 5660–5664, 1991.
- BUJACZ, A. Structures of bovine, equine and leporine serum albumin. *Acta Crystallogr. D*, v. 68, p. 1278–1289, 2012.
- CACCAMO, C. Integral equation theory description of phase equilibria in classical fluids. *Phys. Reports*, v. 274, p. 1–105, 1996.
- CAILLOL, J. M. Critical-point of the Lennard-Jones fluid: a finite size scaling study. *J. Chem. Phys.*, v. 109, p. 4885–4893, 1998.
- CANNAVACCIUOLO, L. et al. Size, flexibility, and scattering functions of semiflexible polyelectrolytes with excluded volume effects: Monte Carlo simulations and neutron scattering experiments. *Phys. Rev. E*, v. 62, p. 5409–5419, 2000.
- CARNAHAN, N.; STARLING, K. Equation of state for nonattracting rigid spheres. *J. Chem. Phys.*, v. 51, p. 635–636, 1969.
- CHACÓN, P. et al. Reconstruction of protein form with X-ray solution scattering and a genetic algorithm. *J. Mol. Biol.*, v. 299, p. 1289–1302, 2000.
- \_\_\_\_\_. Low-resolution structures of proteins in solutions retrieved from X-ray scattering with a genetic algorithm. *Biophys. J.*, v. 74, p. 2760–2775, 1998.
- CHAPELA, G. A. et al. Computer simulation of a gas-liquid surface. *J. Chem. Soc. Faraday Trans.*, v. 73, p. 1133–1141, 1977.
- DARDEN, T.; YORK, D.; PEDERSEN, L. Particle mesh Ewald: an  $N \cdot \log(N)$  method for Ewald sums in large systems. *J. Chem. Phys.*, v. 98, p. 10089–10092, 1993.
- DUDA, Y. Square-well fluid modeling of protein liquid-vapor coexistence. *J. Chem. Phys.*, v. 130, p. 116101, 2009.
- DUNKER, A. K. et al. Intrinsically disordered protein. *J. Mol. Graph. Model.*, v. 19, p. 26–59, 2001.
- FEIGIN, L. A.; SVERGUN, D. I. *Structure Analysis by Small Angle X-ray Scattering and Neutron Scattering*. New York: Plenum Press, 1987.
- FELL, C. J. D.; HUTCHISON, H. P. Diffusion coefficients for sodium and potassium chlorides in water at elevated temperatures. *J. Chem. Eng. Data*, v. 16, p. 427–429, 1971.
- FRANCO, L. F. M.; OLIVEIRA, C. L. P. d.; FILHO, P. d. A. P. Thermodynamics of protein aqueous solutions: from the structure factor to the osmotic pressure. *AIChE J.*, v. 61, p. 2871–2880, 2015.

FRENKEL, D.; SMIT, B. *Understanding Molecular Simulation: from Algorithms to Applications*. 2. ed. San Diego: Academic Press, 2002.

FRENKEL, D. et al. Structure factors of polydisperse systems of hard spheres. a comparison of Monte Carlo simulations and Percus-Yevick theory. *J. Chem. Phys.*, v. 84, p. 4625–4630, 1986.

GEORGE, A.; WILSON, W. Predicting protein crystallization from a dilute solution property. *Acta. Cryst. Sec. D*, v. 50, p. 361–365, 1994.

GIBBS, J. W. *Elementary principles in statistical mechanics*. New York: Charles Scribner's Sons, 1902.

GLATTER, O.; KRATKY, O. *Small Angle X-ray Scattering*. London: Academic Press, 1982.

GUINIER, A. *X-ray Diffraction in crystals, imperfect crystals, and amorphous bodies*. New York: Dover, 1994.

GUNNING, H. E.; GORDON, A. R. The conductance and ionic mobilities for aqueous solutions of potassium and sodium chloride at temperatures from 15°C to 45°C. *J. Chem. Phys.*, v. 10, n. 2, p. 126–131, 1942.

HANSEN, J.; HAYTER, J. A rescaled MSA structure factor for dilute charged colloidal dispersions. *Mol. Phys.*, v. 46, p. 651–656, 1982.

HANSEN, J. P.; MCDONALD, I. *Theory of simple liquids*. 2. ed. London: Academic Press, 1986.

HARRIS, K. R. Relations between the fractional Stokes-Einstein and Nernst-Einstein equations and velocity correlation coefficients in ionic liquids and molten salts. *J. Phys. Chem. B*, v. 114, n. 29, p. 9572–9577, 2010.

HAYTER, J.; PENFOLD, J. An analytic structure factor for macroion solutions. *Mol. Phys.*, v. 42, p. 109–118, 1981.

HERTZ, H. G. Velocity correlations in aqueous electrolyte solutions from diffusion, conductance, and transference data. part 1, theory. *Ber. Bunsen-Ges. Phys. Chem.*, v. 81, n. 7, p. 656–664, 1977.

HILL, T. L. *An Introduction to Statistical Thermodynamics*. New York: Dover, 1986.

\_\_\_\_\_. *Statistical Mechanics*. New York: Dover Publications Inc., 1987.

HOGG, P. J. Disulfide bonds as switches for protein function. *Trends Biochem. Sci.*, v. 28, p. 210–214, 2003.

HOOVER, W. Canonical dynamics: equilibrium phase-space distributions. *Phys. Rev. A*, v. 31, p. 1695–1697, 1985.

HUMPHREY, W.; DALKE, A.; SCHULTEN, K. VMD – Visual Molecular Dynamics. *J. Mol. Graphics*, v. 14, p. 33–38, 1996.

ISONO, T. Density, viscosity, and electrolytic conductivity of concentrated aqueous electrolyte solutions at several temperature. Alkaline-Earth chlorides,  $\text{LaCl}_3$ ,  $\text{Na}_2\text{SO}_4$ ,  $\text{NaNO}_3$ ,  $\text{NaBr}$ ,  $\text{KNO}_3$ ,  $\text{KBr}$ , and  $\text{Cd}(\text{NO}_3)_2$ . *J. Chem. Eng. Data*, v. 29, p. 45–52, 1984.

ISRAELACHVILI, J. *Intermolecular and Surface Forces*. 3. ed. San Diego: Academic Press, 2011.

KASTELIC, M. et al. Protein aggregation in salt solutions. *Proc. Natl. Acad. Sci. USA*, v. 112, p. 6766–6770, 2015.

KATSONIS, P.; BRANDON, S.; VEKILOV, P. G. Corresponding-states laws for protein solutions. *J. Phys. Chem. B*, v. 110, p. 17638–17644, 2006.

KESTIN, J.; KHALIFA, H. E.; CORREIA, R. J. Tables of dynamic and kinematic viscosity of aqueous  $\text{KCl}$  solutions in the temperature range 25–150°C and the pressure range 0.1–35MPa. *J. Phys. Chem. Ref. Data*, v. 10, p. 57–70, 1981.

\_\_\_\_\_. Tables of dynamic and kinematic viscosity of aqueous  $\text{NaCl}$  solutions in the temperature range 20–150°C and the pressure range 0.1–35MPa. *J. Phys. Chem. Ref. Data*, v. 10, p. 71–88, 1981.

KINSLER, P.; FAVARO, A.; MCCALL, M. W. Four Poynting theorems. *Eur. J. Phys.*, v. 30, p. 983–993, 2009.

KNIGHT, C. J.; HUB, J. S. WAXSiS: a web server for the calculation of SAXS/WAXS curves based on explicit-solvent molecular dynamics. *Nucleic Acids Res.*, v. 43, p. W225–W230, 2015.

KOCH, M. H. J.; VACHETTE, P.; SVERGUN, D. I. Small-angle scattering: a view on the properties structures and structural changes of biological macromolecules in solution. *Q. Rev. Biophys.*, v. 36, p. 147–227, 2003.

KOLMOGOROV, A. N. *Foundations of the theory of probability*. New York: Chelsea Publishing Company, 1956.

KOTLARCHYK, M.; CHEN, S. Analysis of small angle neutron scattering spectra from polydisperse interacting colloids. *J. Chem. Phys.*, v. 79, p. 2461–2469, 1983.

KUBO, R. Statistical-mechanical theory of irreversible processes. I. General theory and simple applications to magnetic and conduction problems. *J. Phys. Soc. Jpn.*, v. 12, p. 570–586, 1957.

LANDAU, L. D.; LIFSHITZ, E. M. *Électrodynamique des milieux continus*. Moscow: Éditions MIR, 1969.

\_\_\_\_\_. *The Classical Theory of Fields*. 4. ed. Oxford: Pergamon Press, 1975.

LEBOWITZ, J.; PERCUS, J. Mean spherical model for lattice gases with extended hard cores and continuum fluids. *Phys. Rev.*, v. 144, p. 215–258, 1966.



LEE, S. Y.; UENO, K.; ANGELL, C. A. Lithium salt solutions in mixed sulfone and sulfone-carbonate solvents: A walden plot analysis of the maximally conductive compositions. *J. Phys. Chem. C*, v. 116, n. 45, p. 23915–23920, 2012.

LI, H.; ROBERTSON, A.; JENSEN, J. Very fast empirical prediction and rationalization of protein  $pK_a$  values. *Proteins*, v. 61, p. 704–721, 2005.

LONGSWORTH, L. G. The temperature dependence of the Soret coefficient of aqueous potassium chloride. *J. Phys. Chem.*, v. 61, n. 11, p. 1557–1562, 1957.

LÓPEZ-RENDÓN, R.; REYES, Y.; OREA, P. Thermodynamic properties of short-range square well fluid. *J. Chem. Phys.*, v. 125, p. 084508, 2006.

LU, Y. et al. Study of interactions of bovine serum albumin in aqueous  $(\text{NH}_4)_2\text{SO}_4$  solution at 25 °C by osmotic pressure measurements. *J. Chem. Eng. Data*, v. 54, p. 1975–1980, 2009.

MACFARLANE, D. R. et al. On the concept of ionicity in ionic liquids. *Phys. Chem. Chem. Phys.*, v. 11, p. 4962–4967, 2009.

MALFOIS, M. et al. A model of attractive interactions to account for fluid-fluid phase separation of protein solutions. *J. Chem. Phys.*, v. 105, p. 3290–3300, 1996.

MAUPETIT, J.; TUFFERY, P.; DERREUMAUX, P. A coarse-grained protein force field for folding and structure prediction. *Proteins*, v. 69, p. 394–408, 2007.

MAYER, J. E. Dispersion and polarizability and the van der Waals potential in the alkali halides. *J. Chem. Phys.*, v. 1, p. 270–279, 1933.

MCMILLAN, W.; MAYER, J. The statistical thermodynamics of multicomponent systems. *J Chem Phys*, v. 13, p. 276–305, 1945.

MCQUARRIE, D. *Statistical Mechanics*. California: University Science Books, 2000.

MOLLERUP, J.; BREIL, M. On the thermodynamics of the McMillan-Mayer state function. *Fluid Phase Equilibr.*, v. 276, p. 18–23, 2009.

\_\_\_\_\_. The osmotic second virial coefficient and Gibbs-McMillan-Mayer framework. *Fluid Phase Equilibr.*, v. 286, p. 88–94, 2009.

NARAYANAN, J.; LIU, X. Protein interactions in undersaturated and supersaturated solutions: A study using light and X-ray scattering. *Biophys. J.*, v. 84, p. 523–532, 2003.

NEAL, B.; ASTHAGIRI, D.; LENHOFF, A. Molecular origins of osmotic second virial coefficients of proteins. *Biophys. J.*, v. 75, p. 2469–2477, 1998.

NELDER, J.; MEAD, R. A simplex method for function minimization. *Comp. J.*, v. 7, p. 308–313, 1965.

- NGUYEN, H. T. et al. Accurate small and wide angle x-ray scattering profiles from atomic models of proteins and nucleic acids. *J. Chem. Phys.*, v. 141, p. 22D508, 2014.
- NIELSEN, S. O. et al. Coarse grain models and computer simulation of soft materials. *J. Phys.: Condens. Matter*, v. 16, p. R481–R512, 2004.
- NOSÉ, S. A unified formulation of the constant temperature molecular dynamics methods. *J. Chem. Phys.*, v. 81, p. 511–519, 1984.
- OKORAFOR, O. C. Solubility and density isotherms for the sodium sulfate-water-methanol system. *J. Chem. Eng. Data*, v. 44, p. 488–490, 1999.
- OLIVEIRA, C. *Small angle X-ray scattering from biological systems: Theory and Applications*. Tese (Doutorado) — University of Campinas, 2005.
- OLIVEIRA, C. SUPERSAXS - program package for data treatment, analysis and modeling. 2009.
- OREA, P.; DUDA, Y.; ALEJANDRE, J. Surface tension of a square well fluid. *J. Chem. Phys.*, v. 118, p. 5635–5639, 2003.
- ORNSTEIN, L.; ZERNIKE, F. Accidental deviations of density and opalescence at the critical point of a single substance. *Proc. Sect. Sci. K. ned. Acad. Wet.*, v. 17, p. 793–806, 1914.
- PADRÓ, J.; TRULLÀS, J.; SESÉ, G. Computer simulation study of the dynamic cross-correlations in liquids. *Mol. Phys.*, v. 72, n. 5, p. 1035–1049, 1991.
- PALMER, R.; WEEKS, J. Exact solution of the mean spherical model for charged hard spheres in uniform neutralizing background. *J. Chem. Phys.*, v. 58, p. 4171–4174, 1973.
- PARRINELLO, M.; RAHMAN, A. Polymorphic transitions in single crystal: a new molecular dynamics method. *J. Appl. Phys.*, v. 52, p. 7182–7190, 1981.
- PAU, P. C. F.; BERG, J. O.; MCMILLAN, W. G. Application of Stokes' law to ions in aqueous solution. *J. Phys. Chem.*, v. 94, p. 2671–2679, 1990.
- PAULING, L.; COREY, R. B.; BRANSON, H. R. The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proc. Natl. Acad. Sci. USA*, v. 37, p. 205–211, 1951.
- PEDERSEN, J. Analysis of small-angle scattering data from colloids and polymer solutions: modeling and least-squares fitting. *Adv. Colloid. Interface Sci.*, v. 70, p. 171–210, 1997.
- PERCUS, J.; YEVIK, G. Analysis of classical statistical mechanics by means of collective coordinates. *Phys. Rev.*, v. 110, p. 1–13, 1958.
- PESSÔA-FILHO, P. d. A.; MAURER, G. An extension of the pitzer equation for the excess gibbs energy of aqueous electrolyte systems to aqueous polyelectrolyte solutions. *Fluid Phase Equilib.*, v. 269, p. 25–35, 2008.

PETERS, T. J. *All about albumin: biochemistry, genetics and medical applications*. New York: Academic Press, 1995.

PRAUSNITZ, J. Molecular thermodynamics for some applications in biotechnology. *Pure Appl. Chem.*, v. 75, p. 859–873, 2003.

\_\_\_\_\_. Thermodynamics and the other chemical engineering sciences: old models for new chemical product and processes. *J. Phys.: Condens Matter*, v. 20, p. 1–4, 2008.

PRAUSNITZ, J. M.; LICHTENTHALER, R. N.; AZEVEDO, E. G. *Molecular Thermodynamics of fluid-phase equilibria*. 3. ed. Upper Saddle River: Prentice Hall, 1999.

PROBSTEIN, R. F. *Physicochemical hydrodynamics: an introduction*. 2. ed. New York: John Wiley & Sons, 1994.

REIHER-III, W. H. *Theoretical studies of hydrogen bonding*. Tese (Doutorado) — Harvard University, 1985.

REIS, M. A.; APARICIO, R.; ZHANG, Y. Improving protein template recognition by using small-angle X-ray scattering profiles. *Biophys. J.*, v. 101, p. 2770–2781, 2011.

REY-CASTRO, C.; VEGA, L. F. Transport properties of the ionic liquid 1-ethyl-3-methylimidazolium chloride from equilibrium molecular dynamics simulation. the effect of temperature. *J. Phys. Chem. B*, v. 110, p. 14426–14435, 2006.

ROZANOV, Y. A. *Probability theory: a concise course*. New York: Dover, 1977.

SAUNDERS, M.; VOTH, G. Coarse-graining methods for computational Biology. *Annu. Rev. Biophys.*, v. 42, p. 73–93, 2013.

SCHREINER, C. et al. Fractional walden rule for ionic liquids: Examples from recent measurements and a critique of the so-called ideal kcl line for the Walden plot. *J. Chem. Eng. Data*, v. 55, n. 5, p. 1784–1788, 2010.

SCHRÖDER, C.; STEINHAUSER, O. On the dielectric conductivity of molecular ionic liquids. *J. Chem. Phys.*, v. 131, p. 114504, 2009.

SCHRÖDINGER, E. *Statistical Thermodynamics*. New York: Dover Publications Inc., 1989.

SHARMA, R.; SHARMA, K. On generalization of the hard-sphere model for the the structure factor. *Phys. Lett.*, v. 56, p. 107–108, 1976.

\_\_\_\_\_. The structure factor and the transport properties of dense fluids having molecules with square well potential, possible generalization. *Physica*, v. 89, p. 213–218, 1977.

SHEN, V. K.; SIDERIUS, D. W.; KRELKELBERG, E. P. *NIST Standard Reference Simulation Website, NIST Standard Reference Database Number 173*. Gaithersburg: National Institute of Standards and Technology, 2015.

SHINODA, W.; DEVANE, R.; KLEIN, M. Multi-property fitting and parameterization of a coarse grained model for aqueous surfactants. *Mol. Simulat.*, v. 33, p. 27–36, 2007.

\_\_\_\_\_. Coarse-grained force field for ionic surfactants. *Soft Matter*, v. 7, p. 6178–6186, 2011.

STRIOLO, A. et al. Forces between aqueous nonuniformly charged colloids from molecular simulation. *J. Chem. Phys.*, v. 116, p. 7733–7743, 2002.

\_\_\_\_\_. Analytic calculation of phase diagrams for charged dipolar colloids with orientation-averaged pair potentials. *Phys. Chem. Chem. Phys.*, v. 5, p. 4851–4857, 2003.

SVERGUN, D.; BARBERATO, C.; KOCH, M. *CRY SOL* – a program to evaluate X-ray solution scattering of biological macromolecules from atomic coordinates. *J. Appl. Cryst.*, v. 28, p. 768–773, 1995.

SVERGUN, D. I.; PETOUKHOV, M. V.; KOCH, M. H. J. Determination of domain structure of proteins from X-ray solution scattering. *Biophys. J.*, v. 80, p. 2946–2953, 2001.

SVERGUN, D. I. et al. New developments in direct shape determination from small-angle scattering. 2. Uniqueness. *Acta Cryst. A*, v. 52, p. 419–426, 1996.

TAKADA, S. Coarse-grained molecular simulations of large biomolecules. *Curr. Opin. Struc. Biol.*, v. 22, p. 130–137, 2012.

TARDIEU, A. et al. Proteins in solution: from X-ray scattering intensities to interaction potentials. *J. Cryst. Growth*, v. 196, p. 193–203, 1999.

TAVARES, F. W.; PRAUSNITZ, J. M. Analytical calculation of phase diagrams for solutions containing colloids or globular proteins. *Colloid. Polym. Sci.*, v. 282, p. 620–632, 2004.

TAVARES, F. W.; SANDLER, S. I. Phase equilibria for the mean-force potential of globular protein solutions. *AIChE J.*, v. 43, p. 218–231, 1997.

TESSIER, P. et al. Predictive crystallization of ribonuclease A via rapid screening of osmotic second virial coefficients. *Proteins*, v. 50, p. 303–311, 2003.

TESTER, J. W.; MODELL, M. *Thermodynamics and its applications*. 3. ed. Upper Saddle River: Prentice Hall, 1997.

THIELE, E. Equation of state for hard spheres. *J. Chem. Phys.*, v. 39, p. 474–479, 1963.

- TOMÉ, T.; OLIVEIRA, M. J. *Dinâmica estocástica e irreversibilidade*. São Paulo: EDUSP, 2001.
- TOSI, M. P.; FUMI, F. G. Ionic sizes and Born repulsive parameters in the NaCl-type alkali halides - II The generalizes Huggins-Mayer form. *J. Phys. Chem. Solids*, v. 25, p. 45–52, 1964.
- TOZZINI, V. Coarse-grained models for proteins. *Curr. Opin. Struc. Biol.*, v. 15, p. 144–150, 2005.
- UEMATSU, M.; FRANCK, E. Static dielectric constant of water and steam. *J. Phys. Chem. Ref. Data*, v. 9, p. 1291–1306, 1980.
- URAHATA, S. M.; RIBEIRO, M. C. C. Collective excitations in an ionic liquid. *J. Chem. Phys.*, v. 124, p. 074513, 2006.
- VOICESCU, M.; IONESCU, S.; ANGELESCU, D. G. Spectroscopic and coarse-grained simulation studies of the BSA and HSA protein adsorption on silver nanoparticles. *J. Nanopart. Res.*, v. 14, p. 1174, 2012.
- WALDEN, P. Über organische lösungen und ionisierungsmittel. iii. teil: Innere reibung und deren zusammenhang mit dem leitvermögen. *Z. Phys. Chem.*, v. 55, p. 207–249, 1906.
- WERNESSEN, E.; JUNGWIRTH, P. Effect of water polarizability on the properties of solutions of polyvalent ions: simulations of aqueous sodium sulfate with different force fields. *J. Chem. Theory Comput.*, v. 6, p. 3233–3240, 2010.
- WERTHEIM, E. Exact solution of the Percus-Yevick integral equation for hard spheres. *Phys. Rev. Lett.*, v. 10, p. 321–323, 1963.
- WILLIAMS, T.; KELLEY, C.; many others. *Gnuplot 4.6: an interactive plotting program*. March 2012.
- WU, J.; PRAUSNITZ, J. Osmotic pressures of aqueous bovine serum albumin solutions at high ionic strength. *Fluid Phase Equilibr.*, v. 155, p. 139–154, 1999.
- WU, Y. C. et al. A dc method for the absolute determination of conductivities of the primary standard kcl solutions from 0°C to 50°C. *J. Res. Natl. Inst. Stand. Technol.*, v. 99, n. 3, p. 241–246, 1994.
- XU, W.; COOPER, E. I.; ANGELL, C. A. Ionic liquids: ion mobilities, glass temperatures, and fragilities. *J. Phys. Chem. B*, v. 107, n. 25, p. 6170–6178, 2003.
- YAMAGUCHI, T. et al. Interpretation of the variation of the walden product of ionic liquids with different alkyl chain lengths in terms of relaxation spectra. *J. Phys. Chem. B*, v. 117, n. 15, p. 4121–4126, 2013.
- YOSHIZAWA, M.; XU, W.; ANGELL, C. A. Ionic liquids by proton transfer: vapor pressure, conductivity, and the relevance of  $\Delta pK_a$  from aqueous solutions. *J. Am. Chem. Soc.*, v. 125, n. 50, p. 15411–15419, 2003.

ZHANG, F. et al. Protein interactions studied by SAXS: Effect of ionic strength and protein concentration for BSA in aqueous solutions. *J. Phys. Chem. B*, v. 111, p. 251–259, 2007.

ZHANG, Y.; CREMER, P. Interactions between macromolecules and ions: the Hofmeister series. *Curr. Opin. Chem. Biol.*, v. 10, p. 658–663, 2006.

## APPENDIX A - THE MANY BODY EFFECT

Considering that the radial distribution function may be written as an expansion on densities - the same approach used to derive pressure as a virial expansion (MCQUARRIE, 2000) - the following equation is obtained:

$$g(r, T, \rho) = \sum_{j=0}^{N-2} \rho^j g_j(r, T) \quad (\text{A.1})$$

By the definition of the potential of mean force (MCQUARRIE, 2000),  $W$ , one may rewrite Equation (A.1) as:

$$e^{-W(r, \rho)/k_B T} = \sum_{j=0}^{N-2} \rho^j e^{-W_j(r)/k_B T} \quad (\text{A.2})$$

or:

$$e^{-[W(r, \rho) - W_0(r)]/k_B T} = 1 + \sum_{j=1}^{N-2} \rho^j e^{-[W_j(r) - W_0(r)]/k_B T} \quad (\text{A.3})$$

where  $W_0$  is the two-body contribution. Taking the natural logarithm leads to:

$$W(r, \rho) = W_0(r) - k_B T \ln \left[ 1 + \sum_{j=1}^{N-2} \rho^j e^{-[W_j(r) - W_0(r)]/k_B T} \right] \quad (\text{A.4})$$

The logarithm value on the right side of Equation (A.4) is always positive, thus  $W$  is always lower than  $W_0$ ; they will be equal at infinite dilution. Thus, all many-body contributions are somehow attractive in relation to the two-body interaction potential, provided the radial distribution function can be written as an expansion on densities.

## APPENDIX B – CODE FOR HAYTER-PENFOLD STRUCTURE FACTOR

This code is written in C. It has been successfully compiled with gcc and has been successfully executed in Linux.

```

/*****
/* School of Engineering - University of São Paulo - Brazil */
/* Bioprocess Engineering Group - Department of Chemical Engineering */
/* Structure factor of Hayter and Pendold (1981) */
/* Developed by: Luís Fernando Mercier Franco */
/* Advisor: Prof. Dr. Pedro de Alcântara Pessoa Filho */
/* Processos FAPESP: 2011/22070-5 and 2013/01857-2 */
/* July 3rd, 2013 */
/* Reference: J.B. Hayter, J. Penfold, Mol. Phys. 42 (1981) 109-118. */
*****/

/*****
/* Supporting Information: */
/* */
/* 1) To use the program an input file is required. */
/* This file must contain the diameter, SIGMA, the packing fraction, ETA */
/* and also the values of KAPPA and GAMMA as described in the functional */
/* form of the interaction potential. Moreover, one must declare the */
/* number, N, of points for the calculation and the minimum (QMIN) and */
/* the maximum (QMAX) value of the wavevector, q. */
/* The following is an example, which structure must be obeyed */
/* */
/* SIGMA: */
/* 80.0 */
/* ETA: */
/* 0.05 */
/* KAPPA: */
/* 0.04 */
/* GAMMA: */
/* 10.0 */
*/

```



```

/*      N:                                          */
/*      1000                                       */
/*      QMIN_QMAX:                                 */
/*      0.01   0.5                               */
/*                                          */
/*  2) There is a single output file in which it is saved the wavevector, */
/*  q, on the first column and the calculated structure factor, S(q)      */
/*  on the second column.                                                 */
/*                                          */
/*  3) To compile, save this file as "HAYPEN.c", then                     */
/*  in the terminal:                                                       */
/*      > gcc HAYPEN.c -o HAYPEN.exe -lm                                  */
/*                                          */
/*  4) To run the code:                                                    */
/*      > echo INPUT_FILE_NAME OUTPUT_FILE_NAME | ./HAYPEN.exe           */
/*                                          */
/*  5) The authors do not accept any liability for the use of this        */
/*  program.                                                                */
/*****/

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <complex.h>

#define MAX 1000
#define NITERMAX 1000

void HP(double SIGMA, double ETA, double KAPPA, double GAMMA, double PP[4]);
double INTTRAP(double SIGMA, double ETA, double KAPPA, double GAMMA,
               double PP[4], double X);
double SKHP(double SIGMA, double ETA, double KAPPA, double GAMMA,
             double Q, double PP[4]);
double complex PXHORN(double complex AX[MAX], double complex X, int N);
void BR(double complex CX[MAX], double complex B[MAX], double complex ROOTS, int N);
double complex DERIVPX(double complex AX[MAX], double complex X, int N);
double complex NR(double complex AX[MAX], double complex X0, double TOL, int N);

int main(){
int J, N;
double SIGMA, ETA, KAPPA, GAMMA, QMIN, QMAX, DQ;
double Q[MAX], SQ[MAX], PP[4];
char nfile[40], get[40], nout[40];
FILE *ENT, *OUT;

    scanf("%s %s", nfile, nout);
    ENT = fopen(nfile, "r");
    fscanf(ENT, "%s", get);
    fscanf(ENT, "%lf", &SIGMA);
    fscanf(ENT, "%s", get);
    fscanf(ENT, "%lf", &ETA);
    fscanf(ENT, "%s", get);

```

```

fscanf(ENT, "%lf", &KAPPA);
fscanf(ENT, "%s", get);
fscanf(ENT, "%lf", &GAMMA);
fscanf(ENT, "%s", get);
fscanf(ENT, "%d", &N);
fscanf(ENT, "%s", get);
fscanf(ENT, "%lf", &QMIN);
fscanf(ENT, "%lf", &QMAX);
fclose(ENT);

Q[0] = QMIN;
DQ   = (QMAX - QMIN) / (1.0 * N);

for (J = 1; J < N; J++)
    Q[J] = Q[J-1] + DQ;

HP(SIGMA, ETA, KAPPA, GAMMA, PP);
for (J = 0; J < N; J++)
    SQ[J] = SKHP(SIGMA, ETA, KAPPA, GAMMA, Q[J], PP);

OUT = fopen(nout, "w");
for (J = 0; J < N; J++)
    fprintf(OUT, "%lf %lf\n", Q[J], SQ[J]);
fclose(OUT);

return 0;
}

void HP(double SIGMA, double ETA, double KAPPA, double GAMMA, double PP[4]){
    int K, N, M, J, L;
    double KMIN, KM2, DEL, DEL2, DEL4, ET2, ALPHA1, ALPHA2, ALPHA3;
    double BETA1, BETA2, BETA3, NU1, NU2, NU3, NU4, NU5, PHI1, PHI2, TAU1;
    double TAU2, TAU3, TAU4, TAU5, A1, A2, A3, B1, B2, B3, V1, V2, V3, P1;
    double P2, P3, T1, T2, T3, MU1, MU2, MU3, MU4, MU5, MU6, LBD1, LBD2, LBD3;
    double LBD4, LBD5, LBD6, W12, W13, W14, W15, W16, W23, W24, W25, W26, W34;
    double W35, W36, W45, W46, W56, W4, W3, W2, W1, W0, CHI, ROOT, F;
    double XONR, X, GX, TOL;
    double complex X0;
    double ALR[5], W[5];
    double complex B[5], ALPHA[5], C[5];

    KMIN   = KAPPA * SIGMA;
    KM2    = KMIN * KMIN;
    DEL    = 1.0 - ETA;
    DEL2   = DEL * DEL;
    DEL4   = DEL2 * DEL2;
    ET2    = ETA * ETA;
    ALPHA1 = -1.0 * (2.0 * ETA + 1.0) * DEL / KMIN;
    ALPHA2 = (14.0 * ET2 - 4.0 * ETA - 1.0) / KM2;
    ALPHA3 = 36.0 * ET2 / KM2 / KM2;
    BETA1  = -1.0 * (ET2 + 7.0 * ETA + 1.0) * DEL / KMIN;
    BETA2  = 9.0 * ETA * (ET2 + 4.0 * ETA - 2.0) / KM2;
    BETA3  = 12.0 * ETA * (2.0 * ET2 + 8.0 * ETA - 1.0) / KM2 / KM2;
    NU1    = -1.0 * (ET2 * ETA + 3.0 * ET2 + 45.0 * ETA + 5.0) * DEL / KMIN;
    NU2    = (2.0 * ET2 * ETA + 3.0 * ET2 + 42.0 * ETA - 20.0) / KM2;

```

```

NU3      = (2.0 * ET2 * ETA + 30.0 * ETA - 5.0) / KM2 / KM2;
NU4      = NU1 + 24.0 * ETA * KMIN * NU3;
NU5      = 6.0 * ETA * (NU2 + 4.0 * NU3);
PHI1     = 6.0 * ETA / KMIN;
PHI2     = DEL - 12.0 * ETA / KM2;
TAU1     = (ETA + 5.0) / (5.0 * KMIN);
TAU2     = (ETA + 2.0) / KM2;
TAU3     = -12.0 * ETA * GAMMA * exp(-1.0 * KMIN) * (TAU1 + TAU2);
TAU4     = 3.0 * ETA * KM2 * (TAU1 * TAU1 - TAU2 * TAU2);
TAU5     = 3.0 * ETA * (ETA + 8.0) / 10.0 - 2.0 * (2.0 * ETA + 1.0) *
          (2.0 * ETA + 1.0) / KM2;
A1       = (24.0 * ETA * GAMMA * exp(-1.0 * KMIN) * (ALPHA1 + ALPHA2 +
          (1.0 + KMIN) * ALPHA3) - (2.0 * ETA + 1.0) *
          (2.0 * ETA + 1.0)) / DEL4;
A2       = 24.0 * ETA * (ALPHA3 * (sinh(KMIN) - KMIN * cosh(KMIN)) +
          ALPHA2 * sinh(KMIN) - ALPHA1 * cosh(KMIN)) / DEL4;
A3       = 24.0 * ETA * ((2.0 * ETA + 1.0) * (2.0 * ETA + 1.0) / KM2 -
          DEL2 / 2.0 + ALPHA3 * (cosh(KMIN) - 1.0 - KMIN * sinh(KMIN)) -
          ALPHA1 * sinh(KMIN) + ALPHA2 * cosh(KMIN)) / DEL4;
B1       = (3.0 * ETA * (ETA + 2.0) * (ETA + 2.0) / 2.0 -
          12.0 * ETA * GAMMA * exp(-1.0 * KMIN) * (BETA1 + BETA2 +
          (1.0 + KMIN) * BETA3)) / DEL4;
B2       = 12.0 * ETA * (BETA3 * (KMIN * cosh(KMIN) - sinh(KMIN)) -
          BETA2 * sinh(KMIN) + BETA1 * cosh(KMIN)) / DEL4;
B3       = 12.0 * ETA * (DEL2 * (ETA + 2.0) / 2.0 - 3.0 * ETA * (ETA + 2.0) *
          (ETA + 2.0) / KM2 - BETA3 * (cosh(KMIN) - 1.0 - KMIN * sinh(KMIN)) +
          BETA1 * sinh(KMIN) - BETA2 * cosh(KMIN)) / DEL4;
V1       = ((2.0 * ETA + 1.0) * (ET2 - 2.0 * ETA + 10.0) / 4.0 -
          GAMMA * exp(-1.0 * KMIN) * (NU4 + NU5)) / (5.0 * DEL4);
V2       = (NU4 * cosh(KMIN) - NU5 * sinh(KMIN)) / (5.0 * DEL4);
V3       = ((ET2 * ETA - 6.0 * ET2 + 5.0) * DEL - 6.0 * ETA *
          (2.0 * ET2 * ETA - 3.0 * ET2 + 18.0 * ETA + 10.0) / KM2 +
          24.0 * ETA * NU3 + NU4 * sinh(KMIN) -
          NU5 * cosh(KMIN)) / (5.0 * DEL4);
P1       = (GAMMA * exp(-1.0 * KMIN) * (PHI1 - PHI2) * (PHI1 - PHI2) -
          (ETA + 2.0) / 2.0) / DEL2;
P2       = ((PHI1 * PHI1 + PHI2 * PHI2) * sinh(KMIN) +
          2.0 * PHI1 * PHI2 * cosh(KMIN)) / DEL2;
P3       = ((PHI1 * PHI1 + PHI2 * PHI2) * cosh(KMIN) +
          2.0 * PHI1 * PHI2 * sinh(KMIN) + PHI1 * PHI1 - PHI2 * PHI2) / DEL2;
T1       = TAU3 + TAU4 * A1 + TAU5 * B1;
T2       = TAU4 * A2 + TAU5 * B2 + 12.0 * ETA * (TAU1 * cosh(KMIN) -
          TAU2 * sinh(KMIN));
T3       = TAU4 * A3 + TAU5 * B3 + 12.0 * ETA * (TAU1 * sinh(KMIN) -
          TAU2 * (cosh(KMIN) - 1.0)) - 2.0 * ETA * (ETA + 10.0) / 5.0 - 1.0;
MU1      = T2 * A2 - 12.0 * ETA * V2 * V2;
MU2      = T1 * A2 + T2 * A1 - 24.0 * ETA * V1 * V2;
MU3      = T2 * A3 + T3 * A2 - 24.0 * ETA * V2 * V3;
MU4      = T1 * A1 - 12.0 * ETA * V1 * V1;
MU5      = T1 * A3 + T3 * A1 - 24.0 * ETA * V1 * V3;
MU6      = T3 * A3 - 12.0 * ETA * V3 * V3;
LBD1     = 12.0 * ETA * P2 * P2;
LBD2     = 24.0 * ETA * P1 * P2 - 2.0 * B2;
LBD3     = 24.0 * ETA * P2 * P3;
LBD4     = 12.0 * ETA * P1 * P1 - 2.0 * B1;
LBD5     = 24.0 * ETA * P1 * P3 - 2.0 * B3 - KM2;

```

```

LBD6  = 12.0 * ETA * P3 * P3;
W12   = MU1 * LBD2 - MU2 * LBD1;
W13   = MU1 * LBD3 - MU3 * LBD1;
W14   = MU1 * LBD4 - MU4 * LBD1;
W15   = MU1 * LBD5 - MU5 * LBD1;
W16   = MU1 * LBD6 - MU6 * LBD1;
W23   = MU2 * LBD3 - MU3 * LBD2;
W24   = MU2 * LBD4 - MU4 * LBD2;
W25   = MU2 * LBD5 - MU5 * LBD2;
W26   = MU2 * LBD6 - MU6 * LBD2;
W34   = MU3 * LBD4 - MU4 * LBD3;
W35   = MU3 * LBD5 - MU5 * LBD3;
W36   = MU3 * LBD6 - MU6 * LBD3;
W45   = MU4 * LBD5 - MU5 * LBD4;
W46   = MU4 * LBD6 - MU6 * LBD4;
W56   = MU5 * LBD6 - MU6 * LBD5;
W4     = W16 * W16 - W13 * W36;
W3     = 2.0 * W16 * W15 - W13 * (W35 + W26) - W12 * W36;
W2     = W15 * W15 + 2.0 * W16 * W14 - W13 * (W34 + W25) -
        W12 * (W35 + W26);
W1     = 2.0 * W15 * W14 - W13 * W24 - W12 * (W34 + W25);
W0     = W14 * W14 - W12 * W24;
W[0]   = W0;
W[1]   = W1;
W[2]   = W2;
W[3]   = W3;
W[4]   = W4;

TOL    = pow(10.0, -3.0);
N      = 4;
XONR   = 1.0;

for (K = 0; K <= N; K++)
    B[K] = W[K] + 0.0 * I;
X0 = XONR + (XONR / 1000.0) * I;
M = N;
for (K = 0; K < N; K++){
    ALPHA[K] = NR(B, X0, TOL, N); /* Newton-Raphson */
    for (J = 0; J <= M; J++)
        C[J] = B[J];
    if (K < N - 1)
        BR(C, B, ALPHA[K], M); /* Briott-Ruffini */
    B[M] = 0.0;
    M = M - 1;
}
J = 0;
for (K = 0; K < N; K++){
    if (fabs(cimag(ALPHA[K])) < 5.0 * TOL){
        ALR[J] = creal(ALPHA[K]); /* real roots */
        J = J + 1;
    }
}

/* Search for the right root */
for (K = 0; K < J; K++){
    F = ALR[K];

```

```

        PP[0] = F;
        PP[1] = (W16 * F * F + W15 * F + W14) / (W13 * F + W12);
        PP[2] = B1 + B2 * PP[1] + B3 * F;
        PP[3] = A1 + A2 * PP[1] + A3 * F;
        CHI = 0.0;
        for (L = 0; L < 10; L++){
            X = 1.0 * (L + 1) / 10.0;
            GX = 1.0 + 1.0 / (12.0 * M_PI * ETA * X) *
INTTRAP(SIGMA, ETA, KAPPA, GAMMA, PP, X);
            CHI = CHI + GX * GX;
        }
        if (CHI <= 1000.0 * TOL){
            ROOT = ALR[K];
            K = K + J + 1;
        }
    }
    F = ROOT;
    PP[0] = F;
    PP[1] = (W16 * F * F + W15 * F + W14) / (W13 * F + W12);
    PP[2] = B1 + B2 * PP[1] + B3 * F;
    PP[3] = A1 + A2 * PP[1] + A3 * F;
}

double INTTRAP(double SIGMA, double ETA, double KAPPA, double GAMMA,
               double PP[4], double X){
    int K, N;
    double ALPHA, A, B, H, Q, SK, SUM;

    A = pow(10.0, -6.0);
    B = 100.0;
    N = 500;
    H = (B - A) / (1.0 * (N - 1));
    ALPHA = A;
    Q = ALPHA / SIGMA;
    SK = SKHP(SIGMA, ETA, KAPPA, GAMMA, Q, PP);
    SUM = (SK - 1.0) * ALPHA * sin(ALPHA * X);
    ALPHA = ALPHA + H;

    for (K = 1; K < N; K++){
        Q = ALPHA / SIGMA;
        SK = SKHP(SIGMA, ETA, KAPPA, GAMMA, Q, PP);
        SUM = SUM + 2.0 * (SK - 1.0) * ALPHA * sin(ALPHA * X);
        ALPHA = ALPHA + H;
    }

    Q = ALPHA / SIGMA;
    SK = SKHP(SIGMA, ETA, KAPPA, GAMMA, Q, PP);
    SUM = SUM + (SK - 1.0) * ALPHA * sin(ALPHA * X);
    SUM = SUM * H / 2.0;

    return SUM;
}

double SKHP(double SIGMA, double ETA, double KAPPA, double GAMMA,
            double Q, double PP[4]){

```

```

double KMIN, F, C, B, A, K, K2, K3, TER1, TER2, TER3, TER4, TER5, TER6;
double TER7, AK, SK;

KMIN  = KAPPA * SIGMA;
F      = PP[0];
C      = PP[1];
B      = PP[2];
A      = PP[3];
K      = Q * SIGMA;
K2     = K * K;
K3     = K2 * K;
TER1   = A * (sin(K) - K * cos(K)) / K3;
TER2   = B * ((2.0 / K2 - 1.0) * K * cos(K) + 2.0 * sin(K) - 2.0 / K) / K3;
TER3   = ETA * A * (24.0 / K3 + 4.0 * (1.0 - 6.0 / K2) * sin(K) -
    (1.0 - 12.0 / K2 + 24.0 / K2 / K2) * K * cos(K)) / (2.0 * K3);
TER4   = C * (KMIN * cosh(KMIN) * sin(K) - K * sinh(KMIN) * cos(K)) /
    K / (K2 + KMIN * KMIN);
TER5   = F * (KMIN * sinh(KMIN) * sin(K) -
    K * (cosh(KMIN) * cos(K) - 1.0)) / K / (K2 + KMIN * KMIN);
TER6   = F * (cos(K) - 1.0) / K2;
TER7   = -1.0 * GAMMA * exp(-1.0 * KMIN) * (KMIN * sin(K) +
    K * cos(K)) / K / (K2 + KMIN * KMIN);
AK     = TER1 + TER2 + TER3 + TER4 + TER5 + TER6 + TER7;
SK     = 1.0 / (1.0 - 24.0 * ETA * AK);

return SK;
}

double complex PXHORN(double complex AX[MAX], double complex X, int N){
    int K;
    double complex PX; complex AUX;

    AUX = AX[N] + 0.0 * I;
    for (K = N; K >= 1; K--){
        PX = AUX * X + AX[K-1];
        AUX = PX;
    }

    return PX;
}

void BR(double complex CX[MAX], double complex B[MAX], double complex ROOTS, int N){
    int K;
    double complex R;
    double complex S;

    R = CX[N];
    for (K = N; K > 0; K--){
        S = CX[K-1];
        B[K-1] = R;
        R = S + ROOTS * R;
    }
}

double complex DERIVPX(double complex AX[MAX], double complex X, int N){
    int K, M;

```

```

    double complex DER;
    double complex B[MAX];
    M = N - 1;
    for (K = 0; K <= M; K++)
        B[K] = AX[K+1] * (1.0 * K + 1.0);
    DER = PXHORN(B, X, M);

    return DER;
}

double complex NR(double complex AX[MAX], double complex X0, double TOL, int N){
    int K, NITER;
    double complex XOLD;
    double complex ALPHA;
    double complex XNEW;
    XOLD = X0;
    for (K = 0; K < NITERMAX; K++){
        XNEW = XOLD - PXHORN(AX, XOLD, N) / DERIVPX(AX, XOLD, N);
        if (cabs(XNEW - XOLD) < TOL && cabs(PXHORN(AX, XNEW, N)) < TOL){
            ALPHA = XNEW;
            NITER = K;
            K      = NITERMAX + 1;
        }
        else
            XOLD = XNEW;
    }

    return ALPHA;
}

```

## APPENDIX C – CODES FOR MONTE CARLO SIMULATIONS

The codes presented in this section were developed to be used in the same manner and always in Lennard-Jones units. Therefore, an input file is common to all of them. Such input file called master file is a list of desired file names to be used by the various codes. The structure of this master file is simply a sequence of file names. The following is an example:

```
mc1.log  
conf1.xyz  
mc1.sci  
ratio1.dat  
pos1.xyz  
thermo1.dat  
gofr1.dat  
sofq1.dat  
zpd1.dat
```

Table 8 presents the meaning of each file. While the name of the files are to be freely chosen, the extension as well the order of presentation is not.



Table 8: Description of the list of files in the master file for Monte Carlo simulations

Name in the example	Description
mc1.log	This is a log file for NVT simulation.
conf1.xyz	This is an initial FCC configuration file written in “.xyz” fashion.
mc1.sci	This is control file with the proper information for the simulation. An example will be given later.
ratio1.dat	This file stores the value of acceptance ratio.
pos1.xyz	This is “.xyz” file which stores all the equilibrium configurations of the simulation.
thermo1.dat	This file contains the thermodynamic information at each step. The reduced potential, the compressibility factor and the reduced pressure.
gofr1.dat	This is an output file with the radial distribution function.
sofq1.dat	This is an output file with the structure factor.
zpd1.dat	This is an output file with local particle density along $z$ -direction.

An example of the “.sci” file is:

```

System=          LJ_fluid
Lx=              12.0
Ly=              12.0
Lz=              40.0
Temperature=     1.10
Equil_steps=     40000
Prod_steps=      40000
DRMAX=           0.3

```

```
Rcutoff=          3.0
NADJUST=         5000
```

Table 9 presents the meaning of each command.

Table 9: Description of each command in “sci” files.

Command	Description
System	The name of the system to be simulated.
Lx	The x-dimension of the box in LJ units.
Ly	The y-dimension of the box in LJ units.
Lz	The z-dimension of the box in LJ units.
Temperature	The temperature in LJ units.
Equil_steps	Number which multiplied by the number of particles gives the steps for equilibration. No configuration is stored during equilibration.
Prod_steps	Number which multiplied by the number of particles gives the steps for production.
DRMAX	Initial maximum displacement in LJ units.
Rcutoff	Potential radius of cut-off in LJ units.
NADJUST	Number of steps to adjust the maximum displacement.

## C.1 Code to generate a FCC crystal

```

/*****
/* School of Engineering - University of São Paulo - Brazil */
/* Bioprocess Engineering Group - Department of Chemical Engineering */
/* Generation of a FCC crystal in .xyz file */
/* Developed by: Luís Fernando Mercier Franco */
/* Advisor: Prof. Dr. Pedro de Alcântara Pessoa Filho */
/* Processos FAPESP: 2011/22070-5 and 2013/01857-2 */
/* November 28th, 2012 */
/* Reference: M.P. Allen, D.J. Tildesley, Computer Simulation of Liquids, */
/* Oxford University Press, New York, 1987. */
*****/

/*****
/* Supporting Information: */
/* */
/* 1) To compile, save this file as "FCC.c", then */
/* in the terminal: */
/* > gcc FCC.c -o FCC.exe -lm */
/* */
/* 2) To run the code: */
/* > echo NUMBER_OF_PARTICLES MASTER_FILE_NAME | ./FCC.exe */
*****/

```

```

/*                                                                    */
/*  3) The authors do not accept any liability for the use of this    */
/*  program.                                                            */
/*                                                                    */
/*  Note: if Lz is different from Lx (or Ly) the cube will be placed at */
/*  the very right side of the box.                                     */
/*                                                                    */
/*****

/*****
#include <stdio.h>
#include <math.h>

#define MAX 3000

main(){
    int I, J, K, N, NE, CONT;
    double RX[MAX], RY[MAX], RZ[MAX];
    double RHO, L, BOXL, LX, LY, LZ, DISZ;
    double KB, NA;
    char get[100], LOGFILE[40], CONFFILE[40], SCIFILE[40];
    char MFILE[40], SYS[100];
    FILE *IN, *OUT;

    scanf("%d %s", &N, MFILE);
    IN = fopen(MFILE, "r");
    fscanf(IN, "%s", LOGFILE);
    fscanf(IN, "%s", CONFFILE);
    fscanf(IN, "%s", SCIFILE);
    fclose(IN);

    IN = fopen(SCIFILE, "r");
    fscanf(IN, "%s", get);
    fscanf(IN, "%s", SYS);
    fscanf(IN, "%s", get);
    fscanf(IN, "%lf", &LX);
    fscanf(IN, "%s", get);
    fscanf(IN, "%lf", &LY);
    fscanf(IN, "%s", get);
    fscanf(IN, "%lf", &LZ);
    fclose(IN);

    NE = (int)cbrt(N / 4);
    RHO = (1.0 * N) / LX / LX / LX;
    L = cbrt(4.0 / RHO);

    CONT = 0;
    for (I = 0; I < NE; I++){
        for (J = 0; J < NE; J++){
            for (K = 0; K < NE; K++){
                RX[CONT] = (1.0 * I) * L;
                RY[CONT] = (1.0 * J) * L;
                RZ[CONT] = (1.0 * K) * L;
                CONT = CONT + 1;
            }
        }
    }
}

```

```

for (I = 0; I < NE; I++){
    for (J = 0; J < NE; J++){
        for (K = 0; K < NE; K++){
            RX[CONT] = L / 2.0 + (1.0 * I) * L;
            RY[CONT] = L / 2.0 + (1.0 * J) * L;
            RZ[CONT] = (1.0 * K) * L;
            CONT      = CONT + 1;
        }
    }
}

for (I = 0; I < NE; I++){
    for (J = 0; J < NE; J++){
        for (K = 0; K < NE; K++){
            RX[CONT] = L / 2.0 + (1.0 * I) * L;
            RY[CONT] = (1.0 * J) * L;
            RZ[CONT] = L / 2.0 + (1.0 * K) * L;
            CONT      = CONT + 1;
        }
    }
}

for (I = 0; I < NE; I++){
    for (J = 0; J < NE; J++){
        for (K = 0; K < NE; K++){
            RX[CONT] = (1.0 * I) * L;
            RY[CONT] = L / 2.0 + (1.0 * J) * L;
            RZ[CONT] = L / 2.0 + (1.0 * K) * L;
            CONT      = CONT + 1;
        }
    }
}

BOXL = cbrt((1.0 * N) / RHO);
DISZ = (LZ - LX) / 2.0;

for (I = 0; I < N; I++){
    RX[I] = RX[I] - BOXL / 2.0;
    RY[I] = RY[I] - BOXL / 2.0;
    RZ[I] = RZ[I] - BOXL / 2.0 + DISZ;
}

OUT = fopen(CONFFILE, "w");
fprintf(OUT, "%d\n", N);
fprintf(OUT, "%s\n", SYS);
for (I = 0; I < N; I++){
    fprintf(OUT, "C %e ", RX[I]);
    fprintf(OUT, "%e ", RY[I]);
    fprintf(OUT, "%e\n", RZ[I]);
}
fclose(OUT);

return 0;
}

```

## C.2 Code for NVT ensemble Monte Carlo simulation

```

/*****
/* School of Engineering - University of São Paulo - Brazil */
/* Bioprocess Engineering Group - Department of Chemical Engineering */
/* NVT Monte Carlo code for Lennard-Jones particles */
/* Developed by: Luís Fernando Mercier Franco */
/* Advisor: Prof. Dr. Pedro de Alcântara Pessôa Filho */
/* Processos FAPESP: 2011/22070-5 and 2013/01857-2 */
/* July 27th, 2013 */
/* Reference: M.P. Allen, D.J. Tildesley, Computer Simulation of Liquids, */
/* Oxford University Press, New York, 1987. */
*****/

/*****
/* Supporting Information: */
/* */
/* 1) To compile, save this file as "NVT.c", then */
/* in the terminal: */
/* > gcc NVT.c -o NVT.exe -lm */
/* */
/* 2) To run the code: */
/* > echo MASTER_FILE_NAME | ./NVT.exe */
/* */
/* 3) The authors do not accept any liability for the use of this */
/* program. */
*****/

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>

#define TRUE 1
#define MAX 3000

/*****
/* Declaration of subroutines */
*****/

/* Subroutine to calculate the potential energy and the virial term */
/* of a given configuration. */
void TOTENERGY(double RX[MAX], double RY[MAX], double RZ[MAX],
               double RCUTSQ, double LX, double LY, double LZ,
               double SIGSQ, double *V, double *W, int N);

/* Subroutine to calculate the potential energy and the virial term */
/* of a given particle in relation to others. */
void ENERGY(double RX[MAX], double RY[MAX], double RZ[MAX],
              double RXI, double RYI, double RZI,
              double RCUTSQ, double LX, double LY, double LZ,

```

```

        double SIGSQ, double *V, double *W, int I, int N);

/* Subroutine to calculate the square distance between two particles */
double DISTSQ(double RXI, double RYI, double RZI, double LX, double LY,
              double LZ, double RXJ, double RYJ, double RZJ);

/* Subroutine to calculate the virial term between two particles */
double VIRIJ(double RIJSQ, double SIGSQ);

/* Subroutine to calculate the potential between two particles */
double POTIJ(double RIJSQ, double SIGSQ);

/* Subroutine to generate a random number between 0 and 1 */
double RANF(int SEED[1]);

/*****

/*****
/* Main routine                                     */
/*****

main(){
    int STEP, NSTEPP, NSTEPE, I, J, N, NADJUST, NACCPT, CONT;
    int SEED[1];
    double SUMVSQ, V, RXIOLD, RYIOLD, RZIOLD, RXINEW, RYINEW, RZINEW;
    double RIJSQ, RIJ, RCUT, RHO, T, RATIO, VN, VOLD, VNEW, DV, DVB, DRMAX;
    double W, WOLD, WNEW, DW, Z, PRES, LX, LY, LZ;
    double RX[MAX], RY[MAX], RZ[MAX];
    char get[100], LOGFILE[40], CONFFILE[40], SCIFILE[40];
    char MFILE[40], RATFILE[40], POSFILE[40], THERMOFILE[40], SYS[100];
    time_t t1, t2;
    FILE *IN, *OUT1, *OUT2, *OUT3, *LOGB;

    (void) time(&t1);

    scanf("%s", MFILE);
    IN = fopen(MFILE, "r");
    fscanf(IN, "%s", LOGFILE);
    fscanf(IN, "%s", CONFFILE);
    fscanf(IN, "%s", SCIFILE);
    fscanf(IN, "%s", RATFILE);
    fscanf(IN, "%s", POSFILE);
    fscanf(IN, "%s", THERMOFILE);
    fclose(IN);

    LOGB = fopen(LOGFILE, "w");
    fprintf(LOGB, "\n");
    fprintf(LOGB, "\n\t    SCHOOL OF ENGINEERING OF UNIVERSITY OF SAO PAULO \n");
    fprintf(LOGB, "\n\t          DEPARTAMENT OF CHEMICAL ENGINEERING          \n");
    fprintf(LOGB, "\n\t          BIOPROCESS ENGINEERING GROUP          \n");
    fprintf(LOGB, "\n\t          ..... ");
    fprintf(LOGB, "\n\t          :          MONTE CARLO          :          ");
    fprintf(LOGB, "\n\t          :.....: \n");
    fprintf(LOGB, " \n\t Developed by Luis Fernando Mercier Franco");
    fprintf(LOGB, " \n\t Advisor: Prof. Dr. Pedro de Alcantara Pessoa Filho");

```

```

fprintf(LOGB, "  \n\t Processo FAPESP (2011/22070-5)\n\n");

/*****
/* Reading input files
*****/
IN = fopen(CONFFILE, "r");
fscanf(IN, "%d", &N);
fscanf(IN, "%s", get);
for (J = 0; J < N; J++)
    fscanf(IN, "%s %lf %lf %lf", get, &RX[J], &RY[J], &RZ[J]);
fclose(IN);

IN = fopen(SCIFILE, "r");
fscanf(IN, "%s", get);
fscanf(IN, "%s", SYS);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &LX);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &LY);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &LZ);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &T);
fscanf(IN, "%s", get);
fscanf(IN, "%d", &NSTEPE);
fscanf(IN, "%s", get);
fscanf(IN, "%d", &NSTEPP);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &DRMAX);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &RCUT);
fscanf(IN, "%s", get);
fscanf(IN, "%d", &NADJUST);
fclose(IN);

RHO = (1.0 * N) / LX / LY / LZ;

/*****
Simulation Parameters: \n");
fprintf(LOGB, " Number of particles: %d\n", N);
fprintf(LOGB, " DRMAX*: %.2lf\n", DRMAX);
fprintf(LOGB, " RCUTOFF*: %.2lf\n", RCUT);
fprintf(LOGB, " RHO*: %.2lf\n\n\n", RHO);
fprintf(LOGB, " T*: %.2lf\n\n\n", T);

/*****
/* Inicialization of variables
*****/
int NSTEP      = NSTEPP + NSTEPE;
double RCUTSQ  = RCUT * RCUT;
double BOXL    = LX * LY * LZ;
double VOL     = BOXL * BOXL * BOXL;
double SIGSQ   = 1.0;
double SR3     = 1.0 / RCUT / RCUT / RCUT;

```

```

double SR9      = SR3 * SR3 * SR3;
double VLRC12   = 8.0 * M_PI * RHO * (1.0 * N) * SR9 / 9.0;
double VLRC6    = -8.0 * M_PI * RHO * (1.0 * N) * SR3 / 3.0;
double VLRC     = VLRC12 + VLRC6;
double WLRC12   = 4.0 * VLRC12;
double WLRC6    = 2.0 * VLRC6;
double WLRC     = WLRC12 + WLRC6;
SEED[0]        = 352450618;
/*****/

/*****/
/* Generation of new configuration (Metropolis algorithm) */
/*****/
TOTENERGY(RX, RY, RZ, RCUTSQ, LX, LY, LZ, SIGSQ, &V, &W, N);
V = V + VLRC;
W = W + WLRC;
OUT1  = fopen(RATFILE, "w");
OUT2  = fopen(POSFILE, "w");
OUT3  = fopen(THERMOFILE, "w");
NACCP = 0;
CONT  = 0;
for (STEP = 1; STEP <= NSTEP; STEP++){
    for (I = 0; I < N; I++){

        RXIOLD = RX[I];
        RYIOLD = RY[I];
        RZIOLD = RZ[I];

        /* calculates the potential energy of OLD configuration */
        ENERGY(RX, RY, RZ, RXIOLD, RYIOLD, RZIOLD, RCUTSQ,
                LX, LY, LZ, SIGSQ, &VOLD, &WOLD, I, N);

        /* moves particle i */
        RXINEW = RXIOLD + (2.0 * RANF(SEED) - 1.0) * DRMAX;
        RYINEW = RYIOLD + (2.0 * RANF(SEED) - 1.0) * DRMAX;
        RZINEW = RZIOLD + (2.0 * RANF(SEED) - 1.0) * DRMAX;

        /* selects central image */
        RXINEW = RXINEW - LX * (1.0 * round(RXINEW / LX));
        RYINEW = RYINEW - LY * (1.0 * round(RYINEW / LY));
        RZINEW = RZINEW - LZ * (1.0 * round(RZINEW / LZ));

        /* calculates the potential energy of NEW configuration */
        ENERGY(RX, RY, RZ, RXINEW, RYINEW, RZINEW, RCUTSQ,
                LX, LY, LZ, SIGSQ, &VNEW, &WNEW, I, N);

        /* acceptance criterion */
        DV = VNEW - VOLD;
        DW = WNEW - WOLD;
        DVB = DV / T;

        if (DVB < 75.0){
            if (DV <= 0.0){
                V      = V + DV;
                W      = W + DW;
                RX[I]  = RXINEW;
            }
        }
    }
}

```



```

        RY[I] = RYINew;
        RZ[I] = RZINew;
        NACCPT = NACCPT + 1;
    }
    else{
        if (exp(-1.0 * DVB) > RANF(SEED)){
            V = V + DV;
            W = W + DW;
            RX[I] = RXINew;
            RY[I] = RYINew;
            RZ[I] = RZINew;
            NACCPT = NACCPT + 1;
        }
    }
}

VN = V / (1.0 * N);
CONT = CONT + 1;

PRES = RHO * T + W / VOL;
Z = PRES / RHO / T;

if (CONT % NADJUST == 0){
    RATIO = (1.0 * NACCPT) / (1.0 * NADJUST);
    fprintf(OUT1, "%d %lf %lf\n", CONT, RATIO, DRMAX);
    if (RATIO > 0.45)
        DRMAX = DRMAX * 1.05;
    else
        DRMAX = DRMAX * 0.95;
    NACCPT = 0;
}

}
fprintf(OUT3, "\t %d      %lf %lf %lf\n", STEP, VN, Z, PRES);
printf("\t %d      %lf %lf %lf\n", STEP, VN, Z, PRES);

/* save configurations */
if (CONT > NSTEPE * N){
    fprintf(OUT2, "%d\n", N);
    fprintf(OUT2, "%s\n", SYS);
    for (J = 0; J < N; J++)
        fprintf(OUT2, "C %e %e %e\n", RX[J], RY[J], RZ[J]);
}

}

fclose(OUT1);
fclose(OUT2);
fclose(OUT3);
/*****/

(void) time(&t2);
printf("\n\n Execution time = %ld seconds\n", (int) t2-t1);
fprintf(LOGB, "\n\n Execution time = %ld seconds\n", (int) t2-t1);
/*****/

return 0;

```

```

}

void TOTENERGY(double RX[MAX], double RY[MAX], double RZ[MAX],
               double RCUTSQ, double LX, double LY, double LZ, double SIGSQ,
               double *V, double *W, int N){

    int I, J;
    double RXI, RYI, RZI, RIJSQ;

    *V = 0.0;
    *W = 0.0;
    for (I = 0; I < N; I++){
        RXI = RX[I];
        RYI = RY[I];
        RZI = RZ[I];
        for (J = I + 1; J < N; J++){
            RIJSQ = DISTSQ(RXI, RYI, RZI, LX, LY, LZ,
                           RX[J], RY[J], RZ[J]);
            /* imposes the radius of cut-off */
            if (RIJSQ <= RCUTSQ){
                *V = *V + POTIJ(RIJSQ, SIGSQ);
                *W = *W + VIRIJ(RIJSQ, SIGSQ);
            }
        }
    }
}

void ENERGY(double RX[MAX], double RY[MAX], double RZ[MAX],
              double RXI, double RYI, double RZI,
              double RCUTSQ, double LX, double LY, double LZ, double SIGSQ,
              double *V, double *W, int I, int N){

    int J;
    double RIJSQ;

    *V = 0.0;
    *W = 0.0;
    for (J = 0; J < N; J++){
        if (I != J){
            RIJSQ = DISTSQ(RXI, RYI, RZI, LX, LY, LZ, RX[J], RY[J], RZ[J]);
            /* imposes the radius of cut-off */
            if (RIJSQ <= RCUTSQ){
                *V = *V + POTIJ(RIJSQ, SIGSQ);
                *W = *W + VIRIJ(RIJSQ, SIGSQ);
            }
        }
    }
}

double DISTSQ(double RXI, double RYI, double RZI, double LX, double LY,
               double LZ, double RXJ, double RYJ, double RZJ){

    /* calculates distance in x, y and z for each ij pair */
    double RXIJ = RXI - RXJ;

```

```

double RYIJ = RYI - RYJ;
double RZIJ = RZI - RZJ;
/* applies minimum image convention */
RXIJ = RXIJ - LX * (1.0 * round(RXIJ / LX));
RYIJ = RYIJ - LY * (1.0 * round(RYIJ / LY));
RZIJ = RZIJ - LZ * (1.0 * round(RZIJ / LZ));
/* calculates the square distance between i and j */
double RIJSQ = RXIJ * RXIJ + RYIJ * RYIJ + RZIJ * RZIJ;

return RIJSQ;
}

double VIRIJ(double RIJSQ, double SIGSQ){

    /* calculates the virial */
    double DIST = RIJSQ / SIGSQ;
    double SR2 = 1.0 / DIST;
    double SR6 = SR2 * SR2 * SR2;
    double WIJ = 48.0 * SR6 * (SR6 - 0.5) / 3.0;

    return WIJ;
}

double POTIJ(double RIJSQ, double SIGSQ){

    /* calculates the interparticle potential (Lennard-Jones) */
    double DIST = RIJSQ / SIGSQ;
    double SR2 = 1.0 / DIST;
    double SR6 = SR2 * SR2 * SR2;
    double VIJ = 4.0 * SR6 * (SR6 - 1.0);

    return VIJ;
}

double RANF(int SEED[1]){

    int L = 1029;
    int C = 221591;
    int M = 1048576;

    SEED[0] = (SEED[0] * L + C) % M;
    double FUN = (1.0 * SEED[0]) / (1.0 * M);

    return FUN;
}

```

## C.3 Code to calculate the structure factor and the radial distribution function

```

/*****
/* School of Engineering - University of São Paulo - Brazil */
/* Bioprocess Engineering Group - Department of Chemical Engineering */
/* Calculus of g(r) and S(q) */
/* Developed by: Luís Fernando Mercier Franco */
/* Advisor: Prof. Dr. Pedro de Alcântara Pessôa Filho */
/* Processos FAPESP: 2011/22070-5 and 2013/01857-2 */
/* February 7th, 2014 */
/* Reference: D. Frenkel, R. J. Vos, C. G. Kruif, A. Vrij, J. Chem. Phys., 84*/
/*           (1986) 4625-4630. */
/*           L. Cannavaciolo, C. Sommer, J. S. Pedersen, P. Schurtenberger,*/
/*           Phys. Rev. E, 62 (2000) 5409-5419. */
*****/

/*****
/* Supporting Information: */
/* */
/* 1) To compile, save this file as "SQ.c", then */
/* in the terminal: */
/* > gcc SQ.c -o SQ.exe -lm */
/* */
/* 2) To run the code: */
/* > echo NUBER_OF_PARTICLES MASTER_FILE_NAME | ./SQ.exe */
/* */
/* 3) The authors do not accept any liability for the use of this */
/* program. */
/* */
/* Note: This code requires a file called "dir.in" (at the same directory*/
/* as the executable file) with the 13 independent directions. */
/* Such file must be exactly as the following example: */
/* */
/*      1.0 0.0 0.0 */
/*      0.0 1.0 0.0 */
/*      0.0 0.0 1.0 */
/*      1.0 1.0 0.0 */
/*      1.0 0.0 1.0 */
/*      0.0 1.0 1.0 */
/*      -1.0 1.0 0.0 */
/*      1.0 0.0 -1.0 */
/*      0.0 1.0 -1.0 */
/*      1.0 1.0 1.0 */
/*      -1.0 1.0 1.0 */
/*      1.0 -1.0 1.0 */
/*      1.0 1.0 -1.0 */
*****/

#include <stdlib.h>
#include <stdio.h>
#include <math.h>

```

```

#include <time.h>

#define MAX      3000
#define MAXP      40
#define MAXP2     80
#define MAXP3    120
#define MAXBIN   300

/*****
/* Declaration of subroutines
*****/

/* calculates the histogram for radial distribution function */
void GOFr(double RX[MAX], double RY[MAX], double RZ[MAX], double BOXL,
          double DELR, int HIST[MAX], int N);

/* calculates the structure factor of a configuration */
void SQ(double RX[MAX], double RY[MAX], double RZ[MAX], double BOXL,
        double S[MAXP3], double MD[13][3], int N);

/* feeds the direction matrix for the structure factor calculation */
void DIRECTIONS(double MD[13][3]);

/* arranges the scattering vector in order */
void ORDENA(double Q[MAXP3], double SM[MAXP3], double SIG[MAXP3]);

/*****
/* Main function
*****/

main(){
    int STEP, NSTEPE, NSTEP, I, J, N, P, PP, BIN;
    int HIST[MAX];
    double BOXL, TR, LX, LY, LZ, RHO, VOL, DELR, RLOWER, RUPPER, NIDEAL;
    double RX[MAX], RY[MAX], RZ[MAX], S[MAXP3], Q[MAXP3], MD[13][3];
    double SM[MAXP3], SIG[MAXP3], GR[MAX];
    char get[100], LOGFILE[40], CONFFILE[40], SCIFILE[40];
    char MFILE[40], RATFILE[40], POSFILE[40], THERMOFILE[40], GRFILE[40];
    char SQFILE[40], SYS[100];
    time_t t1, t2;
    FILE *IN, *OUT;

    (void) time(&t1);

    /*****
    /* Reads input files
    *****/
    scanf("%d %s", &N, MFILE);
    IN = fopen(MFILE, "r");
    fscanf(IN, "%s", LOGFILE);
    fscanf(IN, "%s", CONFFILE);
    fscanf(IN, "%s", SCIFILE);
    fscanf(IN, "%s", RATFILE);
    fscanf(IN, "%s", POSFILE);

```

```

fscanf(IN, "%s", THERMOFILE);
fscanf(IN, "%s", GRFILE);
fscanf(IN, "%s", SQFILE);
fclose(IN);

IN = fopen(SCIFILE, "r");
fscanf(IN, "%s", get);
fscanf(IN, "%s", SYS);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &LX);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &LY);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &LZ);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &TR);
fscanf(IN, "%s", get);
fscanf(IN, "%d", &NSTEPE);
fscanf(IN, "%s", get);
fscanf(IN, "%d", &NSTEP);
fclose(IN);

/*****/

/*****/
/* Initialization of variables */
/*****/
BOXL = LX * LY * LZ;
VOL = BOXL * BOXL * BOXL;
DELR = BOXL / 2.0 / (1.0 * MAXBIN);
for (BIN = 1; BIN < MAXBIN; BIN++)
    HIST[BIN] = 0;
DIRECTIONS(MD);
for (P = 0; P < MAXP3; P++){
    SM[P] = 0.0;
    SIG[P] = 0.0;
}
P = 1;
for (PP = 0; PP < MAXP; PP++){
    Q[PP] = (2.0 * P) * M_PI / BOXL; /* Family [100] */
    P++;
}
P = 1;
for (PP = MAXP; PP < MAXP2; PP++){
    Q[PP] = (2.0 * P) * M_PI * M_SQRT2 / BOXL; /* Family [110] */
    P++;
}
P = 1;
for (PP = MAXP2; PP < MAXP3; PP++){
    Q[PP] = (2.0 * P) * M_PI * sqrt(3.0) / BOXL; /* Family [111] */
    P++;
}
/*****/

/*****/
/* Calculus of S(q) */

```

```

/*****/
IN = fopen(POSFILE, "r");
for (STEP = 1; STEP <= NSTEP; STEP++){
    fscanf(IN, "%d", &N);
    fscanf(IN, "%s", get);
    for (J = 0; J < N; J++)
        fscanf(IN, "%s %lf %lf %lf", get, &RX[J], &RY[J], &RZ[J]);
    SQ(RX, RY, RZ, BOXL, S, MD, N);
    for (P = 0; P < MAXP3; P++)
        SM[P] = SM[P] + S[P];
}
fclose(IN);
for (P = 0; P < MAXP3; P++)
    SM[P] = SM[P] / (1.0 * N * NSTEP);

IN = fopen(POSFILE, "r");
for (STEP = 1; STEP <= NSTEP; STEP++){
    fscanf(IN, "%d", &N);
    fscanf(IN, "%s", get);
    for (J = 0; J < N; J++)
        fscanf(IN, "%s %lf %lf %lf", get, &RX[J], &RY[J], &RZ[J]);
    GOFR(RX, RY, RZ, BOXL, DELR, HIST, N);
    SQ(RX, RY, RZ, BOXL, S, MD, N);
    for (P = 0; P < MAXP3; P++)
        SIG[P] = SIG[P] + (S[P] / (1.0 * N) - SM[P]) *
            (S[P] / (1.0 * N) - SM[P]);
}
fclose(IN);
for (P = 0; P < MAXP3; P++)
    SIG[P] = sqrt(SIG[P] / (1.0 * N * NSTEP));

ORDENA(Q, SM, SIG);

OUT = fopen(GRFILE, "w");
double CONST = 4.0 * M_PI * N / VOL / 3.0;
for (BIN = 1; BIN < MAXBIN; BIN++){
    RLOWER = 1.0 * (BIN - 1) * DELR;
    RUPPER = RLOWER + DELR;
    NIDEAL = CONST * (pow(RUPPER, 3.0) - pow(RLOWER, 3.0));
    GR[BIN] = 1.0 * HIST[BIN] / (1.0 * NSTEP) / (1.0 * N) / NIDEAL;
    fprintf(OUT, "%lf %lf\n", 1.0 * BIN * DELR, GR[BIN]);
}
fclose(OUT);

OUT = fopen(SQFILE, "w");
for (P = 0; P < MAXP3; P++)
    fprintf(OUT, "%lf %lf %lf %lf\n", 1.0 / TR, Q[P], SM[P], SIG[P]);
fclose(OUT);
/*****/

(void) time(&t2);
printf("\n\n Execution time = %ld seconds\n", (int) t2-t1);
/*****/

return 0;
}

```

```

void GOFr(double RX[MAX], double RY[MAX], double RZ[MAX], double BOXL,
          double DELR, int HIST[MAX], int N){

    int I, J, BIN;
    double RXI, RYI, RZI, RIJ, RXIJ, RYIJ, RZIJ, RIJSQ;

    for (I = 0; I < N - 1; I++){
        RXI = RX[I];
        RYI = RY[I];
        RZI = RZ[I];
        for (J = I + 1; J < N; J++){
            RXIJ = RXI - RX[J];
            RYIJ = RYI - RY[J];
            RZIJ = RZI - RZ[J];

            RXIJ = RXIJ - BOXL * (1.0 * round(RXIJ / BOXL));
            RYIJ = RYIJ - BOXL * (1.0 * round(RYIJ / BOXL));
            RZIJ = RZIJ - BOXL * (1.0 * round(RZIJ / BOXL));

            RIJSQ = RXIJ * RXIJ + RYIJ * RYIJ + RZIJ * RZIJ;

            RIJ = sqrt(RIJSQ);

            BIN = (int)(RIJ / DELR) + 1;

            if (BIN <= MAXBIN)
                HIST[BIN] = HIST[BIN] + 2;

        }
    }
}

void SQ(double RX[MAX], double RY[MAX], double RZ[MAX], double BOXL,
        double S[MAXP3], double MD[13][3], int N){

    int F, J, P, PP;
    double H, K, L, SCOSA, SSENA, SCOSB, SSENb, COSAJ, SENAJ;
    double COSBJ, SENBJ, COSBI, SENBI, RXJ, RYJ, RZJ;
    double COSA[MAX], SENA[MAX], COSB[MAX], SENB[MAX], SQF[13][MAX];

    for (F = 0; F < 13; F++){
        H = MD[F][0];
        K = MD[F][1];
        L = MD[F][2];
        SCOSA = 0.0;
        SSENA = 0.0;
        SCOSB = 0.0;
        SSENb = 0.0;
        for (J = 0; J < N; J++){
            RXJ = RX[J];
            RYJ = RY[J];
            RZJ = RZ[J];
            COSAJ = cos(2.0 * M_PI * (H * RXJ + K * RYJ + L * RZJ) / BOXL);
            SENAJ = sin(2.0 * M_PI * (H * RXJ + K * RYJ + L * RZJ) / BOXL);

```



```

        COSBJ   = 2.0 * COSAJ * COSAJ - 1.0;
        SENBJ   = 2.0 * SENAJ * COSAJ;
        SCOSA   = SCOSA + COSAJ;
        SSENA   = SSENA + SENAJ;
        SCOSB   = SCOSB + COSBJ;
        SSENB   = SSENB + SENBJ;
        COSA[J] = COSAJ;
        SENA[J] = SENAJ;
        COSB[J] = COSBJ;
        SENB[J] = SENBJ;
    }
    SQF[F][0] = SCOSA * SCOSA + SSENA * SSENA;
    SQF[F][1] = SCOSB * SCOSB + SSENB * SSENB;
    for (P = 2; P < MAXP; P++){
        SCOSB = 0.0;
        SSENB = 0.0;
        for (J = 0; J < N; J++){
            COSAJ   = COSA[J];
            SENAJ   = SENA[J];
            COSBJ   = COSB[J];
            SENBJ   = SENB[J];
            COSBI   = COSAJ * COSBJ - SENAJ * SENBJ;
            SENBI   = SENAJ * COSBJ + COSAJ * SENBJ;
            SCOSB   = SCOSB + COSBI;
            SSENB   = SSENB + SENBI;
            COSB[J] = COSBI;
            SENB[J] = SENBI;
        }
        SQF[F][P] = SCOSB * SCOSB + SSENB * SSENB;
    }
}

/* Average over family [100] */
for (P = 0; P < MAXP; P++){
    S[P] = 0.0;
    for (F = 0; F < 3; F++){
        S[P] = S[P] + SQF[F][P];
    }
    S[P] = S[P] / 3.0;
}

/* Average over family [110] */
PP = 0;
for (P = MAXP; P < MAXP2; P++){
    S[P] = 0.0;
    for (F = 3; F < 9; F++){
        S[P] = S[P] + SQF[F][PP];
    }
    S[P] = S[P] / 6.0;
    PP++;
}

/* Average over family [111] */
PP = 0;
for (P = MAXP2; P < MAXP3; P++){
    S[P] = 0.0;
    for (F = 9; F < 13; F++){
        S[P] = S[P] + SQF[F][PP];
    }
    S[P] = S[P] / 4.0;
    PP++;
}

```

```

}

void DIRECTIONS(double MD[13][3]){

    int I, J;
    FILE *IN;

    IN = fopen("dir.in", "r");
    for (I = 0; I < 13; I++)
        for (J = 0; J < 3; J++)
            fscanf(IN, "%lf", &MD[I][J]);
    fclose(IN);
}

void ORDENA(double Q[MAXP3], double SM[MAXP3], double SIG[MAXP3]){

    int P;
    double AUX, AUX2, AUX3;

    for (P = 1; P < MAXP3; P++){
        while ((P != 0) && (Q[P] < Q[P-1])){
            AUX      = Q[P];
            AUX2     = SM[P];
            AUX3     = SIG[P];
            Q[P]     = Q[P-1];
            SM[P]    = SM[P-1];
            SIG[P]   = SIG[P-1];
            Q[P-1]   = AUX;
            SM[P-1]  = AUX2;
            SIG[P-1] = AUX3;
            P--;
        }
    }
}

```

## C.4 Code to calculate local particle density along $z$ -direction

```

/*****
/* School of Engineering - University of São Paulo - Brazil */
/* Bioprocess Engineering Group - Department of Chemical Engineering */
/* Calculus of local particle density along z-direction */
/* Developed by: Luís Fernando Mercier Franco */
/* Advisor: Prof. Dr. Pedro de Alcântara Pessoa Filho */
/* Processos FAPESP: 2011/22070-5 and 2013/01857-2 */
/* September 1st, 2015 */
*****/

```

```

/*****
/* Supporting Information:
/*
/* 1) To compile, save this file as "LPD.c", then
/* in the terminal:
/* > gcc LPD.c -o LPD.exe -lm
/*
/* 2) To run the code:
/* > echo NUBER_OF_PARTICLES MASTER_FILE_NAME | ./LPD.exe
/*
/* 3) The authors do not accept any liability for the use of this
/* program.
*****/

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>

#define MAX 3000
#define MAXBIN 800

/*****
/* Declaration of subroutines
*****/

void LPD(double RZ[MAX], double LZ,
         double DELZ, int HIST[MAX], int N);

/*****
/* Main routine
*****/

main(){
    int STEP, NSTEPE, NSTEP, I, J, N, BIN;
    int HIST[MAX];
    double LX, LY, LZ, TR, RHO, VOL, DELZ;
    double RX[MAX], RY[MAX], RZ[MAX], GR[MAX];
    char get[100], LOGFILE[40], CONFFILE[40], SCIFILE[40];
    char MFILE[40], RATFILE[40], POSFILE[40], THERMOFILE[40], GRFILE[40];
    char SQFILE[40], SYS[100], LPD[40];
    time_t t1, t2;
    FILE *IN, *OUT;

    (void) time(&t1);

    /*****
    /* Reads input files
    *****/
    scanf("%d %s", &N, MFILE);
    IN = fopen(MFILE, "r");
    fscanf(IN, "%s", LOGFILE);
    fscanf(IN, "%s", CONFFILE);
    fscanf(IN, "%s", SCIFILE);
    fscanf(IN, "%s", RATFILE);
    fscanf(IN, "%s", POSFILE);

```

```

fscanf(IN, "%s", THERMOFILE);
fscanf(IN, "%s", GRFILE);
fscanf(IN, "%s", SQFILE);
fscanf(IN, "%s", LPDFILE);
fclose(IN);

IN = fopen(SCIFILE, "r");
fscanf(IN, "%s", get);
fscanf(IN, "%s", SYS);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &LX);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &LY);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &LZ);
fscanf(IN, "%s", get);
fscanf(IN, "%lf", &TR);
fscanf(IN, "%s", get);
fscanf(IN, "%d", &NSTEPE);
fscanf(IN, "%s", get);
fscanf(IN, "%d", &NSTEP);
fclose(IN);

/*****/

/*****/
/* Initialization of variables */
/*****/
VOL = LX * LY * LZ;
DELZ = LZ / (1.0 * MAXBIN);
for (BIN = 1; BIN < MAXBIN; BIN++)
    HIST[BIN] = 0;
/*****/

/*****/
/* Calculus of local particle density, d(z) */
/*****/
IN = fopen(POSFIL, "r");
for (STEP = 1; STEP <= NSTEP; STEP++){
    fscanf(IN, "%d", &N);
    fscanf(IN, "%s", get);
    for (J = 0; J < N; J++)
        fscanf(IN, "%s %lf %lf %lf", get, &RX[J], &RY[J], &RZ[J]);
    LPD(RZ, LZ, DELZ, HIST, N);
}
fclose(IN);

OUT = fopen(LPDFILE, "w");
for (BIN = 1; BIN < MAXBIN; BIN++){
    GR[BIN] = 1.0 * HIST[BIN] / (1.0 * NSTEP) / LX / LY / DELZ;
    fprintf(OUT, "%lf %lf\n", 1.0 * BIN * DELZ, GR[BIN]);
}
fclose(OUT);
/*****/

(void) time(&t2);

```

```

printf("\n\n Execution time = %ld seconds\n", (int) t2-t1);
/*****/

return 0;
}

void LPD(double RZ[MAX], double LZ,
         double DELZ, int HIST[MAX], int N){

    int J, BIN;
    double RZJ;

    for (J = 0; J < N; J++){
        RZJ = RZ[J];
        BIN = (int)((RZJ + LZ / 2.0) / DELZ) + 1;
        if (BIN <= MAXBIN)
            HIST[BIN] = HIST[BIN] + 1;
    }

}

```

# APPENDIX D – CODE FOR NELDER AND MEAD ALGORITHM

```

/*****
/* School of Engineering - University of São Paulo - Brazil */
/* Bioprocess Engineering Group - Department of Chemical Engineering */
/* Calculus of local particle density along z-direction */
/* Developed by: Luís Fernando Mercier Franco */
/* Advisor: Prof. Dr. Pedro de Alcântara Pessôa Filho */
/* Processos FAPESP: 2011/22070-5 and 2013/01857-2 */
/* February 12th, 2014 */
/* Reference: J.A. Nelder, R.A. Mead, Comp. J. 7 (1965) 308-313. */
*****/

/*****
/* Supporting Information: */
/* */
/* 1) An input file is needed following the same structure as this example:*/
/* */
/*      #Number_of_parameters: */
/*      2 */
/*      #Initial_estimative: */
/*      -1.2 */
/*      1.0 */
/*      #MIN_and_MAX_values: */
/*      -10.0      10.0 */
/*      -10.0      10.0 */
/* */
/* 2) To compile, save this file as "ADJ.c", then */
/* in the terminal: */
/*      > gcc ADJ.c -o ADJ.exe -lm */
/* */
/* 3) To run the code: */
/*      > echo INPUT_FILE_NAME | ./ADJ.exe */
/* */
/* 4) The authors do not accept any liability for the use of this */
/* program. */
/* */
/* Note: this code is supposed to be a optimization program. Here, the */
/* function to be minimized is the Rosenbrock's parabolic valley: */

```

```

/*      f(x) = 100(x2-x1^2)^2 + (1-x1)^2. Obviously, to      */
/*      use this code for other functions one ought to alter the code      */
/*      and write in proper C sintaxe the new function.      */
/*****/

#include <stdio.h>
#include <math.h>

#define NPM 100

double OF(double LAMBDA[NPM], int N);

main(){
    int J, N, LS;
    double X1[NPM], X2[NPM], X3[NPM], XK[NPM], XMIN[NPM], XMAX[NPM], EMAX[NPM];
    double EFAC, Y, Y1, YSTAR, YTEST, YBASE;
    char get[40], NFILE[40];
    FILE *INP, *OUT;

    /*****/
    /* Reading the file name and the initial estimative      */
    /*****/
    scanf("%s", NFILE);
    printf("\n Reading the input file...\n");
    INP = fopen(NFILE, "r");
    fscanf(INP, "%s", get);
    fscanf(INP, "%d", &N);
    fscanf(INP, "%s", get);
    for (J = 0; J < N; J++){
        fscanf(INP, "%lf", &X1[J]);
    }
    fscanf(INP, "%s", get);
    for (J = 0; J < N; J++){
        fscanf(INP, "%lf", &XMIN[J]);
        fscanf(INP, "%lf", &XMAX[J]);
    }
    fclose(INP);
    /*****/

    /*****/
    /* Minimization of the object function      */
    /*****/
    printf(" Minimizing the object function...\n");
    EFAC = pow(10.0, -3.0);
    Y = OF(X1, N);
    for (J = 0; J < N; J++){
        XK[J] = 1.0;
        EMAX[J] = 0.01 * (XMAX[J]-XMIN[J]);
        X2[J] = X1[J];
        X3[J] = X1[J];
    }
    YSTAR = Y;
    LS = 0;
    P100:
    YTEST = YSTAR;
    for (J = 0; J < N; J++){

```

```

    if (X1[J] >= XMAX[J] && XK[J] > 0.0) goto P1010;
    if (X1[J] <= XMIN[J] && XK[J] < 0.0) goto P1010;
    X2[J] = X1[J] + XK[J] * EMAX[J];
    if (X2[J] > XMAX[J]) X2[J] = XMAX[J];
    if (X2[J] < XMIN[J]) X2[J] = XMIN[J];
    Y1 = OF(X2, N);
    if (Y1 >= YTEST) goto P1010;
    XK[J] = 2.0 * XK[J];
    if (XK[J] > 1.0) XK[J] = 1.0;
    if (XK[J] < -1.0) XK[J] = -1.0;
    goto P103;
P1010:
    if (X1[J] <= XMIN[J] && XK[J] > 0.0) goto P1011;
    if (X1[J] >= XMAX[J] && XK[J] < 0.0) goto P1011;
    X2[J] = X1[J] - XK[J] * EMAX[J];
    if (X2[J] < XMIN[J]) X2[J] = XMIN[J];
    if (X2[J] > XMAX[J]) X2[J] = XMAX[J];
    Y1 = OF(X2, N);
    if (Y1 >= YTEST) goto P1011;
    XK[J] = -1.0 * XK[J];
    goto P103;
P1011:
    X2[J] = X1[J];
    XK[J] = 0.5 * XK[J];
    if (XK[J] < 0.0) goto P1015;
    if (XK[J] < 0.5 * EFAC) XK[J] = 0.5 * EFAC;
    goto P1018;
P1015:
    if (fabs(XK[J]) < 0.5 * EFAC) XK[J] = -0.5 * EFAC;
P1018:
    Y1 = YTEST;
P103:
    YTEST = Y1;
}
if (LS == 1) goto P350;
YBASE = YTEST;
if (YBASE < YSTAR) goto P300;
for (J = 0; J < N; J++)
    if (fabs(XK[J]) > 0.5 * EFAC) goto P100;
goto P600;
P300:
    for (J = 0; J < N; J++){
        X1[J] = X3[J] + 2.0 * (X2[J] - X3[J]);
        if (X1[J] < XMIN[J]) X1[J] = XMIN[J];
        if (X1[J] > XMAX[J]) X1[J] = XMAX[J];
        X3[J] = X2[J];
        X2[J] = X1[J];
    }
    YSTAR = OF(X1, N);
    LS = 1;
    goto P100;
P350:
    LS = 0;
    if (YTEST >= YBASE) goto P500;
    YBASE = YTEST;
    goto P300;

```



```

P500:
    for (J = 0; J < N; J++){
        X1[J] = X3[J];
        X2[J] = X1[J];
    }
    YSTAR = YBASE;
    goto P100;
P600:
    Y = OF(X1, N);

printf(" The solution is:\n\n");
for (J = 0; J < N; J++)
    printf(" X[%d] = %e\n", J + 1, X1[J]);
printf(" The optimum value is: Y = %e\n\n", Y);
printf(" The calculation is ended...\n\n");

return 0;
}

double OF(double PAR[NPM], int N){
    double F;

    double X1 = PAR[0];
    double X2 = PAR[1];

    F = 100.0 * (X2 - X1 * X1) * (X2 - X1 * X1) + (1.0 - X1) * (1.0 - X1);

    return F;
}

```

## APPENDIX E – CODE FOR PHASE ENVELOPE CALCULATION

The following code is written in FORTRAN 77. It has been successfully compiled with gfortran and it has been successfully executed in Linux.

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C School of Engineering - University of São Paulo - Brazil          C
C Bioprocess Engineering Group - Department of Chemical Engineering C
C Phase envelope calculation considering the osmotic pressure        C
C derived by Franco et al. (2015)                                   C
C Developed by: Luís Fernando Mercier Franco                       C
C Advisor: Prof. Dr. Pedro de Alcântara Pessoa Filho              C
C Processos FAPESP: 2011/22070-5 and 2013/01857-2                  C
C July 22nd, 2015                                                  C
C Reference: Franco, LFM; Oliveira, CLP; Pessoa Filho, PA.         C
C           Thermodynamics of protein solutions: from the structure C
C           factor to the osmotic pressure, AIChE J, 2015 (in press)C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C Supporting Information:                                          C
C                                                                    C
C   1) To use the program an input file is required. The following C
C   is an example:                                                C
C                                                                    C
C   TREF = 23.0                                                    C
C   TMIN = -149.0                                                  C
C   TMAX = -126.0                                                  C
C   LAMDA = 0.1                                                    C
C   EPS0 = 2.32                                                    C
C   EPS1 = 11.10                                                  C
C                                                                    C
C   TREF is the reference temperature at which the potential      C
C   parameter was fitted. TMIN and TMAX are the minimum and       C
C   the maximum temperature for the calculation of the coexistence C
C   curve. Attention must be paid lest the maximum temperature   C

```

```

C exceeds the critical temperature. LAMDA is the lambda value C
C used in the loop of equilibrium pressure to guarantee C
C the convergence. EPS0 and EPS1 are the potential parameters. C
C
C 2) There are three output files: "dens.dat", pressure.dat", C
C "volume.dat" and "crit.dat". C
C In the very first one, the phase envelope is printed: C
C the temperature in Kelvin on the first column and the protein C
C concentration in mg/mL of equilibrium on the second column. On C
C the third column is plotted the protein concentration in mg/mL C
C of the spinodal curve. C
C In the second file ("pressure.dat"), the temperature in Kelvin, C
C the pressure in Pascal and also the values of the difference C
C between the chemical potential of the two phases and the value C
C of PSI used in Maxwell area rule. C
C In the third file ("volume.dat"), the temperature in Kelvin, C
C the volumes of the light and heavy phases in m3/mol as well as C
C the difference of chemical potential between the two phases C
C are stored. C
C The last file ("crit.dat") shows the critical point data. C
C
C 3) The authors do not accept any liability for the use of this C
C program. C
C
C Note: this code is written specifically to Bovine Serum Albumin C
C and because of it the molar mass is fixed to be 66.5 kDa. C
C If one wants to run to other proteins, one must change C
C the MM value (always in kDa). C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

# PROGRAM MAIN

```

INTEGER N, NMAX, J, K
REAL TOL, TREF, T, EF1, EF2, ZETA, TMAX, ATTO, ATT1, RF
REAL ALPHA, EPS0, EPS1, PIN, EMAX, EMIN, PSI, LAMBDA, VL, VH
REAL MM, NA, KB, SIG, PI, PRES, DMUR, TMIN, PC, VC, CC, FAC
REAL TEMP(10000), HEAV(10000), LIGH(10000)
REAL SPIH(10000), SPIL(10000)
CHARACTER DUM*8, FNAME*15
COMMON EPS0, EPS1, ZETA, T

```

```

TOL = 10.0**(-5.0)
READ(*,300) FNAME
OPEN(1,FILE=FNAME)
READ(1,200) DUM, TREF
READ(1,200) DUM, T
READ(1,200) DUM, TMAX
READ(1,200) DUM, LAMBDA
READ(1,200) DUM, ATTO
READ(1,200) DUM, ATT1
CLOSE(1)
MM = 66.5
NA = 6.0221314
KB = 1.3806488
SIG = 63.3967351934
PI = 3.14159265359
FAC = 6.0*MM/PI/SIG**3.0/NA*10.0**7.0

```



```

      IF (ZMIN .LT. 0.0) THEN
        ZETA = ZMAX / 2.0
      ELSE
        ZETA = (ZETA + ZMAX) / 2.0
      END IF
    END IF
    EF1 = 0.01 * EMAX
    DO WHILE (ABS(FUN(EF1)) .GT. TOL)
      EF1 = EF1 - FUN(EF1) / FDER(EF1)
    END DO
    EF2 = 1.1 * EMIN
    DO WHILE (ABS(FUN(EF2)) .GT. TOL)
      EF2 = EF2 - FUN(EF2) / FDER(EF2)
    END DO
    PSI = ZETA*(EF2-EF1)-PHI(EF2)+PHI(EF1)
  END DO
  TEMP(K) = T + 273.15
  LIGH(K) = EF1*FAC
  HEAV(K) = EF2*FAC
  SPIL(K) = EMAX*FAC
  SPIH(K) = EMIN*FAC
  VL      = MM/LIGH(K)
  VH      = MM/HEAV(K)
  PRES    = 6.0*ZETA*(T+273.15)*KB/SIG**3.0/PI*10.0**7.0
  DMUR    = (LOG(EF1/EF2) + CSI(EF1) - CSI(EF2))*NA*KB*(T+273.15)
  WRITE(2,100), T+273.15, PRES, DMUR, PSI
  WRITE(8,100), T+273.15, VL, VH, DMUR
  T = T + 0.1
  K = K + 1
END DO
CLOSE(2)
CLOSE(8)

```

[illegible]



```
ALPH2 = (4.0*ETA-3.0*ETA**2.0)/(ETA-1.0)**2  
CSI    = ALPH1+ALPH2-8.0*EPS0*ETA+4.0*EPS1*ETA**2  
  
END
```



## APPENDIX F – CODE FOR VISCOSITY VIA GREEN-KUBO

This code is written in C. It has been successfully compiled with gcc and has been successfully executed in Linux.

```

/*****
/* School of Engineering - University of São Paulo - Brazil */
/* Bioprocess Engineering Group - Department of Chemical Engineering */
/* Calculus of local particle density along z-direction */
/* Developed by: Luís Fernando Mercier Franco */
/* Advisor: Prof. Dr. Pedro de Alcântara Pessoa Filho */
/* Processos FAPESP: 2011/22070-5 and 2013/01857-2 */
/* November 11th, 2013 */
/* Reference: J.A. Nelder, R.A. Mead, Comp. J. 7 (1965) 308-313. */
*****/

/*****
/* Supporting Information: */
/*
/* 1) An input file is needed. In this input file the first line is */
/* considered to be a comment. The second line must contain the following */
/* information: the box length in nm, the temperature in Kelvin, */
/* the maximum time for calculation in ps, the TAU value for correction */
/* of dynamics for coarse-grained (if one does not want to consider */
/* such correction, set TAU = 1), the file name with time in ps and */
/* the pressure tensors (PXX, PXY, PXZ, PYY, PYZ, PZZ), the file name */
/* of the file in which the stress correlation function must be stored */
/* and the file name of the file in which the viscosity must be stored. */
/*
/* 2) To compile, save this file as "VISC.c", then */
/* in the terminal: */
/* > gcc VISC.c -o VISC.exe -lm */
/*
/* 3) To run the code: */
/* > echo INPUT_FILE_NAME | ./VISC.exe */
/*
/* 4) The authors do not accept any liability for the use of this */
/* program.

```

```

/*****/

#include <stdio.h>
#include <math.h>

#define TRUE 1
#define MAX 200010

main(){
    int J, K, NDATA, L, NMAX, P;
    double KB, NA, M, D, BOXL, VOL, T, CONST, TMAX, TAU;
    double SKY, SKZ, SYZ, SX1, SY1, SZ1, VIS, PRESXX, PRESYY, PRESZZ;
    double TIME[MAX], PRESXY[MAX], PRESXZ[MAX], PRESYZ[MAX], PRESX1[MAX];
    double PRESY1[MAX], PRESZ1[MAX], SAVG[MAX], AVG[6];
    char NFILE[20], PFILE[20], OFILE[40], QFILE[40], COMMAND[500];
    FILE *INP, *OUT;

    /*****/
    /* Reading input files */
    /*****/
    scanf("%s", NFILE);
    printf("\n Reading input file...\n");
    INP = fopen(NFILE, "r");
    fgets(COMMAND, 500, INP);
    fscanf(INP, "%lf %lf %lf %lf %s %s %s", &BOXL, &T, &TMAX,
        &TAU, PFILE, OFILE, QFILE);
    fclose(INP);

    KB    = 1.3806488 * pow(10.0, -23.0);
    VOL    = BOXL * BOXL * BOXL * pow(10.0, -27.0);
    CONST = 10.0 * VOL / KB / T;

    INP    = fopen(PFILE, "r");
    NDATA = 0;
    while (TRUE){
        K = fscanf(INP, "%lf", &TIME[NDATA]);
        if (K != 1) break;
        fscanf(INP, "%lf", &PRESXX);
        fscanf(INP, "%lf", &PRESXY[NDATA]);
        fscanf(INP, "%lf", &PRESXZ[NDATA]);
        fscanf(INP, "%lf", &PRESYY);
        fscanf(INP, "%lf", &PRESYZ[NDATA]);
        fscanf(INP, "%lf", &PRESZZ);
        PRESX1[NDATA] = 0.5 * (PRESXX - PRESYY);
        PRESY1[NDATA] = 0.5 * (PRESYY - PRESZZ);
        PRESZ1[NDATA] = 0.5 * (PRESXX - PRESZZ);
        if (TIME[NDATA] == TMAX) NMAX = NDATA;
        TIME[NDATA] = TIME[NDATA] * TAU;
        NDATA++;
    }
    fclose(INP);
    /*****/

    for (P = 0; P < 6; P++){
        AVG[P] = 0.0;
        for (K = 0; K < NDATA; K++){

```

```

        AVG[0] = AVG[0] + PRESX1[K];
        AVG[1] = AVG[1] + PRESXY[K];
        AVG[2] = AVG[2] + PRESXZ[K];
        AVG[3] = AVG[3] + PRESY1[K];
        AVG[4] = AVG[4] + PRESYZ[K];
        AVG[5] = AVG[5] + PRESZ1[K];
    }
    for (P = 0; P < 6; P++){
        AVG[P] = AVG[P] / (1.0 * (NDATA - 1));
    }
    for (K = 0; K < NDATA; K++){
        PRESX1[K] = PRESX1[K] - AVG[0];
        PRESXY[K] = PRESXY[K] - AVG[1];
        PRESXZ[K] = PRESXZ[K] - AVG[2];
        PRESY1[K] = PRESY1[K] - AVG[3];
        PRESYZ[K] = PRESYZ[K] - AVG[4];
        PRESZ1[K] = PRESZ1[K] - AVG[5];
    }

    /*****
    /* Stress Auto-Correlation Function calculation */
    /* <PAB(0)*PAB(t)>=1/N * sum_(L=0)^(N) (PAB(L)*PAB(L+t)) */
    /* Averages over PXY,PXZ,PYZ,0.5*(PXX-PYY),0.5*(PYY-PZZ),0.5*(PXX-PZZ) */
    /* Results saved in OFILE */
    /* Reference: */
    /* Rey-Castro, C.; Vega, L. F., J. Phys. Chem. B, 110, 14426-14435, 2006*/
    /*****/
    printf(" Calculating Stress ACF...\n");
    OUT = fopen(OFILE, "w");
    for (K = 0; K < NMAX; K++){
        SXY = 0.0;
        SXZ = 0.0;
        SYZ = 0.0;
        SX1 = 0.0;
        SY1 = 0.0;
        SZ1 = 0.0;
        for (L = K; L < K + (NDATA / 2) ; L++){
            SXY = SXY + PRESXY[L] * PRESXY[L + K];
            SXZ = SXZ + PRESXZ[L] * PRESXZ[L + K];
            SYZ = SYZ + PRESYZ[L] * PRESYZ[L + K];
            SX1 = SX1 + PRESX1[L] * PRESX1[L + K];
            SY1 = SY1 + PRESY1[L] * PRESY1[L + K];
            SZ1 = SZ1 + PRESZ1[L] * PRESZ1[L + K];
        }
        SXY = SXY / (1.0 * (NDATA / 2));
        SXZ = SXZ / (1.0 * (NDATA / 2));
        SYZ = SYZ / (1.0 * (NDATA / 2));
        SX1 = SX1 / (1.0 * (NDATA / 2));
        SY1 = SY1 / (1.0 * (NDATA / 2));
        SZ1 = SZ1 / (1.0 * (NDATA / 2));
        SAVG[K] = 1.0 / 6.0 * (SXY + SXZ + SYZ + SX1 + SY1 + SZ1);
        fprintf(OUT, "%1f %1f\n", TIME[K], SAVG[K]);
    }
    fclose(OUT);
    /*****/

```

```

/*****
/* Integration of Stress Auto-Correlation Function */
/* Green-Kubo equation for viscosity */
/* Reference: */
/* Allen, M. P.; Tildesley, D. J., Computer Simulation of Liquids, */
/* Oxford University Press, New York, 1987. */
/*****
printf(" Integrating Stress ACF to calculate viscosity...\n");
OUT = fopen(QFILE, "w");
for (K = 0; K < NMAX; K++){
    VIS = SAVG[0];
    for (L = 1; L < K - 1; L++){
        VIS = VIS + 2.0 * SAVG[L];
        VIS = CONST * (TIME[L] - TIME[0]) / (2.0 * (K - 1)) *
            (VIS + SAVG[L]);
        if (L > 1)
            fprintf(OUT, "%lf %lf\n", TIME[L], VIS);
    }
fclose(OUT);
printf(" The calculation is ended...\n\n");
/*****

return 0;

}

```

## APPENDIX G – SAXS INTENSITY DATA

Table 10: Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector,  $q$ , for BSA in 1.0 mol·L<sup>-1</sup> NaNO<sub>3</sub> aqueous solutions at pH = 4.9, at several protein concentrations,  $c_p$ , (mg·mL<sup>-1</sup>).

$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 81.79$	$c_p = 76.85$	$c_p = 72.81$	$c_p = 64.20$	$c_p = 56.57$	$c_p = 51.98$	$c_p = 48.24$	$c_p = 44.53$	$c_p = 39.67$
0.0142	1.05 ± 0.03	1.49 ± 0.03	1.62 ± 0.06	1.61 ± 0.06	1.68 ± 0.07	1.78 ± 0.05	1.90 ± 0.08	1.95 ± 0.08	2.02 ± 0.09	2.06 ± 0.09
0.0156	1.12 ± 0.02	1.53 ± 0.02	1.59 ± 0.03	1.73 ± 0.03	1.79 ± 0.04	1.77 ± 0.03	1.95 ± 0.05	2.01 ± 0.05	2.10 ± 0.05	2.08 ± 0.05
0.0171	1.11 ± 0.01	1.56 ± 0.01	1.56 ± 0.03	1.67 ± 0.03	1.72 ± 0.03	1.81 ± 0.02	1.90 ± 0.04	1.96 ± 0.04	2.06 ± 0.05	2.12 ± 0.04
0.0185	1.12 ± 0.01	1.53 ± 0.01	1.55 ± 0.03	1.64 ± 0.03	1.70 ± 0.03	1.82 ± 0.02	1.87 ± 0.04	1.90 ± 0.04	1.99 ± 0.04	2.08 ± 0.04
0.0199	1.13 ± 0.01	1.49 ± 0.01	1.52 ± 0.03	1.59 ± 0.03	1.64 ± 0.03	1.79 ± 0.02	1.79 ± 0.03	1.87 ± 0.04	1.99 ± 0.04	2.01 ± 0.04
0.0213	1.08 ± 0.01	1.43 ± 0.01	1.48 ± 0.02	1.53 ± 0.02	1.63 ± 0.03	1.65 ± 0.02	1.70 ± 0.03	1.81 ± 0.03	1.78 ± 0.04	1.92 ± 0.03
0.0228	1.12 ± 0.01	1.45 ± 0.01	1.55 ± 0.02	1.54 ± 0.02	1.67 ± 0.03	1.74 ± 0.02	1.74 ± 0.03	1.83 ± 0.03	1.92 ± 0.03	1.91 ± 0.03
0.0242	1.10 ± 0.01	1.42 ± 0.01	1.47 ± 0.02	1.47 ± 0.02	1.60 ± 0.02	1.68 ± 0.02	1.76 ± 0.03	1.78 ± 0.03	1.82 ± 0.03	1.87 ± 0.03
0.0256	1.08 ± 0.01	1.39 ± 0.01	1.44 ± 0.02	1.49 ± 0.02	1.56 ± 0.02	1.65 ± 0.02	1.69 ± 0.03	1.70 ± 0.03	1.74 ± 0.03	1.82 ± 0.03
0.0270	1.07 ± 0.01	1.41 ± 0.01	1.44 ± 0.02	1.46 ± 0.02	1.59 ± 0.02	1.61 ± 0.01	1.63 ± 0.03	1.72 ± 0.03	1.67 ± 0.03	1.78 ± 0.02
0.0284	1.14 ± 0.01	1.45 ± 0.01	1.50 ± 0.02	1.52 ± 0.02	1.62 ± 0.02	1.66 ± 0.01	1.64 ± 0.02	1.73 ± 0.03	1.80 ± 0.03	1.82 ± 0.02
0.0299	1.09 ± 0.01	1.36 ± 0.01	1.38 ± 0.02	1.43 ± 0.02	1.51 ± 0.02	1.58 ± 0.01	1.56 ± 0.02	1.65 ± 0.02	1.66 ± 0.03	1.69 ± 0.02
0.0320	1.07 ± 0.01	1.37 ± 0.01	1.39 ± 0.01	1.43 ± 0.01	1.49 ± 0.01	1.56 ± 0.01	1.56 ± 0.02	1.62 ± 0.02	1.64 ± 0.02	1.68 ± 0.02
0.0349	1.03 ± 0.01	1.27 ± 0.01	1.30 ± 0.01	1.33 ± 0.01	1.34 ± 0.01	1.43 ± 0.01	1.44 ± 0.01	1.47 ± 0.02	1.48 ± 0.02	1.52 ± 0.01
0.0377	0.983 ± 0.006	1.18 ± 0.00	1.22 ± 0.01	1.23 ± 0.01	1.28 ± 0.01	1.34 ± 0.01	1.35 ± 0.01	1.37 ± 0.01	1.39 ± 0.01	1.41 ± 0.01
0.0405	0.995 ± 0.005	1.17 ± 0.00	1.19 ± 0.01	1.21 ± 0.01	1.27 ± 0.01	1.30 ± 0.01	1.32 ± 0.01	1.35 ± 0.01	1.34 ± 0.01	1.37 ± 0.01
0.0434	0.930 ± 0.005	1.10 ± 0.00	1.13 ± 0.01	1.14 ± 0.01	1.18 ± 0.01	1.20 ± 0.01	1.22 ± 0.01	1.25 ± 0.01	1.23 ± 0.01	1.28 ± 0.01
0.0462	0.932 ± 0.005	1.07 ± 0.00	1.09 ± 0.01	1.11 ± 0.01	1.12 ± 0.01	1.15 ± 0.01	1.17 ± 0.01	1.18 ± 0.01	1.17 ± 0.01	1.22 ± 0.01
0.0491	0.867 ± 0.005	0.979 ± 0.004	0.987 ± 0.008	1.03 ± 0.01	1.04 ± 0.01	1.05 ± 0.01	1.03 ± 0.01	1.08 ± 0.01	1.07 ± 0.01	1.08 ± 0.01
0.0519	0.839 ± 0.004	0.941 ± 0.004	0.947 ± 0.008	0.957 ± 0.008	0.989 ± 0.009	0.992 ± 0.006	0.992 ± 0.010	1.01 ± 0.01	1.01 ± 0.01	1.03 ± 0.01
0.0548	0.799 ± 0.004	0.872 ± 0.004	0.877 ± 0.007	0.894 ± 0.007	0.914 ± 0.008	0.922 ± 0.005	0.917 ± 0.009	0.943 ± 0.010	0.924 ± 0.010	0.945 ± 0.009
0.0576	0.751 ± 0.004	0.820 ± 0.003	0.831 ± 0.007	0.836 ± 0.007	0.847 ± 0.008	0.850 ± 0.005	0.865 ± 0.009	0.855 ± 0.009	0.859 ± 0.009	0.860 ± 0.008
0.0605	0.719 ± 0.004	0.756 ± 0.003	0.763 ± 0.006	0.770 ± 0.006	0.778 ± 0.007	0.788 ± 0.005	0.785 ± 0.008	0.787 ± 0.009	0.776 ± 0.009	0.790 ± 0.008
0.0647	0.628 ± 0.002	0.658 ± 0.002	0.661 ± 0.004	0.669 ± 0.004	0.664 ± 0.005	0.671 ± 0.003	0.667 ± 0.005	0.678 ± 0.005	0.674 ± 0.006	0.692 ± 0.005
0.0704	0.545 ± 0.002	0.556 ± 0.002	0.554 ± 0.004	0.559 ± 0.004	0.555 ± 0.004	0.566 ± 0.003	0.562 ± 0.005	0.560 ± 0.005	0.557 ± 0.005	0.574 ± 0.004

$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 81.79$	$c_p = 76.85$	$c_p = 72.81$	$c_p = 64.20$	$c_p = 56.57$	$c_p = 51.98$	$c_p = 48.24$	$c_p = 44.53$	$c_p = 39.67$
0.0761	$0.448 \pm 0.002$	$0.453 \pm 0.002$	$0.458 \pm 0.003$	$0.448 \pm 0.003$	$0.455 \pm 0.003$	$0.452 \pm 0.002$	$0.453 \pm 0.004$	$0.451 \pm 0.004$	$0.455 \pm 0.004$	$0.456 \pm 0.004$
0.0818	$0.361 \pm 0.002$	$0.362 \pm 0.001$	$0.353 \pm 0.003$	$0.359 \pm 0.003$	$0.366 \pm 0.003$	$0.359 \pm 0.002$	$0.362 \pm 0.003$	$0.367 \pm 0.004$	$0.356 \pm 0.004$	$0.361 \pm 0.003$
0.0875	$0.289 \pm 0.001$	$0.288 \pm 0.001$	$0.287 \pm 0.002$	$0.287 \pm 0.002$	$0.288 \pm 0.003$	$0.286 \pm 0.002$	$0.284 \pm 0.003$	$0.286 \pm 0.003$	$0.285 \pm 0.003$	$0.282 \pm 0.003$
0.0932	$0.230 \pm 0.001$	$0.228 \pm 0.001$	$0.224 \pm 0.002$	$0.225 \pm 0.002$	$0.224 \pm 0.002$	$0.228 \pm 0.002$	$0.223 \pm 0.003$	$0.226 \pm 0.003$	$0.226 \pm 0.003$	$0.227 \pm 0.003$
0.0989	$0.180 \pm 0.001$	$0.177 \pm 0.001$	$0.180 \pm 0.002$	$0.179 \pm 0.002$	$0.174 \pm 0.002$	$0.179 \pm 0.001$	$0.184 \pm 0.002$	$0.180 \pm 0.003$	$0.176 \pm 0.003$	$0.176 \pm 0.002$
0.1074	$0.121 \pm 0.001$	$0.121 \pm 0.001$	$0.121 \pm 0.001$	$0.122 \pm 0.001$	$0.122 \pm 0.001$	$0.122 \pm 0.001$	$0.121 \pm 0.001$	$0.119 \pm 0.002$	$0.120 \pm 0.002$	$0.121 \pm 0.001$
0.1188	$0.0808 \pm 0.0005$	$0.0799 \pm 0.0004$	$0.0807 \pm 0.0009$	$0.0805 \pm 0.0009$	$0.0791 \pm 0.0010$	$0.0796 \pm 0.0007$	$0.0787 \pm 0.0012$	$0.0779 \pm 0.0013$	$0.0814 \pm 0.0013$	$0.0810 \pm 0.0012$
0.1302	$0.0576 \pm 0.0004$	$0.0583 \pm 0.0004$	$0.0582 \pm 0.0008$	$0.0580 \pm 0.0008$	$0.0570 \pm 0.0009$	$0.0596 \pm 0.0006$	$0.0590 \pm 0.0010$	$0.0592 \pm 0.0011$	$0.0577 \pm 0.0012$	$0.0580 \pm 0.0011$
0.1415	$0.0461 \pm 0.0004$	$0.0472 \pm 0.0003$	$0.0480 \pm 0.0007$	$0.0482 \pm 0.0007$	$0.0470 \pm 0.0008$	$0.0470 \pm 0.0006$	$0.0467 \pm 0.0010$	$0.0476 \pm 0.0010$	$0.0485 \pm 0.0011$	$0.0473 \pm 0.0010$
0.1529	$0.0400 \pm 0.0004$	$0.0401 \pm 0.0003$	$0.0406 \pm 0.0007$	$0.0399 \pm 0.0007$	$0.0392 \pm 0.0007$	$0.0396 \pm 0.0005$	$0.0404 \pm 0.0009$	$0.0396 \pm 0.0010$	$0.0409 \pm 0.0010$	$0.0412 \pm 0.0009$
0.1643	$0.0341 \pm 0.0003$	$0.0340 \pm 0.0003$	$0.0341 \pm 0.0006$	$0.0333 \pm 0.0006$	$0.0349 \pm 0.0007$	$0.0341 \pm 0.0005$	$0.0342 \pm 0.0008$	$0.0343 \pm 0.0009$	$0.0334 \pm 0.0010$	$0.0341 \pm 0.0009$
0.1757	$0.0288 \pm 0.0003$	$0.0284 \pm 0.0003$	$0.0283 \pm 0.0006$	$0.0294 \pm 0.0006$	$0.0290 \pm 0.0007$	$0.0294 \pm 0.0005$	$0.0284 \pm 0.0008$	$0.0295 \pm 0.0009$	$0.0279 \pm 0.0009$	$0.0281 \pm 0.0008$
0.1870	$0.0247 \pm 0.0003$	$0.0245 \pm 0.0003$	$0.0245 \pm 0.0005$	$0.0255 \pm 0.0005$	$0.0242 \pm 0.0006$	$0.0248 \pm 0.0004$	$0.0244 \pm 0.0007$	$0.0243 \pm 0.0008$	$0.0253 \pm 0.0009$	$0.0245 \pm 0.0008$
0.1984	$0.0205 \pm 0.0003$	$0.0208 \pm 0.0002$	$0.0211 \pm 0.0005$	$0.0214 \pm 0.0005$	$0.0214 \pm 0.0006$	$0.0203 \pm 0.0004$	$0.0205 \pm 0.0007$	$0.0193 \pm 0.0008$	$0.0224 \pm 0.0008$	$0.0210 \pm 0.0008$
0.2098	$0.0178 \pm 0.0003$	$0.0174 \pm 0.0002$	$0.0178 \pm 0.0005$	$0.0177 \pm 0.0005$	$0.0174 \pm 0.0006$	$0.0169 \pm 0.0004$	$0.0181 \pm 0.0007$	$0.0160 \pm 0.0007$	$0.0175 \pm 0.0008$	$0.0173 \pm 0.0007$
0.2212	$0.0148 \pm 0.0002$	$0.0141 \pm 0.0002$	$0.0144 \pm 0.0005$	$0.0141 \pm 0.0005$	$0.0149 \pm 0.0005$	$0.0152 \pm 0.0004$	$0.0147 \pm 0.0007$	$0.0146 \pm 0.0007$	$0.0131 \pm 0.0007$	$0.0147 \pm 0.0007$
0.2325	$0.0130 \pm 0.0002$	$0.0132 \pm 0.0002$	$0.0139 \pm 0.0004$	$0.0131 \pm 0.0005$	$0.0134 \pm 0.0005$	$0.0125 \pm 0.0004$	$0.0126 \pm 0.0006$	$0.0141 \pm 0.0007$	$0.0136 \pm 0.0007$	$0.0130 \pm 0.0007$
0.2439	$0.0110 \pm 0.0002$	$0.0106 \pm 0.0002$	$0.0107 \pm 0.0004$	$0.00999 \pm 0.00043$	$0.0100 \pm 0.0005$	$0.0111 \pm 0.0004$	$0.0117 \pm 0.0006$	$0.0126 \pm 0.0007$	$0.0105 \pm 0.0007$	$0.0110 \pm 0.0007$
0.2553	$0.00969 \pm 0.00021$	$0.00954 \pm 0.00019$	$0.00972 \pm 0.00041$	$0.00931 \pm 0.00042$	$0.00934 \pm 0.00048$	$0.00935 \pm 0.00034$	$0.00988 \pm 0.00060$	$0.00889 \pm 0.00064$	$0.00959 \pm 0.00069$	$0.00959 \pm 0.00066$
0.2667	$0.00924 \pm 0.00020$	$0.00890 \pm 0.00019$	$0.00866 \pm 0.00040$	$0.00960 \pm 0.00041$	$0.00925 \pm 0.00048$	$0.00895 \pm 0.00034$	$0.00854 \pm 0.00058$	$0.00844 \pm 0.00063$	$0.00795 \pm 0.00066$	$0.00873 \pm 0.00064$
0.2780	$0.00903 \pm 0.00020$	$0.00992 \pm 0.00019$	$0.0103 \pm 0.0004$	$0.0104 \pm 0.0004$	$0.0102 \pm 0.0005$	$0.00993 \pm 0.00033$	$0.00950 \pm 0.00056$	$0.00935 \pm 0.00061$	$0.0102 \pm 0.0007$	$0.0106 \pm 0.0006$
0.2894	$0.00927 \pm 0.00020$	$0.00946 \pm 0.00018$	$0.00886 \pm 0.00038$	$0.0102 \pm 0.0004$	$0.0103 \pm 0.0005$	$0.00952 \pm 0.00032$	$0.00956 \pm 0.00056$	$0.0103 \pm 0.0006$	$0.0106 \pm 0.0007$	$0.0101 \pm 0.0006$
0.3008	$0.00960 \pm 0.00019$	$0.00958 \pm 0.00018$	$0.00978 \pm 0.00038$	$0.00928 \pm 0.00039$	$0.00978 \pm 0.00045$	$0.00900 \pm 0.00032$	$0.0103 \pm 0.0005$	$0.00965 \pm 0.00060$	$0.00957 \pm 0.00063$	$0.00912 \pm 0.00060$
0.3122	$0.00990 \pm 0.00019$	$0.00952 \pm 0.00018$	$0.00896 \pm 0.00038$	$0.00880 \pm 0.00039$	$0.00884 \pm 0.00044$	$0.00906 \pm 0.00031$	$0.00891 \pm 0.00054$	$0.00992 \pm 0.00059$	$0.00898 \pm 0.00063$	$0.00911 \pm 0.00060$
0.3235	$0.00988 \pm 0.00019$	$0.00973 \pm 0.00018$	$0.00960 \pm 0.00037$	$0.00889 \pm 0.00038$	$0.00937 \pm 0.00044$	$0.0103 \pm 0.0003$	$0.00974 \pm 0.00054$	$0.00954 \pm 0.00059$	$0.00984 \pm 0.00063$	$0.00881 \pm 0.00060$
0.3349	$0.00963 \pm 0.00019$	$0.00984 \pm 0.00017$	$0.00927 \pm 0.00036$	$0.00974 \pm 0.00038$	$0.00977 \pm 0.00043$	$0.00976 \pm 0.00031$	$0.00927 \pm 0.00053$	$0.00955 \pm 0.00058$	$0.00957 \pm 0.00062$	$0.00947 \pm 0.00059$
0.3448	$0.00880 \pm 0.00021$	$0.00942 \pm 0.00020$	$0.00920 \pm 0.00041$	$0.00905 \pm 0.00042$	$0.00881 \pm 0.00049$	$0.00935 \pm 0.00035$	$0.00898 \pm 0.00060$	$0.00868 \pm 0.00065$	$0.00860 \pm 0.00070$	$0.00970 \pm 0.00067$

Table 11: Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector,  $q$ , for BSA in 1.0 mol·L<sup>-1</sup> NaCl aqueous solutions at pH = 4.9, at several protein concentrations,  $c_p$ , (mg·mL<sup>-1</sup>).

$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 67.90$	$c_p = 65.51$	$c_p = 59.30$	$c_p = 54.92$	$c_p = 49.95$	$c_p = 45.85$	$c_p = 40.86$	$c_p = 36.68$	$c_p = 31.10$
0.0142	1.17 ± 0.04	1.95 ± 0.06	2.05 ± 0.07	2.10 ± 0.07	2.11 ± 0.07	2.38 ± 0.08	2.47 ± 0.08	2.50 ± 0.09	2.82 ± 0.10	2.70 ± 0.11
0.0156	1.14 ± 0.03	1.95 ± 0.04	1.96 ± 0.04	2.11 ± 0.04	2.16 ± 0.04	2.28 ± 0.05	2.45 ± 0.05	2.48 ± 0.05	2.51 ± 0.06	2.80 ± 0.07
0.0171	1.17 ± 0.02	1.95 ± 0.03	1.97 ± 0.03	2.15 ± 0.04	2.22 ± 0.04	2.37 ± 0.04	2.44 ± 0.05	2.59 ± 0.05	2.74 ± 0.05	2.81 ± 0.06
0.0185	1.16 ± 0.02	1.90 ± 0.03	1.96 ± 0.03	2.04 ± 0.03	2.16 ± 0.04	2.29 ± 0.04	2.33 ± 0.04	2.45 ± 0.05	2.56 ± 0.05	2.73 ± 0.06
0.0199	1.19 ± 0.02	1.85 ± 0.03	1.98 ± 0.03	2.00 ± 0.03	2.17 ± 0.03	2.26 ± 0.04	2.33 ± 0.04	2.36 ± 0.04	2.49 ± 0.05	2.75 ± 0.05
0.0213	1.19 ± 0.02	1.81 ± 0.03	1.90 ± 0.03	1.98 ± 0.03	2.13 ± 0.03	2.15 ± 0.03	2.32 ± 0.04	2.33 ± 0.04	2.48 ± 0.04	2.56 ± 0.05
0.0228	1.18 ± 0.02	1.80 ± 0.03	1.90 ± 0.03	2.00 ± 0.03	2.07 ± 0.03	2.13 ± 0.03	2.26 ± 0.03	2.28 ± 0.04	2.48 ± 0.04	2.52 ± 0.04
0.0242	1.17 ± 0.02	1.78 ± 0.02	1.88 ± 0.03	1.96 ± 0.03	1.99 ± 0.03	2.08 ± 0.03	2.15 ± 0.03	2.21 ± 0.04	2.40 ± 0.04	2.43 ± 0.04
0.0256	1.17 ± 0.02	1.70 ± 0.02	1.79 ± 0.02	1.87 ± 0.03	1.93 ± 0.03	2.02 ± 0.03	2.10 ± 0.03	2.08 ± 0.03	2.23 ± 0.04	2.30 ± 0.04
0.0270	1.20 ± 0.02	1.81 ± 0.02	1.81 ± 0.02	1.90 ± 0.02	1.94 ± 0.03	2.03 ± 0.03	2.16 ± 0.03	2.19 ± 0.03	2.29 ± 0.03	2.38 ± 0.04
0.0284	1.24 ± 0.02	1.85 ± 0.02	1.86 ± 0.02	1.91 ± 0.02	1.95 ± 0.03	2.06 ± 0.03	2.12 ± 0.03	2.19 ± 0.03	2.33 ± 0.03	2.43 ± 0.04
0.0299	1.17 ± 0.02	1.71 ± 0.02	1.74 ± 0.02	1.80 ± 0.02	1.89 ± 0.02	1.97 ± 0.03	2.03 ± 0.03	2.03 ± 0.03	2.13 ± 0.03	2.26 ± 0.04
0.0320	1.21 ± 0.01	1.70 ± 0.01	1.73 ± 0.01	1.80 ± 0.02	1.88 ± 0.02	1.93 ± 0.02	2.00 ± 0.02	2.06 ± 0.02	2.13 ± 0.02	2.20 ± 0.02
0.0349	1.16 ± 0.01	1.58 ± 0.01	1.64 ± 0.01	1.68 ± 0.01	1.72 ± 0.02	1.76 ± 0.02	1.82 ± 0.02	1.88 ± 0.02	1.95 ± 0.02	1.96 ± 0.02
0.0377	1.11 ± 0.01	1.49 ± 0.01	1.56 ± 0.01	1.56 ± 0.01	1.61 ± 0.01	1.65 ± 0.02	1.70 ± 0.02	1.75 ± 0.02	1.80 ± 0.02	1.87 ± 0.02
0.0405	1.13 ± 0.01	1.46 ± 0.01	1.48 ± 0.01	1.51 ± 0.01	1.59 ± 0.01	1.63 ± 0.01	1.67 ± 0.02	1.67 ± 0.02	1.74 ± 0.02	1.78 ± 0.02
0.0434	1.09 ± 0.01	1.41 ± 0.01	1.44 ± 0.01	1.48 ± 0.01	1.51 ± 0.01	1.54 ± 0.01	1.58 ± 0.01	1.55 ± 0.02	1.62 ± 0.02	1.70 ± 0.02
0.0462	1.07 ± 0.01	1.34 ± 0.01	1.37 ± 0.01	1.40 ± 0.01	1.41 ± 0.01	1.44 ± 0.01	1.49 ± 0.01	1.50 ± 0.01	1.50 ± 0.02	1.56 ± 0.02
0.0491	1.04 ± 0.01	1.27 ± 0.01	1.28 ± 0.01	1.30 ± 0.01	1.34 ± 0.01	1.35 ± 0.01	1.37 ± 0.01	1.38 ± 0.01	1.36 ± 0.01	1.45 ± 0.02
0.0519	1.01 ± 0.01	1.20 ± 0.01	1.20 ± 0.01	1.23 ± 0.01	1.26 ± 0.01	1.27 ± 0.01	1.33 ± 0.01	1.30 ± 0.01	1.33 ± 0.01	1.37 ± 0.01
0.0548	0.981 ± 0.007	1.14 ± 0.01	1.15 ± 0.01	1.13 ± 0.01	1.17 ± 0.01	1.19 ± 0.01	1.20 ± 0.01	1.20 ± 0.01	1.22 ± 0.01	1.24 ± 0.01
0.0576	0.930 ± 0.007	1.05 ± 0.01	1.06 ± 0.01	1.07 ± 0.01	1.07 ± 0.01	1.10 ± 0.01	1.09 ± 0.01	1.11 ± 0.01	1.11 ± 0.01	1.12 ± 0.01
0.0605	0.887 ± 0.007	1.01 ± 0.01	0.997 ± 0.008	0.994 ± 0.009	1.00 ± 0.01	1.01 ± 0.01	1.01 ± 0.01	1.05 ± 0.01	1.02 ± 0.01	1.02 ± 0.01
0.0647	0.792 ± 0.004	0.858 ± 0.005	0.860 ± 0.005	0.866 ± 0.005	0.865 ± 0.006	0.865 ± 0.006	0.884 ± 0.006	0.861 ± 0.007	0.880 ± 0.007	0.884 ± 0.008
0.0704	0.683 ± 0.004	0.729 ± 0.005	0.718 ± 0.005	0.721 ± 0.005	0.714 ± 0.005	0.720 ± 0.005	0.715 ± 0.006	0.718 ± 0.006	0.726 ± 0.006	0.735 ± 0.007



$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 67.90$	$c_p = 65.51$	$c_p = 59.30$	$c_p = 54.92$	$c_p = 49.95$	$c_p = 45.85$	$c_p = 40.86$	$c_p = 36.68$	$c_p = 31.10$
0.0761	$0.572 \pm 0.003$	$0.576 \pm 0.004$	$0.584 \pm 0.004$	$0.586 \pm 0.004$	$0.582 \pm 0.004$	$0.582 \pm 0.005$	$0.585 \pm 0.005$	$0.571 \pm 0.005$	$0.583 \pm 0.006$	$0.576 \pm 0.006$
0.0818	$0.468 \pm 0.003$	$0.465 \pm 0.003$	$0.466 \pm 0.004$	$0.466 \pm 0.004$	$0.466 \pm 0.004$	$0.457 \pm 0.004$	$0.467 \pm 0.004$	$0.471 \pm 0.005$	$0.454 \pm 0.005$	$0.467 \pm 0.005$
0.0875	$0.374 \pm 0.003$	$0.372 \pm 0.003$	$0.375 \pm 0.003$	$0.367 \pm 0.003$	$0.365 \pm 0.003$	$0.368 \pm 0.004$	$0.361 \pm 0.004$	$0.366 \pm 0.004$	$0.375 \pm 0.004$	$0.360 \pm 0.005$
0.0932	$0.295 \pm 0.002$	$0.289 \pm 0.003$	$0.291 \pm 0.003$	$0.285 \pm 0.003$	$0.285 \pm 0.003$	$0.289 \pm 0.003$	$0.292 \pm 0.003$	$0.283 \pm 0.004$	$0.286 \pm 0.004$	$0.288 \pm 0.004$
0.0989	$0.233 \pm 0.002$	$0.225 \pm 0.002$	$0.222 \pm 0.002$	$0.227 \pm 0.002$	$0.221 \pm 0.003$	$0.230 \pm 0.003$	$0.230 \pm 0.003$	$0.228 \pm 0.003$	$0.229 \pm 0.003$	$0.234 \pm 0.004$
0.1074	$0.152 \pm 0.001$	$0.153 \pm 0.001$	$0.150 \pm 0.001$	$0.152 \pm 0.001$	$0.153 \pm 0.002$	$0.155 \pm 0.002$	$0.150 \pm 0.002$	$0.155 \pm 0.002$	$0.151 \pm 0.002$	$0.154 \pm 0.002$
0.1188	$0.0964 \pm 0.0009$	$0.101 \pm 0.001$	$0.0963 \pm 0.0011$	$0.0983 \pm 0.0012$	$0.0991 \pm 0.0013$	$0.0975 \pm 0.0014$	$0.0979 \pm 0.0015$	$0.0986 \pm 0.0016$	$0.101 \pm 0.002$	$0.0997 \pm 0.0019$
0.1302	$0.0656 \pm 0.0008$	$0.0678 \pm 0.0010$	$0.0692 \pm 0.0010$	$0.0680 \pm 0.0010$	$0.0698 \pm 0.0011$	$0.0687 \pm 0.0012$	$0.0681 \pm 0.0013$	$0.0686 \pm 0.0014$	$0.0697 \pm 0.0015$	$0.0666 \pm 0.0017$
0.1415	$0.0517 \pm 0.0007$	$0.0520 \pm 0.0009$	$0.0533 \pm 0.0009$	$0.0536 \pm 0.0009$	$0.0550 \pm 0.0010$	$0.0551 \pm 0.0011$	$0.0522 \pm 0.0012$	$0.0542 \pm 0.0013$	$0.0532 \pm 0.0014$	$0.0527 \pm 0.0016$
0.1529	$0.0438 \pm 0.0007$	$0.0441 \pm 0.0008$	$0.0431 \pm 0.0008$	$0.0441 \pm 0.0009$	$0.0420 \pm 0.0009$	$0.0424 \pm 0.0010$	$0.0444 \pm 0.0011$	$0.0440 \pm 0.0012$	$0.0435 \pm 0.0013$	$0.0450 \pm 0.0015$
0.1643	$0.0358 \pm 0.0006$	$0.0362 \pm 0.0007$	$0.0346 \pm 0.0007$	$0.0353 \pm 0.0008$	$0.0363 \pm 0.0009$	$0.0351 \pm 0.0009$	$0.0348 \pm 0.0010$	$0.0343 \pm 0.0011$	$0.0346 \pm 0.0012$	$0.0364 \pm 0.0014$
0.1757	$0.0281 \pm 0.0005$	$0.0283 \pm 0.0007$	$0.0289 \pm 0.0007$	$0.0289 \pm 0.0007$	$0.0318 \pm 0.0008$	$0.0296 \pm 0.0009$	$0.0297 \pm 0.0009$	$0.0301 \pm 0.0010$	$0.0295 \pm 0.0011$	$0.0286 \pm 0.0013$
0.1870	$0.0252 \pm 0.0005$	$0.0252 \pm 0.0006$	$0.0256 \pm 0.0007$	$0.0249 \pm 0.0007$	$0.0243 \pm 0.0007$	$0.0261 \pm 0.0008$	$0.0249 \pm 0.0009$	$0.0252 \pm 0.0010$	$0.0247 \pm 0.0011$	$0.0246 \pm 0.0012$
0.1984	$0.0198 \pm 0.0005$	$0.0195 \pm 0.0006$	$0.0208 \pm 0.0006$	$0.0197 \pm 0.0007$	$0.0194 \pm 0.0007$	$0.0185 \pm 0.0008$	$0.0217 \pm 0.0008$	$0.0196 \pm 0.0009$	$0.0184 \pm 0.0010$	$0.0184 \pm 0.0012$
0.2098	$0.0156 \pm 0.0005$	$0.0170 \pm 0.0006$	$0.0167 \pm 0.0006$	$0.0162 \pm 0.0006$	$0.0160 \pm 0.0007$	$0.0150 \pm 0.0007$	$0.0166 \pm 0.0008$	$0.0167 \pm 0.0009$	$0.0147 \pm 0.0010$	$0.0183 \pm 0.0011$
0.2212	$0.0149 \pm 0.0004$	$0.0145 \pm 0.0005$	$0.0134 \pm 0.0006$	$0.0146 \pm 0.0006$	$0.0152 \pm 0.0007$	$0.0140 \pm 0.0007$	$0.0130 \pm 0.0008$	$0.0145 \pm 0.0009$	$0.0133 \pm 0.0010$	$0.0140 \pm 0.0011$
0.2325	$0.0128 \pm 0.0004$	$0.0120 \pm 0.0005$	$0.0114 \pm 0.0006$	$0.0127 \pm 0.0006$	$0.0124 \pm 0.0006$	$0.0119 \pm 0.0007$	$0.0134 \pm 0.0008$	$0.0122 \pm 0.0008$	$0.0132 \pm 0.0009$	$0.0120 \pm 0.0011$
0.2439	$0.00997 \pm 0.00042$	$0.0101 \pm 0.0005$	$0.00996 \pm 0.00053$	$0.0106 \pm 0.0006$	$0.00987 \pm 0.00062$	$0.00999 \pm 0.00067$	$0.0113 \pm 0.0007$	$0.0115 \pm 0.0008$	$0.0113 \pm 0.0009$	$0.0108 \pm 0.0011$
0.2553	$0.00922 \pm 0.00041$	$0.00887 \pm 0.00050$	$0.00976 \pm 0.00052$	$0.0104 \pm 0.0006$	$0.0104 \pm 0.0006$	$0.00924 \pm 0.00066$	$0.00880 \pm 0.00072$	$0.0108 \pm 0.0008$	$0.0108 \pm 0.0009$	$0.00879 \pm 0.00103$
0.2667	$0.00893 \pm 0.00040$	$0.00850 \pm 0.00049$	$0.00834 \pm 0.00051$	$0.00917 \pm 0.00055$	$0.00753 \pm 0.00059$	$0.00890 \pm 0.00065$	$0.00896 \pm 0.00071$	$0.00847 \pm 0.00079$	$0.00987 \pm 0.00088$	$0.00837 \pm 0.00100$
0.2780	$0.00868 \pm 0.00039$	$0.00851 \pm 0.00048$	$0.00895 \pm 0.00050$	$0.00856 \pm 0.00054$	$0.00860 \pm 0.00058$	$0.00850 \pm 0.00064$	$0.00915 \pm 0.00070$	$0.00896 \pm 0.00078$	$0.00889 \pm 0.00086$	$0.00823 \pm 0.00099$
0.2894	$0.00952 \pm 0.00038$	$0.00920 \pm 0.00047$	$0.00835 \pm 0.00049$	$0.0100 \pm 0.0005$	$0.00826 \pm 0.00057$	$0.00949 \pm 0.00063$	$0.00842 \pm 0.00068$	$0.0104 \pm 0.0008$	$0.00913 \pm 0.00085$	$0.00899 \pm 0.00098$
0.3008	$0.00958 \pm 0.00038$	$0.00987 \pm 0.00047$	$0.00953 \pm 0.00049$	$0.00853 \pm 0.00052$	$0.00972 \pm 0.00057$	$0.00936 \pm 0.00062$	$0.00895 \pm 0.00067$	$0.00871 \pm 0.00076$	$0.00869 \pm 0.00084$	$0.00898 \pm 0.00097$
0.3122	$0.0104 \pm 0.0004$	$0.00984 \pm 0.00046$	$0.00949 \pm 0.00048$	$0.00922 \pm 0.00052$	$0.0101 \pm 0.0006$	$0.0111 \pm 0.0006$	$0.00938 \pm 0.00067$	$0.00847 \pm 0.00075$	$0.00838 \pm 0.00083$	$0.00941 \pm 0.00095$
0.3235	$0.00984 \pm 0.00037$	$0.00894 \pm 0.00046$	$0.00966 \pm 0.00047$	$0.0102 \pm 0.0005$	$0.00925 \pm 0.00056$	$0.00972 \pm 0.00061$	$0.01000 \pm 0.00066$	$0.00736 \pm 0.00073$	$0.00960 \pm 0.00082$	$0.0112 \pm 0.0009$
0.3349	$0.00907 \pm 0.00036$	$0.00950 \pm 0.00045$	$0.0103 \pm 0.0005$	$0.00892 \pm 0.00050$	$0.00934 \pm 0.00055$	$0.00980 \pm 0.00060$	$0.00873 \pm 0.00065$	$0.00971 \pm 0.00073$	$0.00945 \pm 0.00081$	$0.00826 \pm 0.00092$
0.3448	$0.00851 \pm 0.00041$	$0.00941 \pm 0.00051$	$0.00979 \pm 0.00053$	$0.00717 \pm 0.00056$	$0.00798 \pm 0.00061$	$0.00770 \pm 0.00067$	$0.00847 \pm 0.00073$	$0.00816 \pm 0.00083$	$0.00929 \pm 0.00092$	$0.00981 \pm 0.00105$

Table 12: Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector,  $q$ , for BSA in 1.0 mol·L<sup>-1</sup> Na<sub>2</sub>SO<sub>4</sub> aqueous solutions at pH = 4.9, at several protein concentrations,  $c_p$ , (mg·mL<sup>-1</sup>).

$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 73.10$	$c_p = 62.94$	$c_p = 58.29$	$c_p = 55.14$	$c_p = 52.80$	$c_p = 48.24$	$c_p = 45.93$	$c_p = 40.44$	$c_p = 36.38$
0.0142	0.478 ± 0.033	0.669 ± 0.055	0.826 ± 0.078	0.696 ± 0.057	0.819 ± 0.060	0.878 ± 0.063	0.850 ± 0.070	0.882 ± 0.073	0.952 ± 0.080	0.910 ± 0.046
0.0156	0.479 ± 0.019	0.699 ± 0.037	0.740 ± 0.042	0.723 ± 0.032	0.776 ± 0.034	0.880 ± 0.035	0.832 ± 0.038	0.917 ± 0.041	0.818 ± 0.044	0.894 ± 0.026
0.0171	0.487 ± 0.018	0.688 ± 0.033	0.729 ± 0.037	0.730 ± 0.028	0.789 ± 0.030	0.791 ± 0.031	0.831 ± 0.034	0.876 ± 0.036	0.858 ± 0.040	0.908 ± 0.023
0.0185	0.510 ± 0.016	0.700 ± 0.029	0.714 ± 0.033	0.731 ± 0.025	0.773 ± 0.027	0.780 ± 0.028	0.808 ± 0.030	0.864 ± 0.032	0.852 ± 0.035	0.869 ± 0.021
0.0199	0.526 ± 0.015	0.716 ± 0.027	0.708 ± 0.031	0.724 ± 0.023	0.849 ± 0.026	0.773 ± 0.025	0.835 ± 0.029	0.831 ± 0.029	0.838 ± 0.032	0.880 ± 0.019
0.0213	0.478 ± 0.013	0.703 ± 0.025	0.726 ± 0.028	0.708 ± 0.021	0.730 ± 0.023	0.737 ± 0.023	0.807 ± 0.026	0.773 ± 0.027	0.834 ± 0.028	0.846 ± 0.017
0.0228	0.482 ± 0.012	0.681 ± 0.023	0.723 ± 0.026	0.709 ± 0.020	0.748 ± 0.021	0.769 ± 0.021	0.781 ± 0.023	0.772 ± 0.024	0.810 ± 0.026	0.824 ± 0.015
0.0242	0.495 ± 0.012	0.639 ± 0.022	0.694 ± 0.024	0.718 ± 0.019	0.720 ± 0.019	0.757 ± 0.020	0.740 ± 0.021	0.719 ± 0.022	0.759 ± 0.024	0.802 ± 0.015
0.0256	0.489 ± 0.011	0.607 ± 0.020	0.653 ± 0.023	0.670 ± 0.017	0.684 ± 0.018	0.700 ± 0.018	0.724 ± 0.020	0.685 ± 0.020	0.731 ± 0.023	0.766 ± 0.013
0.0270	0.484 ± 0.011	0.597 ± 0.019	0.687 ± 0.022	0.675 ± 0.017	0.716 ± 0.018	0.710 ± 0.018	0.707 ± 0.019	0.700 ± 0.020	0.740 ± 0.022	0.750 ± 0.013
0.0284	0.481 ± 0.010	0.636 ± 0.019	0.689 ± 0.021	0.667 ± 0.016	0.689 ± 0.017	0.719 ± 0.017	0.708 ± 0.018	0.658 ± 0.019	0.766 ± 0.021	0.752 ± 0.012
0.0299	0.478 ± 0.010	0.576 ± 0.017	0.624 ± 0.020	0.666 ± 0.015	0.642 ± 0.015	0.651 ± 0.016	0.690 ± 0.017	0.661 ± 0.017	0.701 ± 0.020	0.728 ± 0.011
0.0320	0.466 ± 0.007	0.594 ± 0.012	0.623 ± 0.013	0.649 ± 0.010	0.626 ± 0.010	0.649 ± 0.011	0.697 ± 0.012	0.715 ± 0.012	0.697 ± 0.013	0.716 ± 0.008
0.0349	0.430 ± 0.006	0.544 ± 0.011	0.611 ± 0.012	0.576 ± 0.009	0.589 ± 0.009	0.597 ± 0.010	0.635 ± 0.011	0.629 ± 0.011	0.625 ± 0.012	0.649 ± 0.007
0.0377	0.422 ± 0.006	0.500 ± 0.010	0.557 ± 0.011	0.547 ± 0.008	0.559 ± 0.009	0.567 ± 0.009	0.574 ± 0.010	0.572 ± 0.010	0.575 ± 0.011	0.600 ± 0.006
0.0405	0.403 ± 0.006	0.497 ± 0.010	0.518 ± 0.011	0.533 ± 0.008	0.558 ± 0.009	0.538 ± 0.009	0.545 ± 0.009	0.552 ± 0.010	0.569 ± 0.010	0.586 ± 0.006
0.0434	0.407 ± 0.005	0.473 ± 0.009	0.498 ± 0.010	0.508 ± 0.008	0.533 ± 0.008	0.519 ± 0.008	0.506 ± 0.009	0.527 ± 0.009	0.558 ± 0.010	0.548 ± 0.006
0.0462	0.376 ± 0.005	0.429 ± 0.009	0.469 ± 0.010	0.474 ± 0.007	0.488 ± 0.008	0.483 ± 0.008	0.481 ± 0.008	0.482 ± 0.009	0.504 ± 0.009	0.500 ± 0.005
0.0491	0.367 ± 0.005	0.424 ± 0.008	0.442 ± 0.009	0.452 ± 0.007	0.453 ± 0.007	0.445 ± 0.007	0.461 ± 0.008	0.437 ± 0.008	0.461 ± 0.009	0.476 ± 0.005
0.0519	0.358 ± 0.005	0.393 ± 0.008	0.423 ± 0.009	0.424 ± 0.007	0.427 ± 0.007	0.423 ± 0.007	0.445 ± 0.008	0.448 ± 0.008	0.420 ± 0.008	0.440 ± 0.005
0.0548	0.342 ± 0.004	0.372 ± 0.007	0.395 ± 0.008	0.374 ± 0.006	0.395 ± 0.007	0.388 ± 0.007	0.402 ± 0.007	0.390 ± 0.007	0.396 ± 0.008	0.393 ± 0.005
0.0576	0.326 ± 0.004	0.352 ± 0.007	0.350 ± 0.008	0.366 ± 0.006	0.365 ± 0.006	0.370 ± 0.006	0.360 ± 0.007	0.372 ± 0.007	0.379 ± 0.008	0.383 ± 0.004
0.0605	0.306 ± 0.004	0.320 ± 0.007	0.313 ± 0.007	0.335 ± 0.006	0.332 ± 0.006	0.339 ± 0.006	0.337 ± 0.006	0.323 ± 0.007	0.345 ± 0.007	0.338 ± 0.004
0.0647	0.273 ± 0.003	0.283 ± 0.004	0.301 ± 0.005	0.301 ± 0.004	0.294 ± 0.004	0.295 ± 0.004	0.295 ± 0.004	0.297 ± 0.004	0.297 ± 0.005	0.300 ± 0.003
0.0704	0.241 ± 0.002	0.245 ± 0.004	0.251 ± 0.004	0.244 ± 0.003	0.252 ± 0.004	0.253 ± 0.004	0.257 ± 0.004	0.245 ± 0.004	0.257 ± 0.004	0.251 ± 0.003

$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 73.10$	$c_p = 62.94$	$c_p = 58.29$	$c_p = 55.14$	$c_p = 52.80$	$c_p = 48.24$	$c_p = 45.93$	$c_p = 40.44$	$c_p = 36.38$
0.0761	$0.206 \pm 0.002$	$0.205 \pm 0.004$	$0.206 \pm 0.004$	$0.208 \pm 0.003$	$0.207 \pm 0.003$	$0.204 \pm 0.003$	$0.208 \pm 0.003$	$0.212 \pm 0.004$	$0.202 \pm 0.004$	$0.206 \pm 0.002$
0.0818	$0.171 \pm 0.002$	$0.170 \pm 0.003$	$0.167 \pm 0.004$	$0.172 \pm 0.003$	$0.170 \pm 0.003$	$0.172 \pm 0.003$	$0.173 \pm 0.003$	$0.166 \pm 0.003$	$0.172 \pm 0.004$	$0.170 \pm 0.002$
0.0875	$0.140 \pm 0.002$	$0.141 \pm 0.003$	$0.145 \pm 0.003$	$0.138 \pm 0.003$	$0.135 \pm 0.003$	$0.138 \pm 0.003$	$0.135 \pm 0.003$	$0.139 \pm 0.003$	$0.138 \pm 0.003$	$0.138 \pm 0.002$
0.0932	$0.114 \pm 0.002$	$0.114 \pm 0.003$	$0.108 \pm 0.003$	$0.109 \pm 0.002$	$0.112 \pm 0.003$	$0.112 \pm 0.003$	$0.108 \pm 0.003$	$0.108 \pm 0.003$	$0.113 \pm 0.003$	$0.108 \pm 0.002$
0.0989	$0.0903 \pm 0.0016$	$0.0861 \pm 0.0025$	$0.0898 \pm 0.0029$	$0.0946 \pm 0.0022$	$0.0907 \pm 0.0023$	$0.0896 \pm 0.0024$	$0.0931 \pm 0.0026$	$0.0884 \pm 0.0027$	$0.0839 \pm 0.0030$	$0.0914 \pm 0.0018$
0.1074	$0.0634 \pm 0.0010$	$0.0617 \pm 0.0016$	$0.0626 \pm 0.0018$	$0.0587 \pm 0.0014$	$0.0644 \pm 0.0015$	$0.0646 \pm 0.0015$	$0.0616 \pm 0.0016$	$0.0610 \pm 0.0017$	$0.0613 \pm 0.0019$	$0.0627 \pm 0.0011$
0.1188	$0.0403 \pm 0.0009$	$0.0439 \pm 0.0014$	$0.0400 \pm 0.0016$	$0.0406 \pm 0.0013$	$0.0409 \pm 0.0013$	$0.0418 \pm 0.0014$	$0.0409 \pm 0.0015$	$0.0412 \pm 0.0016$	$0.0445 \pm 0.0018$	$0.0425 \pm 0.0010$
0.1302	$0.0288 \pm 0.0008$	$0.0290 \pm 0.0013$	$0.0302 \pm 0.0015$	$0.0308 \pm 0.0012$	$0.0289 \pm 0.0012$	$0.0283 \pm 0.0012$	$0.0303 \pm 0.0014$	$0.0305 \pm 0.0014$	$0.0313 \pm 0.0016$	$0.0299 \pm 0.0010$
0.1415	$0.0227 \pm 0.0007$	$0.0207 \pm 0.0012$	$0.0222 \pm 0.0014$	$0.0223 \pm 0.0011$	$0.0205 \pm 0.0011$	$0.0215 \pm 0.0012$	$0.0225 \pm 0.0013$	$0.0207 \pm 0.0013$	$0.0207 \pm 0.0015$	$0.0221 \pm 0.0009$
0.1529	$0.0167 \pm 0.0007$	$0.0169 \pm 0.0011$	$0.0178 \pm 0.0013$	$0.0188 \pm 0.0010$	$0.0175 \pm 0.0011$	$0.0184 \pm 0.0011$	$0.0182 \pm 0.0012$	$0.0175 \pm 0.0013$	$0.0192 \pm 0.0014$	$0.0176 \pm 0.0008$
0.1643	$0.0144 \pm 0.0006$	$0.0160 \pm 0.0011$	$0.0176 \pm 0.0012$	$0.0165 \pm 0.0009$	$0.0171 \pm 0.0010$	$0.0167 \pm 0.0010$	$0.0152 \pm 0.0011$	$0.0163 \pm 0.0012$	$0.0178 \pm 0.0013$	$0.0170 \pm 0.0008$
0.1757	$0.0126 \pm 0.0006$	$0.0138 \pm 0.0010$	$0.0118 \pm 0.0012$	$0.0123 \pm 0.0009$	$0.0131 \pm 0.0010$	$0.0123 \pm 0.0010$	$0.0119 \pm 0.0011$	$0.0151 \pm 0.0012$	$0.0138 \pm 0.0013$	$0.0112 \pm 0.0008$
0.1870	$0.00944 \pm 0.00057$	$0.0122 \pm 0.0010$	$0.0119 \pm 0.0011$	$0.00962 \pm 0.00086$	$0.0127 \pm 0.0009$	$0.00942 \pm 0.00094$	$0.0119 \pm 0.0010$	$0.0135 \pm 0.0011$	$0.0112 \pm 0.0012$	$0.0116 \pm 0.0007$
0.1984	$0.00856 \pm 0.00054$	$0.00971 \pm 0.00090$	$0.00988 \pm 0.00107$	$0.00974 \pm 0.00083$	$0.00893 \pm 0.00088$	$0.00839 \pm 0.00090$	$0.0109 \pm 0.0010$	$0.00895 \pm 0.00104$	$0.00916 \pm 0.00118$	$0.00943 \pm 0.00071$
0.2098	$0.00783 \pm 0.00053$	$0.00888 \pm 0.00088$	$0.00969 \pm 0.00104$	$0.00679 \pm 0.00080$	$0.00734 \pm 0.00085$	$0.00796 \pm 0.00088$	$0.00752 \pm 0.00095$	$0.00555 \pm 0.00101$	$0.00521 \pm 0.00113$	$0.00805 \pm 0.00069$
0.2212	$0.00574 \pm 0.00051$	$0.00587 \pm 0.00085$	$0.00678 \pm 0.00101$	$0.00744 \pm 0.00078$	$0.00588 \pm 0.00083$	$0.00623 \pm 0.00086$	$0.00564 \pm 0.00094$	$0.00806 \pm 0.00100$	$0.00658 \pm 0.00111$	$0.00679 \pm 0.00067$
0.2325	$0.00598 \pm 0.00050$	$0.00572 \pm 0.00083$	$0.00712 \pm 0.00099$	$0.00699 \pm 0.00076$	$0.00668 \pm 0.00081$	$0.00712 \pm 0.00084$	$0.00638 \pm 0.00091$	$0.00780 \pm 0.00097$	$0.00674 \pm 0.00108$	$0.00605 \pm 0.00065$
0.2439	$0.00484 \pm 0.00048$	$0.00394 \pm 0.00080$	$0.00556 \pm 0.00096$	$0.00468 \pm 0.00074$	$0.00551 \pm 0.00079$	$0.00608 \pm 0.00081$	$0.00604 \pm 0.00089$	$0.00482 \pm 0.00094$	$0.00591 \pm 0.00106$	$0.00432 \pm 0.00063$
0.2553	$0.00568 \pm 0.00047$	$0.00547 \pm 0.00079$	$0.00465 \pm 0.00093$	$0.00543 \pm 0.00072$	$0.00568 \pm 0.00077$	$0.00512 \pm 0.00079$	$0.00697 \pm 0.00088$	$0.00664 \pm 0.00093$	$0.00566 \pm 0.00103$	$0.00565 \pm 0.00062$
0.2667	$0.00428 \pm 0.00046$	$0.00592 \pm 0.00078$	$0.00400 \pm 0.00092$	$0.00468 \pm 0.00071$	$0.00588 \pm 0.00076$	$0.00397 \pm 0.00078$	$0.00574 \pm 0.00086$	$0.00583 \pm 0.00090$	$0.00683 \pm 0.00102$	$0.00689 \pm 0.00061$
0.2780	$0.00489 \pm 0.00046$	$0.00411 \pm 0.00075$	$0.00437 \pm 0.00090$	$0.00625 \pm 0.00070$	$0.00701 \pm 0.00074$	$0.00630 \pm 0.00077$	$0.00636 \pm 0.00083$	$0.00759 \pm 0.00089$	$0.00577 \pm 0.00100$	$0.00664 \pm 0.00060$
0.2894	$0.00646 \pm 0.00045$	$0.00595 \pm 0.00075$	$0.00540 \pm 0.00088$	$0.00760 \pm 0.00069$	$0.00633 \pm 0.00073$	$0.00622 \pm 0.00075$	$0.00674 \pm 0.00082$	$0.00534 \pm 0.00087$	$0.00588 \pm 0.00098$	$0.00779 \pm 0.00059$
0.3008	$0.00693 \pm 0.00044$	$0.00687 \pm 0.00074$	$0.00792 \pm 0.00087$	$0.00723 \pm 0.00067$	$0.00668 \pm 0.00072$	$0.00697 \pm 0.00074$	$0.00563 \pm 0.00081$	$0.00622 \pm 0.00085$	$0.00578 \pm 0.00096$	$0.00689 \pm 0.00057$
0.3122	$0.00769 \pm 0.00043$	$0.00714 \pm 0.00072$	$0.00639 \pm 0.00085$	$0.00704 \pm 0.00066$	$0.00720 \pm 0.00070$	$0.00725 \pm 0.00073$	$0.00671 \pm 0.00080$	$0.00632 \pm 0.00084$	$0.00575 \pm 0.00094$	$0.00636 \pm 0.00057$
0.3235	$0.00800 \pm 0.00043$	$0.00713 \pm 0.00072$	$0.00441 \pm 0.00084$	$0.00491 \pm 0.00065$	$0.00445 \pm 0.00069$	$0.00681 \pm 0.00072$	$0.00591 \pm 0.00078$	$0.00507 \pm 0.00083$	$0.00783 \pm 0.00093$	$0.00540 \pm 0.00056$
0.3349	$0.00711 \pm 0.00042$	$0.00563 \pm 0.00070$	$0.00613 \pm 0.00083$	$0.00615 \pm 0.00064$	$0.00429 \pm 0.00068$	$0.00534 \pm 0.00070$	$0.00507 \pm 0.00077$	$0.00376 \pm 0.00081$	$0.00553 \pm 0.00091$	$0.00434 \pm 0.00055$
0.3448	$0.00619 \pm 0.00048$	$0.00597 \pm 0.00080$	$0.00700 \pm 0.00094$	$0.00461 \pm 0.00072$	$0.00646 \pm 0.00077$	$0.00577 \pm 0.00080$	$0.00398 \pm 0.00087$	$0.00562 \pm 0.00092$	$0.00325 \pm 0.00103$	$0.00445 \pm 0.00062$

Table 13: Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector,  $q$ , for BSA in 1.0 mol·L<sup>-1</sup> NaNO<sub>3</sub> aqueous solutions at pH = 6.3, at several protein concentrations,  $c_p$ , (mg·mL<sup>-1</sup>).

$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 79.88$	$c_p = 75.85$	$c_p = 69.96$	$c_p = 64.90$	$c_p = 59.46$	$c_p = 55.45$	$c_p = 48.53$	$c_p = 43.91$	$c_p = 39.63$
0.0142	1.12 ± 0.04	1.45 ± 0.05	1.49 ± 0.05	1.58 ± 0.05	1.69 ± 0.06	1.73 ± 0.06	1.93 ± 0.07	2.04 ± 0.08	2.03 ± 0.08	2.15 ± 0.09
0.0156	1.11 ± 0.02	1.47 ± 0.03	1.52 ± 0.03	1.57 ± 0.03	1.70 ± 0.04	1.70 ± 0.04	1.86 ± 0.04	1.99 ± 0.05	2.06 ± 0.05	2.12 ± 0.06
0.0171	1.13 ± 0.02	1.46 ± 0.03	1.56 ± 0.03	1.57 ± 0.03	1.68 ± 0.03	1.77 ± 0.04	1.87 ± 0.04	1.93 ± 0.04	2.10 ± 0.05	2.13 ± 0.05
0.0185	1.09 ± 0.02	1.46 ± 0.03	1.55 ± 0.03	1.56 ± 0.03	1.70 ± 0.03	1.78 ± 0.03	1.78 ± 0.03	1.89 ± 0.04	2.00 ± 0.04	2.12 ± 0.05
0.0199	1.12 ± 0.02	1.47 ± 0.02	1.51 ± 0.03	1.56 ± 0.03	1.66 ± 0.03	1.76 ± 0.03	1.74 ± 0.03	1.91 ± 0.04	1.98 ± 0.04	2.09 ± 0.04
0.0213	1.12 ± 0.02	1.42 ± 0.02	1.49 ± 0.02	1.57 ± 0.03	1.62 ± 0.03	1.74 ± 0.03	1.76 ± 0.03	1.90 ± 0.03	1.92 ± 0.04	2.06 ± 0.04
0.0228	1.11 ± 0.02	1.41 ± 0.02	1.49 ± 0.02	1.55 ± 0.02	1.62 ± 0.03	1.68 ± 0.03	1.78 ± 0.03	1.88 ± 0.03	1.87 ± 0.03	1.99 ± 0.04
0.0242	1.09 ± 0.02	1.43 ± 0.02	1.48 ± 0.02	1.53 ± 0.02	1.62 ± 0.02	1.67 ± 0.03	1.73 ± 0.03	1.83 ± 0.03	1.88 ± 0.03	1.97 ± 0.04
0.0256	1.11 ± 0.02	1.39 ± 0.02	1.45 ± 0.02	1.51 ± 0.02	1.60 ± 0.02	1.66 ± 0.03	1.72 ± 0.03	1.78 ± 0.03	1.85 ± 0.03	1.93 ± 0.04
0.0270	1.13 ± 0.02	1.40 ± 0.02	1.46 ± 0.02	1.49 ± 0.02	1.59 ± 0.02	1.62 ± 0.02	1.71 ± 0.03	1.76 ± 0.03	1.82 ± 0.03	1.93 ± 0.03
0.0284	1.10 ± 0.02	1.40 ± 0.02	1.41 ± 0.02	1.49 ± 0.02	1.52 ± 0.02	1.59 ± 0.02	1.60 ± 0.03	1.75 ± 0.03	1.75 ± 0.03	1.87 ± 0.03
0.0299	1.06 ± 0.02	1.40 ± 0.02	1.40 ± 0.02	1.47 ± 0.02	1.51 ± 0.02	1.55 ± 0.02	1.59 ± 0.02	1.74 ± 0.03	1.78 ± 0.03	1.84 ± 0.03
0.0320	1.08 ± 0.01	1.33 ± 0.01	1.40 ± 0.01	1.44 ± 0.01	1.49 ± 0.01	1.55 ± 0.02	1.56 ± 0.02	1.71 ± 0.02	1.68 ± 0.02	1.75 ± 0.02
0.0349	1.07 ± 0.01	1.32 ± 0.01	1.33 ± 0.01	1.39 ± 0.01	1.41 ± 0.01	1.48 ± 0.01	1.51 ± 0.02	1.59 ± 0.02	1.61 ± 0.02	1.71 ± 0.02
0.0377	1.04 ± 0.01	1.27 ± 0.01	1.30 ± 0.01	1.36 ± 0.01	1.39 ± 0.01	1.43 ± 0.01	1.45 ± 0.01	1.49 ± 0.02	1.55 ± 0.02	1.61 ± 0.02
0.0405	1.02 ± 0.01	1.22 ± 0.01	1.25 ± 0.01	1.28 ± 0.01	1.31 ± 0.01	1.34 ± 0.01	1.36 ± 0.01	1.42 ± 0.01	1.44 ± 0.02	1.49 ± 0.02
0.0434	0.998 ± 0.009	1.19 ± 0.01	1.20 ± 0.01	1.25 ± 0.01	1.27 ± 0.01	1.27 ± 0.01	1.30 ± 0.01	1.37 ± 0.01	1.41 ± 0.02	1.44 ± 0.02
0.0462	0.969 ± 0.008	1.12 ± 0.01	1.15 ± 0.01	1.18 ± 0.01	1.19 ± 0.01	1.20 ± 0.01	1.22 ± 0.01	1.27 ± 0.01	1.29 ± 0.01	1.32 ± 0.02
0.0491	0.949 ± 0.008	1.07 ± 0.01	1.10 ± 0.01	1.12 ± 0.01	1.12 ± 0.01	1.15 ± 0.01	1.17 ± 0.01	1.22 ± 0.01	1.20 ± 0.01	1.25 ± 0.01
0.0519	0.907 ± 0.007	1.01 ± 0.01	1.02 ± 0.01	1.06 ± 0.01	1.05 ± 0.01	1.08 ± 0.01	1.09 ± 0.01	1.13 ± 0.01	1.12 ± 0.01	1.18 ± 0.01
0.0548	0.871 ± 0.007	0.955 ± 0.008	0.976 ± 0.009	0.989 ± 0.009	0.978 ± 0.009	1.01 ± 0.01	1.03 ± 0.01	1.06 ± 0.01	1.05 ± 0.01	1.08 ± 0.01
0.0576	0.830 ± 0.007	0.904 ± 0.008	0.922 ± 0.008	0.922 ± 0.008	0.925 ± 0.009	0.941 ± 0.009	0.950 ± 0.010	0.963 ± 0.011	0.972 ± 0.011	0.994 ± 0.012
0.0605	0.793 ± 0.006	0.840 ± 0.007	0.843 ± 0.008	0.859 ± 0.008	0.853 ± 0.008	0.864 ± 0.009	0.870 ± 0.009	0.902 ± 0.010	0.888 ± 0.010	0.916 ± 0.012
0.0647	0.719 ± 0.004	0.750 ± 0.005	0.754 ± 0.005	0.759 ± 0.005	0.760 ± 0.005	0.764 ± 0.006	0.763 ± 0.006	0.775 ± 0.006	0.784 ± 0.007	0.783 ± 0.007
0.0704	0.612 ± 0.004	0.624 ± 0.004	0.618 ± 0.004	0.625 ± 0.005	0.629 ± 0.005	0.637 ± 0.005	0.639 ± 0.005	0.637 ± 0.006	0.629 ± 0.006	0.644 ± 0.006

$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 79.88$	$c_p = 75.85$	$c_p = 69.96$	$c_p = 64.90$	$c_p = 59.46$	$c_p = 55.45$	$c_p = 48.53$	$c_p = 43.91$	$c_p = 39.63$
0.0761	0.515 ± 0.003	0.513 ± 0.004	0.511 ± 0.004	0.520 ± 0.004	0.518 ± 0.004	0.516 ± 0.004	0.518 ± 0.005	0.508 ± 0.005	0.518 ± 0.005	0.517 ± 0.006
0.0818	0.419 ± 0.003	0.421 ± 0.003	0.412 ± 0.003	0.414 ± 0.003	0.418 ± 0.004	0.413 ± 0.004	0.413 ± 0.004	0.412 ± 0.004	0.417 ± 0.005	0.419 ± 0.005
0.0875	0.330 ± 0.003	0.327 ± 0.003	0.330 ± 0.003	0.322 ± 0.003	0.326 ± 0.003	0.323 ± 0.003	0.323 ± 0.004	0.327 ± 0.004	0.323 ± 0.004	0.319 ± 0.004
0.0932	0.259 ± 0.002	0.250 ± 0.002	0.254 ± 0.003	0.253 ± 0.003	0.250 ± 0.003	0.254 ± 0.003	0.249 ± 0.003	0.254 ± 0.003	0.243 ± 0.003	0.249 ± 0.004
0.0989	0.202 ± 0.002	0.196 ± 0.002	0.200 ± 0.002	0.197 ± 0.002	0.195 ± 0.002	0.198 ± 0.003	0.196 ± 0.003	0.196 ± 0.003	0.198 ± 0.003	0.197 ± 0.003
0.1074	0.133 ± 0.001	0.132 ± 0.001	0.131 ± 0.001	0.131 ± 0.001	0.132 ± 0.001	0.133 ± 0.002	0.131 ± 0.002	0.134 ± 0.002	0.134 ± 0.002	0.131 ± 0.002
0.1188	0.0807 ± 0.0009	0.0815 ± 0.0010	0.0811 ± 0.0011	0.0820 ± 0.0011	0.0812 ± 0.0011	0.0800 ± 0.0012	0.0837 ± 0.0013	0.0814 ± 0.0014	0.0813 ± 0.0015	0.0840 ± 0.0018
0.1302	0.0531 ± 0.0007	0.0561 ± 0.0009	0.0562 ± 0.0009	0.0555 ± 0.0009	0.0556 ± 0.0010	0.0559 ± 0.0011	0.0580 ± 0.0012	0.0583 ± 0.0013	0.0572 ± 0.0014	0.0564 ± 0.0015
0.1415	0.0399 ± 0.0006	0.0406 ± 0.0007	0.0421 ± 0.0008	0.0409 ± 0.0008	0.0410 ± 0.0009	0.0418 ± 0.0009	0.0413 ± 0.0010	0.0430 ± 0.0011	0.0425 ± 0.0012	0.0430 ± 0.0014
0.1529	0.0323 ± 0.0006	0.0342 ± 0.0007	0.0351 ± 0.0007	0.0341 ± 0.0008	0.0336 ± 0.0008	0.0353 ± 0.0009	0.0345 ± 0.0009	0.0339 ± 0.0010	0.0333 ± 0.0011	0.0357 ± 0.0013
0.1643	0.0271 ± 0.0005	0.0285 ± 0.0006	0.0287 ± 0.0007	0.0292 ± 0.0007	0.0299 ± 0.0007	0.0281 ± 0.0008	0.0285 ± 0.0009	0.0277 ± 0.0010	0.0291 ± 0.0010	0.0286 ± 0.0012
0.1757	0.0218 ± 0.0005	0.0223 ± 0.0006	0.0218 ± 0.0006	0.0235 ± 0.0007	0.0227 ± 0.0007	0.0231 ± 0.0007	0.0222 ± 0.0008	0.0226 ± 0.0009	0.0228 ± 0.0010	0.0204 ± 0.0011
0.1870	0.0174 ± 0.0004	0.0184 ± 0.0005	0.0181 ± 0.0006	0.0192 ± 0.0006	0.0194 ± 0.0006	0.0186 ± 0.0007	0.0177 ± 0.0008	0.0177 ± 0.0008	0.0167 ± 0.0009	0.0161 ± 0.0010
0.1984	0.0155 ± 0.0004	0.0167 ± 0.0005	0.0171 ± 0.0005	0.0164 ± 0.0006	0.0161 ± 0.0006	0.0159 ± 0.0007	0.0174 ± 0.0007	0.0164 ± 0.0008	0.0164 ± 0.0009	0.0175 ± 0.0010
0.2098	0.0134 ± 0.0004	0.0142 ± 0.0005	0.0128 ± 0.0005	0.0136 ± 0.0006	0.0139 ± 0.0006	0.0135 ± 0.0006	0.0129 ± 0.0007	0.0133 ± 0.0008	0.0141 ± 0.0008	0.0119 ± 0.0009
0.2212	0.0116 ± 0.0004	0.0113 ± 0.0005	0.0110 ± 0.0005	0.0113 ± 0.0005	0.0110 ± 0.0006	0.0112 ± 0.0006	0.0116 ± 0.0007	0.00909 ± 0.00073	0.0119 ± 0.0008	0.0118 ± 0.0009
0.2325	0.00957 ± 0.00036	0.0101 ± 0.0005	0.00959 ± 0.00048	0.0107 ± 0.0005	0.0100 ± 0.0005	0.0109 ± 0.0006	0.00941 ± 0.00064	0.0109 ± 0.0007	0.00905 ± 0.00078	0.0100 ± 0.0009
0.2439	0.00890 ± 0.00035	0.00786 ± 0.00044	0.00803 ± 0.00047	0.00855 ± 0.00050	0.00870 ± 0.00053	0.00857 ± 0.00058	0.00801 ± 0.00063	0.00851 ± 0.00070	0.00869 ± 0.00078	0.00886 ± 0.00089
0.2553	0.00873 ± 0.00035	0.00831 ± 0.00044	0.00785 ± 0.00046	0.00837 ± 0.00050	0.00766 ± 0.00052	0.00791 ± 0.00057	0.00705 ± 0.00061	0.00884 ± 0.00071	0.00833 ± 0.00077	0.00716 ± 0.00087
0.2667	0.00803 ± 0.00034	0.00756 ± 0.00043	0.00686 ± 0.00046	0.00749 ± 0.00049	0.00696 ± 0.00051	0.00818 ± 0.00056	0.00739 ± 0.00061	0.00845 ± 0.00069	0.00719 ± 0.00076	0.00689 ± 0.00086
0.2780	0.00835 ± 0.00034	0.00856 ± 0.00043	0.00750 ± 0.00045	0.00676 ± 0.00048	0.00772 ± 0.00051	0.00757 ± 0.00056	0.00757 ± 0.00060	0.00741 ± 0.00068	0.00861 ± 0.00075	0.00779 ± 0.00085
0.2894	0.00872 ± 0.00034	0.00888 ± 0.00042	0.00923 ± 0.00045	0.00878 ± 0.00048	0.00841 ± 0.00050	0.00863 ± 0.00055	0.00866 ± 0.00060	0.00911 ± 0.00068	0.00904 ± 0.00074	0.00926 ± 0.00086
0.3008	0.00918 ± 0.00034	0.00871 ± 0.00042	0.00875 ± 0.00045	0.00860 ± 0.00048	0.00825 ± 0.00050	0.00823 ± 0.00054	0.00817 ± 0.00059	0.00880 ± 0.00067	0.00854 ± 0.00073	0.00919 ± 0.00084
0.3122	0.00982 ± 0.00034	0.00893 ± 0.00041	0.00957 ± 0.00044	0.00906 ± 0.00047	0.00943 ± 0.00049	0.00926 ± 0.00054	0.0101 ± 0.0006	0.00882 ± 0.00066	0.00837 ± 0.00072	0.0108 ± 0.0008
0.3235	0.00895 ± 0.00033	0.00852 ± 0.00040	0.00934 ± 0.00043	0.00873 ± 0.00046	0.00980 ± 0.00048	0.00915 ± 0.00053	0.00992 ± 0.00057	0.00949 ± 0.00065	0.00869 ± 0.00071	0.00916 ± 0.00081
0.3349	0.00867 ± 0.00032	0.00765 ± 0.00039	0.00848 ± 0.00042	0.00754 ± 0.00044	0.00854 ± 0.00047	0.00790 ± 0.00052	0.00841 ± 0.00056	0.00775 ± 0.00063	0.00803 ± 0.00069	0.00772 ± 0.00079
0.3448	0.00803 ± 0.00036	0.00806 ± 0.00044	0.00849 ± 0.00048	0.00818 ± 0.00051	0.00713 ± 0.00053	0.00703 ± 0.00058	0.00802 ± 0.00064	0.00727 ± 0.00071	0.00847 ± 0.00079	0.00787 ± 0.00090

Table 14: Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector,  $q$ , for BSA in 1.0 mol·L<sup>-1</sup> NaCl aqueous solutions at pH = 6.3, at several protein concentrations,  $c_p$ , (mg·mL<sup>-1</sup>).

$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 80.79$	$c_p = 77.29$	$c_p = 70.35$	$c_p = 65.40$	$c_p = 60.59$	$c_p = 56.05$	$c_p = 50.01$	$c_p = 44.39$	$c_p = 39.53$
0.0142	0.945 ± 0.036	1.57 ± 0.05	1.62 ± 0.06	1.71 ± 0.06	1.82 ± 0.07	2.00 ± 0.07	2.06 ± 0.08	2.01 ± 0.08	2.21 ± 0.09	2.22 ± 0.10
0.0156	1.01 ± 0.02	1.58 ± 0.03	1.62 ± 0.03	1.72 ± 0.03	1.82 ± 0.04	1.94 ± 0.04	2.06 ± 0.04	2.02 ± 0.04	2.18 ± 0.05	2.27 ± 0.06
0.0171	0.988 ± 0.021	1.59 ± 0.03	1.58 ± 0.03	1.69 ± 0.03	1.78 ± 0.03	1.92 ± 0.03	1.97 ± 0.04	1.99 ± 0.04	2.11 ± 0.04	2.21 ± 0.05
0.0185	1.03 ± 0.02	1.51 ± 0.02	1.63 ± 0.03	1.65 ± 0.03	1.82 ± 0.03	1.87 ± 0.03	1.95 ± 0.03	2.00 ± 0.04	2.12 ± 0.04	2.20 ± 0.04
0.0199	1.07 ± 0.02	1.57 ± 0.02	1.66 ± 0.02	1.69 ± 0.03	1.77 ± 0.03	1.86 ± 0.03	1.95 ± 0.03	2.04 ± 0.03	2.12 ± 0.04	2.23 ± 0.04
0.0213	1.09 ± 0.02	1.59 ± 0.02	1.63 ± 0.02	1.74 ± 0.02	1.77 ± 0.03	1.84 ± 0.03	1.94 ± 0.03	1.98 ± 0.03	2.10 ± 0.04	2.17 ± 0.04
0.0228	1.12 ± 0.02	1.54 ± 0.02	1.58 ± 0.02	1.66 ± 0.02	1.74 ± 0.02	1.78 ± 0.03	1.87 ± 0.03	1.97 ± 0.03	1.97 ± 0.03	2.12 ± 0.04
0.0242	1.12 ± 0.02	1.54 ± 0.02	1.59 ± 0.02	1.64 ± 0.02	1.72 ± 0.02	1.78 ± 0.03	1.83 ± 0.03	1.90 ± 0.03	2.02 ± 0.03	2.07 ± 0.04
0.0256	1.10 ± 0.02	1.51 ± 0.02	1.57 ± 0.02	1.63 ± 0.02	1.68 ± 0.02	1.71 ± 0.02	1.81 ± 0.03	1.87 ± 0.03	1.94 ± 0.03	1.99 ± 0.03
0.0270	1.13 ± 0.02	1.52 ± 0.02	1.53 ± 0.02	1.60 ± 0.02	1.67 ± 0.02	1.69 ± 0.02	1.82 ± 0.03	1.88 ± 0.03	1.92 ± 0.03	2.00 ± 0.03
0.0284	1.12 ± 0.01	1.49 ± 0.02	1.49 ± 0.02	1.54 ± 0.02	1.64 ± 0.02	1.66 ± 0.02	1.74 ± 0.02	1.85 ± 0.03	1.87 ± 0.03	1.93 ± 0.03
0.0299	1.12 ± 0.01	1.46 ± 0.02	1.46 ± 0.02	1.55 ± 0.02	1.63 ± 0.02	1.68 ± 0.02	1.73 ± 0.02	1.81 ± 0.03	1.83 ± 0.03	1.88 ± 0.03
0.0320	1.11 ± 0.01	1.45 ± 0.01	1.47 ± 0.01	1.53 ± 0.01	1.59 ± 0.01	1.63 ± 0.01	1.71 ± 0.02	1.71 ± 0.02	1.80 ± 0.02	1.82 ± 0.02
0.0349	1.10 ± 0.01	1.41 ± 0.01	1.41 ± 0.01	1.46 ± 0.01	1.51 ± 0.01	1.57 ± 0.01	1.62 ± 0.01	1.63 ± 0.02	1.74 ± 0.02	1.74 ± 0.02
0.0377	1.08 ± 0.01	1.33 ± 0.01	1.34 ± 0.01	1.42 ± 0.01	1.45 ± 0.01	1.49 ± 0.01	1.55 ± 0.01	1.55 ± 0.01	1.62 ± 0.02	1.65 ± 0.02
0.0405	1.04 ± 0.01	1.28 ± 0.01	1.28 ± 0.01	1.33 ± 0.01	1.39 ± 0.01	1.39 ± 0.01	1.44 ± 0.01	1.46 ± 0.01	1.51 ± 0.01	1.56 ± 0.02
0.0434	1.04 ± 0.01	1.23 ± 0.01	1.24 ± 0.01	1.27 ± 0.01	1.32 ± 0.01	1.34 ± 0.01	1.39 ± 0.01	1.38 ± 0.01	1.45 ± 0.01	1.47 ± 0.02
0.0462	1.00 ± 0.01	1.18 ± 0.01	1.19 ± 0.01	1.20 ± 0.01	1.24 ± 0.01	1.27 ± 0.01	1.27 ± 0.01	1.28 ± 0.01	1.32 ± 0.01	1.33 ± 0.01
0.0491	0.958 ± 0.007	1.11 ± 0.01	1.13 ± 0.01	1.15 ± 0.01	1.18 ± 0.01	1.17 ± 0.01	1.20 ± 0.01	1.22 ± 0.01	1.25 ± 0.01	1.26 ± 0.01
0.0519	0.934 ± 0.007	1.05 ± 0.01	1.04 ± 0.01	1.07 ± 0.01	1.10 ± 0.01	1.12 ± 0.01	1.12 ± 0.01	1.12 ± 0.01	1.17 ± 0.01	1.14 ± 0.01
0.0548	0.887 ± 0.006	0.989 ± 0.008	0.990 ± 0.008	1.01 ± 0.01	1.03 ± 0.01	1.04 ± 0.01	1.03 ± 0.01	1.05 ± 0.01	1.06 ± 0.01	1.06 ± 0.01
0.0576	0.848 ± 0.006	0.923 ± 0.007	0.928 ± 0.007	0.932 ± 0.008	0.948 ± 0.008	0.972 ± 0.009	0.977 ± 0.009	0.979 ± 0.010	0.985 ± 0.010	0.999 ± 0.011
0.0605	0.797 ± 0.006	0.862 ± 0.007	0.856 ± 0.007	0.876 ± 0.007	0.887 ± 0.008	0.885 ± 0.008	0.904 ± 0.008	0.896 ± 0.009	0.912 ± 0.009	0.930 ± 0.011
0.0647	0.717 ± 0.004	0.760 ± 0.004	0.760 ± 0.004	0.766 ± 0.005	0.772 ± 0.005	0.773 ± 0.005	0.782 ± 0.005	0.780 ± 0.006	0.803 ± 0.006	0.795 ± 0.007
0.0704	0.618 ± 0.003	0.639 ± 0.004	0.638 ± 0.004	0.639 ± 0.004	0.648 ± 0.004	0.643 ± 0.005	0.644 ± 0.005	0.650 ± 0.005	0.653 ± 0.005	0.655 ± 0.006

$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 80.79$	$c_p = 77.29$	$c_p = 70.35$	$c_p = 65.40$	$c_p = 60.59$	$c_p = 56.05$	$c_p = 50.01$	$c_p = 44.39$	$c_p = 39.53$
0.0761	0.513 ± 0.003	0.524 ± 0.003	0.527 ± 0.003	0.523 ± 0.004	0.528 ± 0.004	0.525 ± 0.004	0.526 ± 0.004	0.521 ± 0.004	0.519 ± 0.005	0.515 ± 0.005
0.0818	0.415 ± 0.003	0.420 ± 0.003	0.418 ± 0.003	0.423 ± 0.003	0.419 ± 0.003	0.421 ± 0.003	0.421 ± 0.004	0.417 ± 0.004	0.413 ± 0.004	0.417 ± 0.004
0.0875	0.335 ± 0.002	0.332 ± 0.003	0.331 ± 0.003	0.328 ± 0.003	0.332 ± 0.003	0.333 ± 0.003	0.326 ± 0.003	0.325 ± 0.003	0.323 ± 0.004	0.328 ± 0.004
0.0932	0.261 ± 0.002	0.257 ± 0.002	0.259 ± 0.002	0.256 ± 0.002	0.254 ± 0.002	0.255 ± 0.003	0.257 ± 0.003	0.255 ± 0.003	0.260 ± 0.003	0.256 ± 0.003
0.0989	0.204 ± 0.002	0.200 ± 0.002	0.197 ± 0.002	0.201 ± 0.002	0.198 ± 0.002	0.197 ± 0.002	0.198 ± 0.002	0.199 ± 0.003	0.200 ± 0.003	0.196 ± 0.003
0.1074	0.134 ± 0.001	0.131 ± 0.001	0.132 ± 0.001	0.132 ± 0.001	0.130 ± 0.001	0.131 ± 0.001	0.129 ± 0.001	0.130 ± 0.001	0.132 ± 0.002	0.130 ± 0.002
0.1188	0.0832 ± 0.0008	0.0812 ± 0.0009	0.0820 ± 0.0009	0.0815 ± 0.0009	0.0837 ± 0.0010	0.0818 ± 0.0011	0.0825 ± 0.0011	0.0842 ± 0.0012	0.0822 ± 0.0013	0.0836 ± 0.0015
0.1302	0.0550 ± 0.0006	0.0555 ± 0.0007	0.0560 ± 0.0008	0.0564 ± 0.0008	0.0551 ± 0.0009	0.0563 ± 0.0009	0.0572 ± 0.0010	0.0587 ± 0.0011	0.0586 ± 0.0012	0.0580 ± 0.0013
0.1415	0.0418 ± 0.0006	0.0427 ± 0.0006	0.0431 ± 0.0007	0.0426 ± 0.0007	0.0427 ± 0.0008	0.0431 ± 0.0008	0.0434 ± 0.0009	0.0431 ± 0.0009	0.0433 ± 0.0010	0.0429 ± 0.0012
0.1529	0.0347 ± 0.0005	0.0347 ± 0.0006	0.0340 ± 0.0006	0.0341 ± 0.0006	0.0356 ± 0.0007	0.0347 ± 0.0007	0.0355 ± 0.0008	0.0354 ± 0.0009	0.0355 ± 0.0009	0.0368 ± 0.0011
0.1643	0.0276 ± 0.0005	0.0282 ± 0.0005	0.0275 ± 0.0005	0.0274 ± 0.0006	0.0282 ± 0.0006	0.0279 ± 0.0007	0.0290 ± 0.0007	0.0280 ± 0.0008	0.0295 ± 0.0009	0.0293 ± 0.0010
0.1757	0.0220 ± 0.0004	0.0226 ± 0.0005	0.0225 ± 0.0005	0.0226 ± 0.0005	0.0230 ± 0.0006	0.0229 ± 0.0006	0.0225 ± 0.0007	0.0240 ± 0.0007	0.0226 ± 0.0008	0.0232 ± 0.0009
0.1870	0.0192 ± 0.0004	0.0197 ± 0.0005	0.0186 ± 0.0005	0.0191 ± 0.0005	0.0191 ± 0.0005	0.0183 ± 0.0006	0.0181 ± 0.0006	0.0197 ± 0.0007	0.0188 ± 0.0007	0.0201 ± 0.0009
0.1984	0.0151 ± 0.0004	0.0156 ± 0.0004	0.0165 ± 0.0004	0.0165 ± 0.0005	0.0158 ± 0.0005	0.0164 ± 0.0005	0.0157 ± 0.0006	0.0158 ± 0.0006	0.0168 ± 0.0007	0.0175 ± 0.0008
0.2098	0.0127 ± 0.0003	0.0128 ± 0.0004	0.0133 ± 0.0004	0.0136 ± 0.0004	0.0134 ± 0.0005	0.0127 ± 0.0005	0.0129 ± 0.0006	0.0141 ± 0.0006	0.0145 ± 0.0007	0.0150 ± 0.0008
0.2212	0.0116 ± 0.0003	0.0113 ± 0.0004	0.0115 ± 0.0004	0.0106 ± 0.0004	0.0112 ± 0.0005	0.0111 ± 0.0005	0.0110 ± 0.0005	0.0110 ± 0.0006	0.0115 ± 0.0007	0.0114 ± 0.0008
0.2325	0.00956 ± 0.00031	0.00974 ± 0.00037	0.00930 ± 0.00038	0.01000 ± 0.00042	0.00992 ± 0.00045	0.00923 ± 0.00049	0.0105 ± 0.0005	0.0103 ± 0.0006	0.0104 ± 0.0006	0.00979 ± 0.00074
0.2439	0.00860 ± 0.00030	0.00813 ± 0.00036	0.00772 ± 0.00037	0.00877 ± 0.00041	0.00816 ± 0.00044	0.00908 ± 0.00048	0.00824 ± 0.00051	0.00828 ± 0.00057	0.00818 ± 0.00063	0.00788 ± 0.00072
0.2553	0.00788 ± 0.00030	0.00753 ± 0.00036	0.00709 ± 0.00037	0.00689 ± 0.00040	0.00730 ± 0.00043	0.00725 ± 0.00047	0.00705 ± 0.00050	0.00749 ± 0.00056	0.00837 ± 0.00063	0.00822 ± 0.00072
0.2667	0.00736 ± 0.00030	0.00742 ± 0.00035	0.00757 ± 0.00037	0.00740 ± 0.00040	0.00746 ± 0.00043	0.00705 ± 0.00047	0.00762 ± 0.00050	0.00789 ± 0.00056	0.00713 ± 0.00062	0.00749 ± 0.00071
0.2780	0.00732 ± 0.00029	0.00721 ± 0.00035	0.00760 ± 0.00036	0.00817 ± 0.00040	0.00749 ± 0.00043	0.00777 ± 0.00046	0.00711 ± 0.00050	0.00751 ± 0.00055	0.00822 ± 0.00062	0.00641 ± 0.00070
0.2894	0.00787 ± 0.00029	0.00835 ± 0.00035	0.00861 ± 0.00036	0.00856 ± 0.00040	0.00756 ± 0.00042	0.00845 ± 0.00046	0.00796 ± 0.00050	0.00719 ± 0.00054	0.00804 ± 0.00061	0.00849 ± 0.00071
0.3008	0.00846 ± 0.00029	0.00879 ± 0.00035	0.00904 ± 0.00037	0.00864 ± 0.00040	0.00869 ± 0.00043	0.00804 ± 0.00046	0.00854 ± 0.00050	0.00830 ± 0.00055	0.00645 ± 0.00061	0.00723 ± 0.00070
0.3122	0.00890 ± 0.00029	0.00835 ± 0.00034	0.00843 ± 0.00036	0.00800 ± 0.00039	0.00800 ± 0.00042	0.00829 ± 0.00046	0.00923 ± 0.00049	0.00773 ± 0.00054	0.00787 ± 0.00060	0.00794 ± 0.00069
0.3235	0.00839 ± 0.00028	0.00851 ± 0.00034	0.00925 ± 0.00036	0.00833 ± 0.00038	0.00875 ± 0.00042	0.00905 ± 0.00045	0.00918 ± 0.00049	0.00873 ± 0.00053	0.00870 ± 0.00059	0.00781 ± 0.00069
0.3349	0.00834 ± 0.00028	0.00861 ± 0.00033	0.00781 ± 0.00035	0.00775 ± 0.00037	0.00867 ± 0.00041	0.00770 ± 0.00044	0.00835 ± 0.00048	0.00785 ± 0.00053	0.00654 ± 0.00058	0.00669 ± 0.00067
0.3448	0.00801 ± 0.00032	0.00716 ± 0.00038	0.00698 ± 0.00039	0.00758 ± 0.00043	0.00708 ± 0.00046	0.00829 ± 0.00051	0.00588 ± 0.00054	0.00591 ± 0.00060	0.00674 ± 0.00067	0.00681 ± 0.00076

Table 15: Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector,  $q$ , for BSA in 1.0 mol·L<sup>-1</sup> Na<sub>2</sub>SO<sub>4</sub> aqueous solutions at pH = 6.3, at several protein concentrations,  $c_p$ , (mg·mL<sup>-1</sup>).

$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 74.88$	$c_p = 67.87$	$c_p = 65.99$	$c_p = 59.89$	$c_p = 56.70$	$c_p = 49.33$	$c_p = 44.88$	$c_p = 40.65$	$c_p = 31.07$
0.0142	0.388 ± 0.026	0.503 ± 0.042	0.519 ± 0.045	0.518 ± 0.046	0.569 ± 0.051	0.592 ± 0.052	0.605 ± 0.059	0.676 ± 0.066	0.724 ± 0.074	0.735 ± 0.038
0.0156	0.332 ± 0.018	0.491 ± 0.025	0.470 ± 0.027	0.507 ± 0.028	0.531 ± 0.031	0.563 ± 0.032	0.638 ± 0.036	0.638 ± 0.039	0.659 ± 0.044	0.704 ± 0.021
0.0171	0.343 ± 0.016	0.502 ± 0.021	0.501 ± 0.023	0.516 ± 0.023	0.563 ± 0.025	0.591 ± 0.027	0.647 ± 0.030	0.619 ± 0.032	0.635 ± 0.036	0.670 ± 0.017
0.0185	0.344 ± 0.014	0.476 ± 0.018	0.493 ± 0.020	0.516 ± 0.021	0.562 ± 0.023	0.533 ± 0.023	0.591 ± 0.026	0.555 ± 0.028	0.632 ± 0.031	0.646 ± 0.015
0.0199	0.352 ± 0.013	0.506 ± 0.017	0.488 ± 0.018	0.529 ± 0.019	0.566 ± 0.021	0.555 ± 0.021	0.539 ± 0.023	0.564 ± 0.026	0.611 ± 0.029	0.651 ± 0.014
0.0213	0.354 ± 0.012	0.484 ± 0.016	0.505 ± 0.017	0.504 ± 0.018	0.544 ± 0.019	0.564 ± 0.020	0.593 ± 0.022	0.584 ± 0.024	0.598 ± 0.026	0.622 ± 0.012
0.0228	0.340 ± 0.012	0.469 ± 0.014	0.488 ± 0.016	0.516 ± 0.017	0.520 ± 0.018	0.549 ± 0.018	0.553 ± 0.021	0.588 ± 0.022	0.583 ± 0.024	0.600 ± 0.011
0.0242	0.341 ± 0.011	0.477 ± 0.014	0.467 ± 0.015	0.478 ± 0.015	0.518 ± 0.016	0.524 ± 0.017	0.538 ± 0.019	0.561 ± 0.021	0.562 ± 0.022	0.593 ± 0.010
0.0256	0.343 ± 0.010	0.448 ± 0.013	0.488 ± 0.014	0.475 ± 0.014	0.507 ± 0.016	0.534 ± 0.017	0.543 ± 0.018	0.543 ± 0.019	0.555 ± 0.021	0.588 ± 0.010
0.0270	0.356 ± 0.010	0.465 ± 0.013	0.498 ± 0.014	0.481 ± 0.014	0.522 ± 0.015	0.505 ± 0.016	0.519 ± 0.017	0.537 ± 0.019	0.545 ± 0.021	0.566 ± 0.009
0.0284	0.335 ± 0.010	0.458 ± 0.012	0.473 ± 0.013	0.454 ± 0.013	0.484 ± 0.015	0.477 ± 0.015	0.510 ± 0.017	0.506 ± 0.018	0.538 ± 0.020	0.547 ± 0.009
0.0299	0.337 ± 0.009	0.444 ± 0.012	0.454 ± 0.013	0.463 ± 0.013	0.467 ± 0.014	0.472 ± 0.014	0.504 ± 0.016	0.489 ± 0.017	0.514 ± 0.019	0.545 ± 0.008
0.0320	0.337 ± 0.006	0.437 ± 0.008	0.428 ± 0.009	0.444 ± 0.009	0.456 ± 0.009	0.464 ± 0.010	0.486 ± 0.011	0.479 ± 0.011	0.476 ± 0.013	0.508 ± 0.006
0.0349	0.331 ± 0.006	0.418 ± 0.008	0.431 ± 0.008	0.424 ± 0.008	0.453 ± 0.009	0.446 ± 0.009	0.471 ± 0.010	0.471 ± 0.011	0.486 ± 0.012	0.504 ± 0.005
0.0377	0.315 ± 0.006	0.401 ± 0.007	0.419 ± 0.008	0.408 ± 0.008	0.430 ± 0.008	0.428 ± 0.009	0.455 ± 0.010	0.453 ± 0.010	0.475 ± 0.011	0.471 ± 0.005
0.0405	0.315 ± 0.005	0.365 ± 0.007	0.393 ± 0.007	0.407 ± 0.007	0.402 ± 0.008	0.397 ± 0.008	0.421 ± 0.009	0.422 ± 0.010	0.424 ± 0.011	0.440 ± 0.005
0.0434	0.300 ± 0.005	0.360 ± 0.006	0.384 ± 0.007	0.390 ± 0.007	0.394 ± 0.008	0.399 ± 0.008	0.398 ± 0.009	0.404 ± 0.009	0.395 ± 0.010	0.414 ± 0.005
0.0462	0.286 ± 0.005	0.336 ± 0.006	0.353 ± 0.007	0.348 ± 0.007	0.351 ± 0.007	0.364 ± 0.007	0.372 ± 0.008	0.369 ± 0.009	0.376 ± 0.010	0.378 ± 0.004
0.0491	0.284 ± 0.005	0.328 ± 0.006	0.346 ± 0.006	0.336 ± 0.006	0.333 ± 0.007	0.342 ± 0.007	0.367 ± 0.008	0.331 ± 0.008	0.344 ± 0.009	0.361 ± 0.004
0.0519	0.277 ± 0.004	0.317 ± 0.006	0.331 ± 0.006	0.321 ± 0.006	0.347 ± 0.007	0.330 ± 0.007	0.356 ± 0.008	0.341 ± 0.008	0.339 ± 0.009	0.356 ± 0.004
0.0548	0.259 ± 0.004	0.298 ± 0.005	0.309 ± 0.006	0.304 ± 0.006	0.312 ± 0.006	0.300 ± 0.006	0.311 ± 0.007	0.308 ± 0.007	0.302 ± 0.008	0.316 ± 0.004
0.0576	0.251 ± 0.004	0.281 ± 0.005	0.285 ± 0.005	0.285 ± 0.005	0.288 ± 0.006	0.285 ± 0.006	0.299 ± 0.007	0.280 ± 0.007	0.291 ± 0.008	0.296 ± 0.004
0.0605	0.233 ± 0.004	0.259 ± 0.005	0.271 ± 0.005	0.271 ± 0.005	0.258 ± 0.006	0.263 ± 0.006	0.284 ± 0.006	0.265 ± 0.007	0.263 ± 0.007	0.279 ± 0.004
0.0647	0.219 ± 0.003	0.239 ± 0.003	0.240 ± 0.003	0.237 ± 0.003	0.238 ± 0.004	0.243 ± 0.004	0.241 ± 0.004	0.241 ± 0.005	0.238 ± 0.005	0.241 ± 0.002
0.0704	0.193 ± 0.002	0.202 ± 0.003	0.203 ± 0.003	0.196 ± 0.003	0.207 ± 0.003	0.197 ± 0.003	0.213 ± 0.004	0.200 ± 0.004	0.203 ± 0.005	0.203 ± 0.002



$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 74.88$	$c_p = 67.87$	$c_p = 65.99$	$c_p = 59.89$	$c_p = 56.70$	$c_p = 49.33$	$c_p = 44.88$	$c_p = 40.65$	$c_p = 31.07$
0.0761	0.171 ± 0.002	0.171 ± 0.003	0.170 ± 0.003	0.173 ± 0.003	0.172 ± 0.003	0.173 ± 0.003	0.173 ± 0.004	0.176 ± 0.004	0.175 ± 0.004	0.171 ± 0.002
0.0818	0.141 ± 0.002	0.145 ± 0.002	0.144 ± 0.003	0.139 ± 0.003	0.144 ± 0.003	0.139 ± 0.003	0.140 ± 0.003	0.139 ± 0.004	0.140 ± 0.004	0.139 ± 0.002
0.0875	0.122 ± 0.002	0.117 ± 0.002	0.118 ± 0.002	0.118 ± 0.002	0.121 ± 0.003	0.117 ± 0.003	0.113 ± 0.003	0.115 ± 0.003	0.111 ± 0.004	0.116 ± 0.002
0.0932	0.0994 ± 0.0017	0.0969 ± 0.0021	0.0941 ± 0.0022	0.0949 ± 0.0023	0.0962 ± 0.0025	0.0964 ± 0.0026	0.0967 ± 0.0029	0.0937 ± 0.0031	0.0914 ± 0.0035	0.0941 ± 0.0016
0.0989	0.0770 ± 0.0016	0.0771 ± 0.0019	0.0789 ± 0.0021	0.0769 ± 0.0021	0.0745 ± 0.0023	0.0790 ± 0.0024	0.0741 ± 0.0027	0.0769 ± 0.0029	0.0778 ± 0.0033	0.0776 ± 0.0016
0.1074	0.0553 ± 0.0010	0.0549 ± 0.0012	0.0534 ± 0.0013	0.0537 ± 0.0014	0.0525 ± 0.0015	0.0511 ± 0.0016	0.0511 ± 0.0018	0.0547 ± 0.0019	0.0532 ± 0.0021	0.0535 ± 0.0010
0.1188	0.0326 ± 0.0009	0.0330 ± 0.0011	0.0346 ± 0.0012	0.0350 ± 0.0012	0.0348 ± 0.0013	0.0334 ± 0.0014	0.0362 ± 0.0016	0.0340 ± 0.0017	0.0368 ± 0.0020	0.0344 ± 0.0010
0.1302	0.0214 ± 0.0008	0.0238 ± 0.0010	0.0249 ± 0.0011	0.0248 ± 0.0011	0.0220 ± 0.0012	0.0267 ± 0.0013	0.0256 ± 0.0015	0.0247 ± 0.0016	0.0268 ± 0.0018	0.0264 ± 0.0009
0.1415	0.0159 ± 0.0007	0.0145 ± 0.0009	0.0156 ± 0.0010	0.0161 ± 0.0010	0.0148 ± 0.0011	0.0169 ± 0.0012	0.0165 ± 0.0014	0.0145 ± 0.0014	0.0164 ± 0.0017	0.0168 ± 0.0008
0.1529	0.0130 ± 0.0006	0.0140 ± 0.0008	0.0142 ± 0.0009	0.0144 ± 0.0010	0.0128 ± 0.0010	0.0136 ± 0.0011	0.0176 ± 0.0013	0.0141 ± 0.0014	0.0144 ± 0.0016	0.0156 ± 0.0008
0.1643	0.0114 ± 0.0006	0.0112 ± 0.0008	0.0112 ± 0.0009	0.0126 ± 0.0009	0.0112 ± 0.0010	0.0113 ± 0.0010	0.0120 ± 0.0012	0.0115 ± 0.0013	0.0126 ± 0.0015	0.0129 ± 0.0007
0.1757	0.0101 ± 0.0006	0.00978 ± 0.00075	0.00794 ± 0.00081	0.00960 ± 0.00084	0.0105 ± 0.0009	0.0111 ± 0.0010	0.0104 ± 0.0011	0.00883 ± 0.00120	0.00915 ± 0.00137	0.0108 ± 0.0007
0.1870	0.00700 ± 0.00054	0.00825 ± 0.00071	0.00982 ± 0.00078	0.00974 ± 0.00080	0.00993 ± 0.00089	0.00813 ± 0.00093	0.0106 ± 0.0011	0.00927 ± 0.00115	0.00964 ± 0.00131	0.00966 ± 0.00065
0.1984	0.00780 ± 0.00051	0.00695 ± 0.00068	0.00895 ± 0.00075	0.00871 ± 0.00077	0.00851 ± 0.00085	0.00689 ± 0.00088	0.00818 ± 0.00102	0.00890 ± 0.00111	0.00878 ± 0.00126	0.00832 ± 0.00063
0.2098	0.00582 ± 0.00049	0.00803 ± 0.00066	0.00539 ± 0.00072	0.00630 ± 0.00074	0.00577 ± 0.00082	0.00790 ± 0.00086	0.00534 ± 0.00098	0.00839 ± 0.00107	0.00415 ± 0.00121	0.00697 ± 0.00061
0.2212	0.00622 ± 0.00048	0.00447 ± 0.00063	0.00596 ± 0.00070	0.00472 ± 0.00072	0.00751 ± 0.00080	0.00445 ± 0.00083	0.00395 ± 0.00096	0.00386 ± 0.00104	0.00696 ± 0.00119	0.00543 ± 0.00059
0.2325	0.00472 ± 0.00047	0.00458 ± 0.00062	0.00620 ± 0.00069	0.00372 ± 0.00071	0.00478 ± 0.00078	0.00411 ± 0.00081	0.00486 ± 0.00094	0.00515 ± 0.00102	0.00442 ± 0.00116	0.00351 ± 0.00058
0.2439	0.00367 ± 0.00046	0.00499 ± 0.00061	0.00430 ± 0.00067	0.00496 ± 0.00069	0.00464 ± 0.00076	0.00339 ± 0.00080	0.00494 ± 0.00092	0.00533 ± 0.00101	0.00419 ± 0.00114	0.00451 ± 0.00057
0.2553	0.00398 ± 0.00046	0.00437 ± 0.00060	0.00368 ± 0.00066	0.00359 ± 0.00068	0.00305 ± 0.00075	0.00393 ± 0.00078	0.00306 ± 0.00090	0.00534 ± 0.00099	0.00453 ± 0.00112	0.00473 ± 0.00056
0.2667	0.00467 ± 0.00045	0.00644 ± 0.00060	0.00498 ± 0.00065	0.00544 ± 0.00067	0.00593 ± 0.00075	0.00521 ± 0.00078	0.00613 ± 0.00090	0.00531 ± 0.00097	0.00491 ± 0.00110	0.00580 ± 0.00055
0.2780	0.00554 ± 0.00045	0.00573 ± 0.00059	0.00475 ± 0.00064	0.00401 ± 0.00066	0.00415 ± 0.00073	0.00417 ± 0.00076	0.00529 ± 0.00089	0.00366 ± 0.00096	0.00445 ± 0.00109	0.00502 ± 0.00054
0.2894	0.00562 ± 0.00045	0.00517 ± 0.00057	0.00568 ± 0.00063	0.00529 ± 0.00065	0.00581 ± 0.00072	0.00701 ± 0.00076	0.00565 ± 0.00087	0.00447 ± 0.00094	0.00605 ± 0.00107	0.00594 ± 0.00054
0.3008	0.00667 ± 0.00044	0.00597 ± 0.00057	0.00663 ± 0.00063	0.00666 ± 0.00064	0.00596 ± 0.00071	0.00833 ± 0.00075	0.00514 ± 0.00085	0.00634 ± 0.00093	0.00636 ± 0.00106	0.00600 ± 0.00052
0.3122	0.00703 ± 0.00044	0.00725 ± 0.00056	0.00676 ± 0.00061	0.00824 ± 0.00064	0.00801 ± 0.00070	0.00807 ± 0.00073	0.00789 ± 0.00085	0.00862 ± 0.00091	0.00855 ± 0.00104	0.00745 ± 0.00052
0.3235	0.00775 ± 0.00043	0.00682 ± 0.00055	0.00732 ± 0.00060	0.00612 ± 0.00061	0.00637 ± 0.00068	0.00645 ± 0.00072	0.00677 ± 0.00082	0.00757 ± 0.00089	0.00698 ± 0.00102	0.00587 ± 0.00051
0.3349	0.00647 ± 0.00042	0.00550 ± 0.00053	0.00562 ± 0.00059	0.00573 ± 0.00061	0.00578 ± 0.00067	0.00501 ± 0.00069	0.00438 ± 0.00080	0.00544 ± 0.00087	0.00478 ± 0.00099	0.00437 ± 0.00049
0.3448	0.00700 ± 0.00047	0.00585 ± 0.00060	0.00506 ± 0.00066	0.00574 ± 0.00068	0.00574 ± 0.00075	0.00504 ± 0.00078	0.00532 ± 0.00091	0.00351 ± 0.00098	0.00354 ± 0.00112	0.00362 ± 0.00056

Table 16: Small-Angle X-ray Scattering intensities (in arbitrary units) as a function of the wavevector,  $q$ , for BSA in 1.0 mol·L<sup>-1</sup> NaCl aqueous solutions at pH = 3.4, at several protein concentrations,  $c_p$ , (mg·mL<sup>-1</sup>).

$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 73.42$	$c_p = 69.59$	$c_p = 59.63$	$c_p = 55.63$	$c_p = 51.03$	$c_p = 47.84$	$c_p = 42.58$	$c_p = 37.62$	$c_p = 25.93$
0.0142	1.47 ± 0.05	2.44 ± 0.07	2.45 ± 0.07	2.69 ± 0.08	2.89 ± 0.09	2.97 ± 0.09	3.11 ± 0.10	3.18 ± 0.11	3.42 ± 0.12	3.93 ± 0.15
0.0156	1.47 ± 0.03	2.37 ± 0.04	2.46 ± 0.04	2.62 ± 0.04	2.79 ± 0.05	2.85 ± 0.05	3.04 ± 0.05	3.11 ± 0.06	3.31 ± 0.06	3.82 ± 0.07
0.0171	1.47 ± 0.03	2.32 ± 0.03	2.41 ± 0.03	2.63 ± 0.04	2.76 ± 0.04	2.76 ± 0.04	2.90 ± 0.05	3.04 ± 0.05	3.10 ± 0.06	3.63 ± 0.06
0.0185	1.49 ± 0.03	2.35 ± 0.03	2.36 ± 0.03	2.57 ± 0.04	2.62 ± 0.04	2.81 ± 0.04	2.82 ± 0.04	2.93 ± 0.05	3.03 ± 0.05	3.51 ± 0.06
0.0199	1.51 ± 0.03	2.30 ± 0.03	2.35 ± 0.03	2.51 ± 0.03	2.65 ± 0.04	2.76 ± 0.04	2.80 ± 0.04	2.95 ± 0.04	3.12 ± 0.05	3.33 ± 0.05
0.0213	1.51 ± 0.03	2.25 ± 0.03	2.38 ± 0.03	2.48 ± 0.03	2.62 ± 0.03	2.69 ± 0.04	2.74 ± 0.04	2.91 ± 0.04	2.98 ± 0.05	3.17 ± 0.05
0.0228	1.49 ± 0.03	2.18 ± 0.03	2.31 ± 0.03	2.40 ± 0.03	2.55 ± 0.03	2.55 ± 0.03	2.62 ± 0.04	2.68 ± 0.04	2.86 ± 0.04	3.02 ± 0.04
0.0242	1.49 ± 0.02	2.14 ± 0.02	2.23 ± 0.03	2.32 ± 0.03	2.42 ± 0.03	2.43 ± 0.03	2.55 ± 0.03	2.59 ± 0.04	2.79 ± 0.04	2.88 ± 0.04
0.0256	1.50 ± 0.02	2.14 ± 0.02	2.14 ± 0.02	2.28 ± 0.03	2.36 ± 0.03	2.40 ± 0.03	2.45 ± 0.03	2.52 ± 0.03	2.63 ± 0.04	2.78 ± 0.04
0.0270	1.47 ± 0.02	2.10 ± 0.02	2.10 ± 0.02	2.24 ± 0.03	2.31 ± 0.03	2.37 ± 0.03	2.43 ± 0.03	2.50 ± 0.03	2.53 ± 0.04	2.72 ± 0.04
0.0284	1.42 ± 0.02	1.98 ± 0.02	2.04 ± 0.02	2.15 ± 0.02	2.25 ± 0.03	2.30 ± 0.03	2.34 ± 0.03	2.36 ± 0.03	2.45 ± 0.03	2.53 ± 0.03
0.0299	1.41 ± 0.02	1.97 ± 0.02	2.00 ± 0.02	2.07 ± 0.02	2.17 ± 0.03	2.28 ± 0.03	2.23 ± 0.03	2.31 ± 0.03	2.40 ± 0.03	2.41 ± 0.03
0.0320	1.41 ± 0.01	1.88 ± 0.01	1.89 ± 0.01	1.99 ± 0.02	2.07 ± 0.02	2.12 ± 0.02	2.16 ± 0.02	2.17 ± 0.02	2.21 ± 0.02	2.29 ± 0.02
0.0349	1.38 ± 0.01	1.78 ± 0.01	1.81 ± 0.01	1.89 ± 0.01	1.95 ± 0.02	1.97 ± 0.02	2.03 ± 0.02	2.00 ± 0.02	2.06 ± 0.02	2.01 ± 0.02
0.0377	1.31 ± 0.01	1.69 ± 0.01	1.70 ± 0.01	1.76 ± 0.01	1.81 ± 0.01	1.82 ± 0.01	1.84 ± 0.02	1.89 ± 0.02	1.87 ± 0.02	1.84 ± 0.02
0.0405	1.24 ± 0.01	1.56 ± 0.01	1.59 ± 0.01	1.60 ± 0.01	1.65 ± 0.01	1.65 ± 0.01	1.71 ± 0.02	1.68 ± 0.02	1.68 ± 0.02	1.61 ± 0.02
0.0434	1.21 ± 0.01	1.47 ± 0.01	1.47 ± 0.01	1.50 ± 0.01	1.55 ± 0.01	1.56 ± 0.01	1.57 ± 0.01	1.56 ± 0.01	1.52 ± 0.02	1.48 ± 0.02
0.0462	1.13 ± 0.01	1.35 ± 0.01	1.37 ± 0.01	1.38 ± 0.01	1.44 ± 0.01	1.43 ± 0.01	1.44 ± 0.01	1.42 ± 0.01	1.42 ± 0.01	1.33 ± 0.01
0.0491	1.08 ± 0.01	1.25 ± 0.01	1.26 ± 0.01	1.29 ± 0.01	1.33 ± 0.01	1.32 ± 0.01	1.32 ± 0.01	1.32 ± 0.01	1.31 ± 0.01	1.22 ± 0.01
0.0519	1.02 ± 0.01	1.17 ± 0.01	1.18 ± 0.01	1.19 ± 0.01	1.23 ± 0.01	1.21 ± 0.01	1.22 ± 0.01	1.20 ± 0.01	1.19 ± 0.01	1.12 ± 0.01
0.0548	0.968 ± 0.009	1.08 ± 0.01	1.08 ± 0.01	1.09 ± 0.01	1.10 ± 0.01	1.10 ± 0.01	1.10 ± 0.01	1.10 ± 0.01	1.07 ± 0.01	0.988 ± 0.011
0.0576	0.898 ± 0.008	0.996 ± 0.008	0.989 ± 0.008	0.991 ± 0.008	0.992 ± 0.009	1.01 ± 0.01	0.992 ± 0.010	0.972 ± 0.010	0.987 ± 0.011	0.918 ± 0.011
0.0605	0.847 ± 0.008	0.909 ± 0.007	0.912 ± 0.007	0.905 ± 0.008	0.915 ± 0.008	0.911 ± 0.009	0.900 ± 0.009	0.901 ± 0.010	0.882 ± 0.010	0.818 ± 0.010
0.0647	0.757 ± 0.005	0.789 ± 0.005	0.782 ± 0.005	0.787 ± 0.005	0.799 ± 0.005	0.790 ± 0.005	0.779 ± 0.006	0.761 ± 0.006	0.760 ± 0.007	0.703 ± 0.006
0.0704	0.633 ± 0.005	0.649 ± 0.004	0.651 ± 0.004	0.646 ± 0.004	0.639 ± 0.005	0.635 ± 0.005	0.642 ± 0.005	0.628 ± 0.005	0.613 ± 0.006	0.579 ± 0.006

$q / \text{\AA}^{-1}$	$c_p = 100.00$	$c_p = 73.42$	$c_p = 69.59$	$c_p = 59.63$	$c_p = 55.63$	$c_p = 51.03$	$c_p = 47.84$	$c_p = 42.58$	$c_p = 37.62$	$c_p = 25.93$
0.0761	$0.523 \pm 0.004$	$0.523 \pm 0.003$	$0.522 \pm 0.004$	$0.517 \pm 0.004$	$0.519 \pm 0.004$	$0.518 \pm 0.004$	$0.510 \pm 0.004$	$0.505 \pm 0.005$	$0.497 \pm 0.005$	$0.462 \pm 0.005$
0.0818	$0.423 \pm 0.003$	$0.418 \pm 0.003$	$0.422 \pm 0.003$	$0.421 \pm 0.003$	$0.417 \pm 0.004$	$0.416 \pm 0.004$	$0.413 \pm 0.004$	$0.404 \pm 0.004$	$0.400 \pm 0.004$	$0.370 \pm 0.004$
0.0875	$0.345 \pm 0.003$	$0.330 \pm 0.003$	$0.330 \pm 0.003$	$0.325 \pm 0.003$	$0.326 \pm 0.003$	$0.327 \pm 0.003$	$0.323 \pm 0.003$	$0.312 \pm 0.004$	$0.318 \pm 0.004$	$0.305 \pm 0.004$
0.0932	$0.269 \pm 0.003$	$0.265 \pm 0.002$	$0.260 \pm 0.002$	$0.259 \pm 0.003$	$0.256 \pm 0.003$	$0.256 \pm 0.003$	$0.259 \pm 0.003$	$0.256 \pm 0.003$	$0.261 \pm 0.003$	$0.258 \pm 0.004$
0.0989	$0.213 \pm 0.002$	$0.208 \pm 0.002$	$0.209 \pm 0.002$	$0.207 \pm 0.002$	$0.212 \pm 0.002$	$0.205 \pm 0.002$	$0.210 \pm 0.003$	$0.208 \pm 0.003$	$0.207 \pm 0.003$	$0.217 \pm 0.003$
0.1074	$0.145 \pm 0.001$	$0.143 \pm 0.001$	$0.145 \pm 0.001$	$0.146 \pm 0.001$	$0.146 \pm 0.001$	$0.146 \pm 0.001$	$0.148 \pm 0.002$	$0.148 \pm 0.002$	$0.150 \pm 0.002$	$0.159 \pm 0.002$
0.1188	$0.0922 \pm 0.0011$	$0.0951 \pm 0.0010$	$0.0953 \pm 0.0010$	$0.0967 \pm 0.0011$	$0.0989 \pm 0.0012$	$0.0978 \pm 0.0012$	$0.0996 \pm 0.0013$	$0.103 \pm 0.001$	$0.102 \pm 0.002$	$0.112 \pm 0.002$
0.1302	$0.0678 \pm 0.0009$	$0.0692 \pm 0.0008$	$0.0694 \pm 0.0009$	$0.0716 \pm 0.0009$	$0.0694 \pm 0.0010$	$0.0718 \pm 0.0011$	$0.0712 \pm 0.0012$	$0.0757 \pm 0.0013$	$0.0747 \pm 0.0014$	$0.0783 \pm 0.0016$
0.1415	$0.0508 \pm 0.0008$	$0.0550 \pm 0.0007$	$0.0554 \pm 0.0008$	$0.0560 \pm 0.0008$	$0.0549 \pm 0.0009$	$0.0561 \pm 0.0010$	$0.0562 \pm 0.0010$	$0.0566 \pm 0.0011$	$0.0578 \pm 0.0012$	$0.0583 \pm 0.0014$
0.1529	$0.0398 \pm 0.0007$	$0.0436 \pm 0.0007$	$0.0426 \pm 0.0007$	$0.0425 \pm 0.0008$	$0.0428 \pm 0.0008$	$0.0427 \pm 0.0009$	$0.0438 \pm 0.0009$	$0.0453 \pm 0.0010$	$0.0457 \pm 0.0011$	$0.0416 \pm 0.0013$
0.1643	$0.0317 \pm 0.0006$	$0.0341 \pm 0.0006$	$0.0328 \pm 0.0006$	$0.0335 \pm 0.0007$	$0.0340 \pm 0.0007$	$0.0342 \pm 0.0008$	$0.0344 \pm 0.0009$	$0.0347 \pm 0.0009$	$0.0331 \pm 0.0010$	$0.0333 \pm 0.0012$
0.1757	$0.0256 \pm 0.0006$	$0.0262 \pm 0.0005$	$0.0270 \pm 0.0006$	$0.0260 \pm 0.0006$	$0.0263 \pm 0.0007$	$0.0268 \pm 0.0007$	$0.0256 \pm 0.0008$	$0.0247 \pm 0.0008$	$0.0268 \pm 0.0010$	$0.0267 \pm 0.0011$
0.1870	$0.0204 \pm 0.0005$	$0.0215 \pm 0.0005$	$0.0210 \pm 0.0005$	$0.0205 \pm 0.0006$	$0.0209 \pm 0.0006$	$0.0222 \pm 0.0007$	$0.0207 \pm 0.0007$	$0.0205 \pm 0.0008$	$0.0204 \pm 0.0009$	$0.0197 \pm 0.0010$
0.1984	$0.0171 \pm 0.0005$	$0.0161 \pm 0.0005$	$0.0158 \pm 0.0005$	$0.0169 \pm 0.0005$	$0.0162 \pm 0.0006$	$0.0151 \pm 0.0006$	$0.0155 \pm 0.0007$	$0.0154 \pm 0.0007$	$0.0154 \pm 0.0008$	$0.0140 \pm 0.0010$
0.2098	$0.0141 \pm 0.0005$	$0.0132 \pm 0.0004$	$0.0141 \pm 0.0005$	$0.0138 \pm 0.0005$	$0.0137 \pm 0.0006$	$0.0126 \pm 0.0006$	$0.0131 \pm 0.0006$	$0.0133 \pm 0.0007$	$0.0147 \pm 0.0008$	$0.0124 \pm 0.0010$
0.2212	$0.0128 \pm 0.0004$	$0.0115 \pm 0.0004$	$0.0113 \pm 0.0004$	$0.0113 \pm 0.0005$	$0.0107 \pm 0.0005$	$0.0116 \pm 0.0006$	$0.0118 \pm 0.0006$	$0.0110 \pm 0.0007$	$0.0107 \pm 0.0008$	$0.0125 \pm 0.0009$
0.2325	$0.00999 \pm 0.00040$	$0.00996 \pm 0.00041$	$0.0105 \pm 0.0004$	$0.0101 \pm 0.0005$	$0.0101 \pm 0.0005$	$0.00947 \pm 0.00056$	$0.00903 \pm 0.00061$	$0.0100 \pm 0.0007$	$0.00925 \pm 0.00076$	$0.00970 \pm 0.00091$
0.2439	$0.00956 \pm 0.00040$	$0.00873 \pm 0.00039$	$0.00884 \pm 0.00042$	$0.00871 \pm 0.00047$	$0.00901 \pm 0.00051$	$0.00964 \pm 0.00055$	$0.00947 \pm 0.00061$	$0.0100 \pm 0.0007$	$0.00845 \pm 0.00074$	$0.0109 \pm 0.0009$
0.2553	$0.00882 \pm 0.00039$	$0.00775 \pm 0.00039$	$0.00796 \pm 0.00041$	$0.00808 \pm 0.00046$	$0.00864 \pm 0.00051$	$0.00822 \pm 0.00054$	$0.00826 \pm 0.00059$	$0.00836 \pm 0.00067$	$0.00850 \pm 0.00074$	$0.00813 \pm 0.00090$
0.2667	$0.00789 \pm 0.00038$	$0.00752 \pm 0.00038$	$0.00776 \pm 0.00041$	$0.00804 \pm 0.00046$	$0.00840 \pm 0.00050$	$0.00737 \pm 0.00053$	$0.00858 \pm 0.00059$	$0.00839 \pm 0.00066$	$0.00832 \pm 0.00074$	$0.00653 \pm 0.00088$
0.2780	$0.00870 \pm 0.00038$	$0.00845 \pm 0.00038$	$0.00898 \pm 0.00041$	$0.00853 \pm 0.00046$	$0.00856 \pm 0.00050$	$0.00844 \pm 0.00053$	$0.00905 \pm 0.00059$	$0.00832 \pm 0.00065$	$0.00892 \pm 0.00072$	$0.00832 \pm 0.00088$
0.2894	$0.00909 \pm 0.00038$	$0.00924 \pm 0.00039$	$0.00881 \pm 0.00040$	$0.00864 \pm 0.00045$	$0.00817 \pm 0.00049$	$0.00810 \pm 0.00053$	$0.00853 \pm 0.00058$	$0.00825 \pm 0.00064$	$0.00830 \pm 0.00072$	$0.00910 \pm 0.00088$
0.3008	$0.00904 \pm 0.00038$	$0.00933 \pm 0.00038$	$0.00872 \pm 0.00040$	$0.00892 \pm 0.00045$	$0.00903 \pm 0.00049$	$0.00919 \pm 0.00052$	$0.00908 \pm 0.00058$	$0.00914 \pm 0.00064$	$0.00778 \pm 0.00071$	$0.00884 \pm 0.00087$
0.3122	$0.00930 \pm 0.00038$	$0.00921 \pm 0.00038$	$0.00883 \pm 0.00039$	$0.00917 \pm 0.00044$	$0.00887 \pm 0.00048$	$0.00886 \pm 0.00051$	$0.00816 \pm 0.00056$	$0.00853 \pm 0.00063$	$0.00950 \pm 0.00071$	$0.00821 \pm 0.00085$
0.3235	$0.00832 \pm 0.00037$	$0.00902 \pm 0.00037$	$0.00935 \pm 0.00039$	$0.00898 \pm 0.00044$	$0.00934 \pm 0.00048$	$0.00913 \pm 0.00051$	$0.00932 \pm 0.00056$	$0.00927 \pm 0.00062$	$0.00893 \pm 0.00069$	$0.00976 \pm 0.00084$
0.3349	$0.00837 \pm 0.00036$	$0.00831 \pm 0.00036$	$0.00862 \pm 0.00038$	$0.00818 \pm 0.00043$	$0.00861 \pm 0.00047$	$0.00886 \pm 0.00050$	$0.00763 \pm 0.00054$	$0.00791 \pm 0.00060$	$0.00804 \pm 0.00068$	$0.00781 \pm 0.00083$
0.3448	$0.00813 \pm 0.00041$	$0.00787 \pm 0.00041$	$0.00722 \pm 0.00043$	$0.00769 \pm 0.00048$	$0.00693 \pm 0.00053$	$0.00763 \pm 0.00056$	$0.00787 \pm 0.00063$	$0.00614 \pm 0.00069$	$0.00703 \pm 0.00077$	$0.00637 \pm 0.00094$

## APPENDIX H - FITTED SAXS INTENSITY DATA

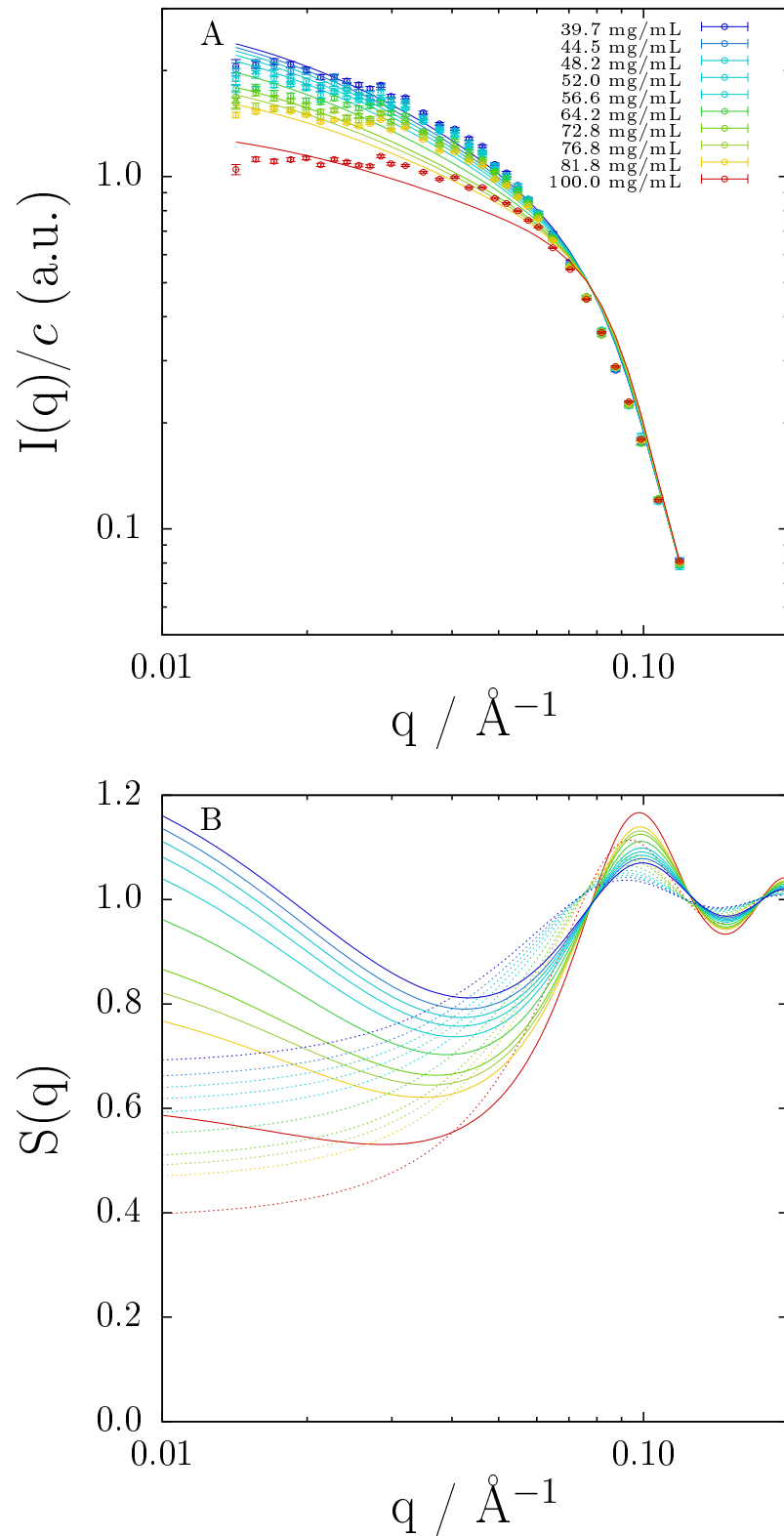


Figure 39: A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L<sup>-1</sup> NaCl aqueous solution at pH=4.9 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with  $\varepsilon$  calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with  $\varepsilon$  calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution.

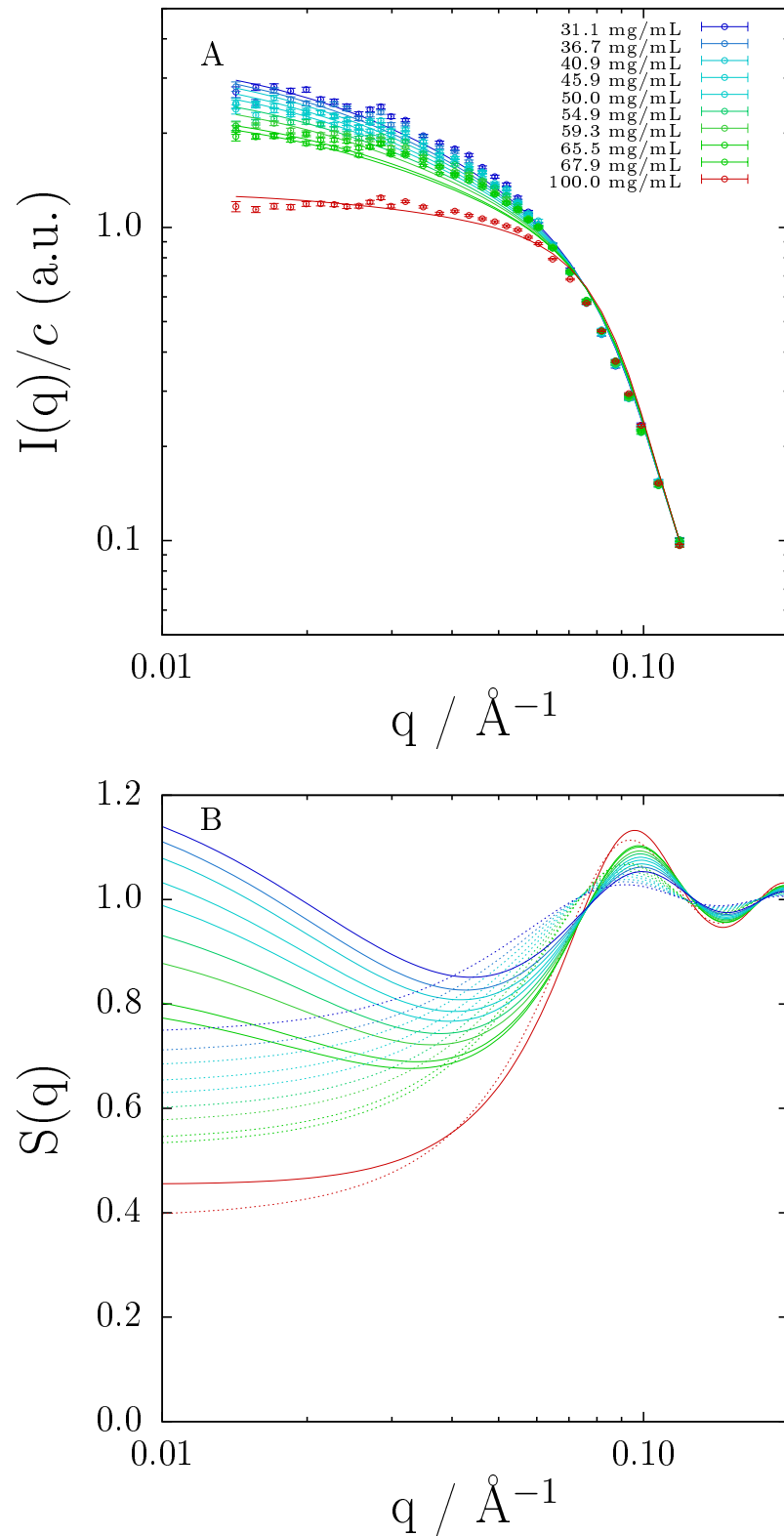


Figure 40: A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L<sup>-1</sup> NaNO<sub>3</sub> aqueous solution at pH=4.9 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with  $\varepsilon$  calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with  $\varepsilon$  calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution.

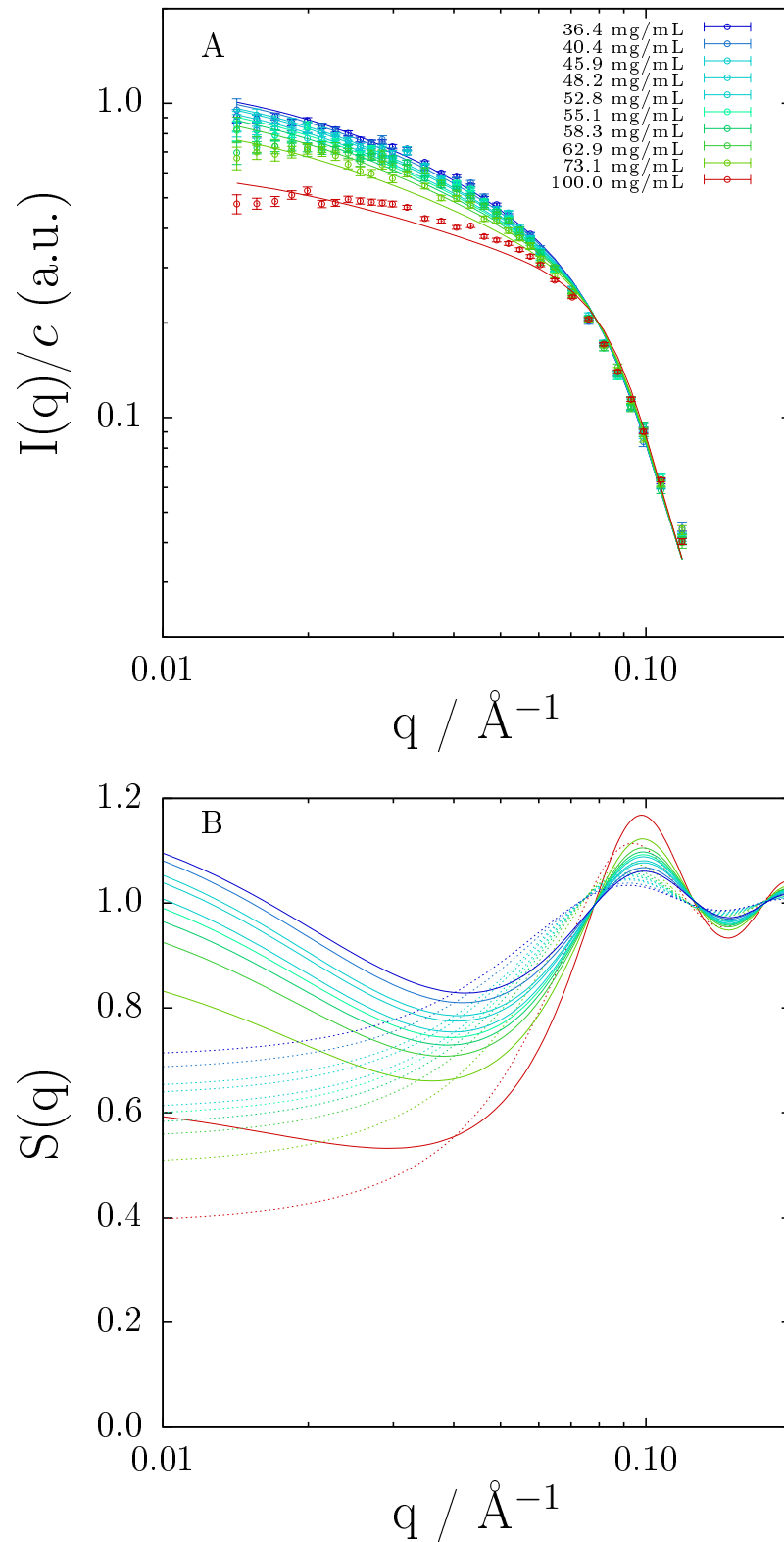


Figure 41: A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L<sup>-1</sup> Na<sub>2</sub>SO<sub>4</sub> aqueous solution at pH=4.9 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with  $\varepsilon$  calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with  $\varepsilon$  calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution.

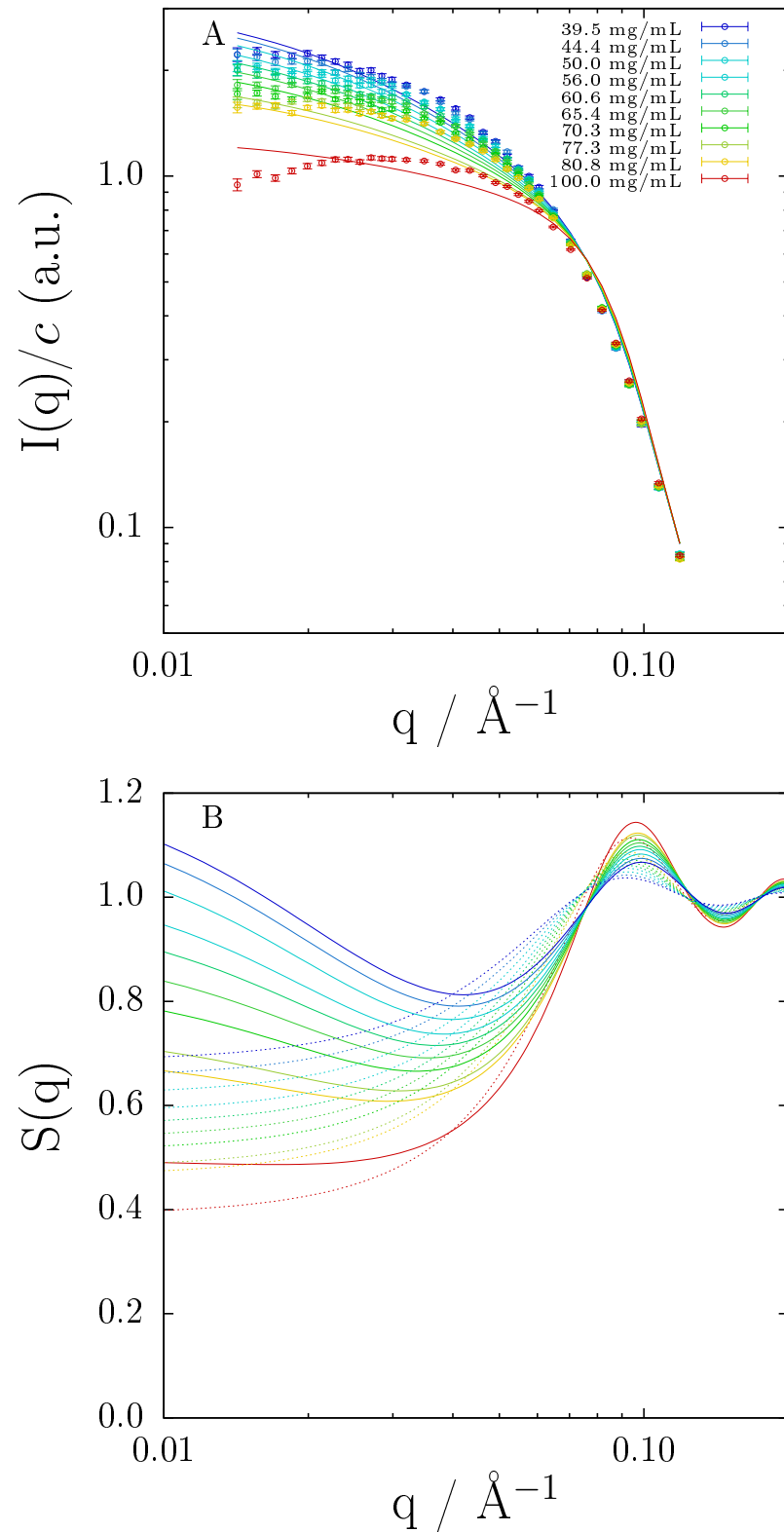


Figure 42: A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L<sup>-1</sup> NaCl aqueous solution at pH=6.3 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with  $\varepsilon$  calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with  $\varepsilon$  calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution.



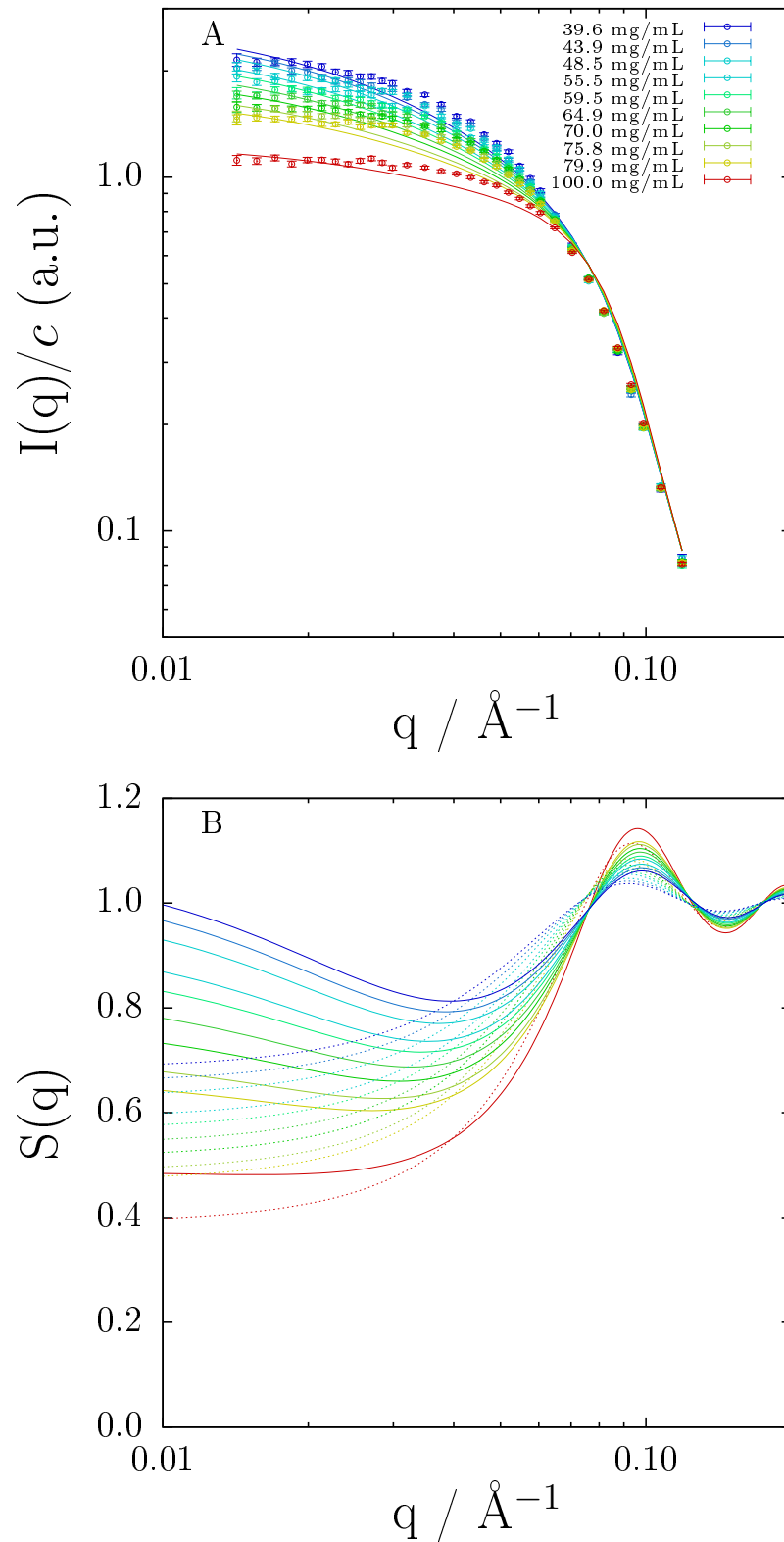


Figure 43: A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L<sup>-1</sup> NaNO<sub>3</sub> aqueous solution at pH=6.3 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with  $\varepsilon$  calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with  $\varepsilon$  calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution.

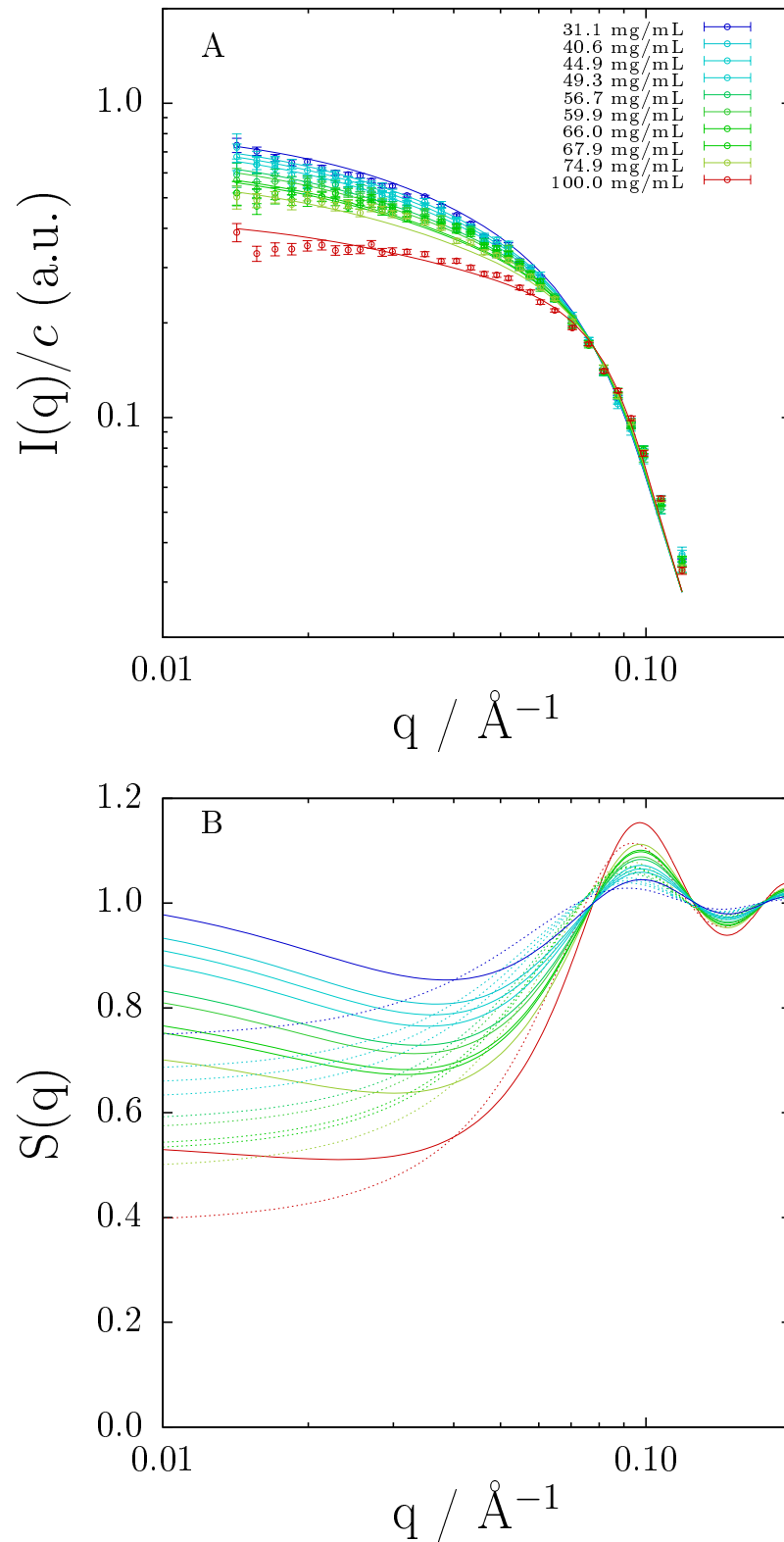


Figure 44: A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L<sup>-1</sup> Na<sub>2</sub>SO<sub>4</sub> aqueous solution at pH=6.3 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with  $\varepsilon$  calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with  $\varepsilon$  calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution.

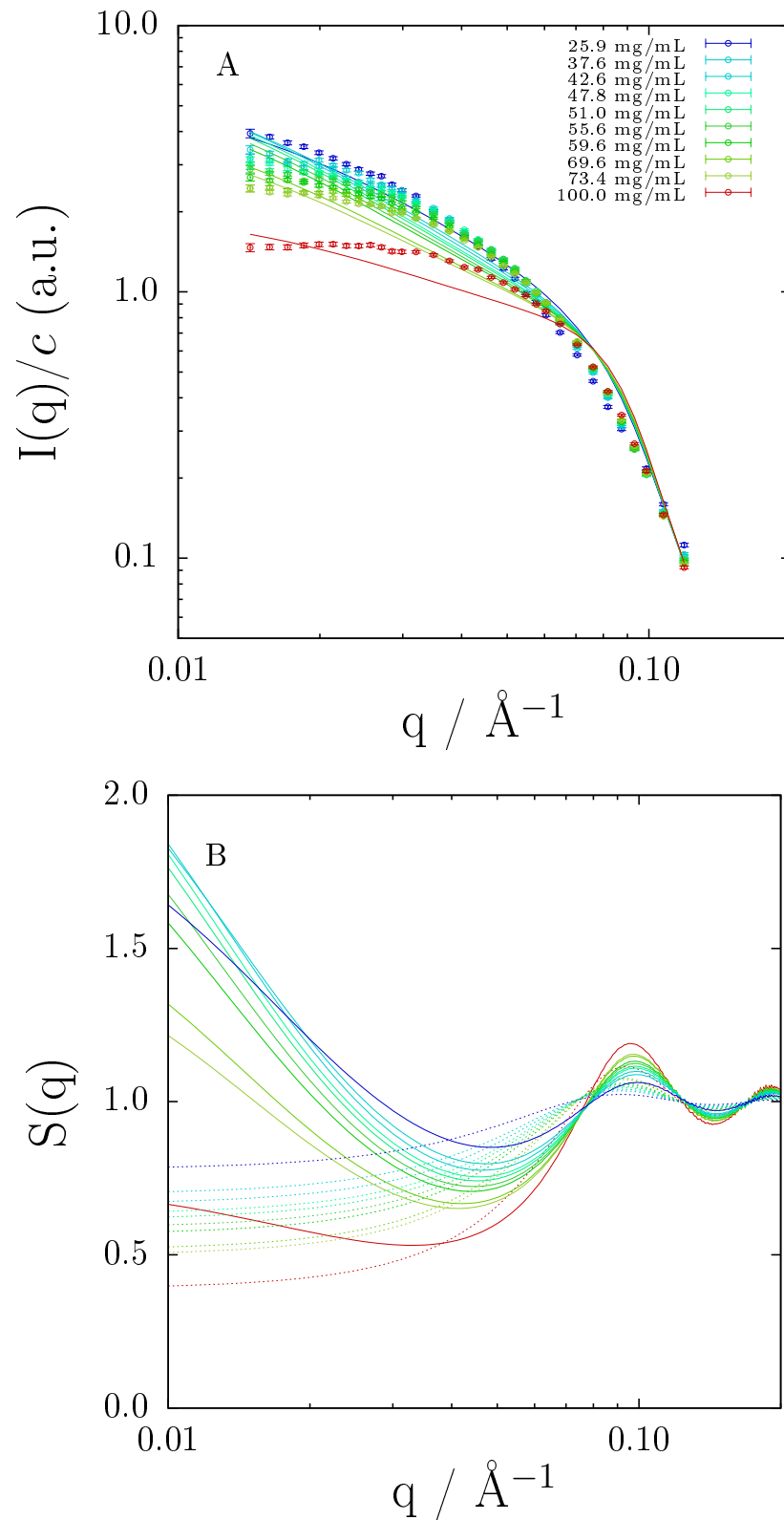


Figure 45: A) X-ray scattering intensity (in a.u., arbitrary units) of BSA in 1.0 mol·L<sup>-1</sup> NaCl aqueous solution at pH=3.4 and 23°C. Open circles, experimental data; continuous line, Equation (5.32) with  $\varepsilon$  calculated using Equation (5.38). B) Calculated structure factor. Continuous line, Equations (5.14) and (5.32) with  $\varepsilon$  calculated using Equation (5.38); dashed line, Hard-Sphere Percus-Yevick solution.

## APPENDIX I - FIGURES FOR MOLTEN NaCl SIMULATIONS

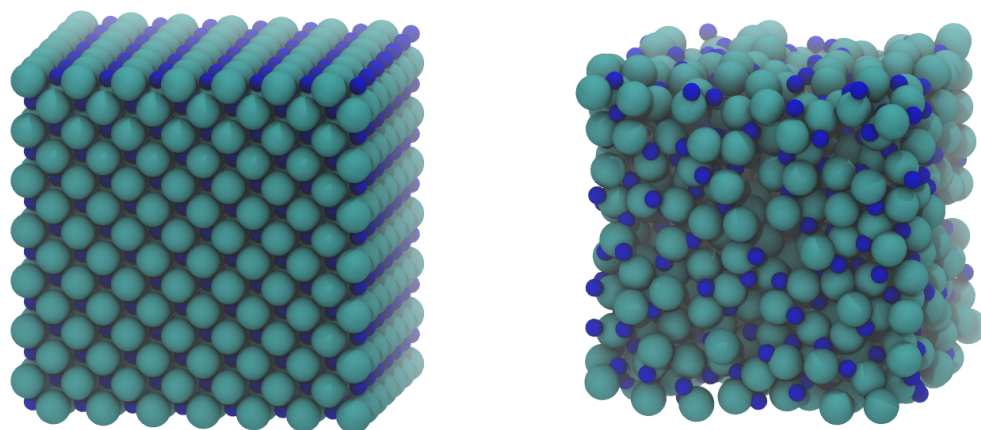


Figure 46: Initial FCC configuration at the left side and final molten NaCl configuration at the right side. Blue beads represents sodium ions and green beads represents chloride.

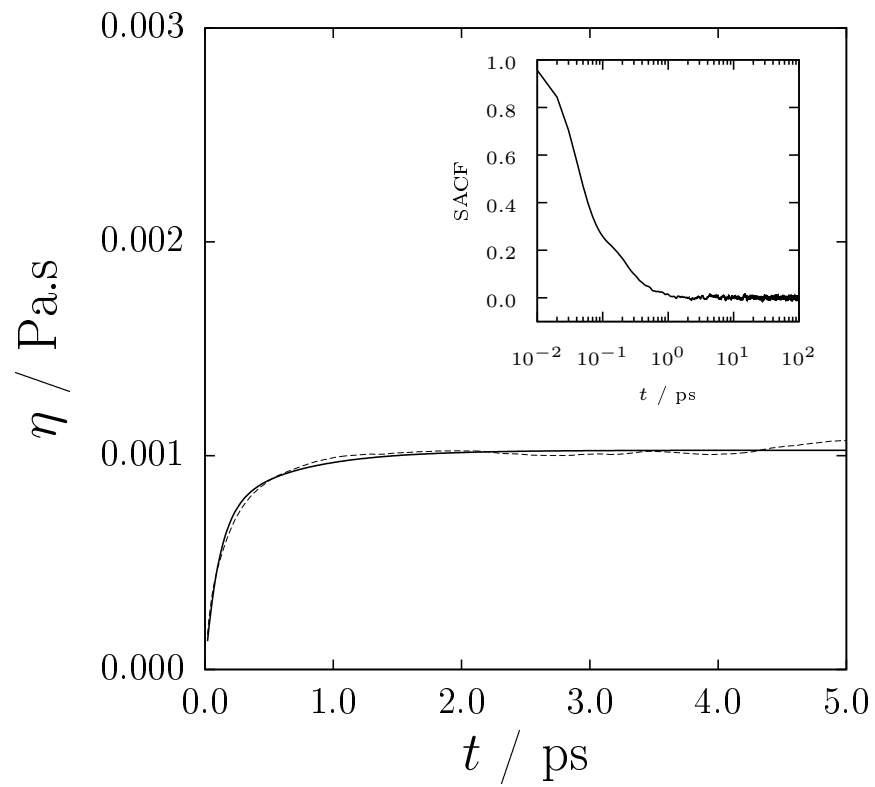


Figure 47: Viscosity of molten NaCl calculated from equilibrium molecular dynamics simulations using Green-Kubo relations at 1100 K. The inset plot shows the stress autocorrelation function.

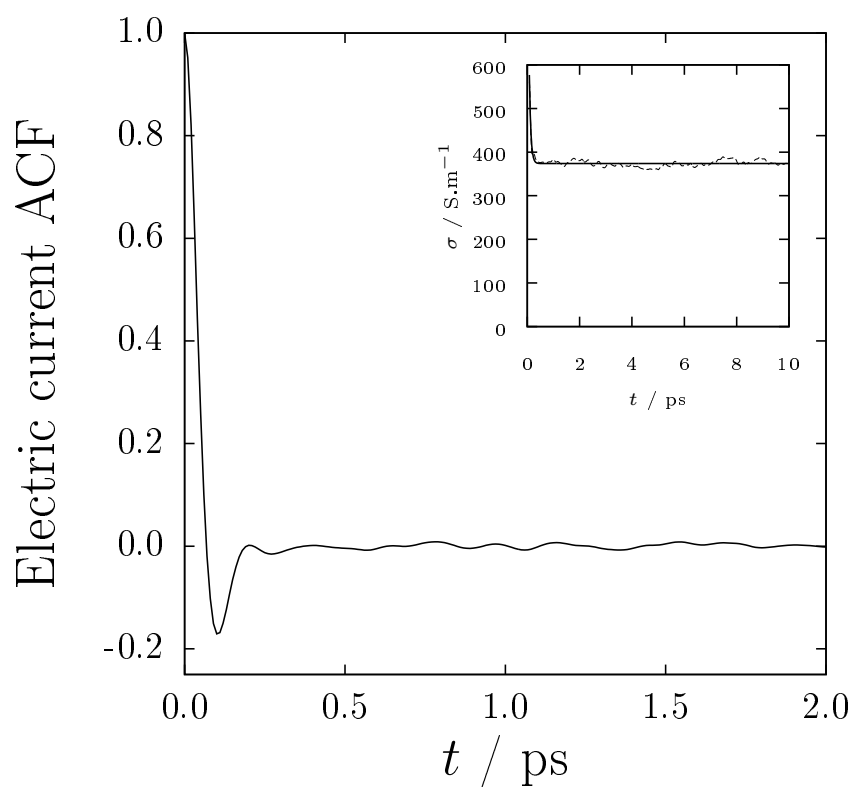


Figure 48: Electric current autocorrelation function of molten NaCl calculated from equilibrium molecular dynamics simulations at 1300 K. The inset plot shows the electric conductivity using Green-Kubo relations.

## APPENDIX J – PLOTS FOR VISCOSITY AND ELECTRIC CONDUCTIVITY

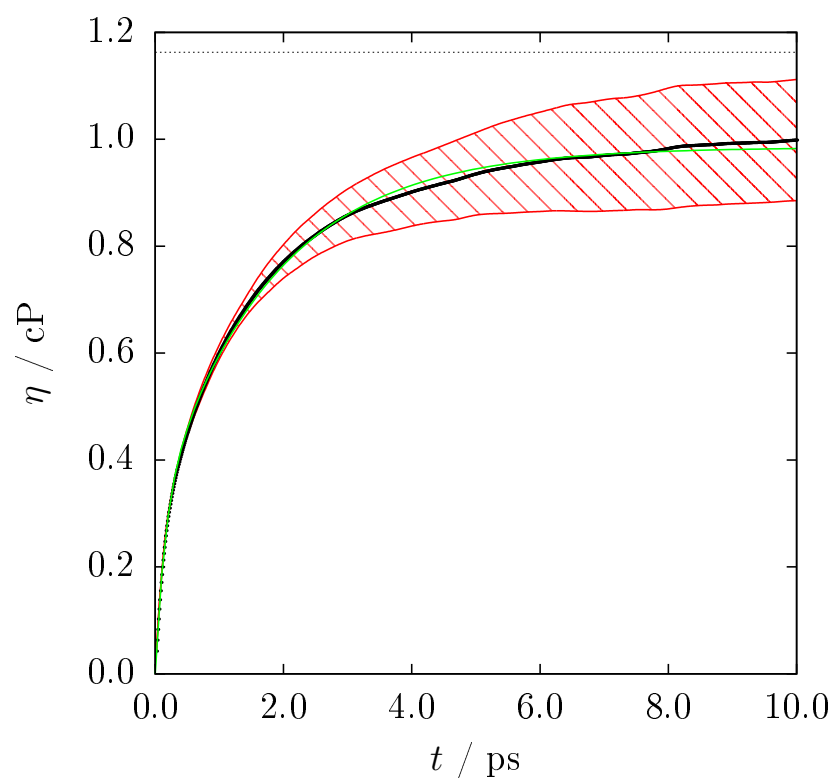


Figure 49: Viscosity of 0.05 mol·kg<sup>-1</sup> sodium sulfate aqueous solution at 293.15 K. Red lines represent the standard deviation calculated by five different initial configurations. Continuous black line, average viscosity. Continuous green line, fitted viscosity by Equation (7.15). Dotted line, experimental value (ISONO, 1984).

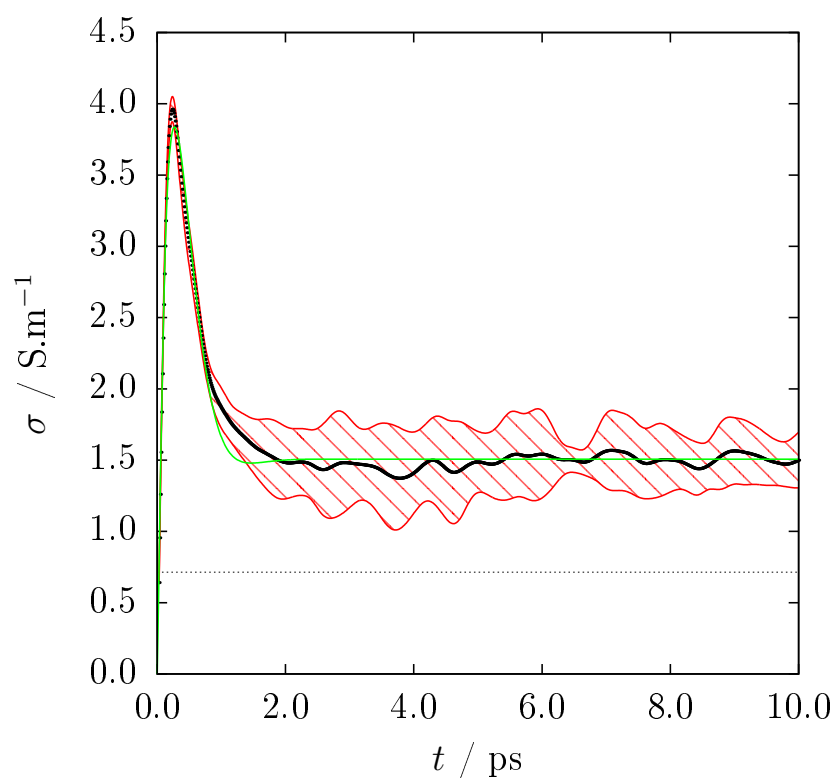


Figure 50: Ionic conductivity of  $0.05 \text{ mol}\cdot\text{kg}^{-1}$  sodium sulfate aqueous solution at  $293.15 \text{ K}$ . Red lines represent the standard deviation calculated by five different initial configurations. Continuous black line, average ionic conductivity. Continuous green line, fitted ionic conductivity by Equation (7.16). Dotted line, experimental value (ISONO, 1984).



## APPENDIX K - RELATION BETWEEN THE MEAN DIFFUSIVITY COEFFICIENT AND IONIC DIFFUSIVITIES

The mean diffusivity coefficient,  $D$ , may be expressed as the harmonic mean of the ionic diffusivities weighted by the stoichiometric coefficients, and this expression may be rewritten as (PROBSTEN, 1994):

$$D = D_+ \left( \frac{1 - z_+/z_-}{1 - z_+D_+/z_-D_-} \right) \quad (\text{K.1})$$

where  $D_+$  is the cation diffusivity,  $D_-$  is the anion diffusivity,  $z_+$  is the cation valence and  $z_-$  is the anion valence.

To calculate the Nernst-Einstein conductivity in terms of the mean diffusivity coefficient, one may replace the summation in Nernst-Einstein expression (URAHATA; RIBEIRO, 2006) for a parameter multiplied by the mean diffusivity coefficient:

$$\sum_i \nu_i z_i^2 D_i = \nu_+ z_+^2 D_+ + \nu_- z_- D_- = \zeta D \quad (\text{K.2})$$

where  $\nu_i$  is the stoichiometric coefficient of ion  $i$  and  $\zeta$  is the new parameter which relates the mean diffusivity coefficient to the ionic diffusivities.

Rearranging Equation (K.2), one has:

$$\nu_+ z_+^2 D_+ + \nu_- z_- D_- = D_+ \left( \nu_+ z_+^2 + \nu_- z_-^2 \frac{D_-}{D_+} \right) = \zeta D \quad (\text{K.3})$$

Comparing Equation (K.1) to Equation (K.3), one has that  $\zeta$  is given by the following relation:

$$\zeta = \frac{(\nu_+ z_+^2 + \nu_- z_-^2 D_- / D_+)}{\left( \frac{1 - z_+ / z_-}{1 - z_+ D_+ / z_- D_-} \right)} \quad (\text{K.4})$$

For the sake of simplicity, let us define some parameters:

$$\theta = \frac{z_+}{z_-} \quad (\text{K.5})$$

$$\delta = \frac{D_+}{D_-} \quad (\text{K.6})$$

$$\nu = \frac{\nu_+}{\nu_-} \quad (\text{K.7})$$

Therefore:

$$\zeta = z_+^2 \left( 1 + \frac{1}{\nu \theta^2 \delta} \right) \left( \frac{1 - \theta \delta}{1 - \theta} \right) \quad (\text{K.8})$$

Or:

$$\zeta = z_+^2 \frac{(1 + \nu \theta^2 \delta)(1 - \theta \delta)}{\nu \theta^2 \delta (1 - \theta)} \quad (\text{K.9})$$

Evidently, for a 1:1 salt:

$$\zeta = \frac{(1 + \delta)^2}{2\delta} \quad (\text{K.10})$$

It remains an issue, however, how to evaluate the value of parameter  $\delta$ . Assuming Stokes-Einstein law, this parameter could be given by the inverse ratio of the Stokes radii:

$$\delta = \frac{D_+}{D_-} = \frac{r_-}{r_+} \quad (\text{K.11})$$

Table 17 presents the values of Stokes radii for a series of monovalent ions

obtained in the literature (PAU; BERG; MCMILLAN, 1990) and the resulting value of  $\delta$  and  $\zeta$  for a variety of 1:1 salts.

Table 17: Stokes radii for monovalent ions obtained in the literature (PAU; BERG; MCMILLAN, 1990) and values of  $\delta$  and  $\zeta$  for alkali halides according to Equations (K.11) and (K.10) respectively.

Salt	$r_+ / \text{\AA}$	$r_- / \text{\AA}$	$\delta$	$\zeta$
LiF	3.58	2.49	1.44	2.07
LiCl	3.58	1.81	1.98	2.24
LiBr	3.58	1.77	2.02	2.26
LiI	3.58	1.8	1.99	2.25
NaF	2.76	2.49	1.11	2.01
NaCl	2.76	1.81	1.52	2.09
NaBr	2.76	1.77	1.56	2.10
NaI	2.76	1.8	1.53	2.09
KF	1.88	2.49	0.76	2.04
KCl	1.88	1.81	1.04	2.00
KBr	1.88	1.77	1.06	2.00
KI	1.88	1.8	1.04	2.00
RbF	1.78	2.49	0.71	2.06
RbCl	1.78	1.81	0.98	2.00
RbBr	1.78	1.77	1.01	2.00
RbI	1.78	1.8	0.99	2.00
CsF	1.79	2.49	0.72	2.05
CsCl	1.79	1.81	0.99	2.00
CsBr	1.79	1.77	1.01	2.00
CsI	1.79	1.8	0.99	2.00