BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

BIOMOLECULAR ELECTROSTATICS WITH CONTINUUM MODELS: A BOUNDARY INTEGRAL IMPLEMENTATION AND APPLICATIONS TO BIOSENSORS

by

CHRISTOPHER COOPER VILLAGRÁN

B.Sc., Universidad Técnica Federico Santa María, 2007
Engineer's Degree, Universidad Técnica Federico Santa María, 2009
M.S., Boston University, 2012

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2015

UMI Number: 3684993

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

UMI 3684993

Published by ProQuest LLC (2015). Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.
All rights reserved. This work is protected against
unauthorized copying under Title 17, United States Code

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

© 2014 by
Christopher Cooper Villagrán
All rights reserved

Approved by

First Reader

Lorena A. Barba, PhD
Adjunct Professor of Mechanical Engineering
Boston University
Associate Professor of Mechanical and Aerospace Engineering
The George Washington University

Second Reader

Harold Park, PhD
Associate Professor of Mechanical Engineering

Third Reader

Paul Barbone, PhD
Professor of Mechanical Engineering

Fourth Reader

Matthias Schneider, PhD
Assistant Professor of Mechanical Engineering

Acknowledgments

¡Gracias totales! was the phrase used by argentinian musician Gustavo Cerati (b. 1959, d. 2014), lead singer of Soda Stereo, to thank fans and friends in what was supposed to be their last presentation, in 1997 — they did a reunion tour in 2007. This expression became very popular in South America to express gratitude. Even though I am not a die-hard fan of Soda, I do not plan to do another PhD in ten years, and this is a different context, I use it here because it represents how deeply grateful I am of the people who helped me through.

I would like to express my deepest gratitude to my advisor, Prof. Lorena Barba, who encouraged me to apply for the PhD, and has given me guidance and support these five years. I am mostly grateful for the trust and freedom she gave me to develop my research, even when we were both new to the topic. Her mentoring and working style has shaped me in the way I approach research, and I feel very lucky and honored to be her student.

I am also very grateful of my thesis committee: Prof. Paul Barbone, Prof. Harold Park, and Prof. Matthias Schneider, for lending their time and insights to this project. To professor Schneider in particular for showing interest in my progress and well being, and for some very enriching conversations. Also, special thanks to Prof. Harold Park, for his enthusiasm and kindly agreeing to be a part of this committee in such short notice.

While at BU, I realized that most of the value of the PhD is in the interactions with the research group, and I was part of a very special one. Many thanks to Prof. Barba's research group: Simon, Anush, Olivier, Brent, Paul and Gil, are the ones I shared the most with, past members Rio and Felipe, were very helpful in the early stages, and new members Naty, Piyueh and Tingyu, have been of great assistance recently. Interactions with them has affected this work tremendously, by means of technical discussions, help debugging code, or just relaxing in the pub. I cannot overstate how important their friendship has been over these years!

Collaborations have been key in my developments. I am deeply grateful of the interactions with Prof. Jaydeep Bardhan, from Northeastern University, which went beyond purely technical matters. His deep knowledge of biophysics and BEM has been a great guide to this work. I am also very thankful of communications with people from the Navy Research Laboratory, specially Dr. Marc Raphael, Dr. Jeff Byers, Dr. Ellen Goldman, and Dr. George Anderson. They were always very keen (and fast!) on answering our questions on biosensors and molecular biology.

I would like to thank Boston University, the Office of Naval Research (ONR), and the Becas Chile program from CONICYT. Their financial support is greatly appreciated.

I wish to express my gratitude to my parents, who's support and example have had a deep influence on me. Thank you for always being there for me.

I was blessed to find my soul mate early in the game. Without Daniella's endless and unconditional love and support none of this would have been possible. Her selflessness, even in difficult times away from the family, is an inspiration to me. The last few months have been very special with the arrival of Amaya, who became the center of our lives, and makes me feel excited when I think about the future. This thesis is dedicated, with love, to them.

BIOMOLECULAR ELECTROSTATICS WITH CONTINUUM MODELS: A BOUNDARY INTEGRAL IMPLEMENTATION AND APPLICATIONS TO BIOSENSORS

CHRISTOPHER COOPER VILLAGRÁN

Boston University, College of Engineering, 2015

Major Professor: Lorena A. Barba, PhD,
Adjunct Professor of Mechanical Engineering
Boston University
Associate Professor of Mechanical & Aerospace Engineering
The George Washington University

ABSTRACT

The implicit-solvent model uses continuum electrostatic theory to represent the salt solution around dissolved biomolecules, leading to a coupled system of the Poisson-Boltzmann and Poisson equations. This thesis uses the implicit-solvent model to study solvation, binding and adsorption of proteins.

We developed an implicit-solvent model solver that uses the boundary element method (BEM), called `PyGBe`. BEM numerically solves integral equations along the biomolecule-solvent interface only, therefore, it does not need to discretize the entire domain. `PyGBe` accelerates the BEM with a treecode algorithm and runs on graphic processing units. We performed extensive verification and validation of the code, comparing it with experimental observations, analytical solutions, and other numerical tools. Our results suggest that a BEM approach is more appropriate than volumetric-

based methods, like finite-difference or finite-element, for high accuracy calculations. We also discussed the effect of features like solvent-filled cavities and Stern layers in the implicit-solvent model, and realized that they become relevant in binding energy calculations.

The application that drove this work was nano-scale biosensors —devices designed to detect biomolecules. Biosensors are built with a functionalized layer of ligand molecules, to which the target molecule binds when it is detected. With our code, we performed a study of the orientation of proteins near charged surfaces, and investigated the ideal conditions for ligand molecule adsorption. Using immunoglobulin G as a test case, we found out that low salt concentration in the solvent and high positive surface charge density leads to favorable orientations of the ligand molecule for biosensing applications.

We also studied the plasmonic response of localized surface plasmon resonance (LSPR) biosensors. LSPR biosensors monitor the plasmon resonance frequency of metallic nanoparticles, which shifts when a target molecule binds to a ligand molecule. Electrostatics is a valid approximation to the LSPR biosensor optical phenomenon in the long-wavelength limit, and BEM was able to reproduce the shift in the plasmon resonance frequency as proteins approach the nanoparticle.

Contents

1 Introduction **1**

2 Background **4**

2.1 Implicit solvent model 4

2.1.1 Electrostatics in vacuum 4

2.1.2 Electrostatics in a continuum dielectric medium 9

2.1.3 The effect of an electrolyte solution 13

2.1.4 Implicit solvent models in molecular systems 18

2.1.5 Energy calculation with the implicit solvent model 22

2.1.6 Calculations using the implicit-solvent model 25

2.2 Boundary Element Methods (BEM) 28

2.2.1 Derivation of a Laplace problem 29

2.2.2 Layer operators 33

2.2.3 Discretization and linear system 38

2.2.4 Acceleration strategies 39

2.3 Treecode 42

2.4 Treecode in BEM 47

2.5 Biosensors 48

2.5.1 Biosensor models with implicit-solvent 49

3 Methods **52**

3.1 Integral formulation of molecular systems 52

3.1.1 Single-surface models 52

3.1.2 Multi-surface models 53

3.2 Integral formulation for protein surface interaction 57

3.3 Numerical solution through the boundary element method 58

3.3.1 Discretization of the equations 58

3.3.2 Energy calculation with the boundary element method 60

3.3.3 Integral evaluation 61

3.3.4 Preconditioning 64

3.4 Implementation details of the treecode 66

3.4.1 Computation of the normal derivative 66

3.4.2 Efficiency considerations in BEM 67

3.4.3 GPU implementation 67

3.5 Richardson extrapolation 69

3.6 `PyGBe` 70

4 Calculation of solvation and binding energies 72

4.1 Verification of `PyGBe` 73

4.1.1 Comparison with analytical solutions 73

4.1.2 Comparison with existing codes 75

4.2 Effect of solvent-filled cavities and Stern layers 77

4.2.1 Solvation energy 77

4.2.2 Binding energy 78

4.2.3 Peptide-RNA complex 84

5 Protein-surface interaction calculations 87

5.1 Analytical solution for spheres 88

5.1.1 Expansion in Legendre polynomials 88

5.1.2 Energy calculation 93

5.2 Verification of `PyGBe`'s extension 96

5.3 Protein G B1 D4′ near a charged surface 97
5.3.1 Setup of experiments 98
5.3.2 Results for protein G B1 D4′ 100

6 Ligand orientation in biosensors 104

6.1 Results for an antibody near a charged surface 105
6.1.1 Mesh refinement study for immunoglobulin G 105
6.1.2 Probing orientation of immunoglobulin G 107

7 Response of a localized surface plasmon resonance biosensor 112

7.1 Scattering in the far-field 113
7.1.1 The scattered electric field 113
7.1.2 The extinction cross-section 118
7.1.3 The optical theorem 119
7.2 Electrostatic approximation of localized surface plasmon resonance . . 120
7.3 Boundary integral formulation 122
7.3.1 Electrostatic potential under an incoming electric field 122
7.3.2 Boundary integral expression of the dipole moment 124
7.4 Results for an isolated nanoparticle 125
7.5 LSPR sensor response calculations 127

8 Conclusions 132

References 138

Curriculum Vitae 150

List of Tables

4.1 Coulombic energies for the trypsin-BPTI complex. 78

4.2 Extrapolated solvation energies. 84

5.1 Extrapolated values of energy for Protein G B1 D4′. 100

5.2 Average orientation. 103

List of Figures

2·1 Jump condition for surface charge. 7

2·2 Surface definitions. 19

2·3 Representation of a protein in a solvent. 20

2·4 Protein with solvent-filled cavities and Stern layer. 22

2·5 Treatment of the singular integral 31

2·6 Multipole expansion in the treecode 43

2·7 Steps of the treecode 46

2·8 Sketch of a biosensor. 48

3·1 Representation of a protein with a Stern layer and two solvent-filled cavities. Gray regions contain salt, white regions don't. 54

3·2 Representation of a protein interacting with a surface with prescribed charge or potential. 58

3·3 Discretization of a molecular surface. Γ_i is the panel where the collocation point resides and Γ_j the panel being integrated. 59

3·4 Close-up view of the SES of Lysozyme discretized in flat triangles using `msms`. Colors represent potential on the surface. 60

3·5 Sketch of semi analytical integration scheme from (Zhu et al., 2001) . 63

3·6 GPU versus CPU implementations of the treecode. 69

4·1 Sketch of the Kirkwood sphere with a Stern layer and an off center charge. 73

4·2 Results for for Kirkwood sphere with Stern layer using 2048, 8192, 32768 and 131072 boundary elements per surface. 74

4·3 Results for the lysozyme test, using `PyGBe` and `APBS`. Data sets, figure files and plotting scripts available under CC-BY (Cooper et al., 2013d). 76

4·4 Results for the solvation energy of lysozyme for various `PyGBe` models. Data sets, figure files and plotting scripts available under CC-BY (Cooper et al., 2013d). 78

4·5 `APBS` and `PyGBe` results for trypsin-BPTI complex. Data sets, figure files and plotting scripts available under CC-BY (Cooper et al., 2013b). 80

4·6 `APBS` and `PyGBe` results for trypsin as receptor of trypsin-BPTI complex. Data sets, figure files and plotting scripts available under CC-BY (Cooper et al., 2013b). 82

4·7 `APBS` and `PyGBe` results for BPTI as ligand of trypsin-BPTI complex. Data sets, figure files and plotting scripts available under CC-BY (Cooper et al., 2013b). 83

4·8 `APBS` and `PyGBe` results for binding energies of the trypsin-BPTI complex. The multi-surface model of `PyGBe` considers cavities and Stern layer, and the single-surface model considers neither. Data sets, figure files and plotting scripts available under CC-BY (Cooper et al., 2013b). 84

4·9 `APBS` and `PyGBe` results for binding energies of a 22-residue long α helix peptide with "box B" hairpin structure of RNA. `PyGBe` runs used a multi-surface model (considers Stern layer), and a single-surface model. Data sets, figure file and plotting script available under CC-BY (Cooper et al., 2013a). 85

5·1 Sketch of system solved with Legendre polynomials expansions. . . . 89

5·2 Sketch of system used in the convergence study of Figure 5·3. 96

5·3 Mesh convergence study for the interaction free energy between a spherical molecule with a centered charge and a sphere with potential $\phi = 1$. 97
5·4 Structure of Protein G B1 (PDB code: 1PGB). 98
5·5 Setup of orientation experiment. 99
5·6 Mesh convergence study of the solvation and surface energy for protein G B1 D4′ mutant interacting with a surface with charge density 0.05C/m^2. 101
5·7 Orientation distribution of protein G B1 D4′. Figures 5·7a and 5·7b are the probability with respect to the tilt angle and its cosine, respectively. Figures 5·7c and 5·7d are the orientation with respect to both the tilt and rotation angle. 102
5·8 Electrostatic potential of protein G B1 D4′ for the preferred orientations according to Figure 5·7. Black arrow indicates direction of dipole moment vector. 102

6·1 Structure of immunoglobulin G (PDB code: 1IGT). 106
6·2 Mesh convergence study of the solvation plus surface energy for immunoglobulin G interacting with a surface with charge density 0.05C/m^2.107
6·3 Orientation probability and surface potential of immunoglobulin's preferred orientation on a negative surface charge. The black arrow is the dipole vector. 108
6·4 Orientation probability and surface potential of immunoglobulin's preferred orientation on a positive surface charge. The black arrow is the dipole vector. 109

7·1 Sketch of a LSPR biosensor 113
7·2 Nanoparticle subject to an electromagnetic wave. 114

7·3 Spherical nanoparticle subject to a constant electric field. 126

7·4 Convergence of extinction cross-section of single silver nanoparticle. . 126

7·5 Extinction cross-section with wavelength for a 20nm silver sphere immersed in water. 127

7·6 Extinction cross-section with wavelength for a 20nm gold sphere immersed in water. 128

7·7 Setup for LSPR response calculations. Two spherical proteins are not represented in this figure: they are also placed a distance d from the silver surface, on the axis going into the paper. 129

7·8 Convergence of the extinction cross-section of a 20nm silver sphere with six spherical proteins placed 1nm away from the silver surface, versus the average number of elements per square nanometer. This setup is sketched in Figure 7·7. 130

7·9 Extinction-cross section with wavelength for a 20nm silver sphere with six spherical proteins placed 1, 2 and 4nm away from the silver surface, and at infinity. This setup is sketched in Figure 7·7. 131

List of symbols

$\mathbf{A}$	Magnetic vector potential
A_D	Area of detector in LSPR biosensor
$a^{\mathbf{k}}$	Coefficients of Taylor expansion of order $\mathbf{k} = (k_1, k_2, k_3)$
a_n, b_n, c_n	Coefficients of expansion in Legendre polynomials
$\mathbf{B}$	Magnetic field
c	Concentration
c_0	Mean concentration
C_{abs}	Absorption cross-section
C_{ext}	Extinction cross-section
C_{sca}	Scatter cross-section
$\mathbf{D}$	Electric displacement
$\mathbf{d}$	Distance vector between two point charges
$\mathbf{E}$	Electric field
E	Energy
e^+	Positive charge of an electron
e^-	Negative charge of an electron
$\mathbf{e}_l$	Unit vector in the l direction
F_{Coulomb}	Coulomb energy
F_{GB}	Solvation free energy with Generalized-Born model
$F_{\text{interaction}}$	Interaction free energy
F_{solve}	Solvation free energy
F_{surface}	Surface free energy
F_{total}	Total free energy
g	Radial distribution function
G_L	Free space Green's function of the Laplace equation
G_Y	Free space Green's function of the linearized Poisson-Boltzmann equation
$\mathbf{H}$	Magnetizing field
H^s	Sobolev space for $p = 2$, with real order s
I_i	Incoming intensity
i_m	Modified spherical Bessel function of the first kind
$\mathbf{J}_f$	Free current density
k	Wave number
k_B	Boltzmann constant

$K_L^{\mathbf{r}}$	Double-layer potential of the Laplace potential, evaluated at $\mathbf{r}$
$K_{L,\text{disc}}^{\mathbf{r}}$	Discretized double-layer potential of the Laplace potential at $\mathbf{r}$
k_n	Modified spherical Bessel function of the second kind
$K_Y^{\mathbf{r}}$	Double-layer potential of the Yukawa potential, evaluated at $\mathbf{r}$
$K_{Y,\text{disc}}^{\mathbf{r}}$	Discretized double-layer potential of the Yukawa potential at $\mathbf{r}$
$K_{Y,\text{disc}}^{\mathbf{r}}$	Discretized double-layer potential of the Yukawa potential at $\mathbf{r}$
l	Line
$\mathbf{l}$	Unit vector tangent to l
L_p	Space of functions with integrable p^{th} power of the absolute value
m	Mass
$m_c^{\mathbf{k}}$	Multipole expansions
$\mathbf{n}$	Normal unit vector
n	Refraction index
$\mathbb{P}$	Preconditioner matrix
$\mathbf{P}$	Electric polarization vector
P	Order of Taylor expansion
$\mathbf{p}$	Dipole moment vector
p_m	Momentum
p	Observed order of convergence
P_n	n^{th} degree Legendre polynomial
$P(\lambda)$	Probability of microstate λ
q_i	Strength of point charge i
$\mathbf{r}$	Position vector
r_{vdW}	van der Waals radius
r_W	Radius of water molecule ($\sim 1.4\mathring{A}$)
T	Temperature
$\mathbf{t}$	Unit vector tangent to surface
U	Potential energy
$V_L^{\mathbf{r}}$	Single-layer potential of the Laplace potential, evaluated at $\mathbf{r}$
$V_{L,\text{disc}}^{\mathbf{r}}$	Discretized single-layer potential of the Laplace potential at $\mathbf{r}$
$V_Y^{\mathbf{r}}$	Single-layer potential of the Yukawa potential, evaluated at $\mathbf{r}$
$V_{Y,\text{disc}}^{\mathbf{r}}$	Discretized single-layer potential of the Yukawa potential at $\mathbf{r}$
W	Work, energy
w	Potential of mean force
W_{abs}	Absorbed energy
W_{ext}	Extinct energy
W_p^r	Sobolev space of integer order r, based on L_p
W_p^s	Sobolev space of real order s, based on L_p
W_{sca}	Scattered energy
W_d	Energy measured by detector in LSPR biosensor
Y_{lm}	Spherical harmonic function of degree l and order m
Z_N	Configuration integral

α_{rot}	Rotation angle of protein adsorbed on surface
α_{tilt}	Tilt angle of protein adsorbed on surface
Γ	Surface
$\mathbf{\Gamma}$	Gamma function
γ_0	Dirichlet trace operator
γ_1	Neumann trace operator
ϵ	Relative permittivity
ϵ_0	Vacuum permittivity
θ	Multipole-acceptance criterion
κ	Debye length
μ	Relative permeability
ρ	Volume charge density
ρ_b	Bound or polarization charge
ρ_f	Free volume charge density
ϕ	Electrostatic potential
σ	Surface charge density
χ_e	Electric susceptibility
Ω	Volume

List of Acronyms and Abbreviations

ACA	Adaptive-cross approximation
AFMPB	Adaptive Fast Multipole Poisson-Boltzmann Solver
APBS	Adaptive Poisson Boltzmann Solver
BEM	Boundary Element Method
BIE	Boundary integral equation
CPU	Central processing unit
CUDA	Computer Unified Device Architecture
FET	Field-effect transistor
FFT	Fast Fourier transform
FFTSVD	Fast Fourier transform singular value decomposition
pFFT	Precorrected fast Fourier transform
FMM	Fast Multipole Method
GMRES	Generalized minimal residual
GPU	Graphics processing unit
L2L	Local-to-Local
L2P	Local-to-Particle
LSPR	Localized surface plasmon resonance
M2M	Multipole-to-Multipole
M2P	Multipole-to-Particle
MAC	Multipole-acceptance criterion
MD	Molecular dynamics
MEAD	Macroscopic Electrostatics with Atomic Detail
MIBPB	Matched interface boundary Poisson-Boltzmann
P2M	Particle-to-Multipole
P2P	Particle-to-Particle
PDE	Partial-differential equation
SAM	Self-assembled monolayer
SAS	Solvent-accessible surface
SES	Solvent-excluded surface
TABI	Treecode-Accelerated Boundary Integral

Chapter 1

Introduction

Electrostatics plays a major role in many biomolecular processes, like solvation, binding and adsorption. In biological settings, these occur in an aqueous environment with salt, and in any analysis, it is key to account for the solvent appropriately.

Molecular dynamics (MD) is perhaps the most popular way to analyze these systems. MD considers all water molecules explicitly, however, there are some situations where these simulations may take too long. Current cutting-edge MD codes are capable of simulating on the order of a few micro seconds a day, which is very good for some applications, like folding of small proteins, but they are not well suited to study equilibrium parameters. Continuum electrostatics with implicit-solvent models offer a different approach: they separate the domain into a solvent and protein region, and apply the equations of electrostatics in a dielectric. We use an implicit-solvent model that solves the Poisson equation of Gauss's law in the protein region, which contains point-charge like atoms. In the solvent region, the presence of salt transforms Gauss's law into the Poisson-Boltzmann equation, which interfaces with the Poisson equation at the molecular surface.

The coupled Poisson and Poisson-Boltzmann equation system is of interest in the biophysics community, and it is solved numerically in many widely-used software packages, with notable examples in `Delphi` (Gilson et al., 1987) and `APBS` (Baker et al., 2001). Even though volumetric methods, like finite-element or finite-difference, are more popular, boundary integral methods have successfully been used in this application (Lu et al., 2006; Altman et al., 2009; Geng and Krasny, 2013).

In this work, we simulate biomolecules in salty water with the implicit-solvent model using a boundary integral approach. Our implementation of this model is called `PyGBe`, [1] a Python code that interfaces with GPUs in the most computationally intensive parts with PYCUDA. `PyGBe` is open and released under the MIT license.

One main motivation of this work was to apply the implicit-solvent model for biosensing applications. Biosensors are nanoscale devices that are designed to detect biomolecules. They consist of a sensing element, such as a nanoparticle, functionalized with a self-assembled monolayer of a small charged group, to which a ligand molecule attaches. The ligand molecule, in many cases an antibody, captures the target molecule because they have high binding specificity. Computer simulations are rarely used in the biosensing community, where experimental observations guide the progress, mostly by trial and error. This work intends to fill this gap, providing experimentalists with computational tools to guide their research.

We recognized two areas in which electrostatic modeling was relevant in biosensing applications:

> *Ligand molecule orientation.* For a biosensor to work well, the binding sites of the ligand molecule must be accessible by the target molecule. Then, the orientation at which the ligand molecule adsorbs has a large effect on the sensitivity of the biosensor (Trilling et al., 2013b), and it is not easy to control. Ligand molecule adsorption is dominated by electrostatic effects, making our approach suitable to study it.
>
> *Localized surface plasmon resonance (LSPR) biosensor response.* LSPR biosensors measure the shift of plasmon resonance frequency in metallic nanoparticles as the target molecule binds to it. The distance between the sensor and the

---

[1] `https://github.com/barbagroup/pygbe`

target molecule has a big impact on the sensitivity, but this relation is not well understood. Even though this is an optic phenomenon, we can study this problem using electrostatics because it is a good approximation in the long-wavelength limit.

The objectives of this thesis are:

- Generate a desktop-level tool for biomolecular electrostatics calculations, that is open-source, easy to use, and efficient.
- Review the use of BEM in biomolecular applications, and analyze when it is an appropriate tool.
- Discuss the impact of solvent-filled cavities and Stern layers in the implicit-solvent model.
- Verify and validate the electrostatic approach for protein-surface interaction calculations.
- Study the orientation of adsorbed ligand molecules in biosensors.
- Investigate the response of LSPR biosensors using an electrostatic approach.

After the introduction, Chapter 2 reviews some background in terms of the model, algorithms, and applications. Details of how the background applies to this work, and the specific methods that we used, are included in Chapter 3. The first results appear in Chapter 4, where we verify and validate our code implementation, and discuss some details of the model. Next, Chapter 5 extends the model to account for charged surfaces, and Chapter 6 uses this extension to study the orientation of ligand molecules in biosensors. Then, Chapter 7 applies the electrostatic BEM approach to LSPR biosensor response. Finally, Chapter 8 wraps up with the conclusions and possible research avenues for the future.

Chapter 2

Background

2.1 Implicit solvent model

In biological settings, proteins live in an aqueous environment which determines their functionality, and it is crucial to include the water (solvent) effect in any model. The implicit solvent model uses electrostatics in a continuum dielectric medium to represent the aqueous environment, and solves for the mean-field electrostatic potential. In this section, we will briefly review electrostatic theory (Jackson, 1998; Griffiths, 1999), to then show how it is applied to molecular systems.

2.1.1 Electrostatics in vacuum

The electric field and electrostatic potential

To describe electrostatic theory, it is useful to introduce the concept of an electric field ($\mathbf{E}$), which is the vector field given by the force per unit charge acting at a given point. If we were able to put a test charge q that is small enough to not perturb the rest of the field, it would be subjected to a force $\mathbf{F} = q\mathbf{E}$. In Maxwell's equations, the electric field is related to the charge density distribution (ρ) by Gauss's law

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0}, \tag{2.1}$$

where ϵ_0 is the permittivity of vacuum. Gauss's law is commonly presented in its integral form, which we can obtain by taking the integral on both sides of Equation (2.1) over the volume Ω, and applying the divergence theorem on the left-hand side term, yielding

$$\oint_{\Gamma} \mathbf{E} \cdot \mathbf{n} \mathrm{d}\Gamma = \int_{\Omega} \frac{\rho}{\epsilon_0} \mathrm{d}\Omega. \tag{2.2}$$

where Γ is the surface that encloses Ω, and $\mathbf{n}$ is a unit vector normal to Γ. The term on the left side of Equation (2.2) is known as the electric flux.

Electrostatics does not consider the contribution of the magnetic field in Maxwell's equations, therefore, the electric field is irrotational

$$\nabla \times \mathbf{E} = 0, \tag{2.3}$$

which indicates that there is a scalar potential ϕ such that

$$-\nabla\phi = \mathbf{E}. \tag{2.4}$$

Using Equation (2.1) and Equation (2.4), we arrive at the well known Poisson equation of electrostatics

$$\nabla^2\phi = -\frac{\rho}{\epsilon_0}, \tag{2.5}$$

where ϕ is the electrostatic potential.

If the charge distribution is described by a single point charge ($\rho = q\delta(\mathbf{r}')$), the solution to Equation (2.5) yields Coulomb's law

$$\phi(\mathbf{r}) = \frac{q}{4\pi\epsilon_0 |\mathbf{r} - \mathbf{r}'|}, \tag{2.6}$$

which can be extended to a general charge distribution with

$$\phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \int_{\Omega} \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \mathrm{d}\Omega', \tag{2.7}$$

Jump conditions in electrostatics

It is common in electrostatics to find charges confined on a surface. Let us consider the system sketched in Figure 2·1, where $\mathbf{E}_1$ and $\mathbf{E}_2$ are the electric fields evaluated just inside and outside, respectively, of a surface Γ that has a charge density $\sigma(\mathbf{r})$. If we evaluate $\mathbf{E}_1$ and $\mathbf{E}_2$ with Equation (2.2) the only difference in the charge enclosed by those surfaces is given by $\sigma(\mathbf{r})$, hence, subtracting these two evaluations yields the jump condition for the normal component of the electric field across Γ:

$$(\mathbf{E}_2 - \mathbf{E}_1) \cdot \mathbf{n} = \frac{\sigma}{\epsilon_0}. \tag{2.8}$$

Since $\mathbf{E}$ is an irrotational field, the line integral along l depends only on the electrostatic potential at the end points

$$\int_A^B \mathbf{E} \cdot \mathbf{l} \mathrm{d}l = -(\phi(\mathbf{r}_B) - \phi(\mathbf{r}_A)), \tag{2.9}$$

where $\mathbf{l}$ is a unit vector tangent to l. Then, if l is a closed curve, the integral in Equation (2.9) is 0.

Consider there is a closed curve l like the one in Figure 2·1 that contains four parts: the first part is parallel to $\mathbf{n}$ and crosses Γ from inside to outside, the second one goes parallel to the surface on the outside, the third section crosses back inside, normal to the surface, and the last part goes parallel to the surface back to the origin. Equation (2.9) tells us that the line integral of $\mathbf{E}$ along l is zero, and we know that the integral on the sections that cross Γ are equal to the jump condition in Equation (2.8), with opposite signs. Then, the integrals along the sides that are parallel to Γ must add to zero, which happens only if the tangential component of $\mathbf{E}$ is continuous across Γ, ie.

$$\mathbf{E}_1 \cdot \mathbf{t} = \mathbf{E}_2 \cdot \mathbf{t}, \tag{2.10}$$

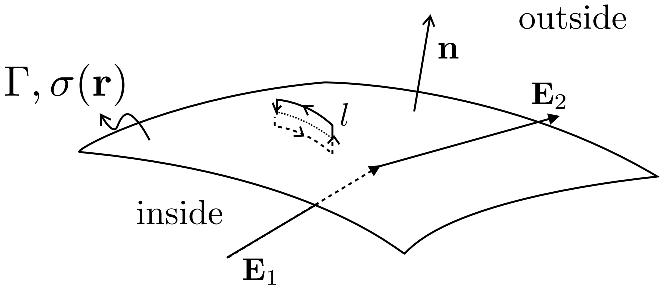


Figure 2·1: Jump condition for surface charge.

where $\mathbf{t}$ is a vector parallel to Γ.

Even though a surface charge generates a discontinuity in the electric field, it is bounded, hence the electrostatic potential is continuous throughout.

Electrostatic potential energy

The work done in moving a charge between two points A and B is given by

$$\begin{aligned} W = -\int_A^B \mathbf{F} \cdot \mathbf{l} \mathrm{d}l &= -q \int_A^B \mathbf{E} \cdot \mathbf{l} \mathrm{d}l \\ &= q \int_A^B \nabla\phi \cdot \mathbf{l} \mathrm{d}l \\ &= q \int_A^B \mathrm{d}\phi = q(\phi(\mathbf{r}_B) - \phi(\mathbf{r}_A)), \end{aligned} \tag{2.11}$$

where $\mathbf{l}$ is a vector tangent to l. Then, the work of bringing charge i from infinity, where the electrostatic potential decays to zero, to $\mathbf{r}_i$, is

$$W_i = q_i \phi(\mathbf{r}_i). \tag{2.12}$$

We can generalize the result from Equation (2.12) to obtain the electrostatic potential energy for any charge distribution (ρ),

$$W = \frac{1}{2} \int\limits_{\Omega} \rho(\mathbf{r}) \phi(\mathbf{r}) \mathrm{d}\Omega. \tag{2.13}$$

The 1/2 term in Equation (2.13) avoids double counting interactions. For example, if the charge distribution were two point charges, the electrostatic potential energy is given by $q_1 q_2 / 4\pi\epsilon_0$ and not $q_1 q_2 / 4\pi\epsilon_0 + q_2 q_1 / 4\pi\epsilon_0$.

The multipole expansion

The electrostatic potential can be analyzed using multipole expansions, usually with spherical harmonics

$$\phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \sum_{l=0}^{\infty} \sum_{m=-l}^{l} \frac{4\pi}{2l+1} q_{lm} \frac{Y_{lm}(\theta, \varphi)}{|\mathbf{r}|^{l+1}}, \tag{2.14}$$

where

$$q_{lm} = \int\limits_{\Omega} Y^{\star}_{lm}(\theta', \varphi') |\mathbf{r}'|^{l} \rho(\mathbf{r}') \mathrm{d}\Omega' \tag{2.15}$$

are the multipole moments, Y_{lm} is the spherical harmonic function of degree l and order m, and $Y^{\star}_{lm}$ is the complex conjugate of Y_{lm}. This analysis is useful to see, for example, how much a charge distribution resembles a point charge, given by the importance of the monopole moment (q_{00}).

Another popular distribution is the dipole moment, which appears in Equation (2.14) for l=1,

$$\mathbf{p} = \int\limits_{\Omega} \mathbf{r}' \rho(\mathbf{r}') \mathrm{d}\Omega'. \tag{2.16}$$

The simplest system with a dipole consists of two opposite-sign point charges that are close together. In this case, the dipole moment is

$$\mathbf{p} = q\mathbf{d}, \tag{2.17}$$

where $\mathbf{d}$ is the position vector between the two charges. If $|\mathbf{d}|$ is small enough, the dipole moment will dominate in the spherical harmonic expansion, and we can represent the system as a point dipole. The electrostatic potential of two infinitesimally close charges with opposite signs, centered at $\mathbf{r}'$ is

$$\phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \frac{\mathbf{p} \cdot (\mathbf{r} - \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|^3}. \tag{2.18}$$

In the presence of an external field, electrostatic forces will try to orient the dipole moment parallel to the electric field.

2.1.2 Electrostatics in a continuum dielectric medium

Polarizability and electric displacement

The theory outlined by Section 2.1.1 is useful when there is vacuum between charges. This happens, for example, in molecular dynamics (MD) simulations, where molecules are modeled with point charges and dipoles in vacuum. To use this theory in macroscopic settings, we need to account for every molecule in the system to obtain a correct representation, which quickly becomes very expensive to compute. This is a motivation to develop a macroscopic theory of electrostatics, where the electric field acts on a dielectric material, which approximates microscopic effects of molecules and atoms of the medium.

In a medium comprised of many molecules and atoms, if no electric field is applied, all multipole moments are zero. However, with an applied electric field, the dominant molecular multipole is the dipole. This produces a electric polarization $\mathbf{P}$ in the medium, which is the dipole moment density per unit volume. This concept generalizes Equation (2.18) into

$$\phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \int\limits_{\Omega} \frac{\mathbf{P}(\mathbf{r}') \cdot (\mathbf{r} - \mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|^3} \mathrm{d}\Omega', \tag{2.19}$$

which can be rewritten as

$$\phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \int\limits_{\Omega} \mathbf{P}(\mathbf{r}') \cdot \nabla' \left[\frac{1}{|\mathbf{r} - \mathbf{r}'|} \right] \mathrm{d}\Omega'. \tag{2.20}$$

We can compute the electrostatic potential from a macroscopic point of view by superimposing the potential due to a charge distribution (Equation (2.7)), and the potential due to a distribution of dipoles in the medium (Equation (2.20)), which gives

$$\phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \int\limits_{\Omega} \left(\frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} + \mathbf{P}(\mathbf{r}') \cdot \nabla' \left[\frac{1}{|\mathbf{r} - \mathbf{r}'|} \right] \right) \mathrm{d}\Omega'. \tag{2.21}$$

Using integration by parts for the second term of the integrand, and assuming the only boundary condition is that $\mathbf{P}$ is bounded as $\mathbf{r}' \to \infty$, we can rewrite Equation (2.21) as

$$\phi(\mathbf{r}) = \frac{1}{4\pi\epsilon_0} \int\limits_{\Omega} \left(\frac{1}{|\mathbf{r} - \mathbf{r}'|} \left[\rho(\mathbf{r}') - \nabla' \cdot \mathbf{P}(\mathbf{r}') \right] \right) \mathrm{d}\Omega', \tag{2.22}$$

Equation (2.22) is identical to Equation (2.7) for a charge distribution $\rho - \nabla \cdot \mathbf{P}$, where the term $-\nabla \cdot \mathbf{P}$ is sometimes referred to as bound or polarization charge (ρ_b). In this context, Gauss's law would look like

$$\nabla \cdot \mathbf{E} = \frac{1}{\epsilon_0} \left[\rho - \nabla \cdot \mathbf{P} \right], \tag{2.23}$$

motivating the definition of the electric displacement $\mathbf{D}$,

$$\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P}. \tag{2.24}$$

Using $\mathbf{D}$, we can rewrite Equation (2.23) as

$$\nabla \cdot \mathbf{D} = \rho, \tag{2.25}$$

which is the macroscopic version of Gauss's law. Equation (2.25) has the integral form

$$\oint_{\Gamma} \mathbf{D} \cdot \mathbf{n} \mathrm{d}\Gamma = \int_{\Omega} \rho \mathrm{d}\Omega. \tag{2.26}$$

Linear dielectric

We will assume an isotropic dielectric material that reacts linearly to an applied electric field,

$$\mathbf{P} = \epsilon_0 \chi_e \mathbf{E}, \tag{2.27}$$

where χ_e is the electric susceptibility of the medium. This linear constitutive relation is widely used, but fails when the electric field becomes too large. We can rewrite Equation (2.24) as

$$\mathbf{D} = \epsilon \mathbf{E}, \tag{2.28}$$

where

$$\epsilon = \epsilon_0 (1 + \chi_e), \tag{2.29}$$

is the dielectric constant of the medium. Using Equation (2.29) and Equation (2.4), we obtain the macroscopic version of the Poisson equation of electrostatics

$$\nabla^2 \phi = -\nabla \cdot \mathbf{E} = -\frac{\rho}{\epsilon}. \tag{2.30}$$

Surface discontinuities in dielectric media

In electrostatics, many times the domain contains more than one dielectric medium. We can derive the conditions at the interface of two dielectric materials starting from Equation (2.26) and using the same reasoning that led to Equation (2.8) and Equation (2.10). This leads to

$$\begin{aligned} &(\mathbf{D}_2 - \mathbf{D}_1) \cdot \mathbf{n} = \sigma \quad \text{and,} \\ &\mathbf{E}_1 \cdot \mathbf{t} = \mathbf{E}_2 \cdot \mathbf{t}. \end{aligned} \tag{2.31}$$

In Equation (2.31), the surface charge σ does not include the polarization surface charge.

Energy calculation in dielectric media

The energy calculation in Equation (2.13) is valid in vacuum, but for dielectrics, we need to account for the energy used to polarize the medium. The work done by an infinitesimal change in the macroscopic charge distribution, that does not affect the electrostatic potential, is

$$\delta W = \int_{\Omega} \delta\rho(\mathbf{r})\phi(\mathbf{r})\mathrm{d}\Omega. \tag{2.32}$$

Using Equation (2.25), we realize that $\delta\rho = \nabla \cdot (\delta\mathbf{D})$, which allows us to rewrite Equation (2.32) as

$$\delta W = \int_{\Omega} \delta\mathbf{D} \cdot \mathbf{E}\mathrm{d}\Omega, \tag{2.33}$$

after integration by parts with vanishing boundary integrals, and using $\mathbf{E} = -\nabla\phi$. The total energy, is then given by the integration of Equation (2.33) as,

$$W = \int \delta W = \int \int_{\Omega} \delta\mathbf{D} \cdot \mathbf{E}\mathrm{d}\Omega. \tag{2.34}$$

So far, to arrive at Equation (2.34), we have not used any information from the material properties. However, to continue further we need to use the linear constitutive law (Equation (2.28)), which allows us to write

$$\delta\mathbf{D} \cdot \mathbf{E} = \frac{1}{2}\delta(\mathbf{D} \cdot \mathbf{E}). \tag{2.35}$$

Using Equation (2.35) in Equation (2.34) yields the following expression for energy

$$W = \frac{1}{2}\int_{\Omega} \mathbf{E} \cdot \mathbf{D}\mathrm{d}\Omega. \tag{2.36}$$

In Equation (2.36), we can replace $\mathbf{E} = -\nabla\phi$ and $\nabla \cdot \mathbf{D} = \rho$, and then integrate by parts with vanishing conditions at the boundaries to obtain

$$W = \frac{1}{2} \int_{\Omega} \phi(\mathbf{r})\rho(\mathbf{r})\mathrm{d}\Omega, \tag{2.37}$$

which is exactly the same as Equation (2.13). This is only valid for a linear dielectric material.

2.1.3 The effect of an electrolyte solution

Physiological fluid consists of water and salt. When dissolved, salt produces ions that contribute to the electric field and are free to move in the water. An applied field will affect the spatial distribution of salt ions, making the charge density distribution from Equation (2.30) dependent on the electrostatic potential ($\rho = \rho(\phi)$). Hill outlined the mathematical analysis of this system (Hill, 1960), which yields the Poisson-Boltzmann equation. Next, we will review this derivation.

The radial distribution function

Let us consider N particles that are free to move in a continuum medium with volume V. If we place the origin at a specific particle i, we can write the time-averaged number density distribution or concentration ($c(\mathbf{r})$), as

$$c(\mathbf{r}) = \frac{N}{V} g(r) = c_0 g(r). \tag{2.38}$$

The radial distribution function ($g(r)$) is the deviation of the number density from the mean (c_0), a distance r from particle i. This is a convenient way to represent number density because the time-averaged distribution will tend to be spherically symmetric with respect to the reference particle i.

The number of particles in a volume $\mathrm{d}V_\alpha$ located at $\mathbf{r}_\alpha$ is

$$N_\alpha = c(\mathbf{r}_\alpha)\mathrm{d}V_\alpha, \tag{2.39}$$

therefore, N_α/N represents the probability of a particle to be in $\mathrm{d}V_\alpha$. To find the probability that any particle is in $\mathrm{d}V_\alpha$, we need to sample all N particles, each one with probability N_α/N, leaving

$$N\frac{N_\alpha}{N} = N\frac{c(\mathbf{r}_\alpha)\mathrm{d}V_\alpha}{N} = c_0 g(|\mathbf{r}_\alpha|)\mathrm{d}V_\alpha. \tag{2.40}$$

Let us now consider that we have two species, α and β, in a continuum medium. The probability of finding a particle α in $\mathrm{d}V_\alpha$ is $c_\alpha \mathrm{d}V_\alpha$. On the other hand, the probability to find a particle β in $\mathrm{d}V_\beta$, a distance $r_{\alpha\beta}$ from the α particle, is $c_\beta g_{\alpha\beta}(r_{\alpha\beta})\mathrm{d}V_\beta$. Then, the probability to find α in $\mathrm{d}V_\alpha$ and β in $\mathrm{d}V_\beta$ is the multiplication of both probabilities

$$c_\alpha c_\beta g_{\alpha\beta}(r_{\alpha\beta})\mathrm{d}V_\alpha \mathrm{d}V_\beta. \tag{2.41}$$

The argument that led to Equation (2.41) can be generalized to any number of species as

$$c_\alpha c_\beta c_\gamma \ldots g_{\alpha\beta\gamma\ldots}(r_{\alpha\beta}, r_{\alpha\gamma}, \ldots)\mathrm{d}V_\alpha \mathrm{d}V_\beta \mathrm{d}V_\gamma \ldots. \tag{2.42}$$

If we consider that α, β, γ, $\ldots$, are particles of the same species, Equation (2.42) becomes

$$c^n g_n(r_{12}, r_{13}, \ldots, r_{1n})\mathrm{d}V_1 \mathrm{d}V_2 \ldots \mathrm{d}V_n, \tag{2.43}$$

and corresponds to the probability of finding particle 1 in $\mathrm{d}V_1$, particle 2 in $\mathrm{d}V_2$ (r_{12} from particle 1), particle 3 in $\mathrm{d}V_3$ (r_{13} from particle 1), $\ldots$, and particle n in $\mathrm{d}V_n$ (r_{1n} from particle 1).

The probability to find a system of N particles with energy $E = p_m^2/2m + U$ is given by the Maxwell-Boltzmann distribution

$$P(E) = \frac{e^{-E/k_BT}\mathrm{d}q\mathrm{d}p}{\int e^{-E/k_BT}\mathrm{d}q\mathrm{d}p}, \tag{2.44}$$

where p_m is the momentum, q the spatial configuration, U the potential energy, k_B the Boltzmann constant and T the temperature. If we integrate over momentum and replace dq by the position of each particle dV_1dV_2d$V_3 \ldots$dV_N, we get

$$P(U) = \frac{e^{-U/k_BT}\mathrm{d}V_1\mathrm{d}V_2\ldots\mathrm{d}V_N}{Z_N}, \tag{2.45}$$

where Z_N is the configuration integral

$$Z_N = \int e^{-U/k_BT}\mathrm{d}V_1\mathrm{d}V_2\ldots\mathrm{d}V_N \tag{2.46}$$

over all possible configurations.

The probability of finding particles 1 through n in dV_1 to dV_n, regardless of the position of the remaining $N-n$ particles is

$$\frac{\mathrm{d}V_1\mathrm{d}V_2\ldots\mathrm{d}V_n\int e^{-U/k_BT}\mathrm{d}V_{n+1}\mathrm{d}V_{n+2}\ldots\mathrm{d}V_N}{Z_N}, \tag{2.47}$$

however, given that particles are indistinguishable —any particle can be 1, 2, …, n—, then we need to consider all permutations

$$P(U) = \frac{N!}{(N-n)!}\frac{\mathrm{d}V_1\mathrm{d}V_2\ldots\mathrm{d}V_n\int e^{-U/k_BT}\mathrm{d}V_{n+1}\mathrm{d}V_{n+2}\ldots\mathrm{d}V_N}{Z_N}. \tag{2.48}$$

Equation (2.48) is the probability to find particle 1 in dV_1, particle 2 in dV_2, … and particle n in dV_n. This is equivalent to Equation (2.43), and equating them gives

$$\rho^n g_n(r_{12}, r_{13}, \ldots, r_{1n}) = \frac{N!}{(N-n)!}\frac{\int e^{-U/k_BT}\mathrm{d}V_{n+1}\mathrm{d}V_{n+2}\ldots\mathrm{d}V_N}{Z_N}. \tag{2.49}$$

For n small, we can use the approximation

$$\frac{N!}{c^n(N-n)!} = V^n\left[1+\mathcal{O}\left(\frac{1}{N}\right)\right] \tag{2.50}$$

to obtain an expression for the radial distribution function

$$g_n(r_{12}, r_{13}, \ldots, r_{1n}) = \frac{V^n \int e^{-U/k_BT}\mathrm{d}V_{n+1}\mathrm{d}V_{n+2}\ldots\mathrm{d}V_N}{Z_N}. \tag{2.51}$$

It is convenient to define a function w_n such that

$$g_n = e^{-w_n/k_BT}, \tag{2.52}$$

which applied in Equation (2.51) gives

$$e^{-w_n/k_BT} = \frac{V^n \int e^{-U/k_BT}\mathrm{d}V_{n+1}\mathrm{d}V_{n+2}\ldots\mathrm{d}V_N}{Z_N}. \tag{2.53}$$

The physical meaning of w_n becomes evident when we take the logarithm of Equation (2.53) and derive it with respect to the position of particle i, where $1 \leq i \leq n$, leading to

$$-\nabla_i w_n = \frac{-\int e^{-U/k_BT}(\nabla_i U)\mathrm{d}V_{n+1}\mathrm{d}V_{n+2}\ldots\mathrm{d}V_N}{\int e^{-U/k_BT}\mathrm{d}V_{n+1}\mathrm{d}V_{n+2}\ldots\mathrm{d}V_N}, \tag{2.54}$$

which is the mean force on particle i. Equation (2.54) reveals that the function w_n is the potential of the mean force on particle i.

The Poisson-Boltzmann equation

We can use the radial distribution function to describe the distribution of salt ions in a solvent by assuming that ions are hard spheres, and that the solvent has no molecular behavior: it is a continuum even at small distances. To obtain the Poisson-Boltzmann equation, we need to consider that the force on the ions is purely electrostatic in nature, meaning the potential of mean force is

$$w = \phi q, \tag{2.55}$$

where ϕ is the electrostatic potential and q the charge of the ion. There are a few assumptions made when writing Equation (2.55). In general, ions not only feel forces due to electrostatics, but also from van der Waals and solvation effects (entropy); however, they are short range interactions, and Equation (2.55) becomes exact for highly diluted systems (ions are far apart). In Equation (2.55) we are also neglecting activity coefficients, implying an ideal ion mixture. In an ideal mixture, the interaction between each pair of chemical species are the same, making the enthalpy of mixing equal to zero. The activity coefficient is a measure of how far a mixture is from ideal.

Using the radial distribution function in Equation (2.30) for N_s species of ions, we obtain

$$\nabla^2\phi = -\frac{\rho}{\epsilon} = -\frac{1}{\epsilon}\sum_i^{N_s} q_i c_i(\mathbf{r}) = -\frac{1}{\epsilon}\sum_i^{N_s} q_i c_{0i} g_i(r) = -\frac{1}{\epsilon}\sum_i^{N_s} q_i c_{0i} e^{-\phi q_i/k_B T}, \tag{2.56}$$

where q_i and c_{0i} are the charge and mean number density of ion species i. In the case of salt —for example, sodium chloride—, there are two types of ions: the sodium ion with charge e^+ and the chlorine ion with charge e^-. These ions usually come from the same source, so it is safe to assume the same mean number density c_0. In this specific case, Equation (2.56) becomes

$$\begin{aligned}\nabla^2\phi &= -\frac{n_0}{\epsilon}\left(e^+ e^{-\phi e^+/k_B T} + e^- e^{-\phi e^-/k_B T}\right)\\ &= -\frac{e^+ n_0}{\epsilon}\left(e^{-\phi e^+/k_B T} - e^{\phi e^+/k_B T}\right) = \frac{2n_0 e^+}{\epsilon}\sinh\left(\frac{\phi e^+}{k_B T}\right).\end{aligned} \tag{2.57}$$

In cases where $\phi q << k_B T$, the first order approximation of the exponential reads

$$e^{-\phi q/k_B T} \approx 1 + \frac{\phi q}{k_B T}, \tag{2.58}$$

and we can write a linearized version of Equation (2.56)

$$\nabla^2 \phi = \frac{\phi}{\epsilon k T} \sum_i^{N_s} n_{0i} q_i = \kappa^2 \phi. \tag{2.59}$$

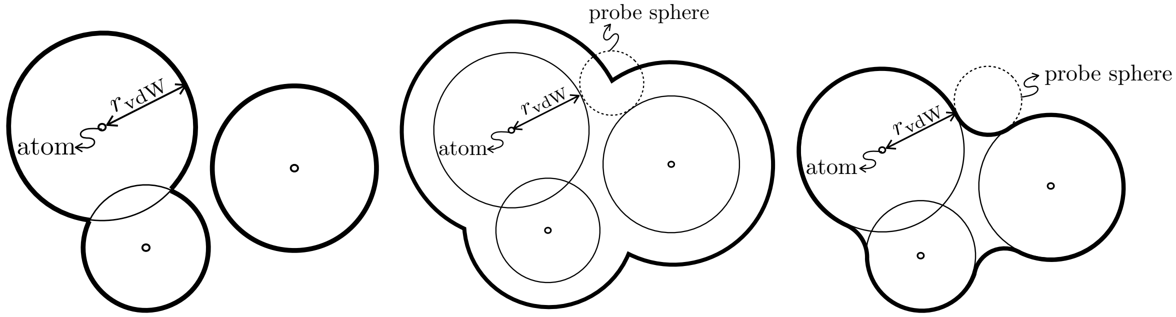
The screening parameter κ is the inverse of the Debye length, and for the salt case that led to Equation (2.57), it becomes

$$\kappa^2 = \frac{2 n_0 e^+}{\epsilon k T}. \tag{2.60}$$

As an example, normal serum salt concentration is around 150mM, which gives $\kappa = 0.125$ Å^{-1}.

2.1.4 Implicit solvent models in molecular systems

The implicit-solvent model uses continuum electrostatic theory in dielectric media to represent the solvent around a protein. With this approach, there is no need to explicitly account for each water molecule, and the solvent polarization is considered by the dielectric constant, decreasing the computational effort. This model has been used for calculations in a variety of applications, like acid dissociation constant computations, binding affinities, solvation energies, and catalysis, among others. There are several thorough reviews on this subject (Sharp and Honig, 1990; Roux and Simonson, 1999; Orozco and Luque, 2000; Simonson, 2003; Baker, 2004; Tomasi et al., 2005; Lu et al., 2008; Xu and Cai, 2011; Bardhan, 2012), showing the large amount of work that has been done using implicit-solvent models for molecular systems.



(a) van de Waals surface (b) Solvent-accessible surface (c) Solvent-excluded surface

Figure 2·2: Surface definitions.

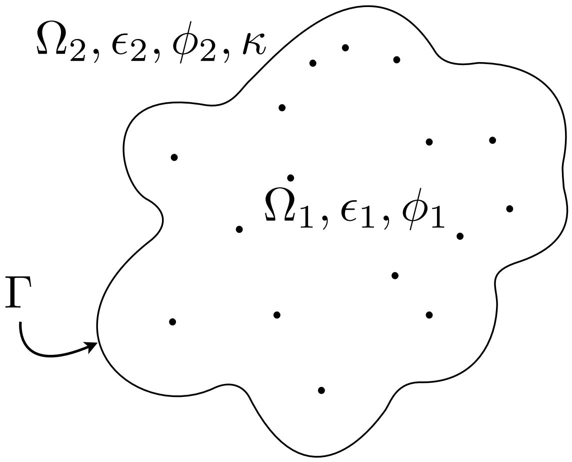
Continuum electrostatics in a biomolecular system

In a biological setting, biomolecules are generally dissolved in water with salt. An important family of biomolecules are proteins, which are chains of amino acids that are usually found tightly folded in very specific ways. Inside the protein there is a charge distribution from the pH-dependent charge of amino acids and partial charges from chemical bonds. Fortunately, the crystal structure information of many folded proteins of interest are freely available in the Protein Data Bank (PDB),[1] and force field models can compute the van der Waals radii and charge at each atom location.

One way to represent the protein-solvent system is by considering the solvent as a dielectric continuum. In the context of electrostatics, we can identify two dielectric regions in a dissolved protein: a solvent and a protein region. The solvent region consists of water and salt, it has the dielectric constant of water ($\epsilon \approx 80$), and salt ions are considered by the Poisson-Boltzmann equation, in our case, in the linearized form of Equation (2.59). The protein region has a much lower dielectric constant ($\epsilon \approx 2-4$) because the reorientation of its dipolar groups is much more restricted (Gilson and Honig, 1986; Gilson and Honig, 1987), and contains a charge distribution, usually modeled as point charges at the atoms' locations.

---

[1] `http://www.rcsb.org/`



$\Omega_2, \epsilon_2, \phi_2, \kappa$
$\Omega_1, \epsilon_1, \phi_1$
Γ

Figure 2·3: Representation of a protein in a solvent.

Figure 2·2 outlines three commonly used interface definitions, all of which depend on the van der Waals radii of the atom, computed from a force field. The van der Waals surface of Figure 2·2a considers the protein region to be anything within the atomic radius of any atom. This definition is used in Generalized Born methods, where the protein is approximated as a set of spheres. Figure 2·2b shows the solvent-accessible surface (SAS), which is generated by tracing the center of a spherical probe of the size of a water molecule (∼1.4Å), as it rolls around the protein. Finally, the solvent-excluded surface (SES) sketched in Figure 2·2c defines the closest a water molecule can get to the protein. In this work, we use the SES, computed using the freely available software `MSMS`[2] (Sanner et al., 1995).

There is another well-known molecular surface not shown in Figure 2·2 called the Gaussian surface. This surface is implicitly defined by

$$F(\mathbf{r}) = \sum_{i=1}^{Nq} e^{-|\mathbf{r}-\mathbf{r}_i|/r_W r_{vdW,i}^2} - e^{-1/r_W} \tag{2.61}$$

where $\mathbf{r}$ is the surface, $\mathbf{r}_i$ the location of atom i, $r_{vdW,i}$ the van de Waals radii of atom i and r_W the size of the spherical probe, or water molecule.

---

[2] `http://mgl.scripps.edu/people/sanner/html/msms_home.html`

Figure 2·3 sketches a dissolved protein. The protein region (Ω_1) has permittivity ϵ_1, point charges at the locations of the atoms, and the electrostatic potential is represented by ϕ_1. On the other hand, in the solvent region (Ω_2) the electrostatic potential is ϕ_2, the permittivity ϵ_2, and there are salt ions that produce a Debye length κ^{-1}. On the interface Γ, in this case the SES, two conditions are met. The first one is that the electrostatic potential must be continuous, because, as outlined by Section 2.1.1, the electric field must be bounded. Secondly, given that there are no explicit charges on the SES, Equation (2.31) indicates that the electric displacement must be continuous. Mathematically, this can be written as

$$\begin{aligned}
\nabla^2\phi_1(\mathbf{r}) &= -\sum_k \frac{q_k}{\epsilon_1}\delta(\mathbf{r},\mathbf{r}_k) && \text{in solute } (\Omega_2)\\
\nabla^2\phi_2(\mathbf{r}) &= \kappa^2\phi_2(\mathbf{r}) && \text{in solvent } (\Omega_1)\\
\phi_1 &= \phi_2 && \text{on interface, } \Gamma\\
\epsilon_1\frac{\partial\phi_1}{\partial\mathbf{n}} &= \epsilon_2\frac{\partial\phi_2}{\partial\mathbf{n}}. &&
\end{aligned}\tag{2.62}$$

Solvent-filled cavities and Stern layers. Two important features that the implicit-solvent model is capable of resolving are solvent-filled cavities and Stern layers. Even though proteins fold very tightly, solvent can be trapped inside it in pockets or cavities. In the context of the implicit-solvent model, cavities are regions with high dielectric constant inside the protein. A continuum treatment is not ideal when the cavity is small and can only accommodate a few water molecules, but this work does not consider a special treatment of such cases.

The continuum treatment of ions in the Poisson-Boltzmann theory does not allow it to consider steric effects, which leads to an overestimation of ions near charged surfaces. The Stern (Stern, 1924) or ion-exclusion layer is a thin (∼2Å), ion-free region around the SES, that tries to account for this effect. The common practice is to use the dielectric constant of the solvent in the Stern layer, however, recent work

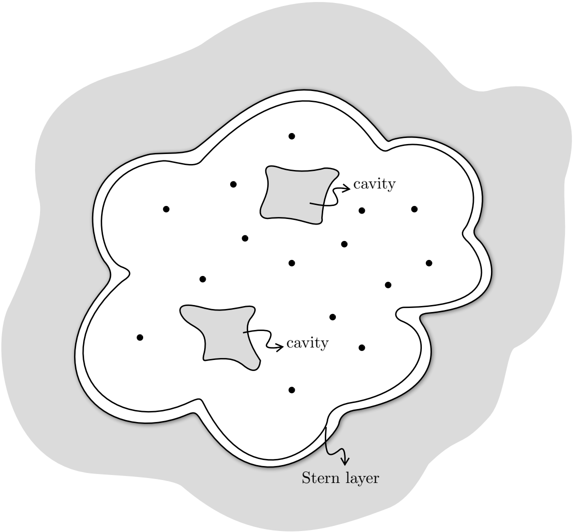


Figure 2·4: Protein with solvent-filled cavities and Stern layer.

extends this idea to vary permittivity and even viscosity (Bonthuis and Netz, 2013).

Figure 3·1 shows a dissolved protein that contains two solvent-filled cavities and a Stern layer around the outmost surface. The two-region model from Equation (2.62) is unable to capture those details, and we will discuss how to include them in Section 3.1.2.

2.1.5 Energy calculation with the implicit solvent model

For a system with dissolved proteins, it is useful to separate the free energy into polar and non-polar components. The electrostatic energy is the polar contribution to free energy, whereas the non-polar component considers van der Waals and entropic effects.

In the results of this thesis, we assume that all free energy is of electrostatic nature, neglecting the non-polar component. This is a good approximation in many cases, especially when proteins have charged regions. For simplicity, throughout this thesis we will use the term energy to refer to the polar component of free energy, unless otherwise noted.

Polar contribution to energy

In this work, we look at systems with three possible sources of electrostatic energy: Coulombic energy from the point charges, solvation energy caused by the influence of the solvent outside the protein region, and surface energy from a surface with prescribed charge or potential.

Coulombic energy The implicit-solvent model accounts for the charge distribution of the protein using point charges at the locations of the atoms. Using delta functions as the charge distribution, Equation (2.13) reads

$$F_{\text{Coulomb}} = \frac{1}{2}\sum_i \sum_j \frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j|}, \tag{2.63}$$

which is the well-known Coulomb energy.

Solvation free energy. When a protein is disolved, the solvent reacts by polarizing, giving rise to what we commonly know as the solvation energy. If the system contains only one protein, the solvation energy is given by the difference in energy between the protein in vacuum —with only the Coulombic effect of the point charges— and the dissolved protein. If there are other entities in the solvent, such as other proteins or charged surfaces, they will also be affected by the solvated protein and contribute to the solvation energy. This is why, in this case, we want to generalize the concept of solvation energy to the energy generated by the reaction of all of the protein's surroundings: solvent and other entities. In single-molecule settings, this definition of solvation energy coincides with the energy required to dissolve a molecule. This said, the solvation energy is generated by the reaction potential (ϕ_{reac}) inside the protein, i.e.,

$$F_{\text{solv}} = \frac{1}{2}\int_{\Omega} \rho\phi_{\text{reac}} = \sum_{k=0}^{N_q} q_k \phi_{\text{reac}}(\mathbf{r}_k), \tag{2.64}$$

where ρ is the charge distribution. In this case the charge distribution consists of point charges which transforms the integral into a sum. On the other hand, the reaction potential is

$$\phi_{\text{reac}} = \phi_{\text{total}} - \phi_{\text{Coulomb}}, \tag{2.65}$$

and we can interpret it as the potential generated by the boundary of the molecular region (Γ and Ω_1 in Figure 2·3).

Surface free energy. Chan and co-workers (Chan and Mitchell, 1983; Carnie and Chan, 1993) derived the free energy for a surface with a set charge or potential. They describe the free energy on a surface as

$$\begin{aligned} F_{\text{surface}} &= \frac{1}{2}\int\limits_{\Gamma} G_c \sigma_0^2 \mathrm{d}\Gamma \quad \text{for set charge distribution, and} \\ F_{\text{surface}} &= -\frac{1}{2}\int\limits_{\Gamma} G_p \phi_0^2 \mathrm{d}\Gamma \quad \text{for set potential,} \end{aligned} \tag{2.66}$$

where ϕ_0 and σ_0 are the prescribed potential and surface charge, respectively. The potential is given by $\phi(\sigma, R, \mathbf{x}) = G_c(R, \mathbf{x})\sigma$ for the first expression and the surface charge by $\sigma(\phi, R, \mathbf{x}) = G_p(R, \mathbf{x})\phi$ for the second one. This is valid because we are using a linearized Poisson-Boltzmann model.

Non-polar contribution to energy

The non-polar contribution to free energy ($F_{\text{non-polar}}$) considers short-range dispersion interactions between molecules, given by the van der Waals interaction, and conformational effects, represented by the entropy.

A dissolved protein occupies space in the solvent, and to accommodate it, water molecules need to be pulled away from the protein region. Also, water molecules will have a very organized structure in the first few solvation shells, near the solvent-excluded surface. These two effects work against entropy, and they need to be considered to have a complete representation of the free energy.

There are well established models for the non-polar contribution of free energy (Roux and Simonson, 1999; Wagoner and Baker, 2006). Most of those models assume a linear relation of $F_{\text{non-polar}}$ with the molecular surface, given by the solvent-excluded surface:

$$F_{\text{non-polar}} = \gamma A, \tag{2.67}$$

where A is the molecular surface area and γ the energetic coefficient or surface tension.

Interaction free energy.

When there are two or more bodies in the solvent, these will interact. In order to compute the energy of interaction, we need to take the difference between the total energy of the interacting system and the total energy of each isolated component. Neglecting the non-polar contributions, the total free energy is given by

$$F_{\text{total}} = F_{\text{Coulomb}} + F_{\text{surface}} + F_{\text{solv}}. \tag{2.68}$$

Then, the interaction free energy is

$$F_{\text{interaction}} = F_{\text{total}}^{\text{assembly}} - \sum_{i=1}^{N_c} F_{\text{total}}^{\text{comp}_i}, \tag{2.69}$$

where N_c is the number of components in the system and $F_{\text{total}}^{\text{comp}_i}$ is calculated over the isolated component i.

When the interacting system is an assembly of two bound molecules, the interaction energy corresponds to the binding energy.

2.1.6 Calculations using the implicit-solvent model

The solution of the coupled Poisson and Poisson-Boltzmann equations (Equation (2.62)) in molecular settings has been of interest to the biophysics community for a long time. In 1934, Kirkwood derived a closed expression for the solvation energy of a spherical molecule, based on spherical harmonics (Kirkwood, 1934). This expression can handle an arbitrary distribution of point charges inside the molecule and a Stern layer. Unfortunately, analytical forms are available only for spheres, and more complicated geometries require numerical approximations.

Numerical solution of the Poisson-Boltzmann equation

There has been substantial work in solving the coupled system of PDE's in Equation (2.62) for realistic molecular geometries. The first numerical calculations in a biomolecular setting were done in 1982 by Warwicker and Watson (Warwicker and Watson, 1982) using the finite-difference method, however, they considered a salt-free solvent. Also using the finite-difference method, Honig's group started developing `Delphi` in 1985 (Gilson et al., 1985), which became a standard in the community and is still actively supported, and later Bashford wrote the finite-difference code **`MEAD`** to perform pK_a calculations (Bashford and Gerwert, 1992; Bashford, 1997). Another numerical software that is actively developed and a standard in the biophysics community is Baker's **`APBS`** (Baker et al., 2001), which contains both finite-difference and finite-element solvers. The main advantage of using a finite-element method over a finite-difference method is that the mesh can conform to the molecular surface, avoiding the staircase representation of the solvent-excluded surface inherent of the finite-difference mesh. Along this line, the group led by Wei introduced the **`MIBPB`** package (Geng et al., 2007), a finite-difference solver with a special treatment of the interface called *the matched interface and boundary*, where they are able to explicitly enforce the interface conditions, getting second order accuracy.

On the other hand, the elliptic nature of Equation (2.62) allows its formulation in terms of an integral along the interface, which motivated several developments using the boundary-element method. This formulation was first derived by Shaw (Shaw, 1985), expressing the integral in terms of the polarization surface charge, which is valid for a salt-free solvent. Later, Yoon and Lenhoff (Yoon and Lenhoff, 1990) considered the linearized Poisson-Boltzmann equation in the solvent, and Juffer and co-workers (Juffer et al., 1991) found a formulation with hypersingular operators. Even though hypersingular operators may be more cumbersome, and the equation presented by Juffer contains more terms, the final matrix has a condition number that does not depend on the number of discretization elements, whereas in Yoon and Lenhoff's formulation the conditioning is size dependent (Liang and Subramaniam, 1997). Our work is based on the expression derived by Yoon and Lenhoff (Yoon and Lenhoff, 1990), because we use preconditioners and problem sizes where this is a competitive alternative.

An example of a boundary-element implementation of Equation (2.62) is the work of the group led by White and Tidor, that use the FFT-accelerated boundary-element code `FFTSVD` for drug design (Altman et al., 2005). Also, Lu and co-workers released the fast-multipole-based `AFMPB` in 2006 (Lu et al., 2006), and more recently Krasny and Geng developed a treecode-accelerated boundary-element code called `TABI` (Geng and Krasny, 2013). During the last couple of years, our group developed `PyGBe` (Cooper and Barba, 2013; Cooper et al., 2013c), a boundary-element method code that is accelerated with a treecode and runs on GPU. Results in this work were obtained using `PyGBe`, unless otherwise noted.

The main advantages of boundary-integral based methods over finite-difference or finite-elements are a more faithful representation of the molecular surface, boundary conditions matched exactly at infinity, and fewer degrees of freedom. However, boundary-element techniques yield dense matrices, requiring special acceleration techniques to obtain reasonable computing times. Some advantages of volume-based methods, like finite-elements and finite-differences, are that they can solve the nonlinear Poisson-Boltzmann equation, and easily deal with Stern layers and cavities.

Generalized-Born model

The generalized-Born model considers the solute as a set of interacting spheres with centered charges, and uses the analytical result from Kirkwood (Kirkwood, 1934) to find the solvation energy. The best-known form of this model was presented by Still and co-workers (Still et al., 1990), which is

$$F_{\text{GB}} = -166\left(1 - \frac{1}{\epsilon_{\text{solv}}}\right)\sum_i \sum_j \frac{q_i q_j}{f_{\text{GB}}}, \tag{2.70}$$

with,

$$\begin{aligned} f_{\text{GB}} &= (|\mathbf{r}_i - \mathbf{r}_j|^2 + \alpha_{ij}^2 e^D), \\ \alpha_{ij} &= (\alpha_i \alpha_j)^{0.5}, \text{ and} \\ D &= |\mathbf{r}_i - \mathbf{r}_j|^2/(2\alpha_{ij})^2 \end{aligned} \tag{2.71}$$

where α_i is a fitting parameter called the Born radius. This parameter is case dependent and usually obtained from experiments.

2.2 Boundary Element Methods (BEM)

The boundary-element method (BEM) solves boundary-integral equations (BIE) numerically. This is useful in the context of PDEs when the differential operator has a fundamental solution, and can be written as a BIE. The idea of representing PDEs as BIEs comes from the late 1800s and early 1900s, with the work by Betti, Somigliana and Fredholm (Betti, 1872; Somigliana, 1885; Fredholm, 1903), which were based on earlier work by Green (Green, 1828). In fact, the work by Fredholm is considered to be the starting point of BEM, as he was the first person to compute boundary data in a potential theory application. The work used analytical techniques which were valid for simple geometries. The first numerical solutions of Fredholm's equations appeared in the 1960s for two-dimensional potential theory (Jawson, 1963; Symm, 1963), and later for two- and three-dimensional potential theory and elasticity (Rizzo, 1967; Cruse, 1969). From then, BEM has evolved to be a widely used numerical technique.

There are several references that present good introductions to BEM. Very thorough overviews can be found in the monographs by Atkinson, McLean and Steinbach (Atkinson, 1997; McLean, 2000; Steinbach, 2008), that go into detail of the mathematics behind the method. Less exhaustive reviews can be found in the work by Brebbia (Brebbia and Dominguez, 1992) and Katsikadelis (Katsikadelis, 2002)

2.2.1 Derivation of a Laplace problem

A simple PDE that can be solved with BEM is the Laplace equation. For a scalar quantity ϕ in a domain Ω with boundary Γ, and known boundary conditions ϕ_0 or $\partial\phi_0/\partial\mathbf{n}$, the Laplace equation reads

$$\begin{aligned} \nabla^2\phi = 0 \quad & \text{on } \Omega \\ \phi = \phi_0 \text{ or } \frac{\partial\phi}{\partial\mathbf{n}} = \frac{\partial\phi_0}{\partial\mathbf{n}} \quad & \text{on } \Gamma \end{aligned} \tag{2.72}$$

The weak formulation of Equation (2.72) with test function w is

$$\int\limits_{\Omega} \nabla^2\phi(\mathbf{r}'_\Omega)w(\mathbf{r}'_\Omega)\mathrm{d}\Omega' = 0. \tag{2.73}$$

where the evaluation point $\mathbf{r}_\Omega$ is a location in the domain Ω, and the integral runs over the primed quantities.

The result of applying a linear differential operator on its free-space Green's function, is minus the Dirac delta function ($-\delta(\mathbf{r})$). For example,

$$\nabla^2 G_L(\mathbf{r},\mathbf{r}') = -\delta(\mathbf{r}) \tag{2.74}$$

defines the free-space Green's function for the Laplace operator, which is

$$G_L(\mathbf{r},\mathbf{r}') = \frac{1}{4\pi|\mathbf{r}-\mathbf{r}'|}. \tag{2.75}$$

Using Laplace's free-space Green's function as the test function in Equation (2.73), we get

$$\int\limits_{\Omega} \nabla^2\phi(\mathbf{r}'_\Omega)G_L(\mathbf{r}_\Omega,\mathbf{r}'_\Omega)\mathrm{d}\Omega' = 0. \tag{2.76}$$

We can manipulate the integrand in Equation (2.76) to obtain

$$\int\limits_{\Omega} \left[\nabla\cdot(\nabla\phi(\mathbf{r}'_\Omega)G_L(\mathbf{r}_\Omega,\mathbf{r}'_\Omega)) - \nabla\phi(\mathbf{r}'_\Omega)\nabla G_L(\mathbf{r}_\Omega,\mathbf{r}'_\Omega)\right]\mathrm{d}\Omega' = 0, \tag{2.77}$$

and further manipulating the second term of Equation (2.77), yields

$$\int\limits_{\Omega} \left[\nabla\cdot(\nabla\phi(\mathbf{r}'_\Omega)G_L(\mathbf{r}_\Omega,\mathbf{r}'_\Omega)) - \nabla\cdot(\phi(\mathbf{r}'_\Omega)\nabla G_L(\mathbf{r}_\Omega,\mathbf{r}'_\Omega)) + \phi(\mathbf{r}'_\Omega)\nabla^2 G_L(\mathbf{r}_\Omega,\mathbf{r}'_\Omega)\right]\mathrm{d}\Omega' = 0. \tag{2.78}$$

The divergence operator affects the first two terms of Equation (2.78), therefore we can apply the divergence theorem on them. Also, using Equation (2.74), we get

$$\int\limits_{\Gamma} \nabla\phi(\mathbf{r}'_\Gamma)\cdot\mathbf{n}G_L(\mathbf{r}_\Omega,\mathbf{r}'_\Gamma)\mathrm{d}\Gamma' - \int\limits_{\Gamma} \phi(\mathbf{r}'_\Gamma)\nabla G_L(\mathbf{r}_\Omega,\mathbf{r}'_\Gamma)\cdot\mathbf{n}\mathrm{d}\Gamma' - \int\limits_{\Omega} \phi(\mathbf{r}'_\Omega)\delta(\mathbf{r}_\Omega)\mathrm{d}\Omega' = 0 \tag{2.79}$$

where $\mathbf{n}$ is the vector normal to Γ pointing out of the domain Ω, and $\mathbf{r}_\Gamma$ is a location on the boundary Γ. We can rewrite Equation (2.79) as

$$\phi(\mathbf{r}_\Omega) = \int\limits_{\Gamma} G_L(\mathbf{r}_\Omega,\mathbf{r}'_\Gamma)\frac{\partial}{\partial\mathbf{n}}\phi(\mathbf{r}'_\Gamma)\mathrm{d}\Gamma' - \int\limits_{\Gamma} \phi(\mathbf{r}'_\Gamma)\frac{\partial}{\partial\mathbf{n}}G_L(\mathbf{r}_\Omega,\mathbf{r}'_\Gamma)\mathrm{d}\Gamma' \tag{2.80}$$

The derivation from Equation (2.76) to Equation (2.79) also serves as proof of

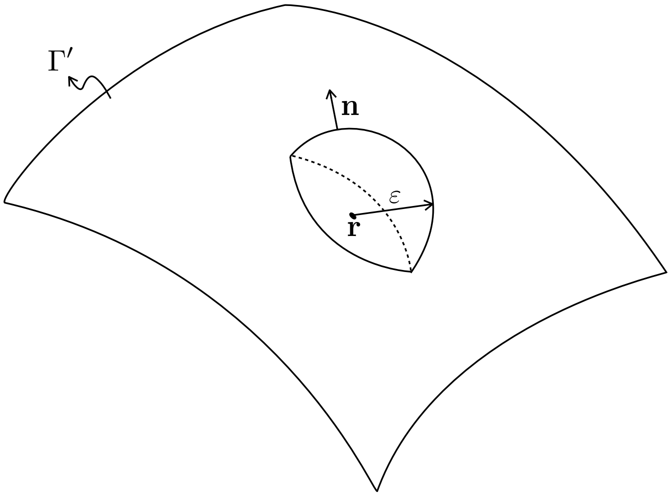


Figure 2·5: Treatment of the singular integral

Green's second identity, which for $\phi(\mathbf{r}')$ and $G_L(\mathbf{r},\mathbf{r}')$ reads

$$\int_{\Omega} (\nabla^2\phi G_L - \phi\nabla^2 G_L)\mathrm{d}\Omega = \int_{\Gamma} G_L \frac{\partial}{\partial\mathbf{n}}\phi - \phi\frac{\partial}{\partial\mathbf{n}}G_L\mathrm{d}\Gamma. \tag{2.81}$$

In Equation (2.80), $\mathbf{r}$ can be anywhere in the domain Ω, and $\mathbf{r}'$ runs only on the boundary Γ. If we take the limit as $\mathbf{r}\to\Gamma$, G_L has a singularity when $\mathbf{r}=\mathbf{r}'$. To handle this issue, we perform the integral on a surface Γ' that is exactly equal to Γ everywhere but contains a hemisphere of radius ε centered at $\mathbf{r}$, and split the integrals over Γ' into the integral over the region where Γ and Γ' match, plus the integral over the hemisphere (Γ'_{hem}). The surface Γ' avoids the singularity, and taking $\varepsilon\to 0$ recovers the original surface Γ. This setting is sketched in Figure 2·5.

The integral involving G_L over the hemisphere Γ'_{hem} is

$$\begin{aligned}\int_{\Gamma'_{\text{hem}}} \frac{\partial}{\partial\mathbf{n}}\phi(\mathbf{r}')G_L(\mathbf{r},\mathbf{r}')\mathrm{d}\Gamma' &= \frac{1}{4\pi}\int_{\Gamma'_{\text{hem}}} \frac{\partial}{\partial\mathbf{n}}\phi(\mathbf{r}')\frac{1}{|\mathbf{r}-\mathbf{r}'|}\mathrm{d}\Gamma' \\ &= \frac{1}{4\pi\varepsilon}\int_{\Gamma'_{\text{hem}}} \frac{\partial}{\partial\mathbf{n}}\phi(\mathbf{r}')\mathrm{d}\Gamma',\end{aligned} \tag{2.82}$$

because we know that on the hemisphere Γ'_{hem} the distance $|\mathbf{r}-\mathbf{r}'| = \varepsilon$ is constant. Taking the limit $\varepsilon \to 0$ to recover the original Γ, we get

$$\begin{aligned}\lim_{\varepsilon\to 0}\frac{1}{4\pi\varepsilon}\int\limits_{\Gamma'_{\text{hem}}}\frac{\partial}{\partial\mathbf{n}}\phi(\mathbf{r}')\mathrm{d}\Gamma' &= \lim_{\varepsilon\to 0}\frac{1}{4\pi\varepsilon}\frac{\partial}{\partial\mathbf{n}}\phi(\mathbf{r})\int\limits_{\Gamma'_{\text{hem}}}\mathrm{d}\Gamma' \\ &= \lim_{\varepsilon\to 0}\frac{1}{4\pi\varepsilon}\frac{\partial}{\partial\mathbf{n}}\phi(\mathbf{r})2\pi\varepsilon^2 = 0,\end{aligned} \tag{2.83}$$

considering that $\mathbf{r}' \to \mathbf{r}$ as $\varepsilon \to 0$. The approximation that the area around the singularity is a hemisphere is valid because at small ε, for a smooth surface Γ, the surface in the vicinity of Γ'_{hem} is approximately flat.

We can do a similar calculation for the integrals involving $\partial G_L/\partial\mathbf{n}$, which on the hemisphere Γ'_{hem} is

$$\begin{aligned}\int\limits_{\Gamma'_{\text{hem}}}\phi(\mathbf{r}')\frac{\partial}{\partial\mathbf{n}}G_L(\mathbf{r},\mathbf{r}')\mathrm{d}\Gamma' &= -\frac{1}{4\pi}\int\limits_{\Gamma'_{\text{hem}}}\phi(\mathbf{r}')\frac{(\mathbf{r}-\mathbf{r}')\cdot\mathbf{n}}{|\mathbf{r}-\mathbf{r}'|^3}\mathrm{d}\Gamma' \\ &= -\frac{1}{4\pi}\int\limits_{\Gamma'_{\text{hem}}}\phi(\mathbf{r}')\frac{\varepsilon}{\varepsilon^3}\mathrm{d}\Gamma',\end{aligned} \tag{2.84}$$

because $(\mathbf{r}-\mathbf{r}')$ is always aligned with $\mathbf{n}$ and has magnitude ε. Then, the limit as $\varepsilon \to 0$ gives

$$\begin{aligned}\lim_{\varepsilon\to 0}-\frac{1}{4\pi}\int\limits_{\Gamma'_{\text{hem}}}\phi(\mathbf{r}')\frac{\varepsilon}{\varepsilon^3}\mathrm{d}\Gamma' &= -\frac{\phi(\mathbf{r})}{4\pi}\lim_{\varepsilon\to 0}\frac{1}{\varepsilon^2}\int\limits_{\Gamma'_{\text{hem}}}\mathrm{d}\Gamma' \\ &= -\frac{\phi(\mathbf{r})}{4\pi}\lim_{\varepsilon\to 0}\frac{1}{\varepsilon^2}2\pi\varepsilon^2 = -\frac{\phi(\mathbf{r})}{2}.\end{aligned} \tag{2.85}$$

Using the results from Equation (2.83) and Equation (2.85), we can write the limit of Equation (2.80) as $\mathbf{r} \to \Gamma$ as

$$\frac{\phi(\mathbf{r}_\Gamma)}{2} + \int\limits_{\Gamma}\phi(\mathbf{r}'_\Gamma)\frac{\partial}{\partial\mathbf{n}}G_L(\mathbf{r}_\Gamma,\mathbf{r}'_\Gamma)\mathrm{d}\Gamma' = \int\limits_{\Gamma}G_L(\mathbf{r}_\Gamma,\mathbf{r}'_\Gamma)\frac{\partial}{\partial\mathbf{n}}\phi(\mathbf{r}'_\Gamma)\mathrm{d}\Gamma', \tag{2.86}$$

where these are Cauchy principal value integrals.

The exact same derivation that led from Equation (2.72) to Equation (2.86) is valid for the linearized Poisson-Boltzmann equation described by Equation (2.59), where the differential operator is $(\nabla^2 - \kappa^2)$. The free space Green's function of the linearized Poisson-Boltzmann equation, also known as the Yukawa potential, is

$$G_Y(\mathbf{r}, \mathbf{r}') = \frac{\exp(-\kappa|\mathbf{r} - \mathbf{r}'|)}{4\pi|\mathbf{r} - \mathbf{r}'|}. \tag{2.87}$$

2.2.2 Layer operators

The integral operators in Equation (2.86) and Equation (2.80) are recurring in boundary integral techniques. That is why they have been classified as the single-layer potential and double-layer potential operators, and thoroughly studied to assure their solvability (Steinbach, 2008).

Sobolev spaces

The integer order r Sobolev space based on $L_p(\Omega)$ is defined as

$$W_p^r(\Omega) = \left\{\psi \in L_p(\Omega) : \partial^\alpha \psi \in L_p(\Omega) \text{ for } |\alpha| \leq r\right\}, \tag{2.88}$$

where $L_p(\Omega)$ is the space of functions ψ for which the p^{th} power of the absolute value is integrable

$$||\psi||_p = \left(\int\limits_\Omega |\psi|^p \mathrm{d}\Omega \right)^{\frac{1}{p}} < \infty, \tag{2.89}$$

and $W_p^r(\Omega)$ is equipped with the norm

$$||\psi||_{W_p^r(\Omega)} = \left(\sum_{|\alpha|\leq r} \int\limits_\Omega |\partial^\alpha \psi(\mathbf{r})|^p \mathrm{d}\Omega \right)^{\frac{1}{p}}. \tag{2.90}$$

The Sobolev $W_p^s(\Omega)$ space of real order $s = r + \mu$, where r is an integer and

$0 < \mu < 1$, based on $L_p(\Omega)$, corresponds to

$$W_p^s(\Omega) = \left\{\psi \in W_p^r(\Omega) : |\partial^\alpha \psi|_{\mu,p,\Omega} < \infty \text{ for } |\alpha| = r\right\}, \tag{2.91}$$

with norm

$$||\psi||_{W_p^s(\Omega)} = \left(||\psi||^p_{W_p^r(\Omega)} + \sum_{|\alpha|=r} |\partial^\alpha \psi|^p_{\mu,p,\Omega}\right)^{\frac{1}{p}} \tag{2.92}$$

where

$$|\psi|_{\mu,p,\Omega} = \left(\int\limits_\Omega \int\limits_\Omega \frac{|\psi(\mathbf{r}) - \psi(\mathbf{r}')|}{|\mathbf{r} - \mathbf{r}'|^{n+p\mu}} \mathrm{d}\Omega \mathrm{d}\Omega\right)^{\frac{1}{p}} \tag{2.93}$$

is the Slobodeckiĭ semi norm.

The order of a Sobolev space can also be negative. $W_p^{-r}(\Omega)$ comprises distributions that admit the representation

$$\psi = \sum_{|\alpha| \le r} \partial^\alpha f_\alpha \text{ with } f_\alpha \in L_p(\Omega), \tag{2.94}$$

with norm

$$||\psi||_{W_p^{-r}(\Omega)} = \inf \left(\sum_{|\alpha| \le r} ||f_\alpha||^p_{L_p(\Omega)}\right). \tag{2.95}$$

Since we use the weak formulation of the PDE, shown in Equation (2.73), we require ϕ to be a square-integrable function, *i.e.* $\phi \in L_2$, and for $p = 2$ we will call the Sobolev space $W_2^s = H^s$.

For integer order r, the inner product of H^r is

$$(\varphi, \psi)_{H^r(\Omega)} = \sum_{|\alpha| \le r} \int\limits_\Omega \overline{\partial^\alpha \varphi(\mathbf{x})} \partial^\alpha \psi(\mathbf{x}) \mathrm{d}\Omega, \tag{2.96}$$

and for real order s it is

$$(\varphi,\psi)_{H^s(\Omega)} = (\varphi,\psi)_{H^r(\Omega)} + \sum_{|\alpha|=r} \int\limits_{\Omega}\int\limits_{\Omega} \frac{\overline{[\partial^\alpha\varphi(\mathbf{r}) - \partial^\alpha\varphi(\mathbf{r}')]}[\partial^\alpha\psi(\mathbf{r}) - \partial^\alpha\psi(\mathbf{r}')]}{|\mathbf{r}-\mathbf{r}'|^{n+2\mu}} \mathrm{d}\Omega\mathrm{d}\Omega. \tag{2.97}$$

In the definitions from Equation (2.88) to Equation (2.97), the derivative $\partial^\alpha\psi$ is taken in the weak sense, that is, if

$$\int\limits_{\Omega} \varphi D^\alpha \zeta \mathrm{d}\Omega = \int\limits_{\Omega} \psi\zeta\mathrm{d}\Omega \quad \forall\ \zeta \in C^\infty \tag{2.98}$$

where D^α is the classic derivative operator, then ψ is the order α derivative of φ. In cases when $\psi \in C^\alpha$, $\partial^\alpha\psi$ matches the classical derivative.

We can also define Sobolev space $\widetilde{H}^s(\Omega)$ as the closure of $C_0^\infty(\Omega)$ with respect to the norm in $H^s(\Omega)$ space $(||\psi||_{H^s(\Omega)})$.

Trace operator

The trace operator γ_0 takes a distribution $\psi(\mathbf{r}_\Omega)$ valid on the whole domain and evaluates it on the boundary

$$\gamma_0\psi(\mathbf{r}_\Omega) = \psi(\mathbf{r}_\Gamma). \tag{2.99}$$

If s >1/2, then γ_0 has a unique extension to a bounded operator

$$\gamma_0 : H^s(\mathbb{R}^n) \to H^{s-1/2}(\mathbb{R}^{n-1}). \tag{2.100}$$

The operator γ_0 is also known as the Dirichlet trace operator, and we find it in the second term in Equation (2.86), where $\phi(\mathbf{r}) = \phi(\mathbf{r}_\Gamma)$.

The term on the right-hand side of Equation (2.86) contains the normal derivative of the trace. This combined operator is known as the Neumann trace, γ_1:

$$\gamma_1 = \partial/\partial\mathbf{n}\gamma_0 : H^s(\mathbb{R}^n) \to H^{s-3/2}(\mathbb{R}^{n-1}). \tag{2.101}$$

With this new notation, we can rewrite Equation (2.73) using Green's second identity for distributions ϕ and w as

$$\int_\Gamma \gamma_1\phi(\mathbf{r}')\gamma_0 w(\mathbf{r}')\mathrm{d}\Gamma - \int_\Gamma \gamma_0\phi(\mathbf{r}')\gamma_1 w(\mathbf{r}')\mathrm{d}\Gamma - \int_\Omega \phi(\mathbf{r}')\nabla^2 w(\mathbf{r}')\mathrm{d}\Omega = 0 \tag{2.102}$$

which is valid and has a unique solution for $\phi(\mathbf{r}_\Omega) \in H^1$ and $w(\mathbf{r}_\Omega) \in H^1$ (McLean, 2000; Steinbach, 2008). Then, $\nabla^2 w(\mathbf{r}_\Omega) \in \widetilde{H}^{-1}$, $\gamma_0\phi(\mathbf{r}_\Omega) \in H^{\frac{1}{2}}$ and $\gamma_1\phi(\mathbf{r}_\Omega) \in H^{-\frac{1}{2}}$.

Single-layer potential operator

For a density function $\psi \in H^{-\frac{1}{2}}$ on a Lipshitz surface Γ that bounds a domain Ω, the single-layer potential operator evaluated at $\mathbf{r}_\Omega$ is

$$\phi(\mathbf{r}_\Omega) = \widetilde{V}_L^{\mathbf{r}_\Omega}(\psi(\mathbf{r}_\Gamma)) = \int_\Gamma \psi(\mathbf{r}_\Gamma)G_L(\mathbf{r}_\Omega, \mathbf{r}_\Gamma)\mathrm{d}\Gamma, \tag{2.103}$$

and judging from Equation (2.80), the resulting distribution $\phi(\mathbf{r}_\Omega)$ resides H^1. Hence, Equation (2.107) defines a linear map

$$\widetilde{V} : H^{-\frac{1}{2}}(\Gamma) \to H^1(\Omega), \tag{2.104}$$

bounded by

$$||\phi||_{H^1(\Omega)} = ||\widetilde{V}_L^{\mathbf{r}_\Omega}(\psi(\mathbf{r}_\Gamma))||_{H^1(\Omega)} < c||\psi||_{H^{-\frac{1}{2}}(\Gamma)}. \tag{2.105}$$

Since $\widetilde{V}_L^{\mathbf{r}_\Omega}(\psi(\mathbf{r}_\Gamma)) \in H^1$, the Dirichlet trace of $\widetilde{V}$ is well defined and bounded, defining a new single-layer potential operator

$$V = \gamma_0\widetilde{V} : H^{-\frac{1}{2}} \to H^{\frac{1}{2}} \tag{2.106}$$

and is

$$\phi(\mathbf{r}_\Gamma) = V_L^{\mathbf{r}_\Gamma}(\psi(\mathbf{r}_\Gamma)) = \int\limits_\Gamma \psi(\mathbf{r}'_\Gamma) G_L(\mathbf{r}_\Gamma, \mathbf{r}'_\Gamma) \mathrm{d}\Gamma. \tag{2.107}$$

The fact that the operator V results in a distribution in $H^{\frac{1}{2}}$ agrees with Equation (2.86), where the term $\phi(\mathbf{r})/2$ also resides in $H^{\frac{1}{2}}$ because it is evaluated on the boundary, and the distribution $\partial\phi/\partial\mathbf{n} \in H^{-\frac{1}{2}}$.

Double-layer potential operator

For a density function $\psi \in H^{\frac{1}{2}}$ on a Lipshitz surface Γ that bounds a domain Ω, the double-layer potential operator evaluated at $\mathbf{r}_\Omega$ is

$$\phi(\mathbf{r}_\Omega) = W_L^{\mathbf{r}_\Omega}(\psi(\mathbf{r}_\Gamma)) = \int\limits_\Gamma \psi(\mathbf{r}_\Gamma) \frac{\partial}{\partial \mathbf{n}} G_L(\mathbf{r}_\Omega, \mathbf{r}_\Gamma) \mathrm{d}\Gamma, \tag{2.108}$$

and judging from Equation (2.80), the resulting distribution $\phi(\mathbf{r}_\Omega)$ resides H^1. Hence, Equation (2.108) defines a linear map

$$W : H^{\frac{1}{2}}(\Gamma) \to H^1(\Omega), \tag{2.109}$$

bounded by

$$||\phi||_{H^1(\Omega)} = ||W_L^{\mathbf{r}_\Omega}(\psi(\mathbf{r}_\Gamma))||_{H^1(\Omega)} < c||\psi||_{H^{\frac{1}{2}}(\Gamma)}. \tag{2.110}$$

Since $W_L^{\mathbf{r}_\Omega}(\psi(\mathbf{r}_\Gamma)) \in H^1$, the Dirichlet trace of W is well defined and bounded, defining a new double-layer potential operator

$$\gamma_0 W : H^{\frac{1}{2}} \to H^{\frac{1}{2}}. \tag{2.111}$$

We saw in Equation (2.85) that a $\phi/2$ term appears by taking the Dirichlet trace of the double layer potential. It is useful to define the operator K by

$$\phi(\mathbf{r}_\Gamma) = K_L^{\mathbf{r}\Gamma}(\psi(\mathbf{r}_\Gamma)) = \int\limits_\Gamma \psi(\mathbf{r}'_\Gamma)\frac{\partial}{\partial \mathbf{n}}G_L(\mathbf{r}_\Gamma, \mathbf{r}'_\Gamma)\mathrm{d}\Gamma. \tag{2.112}$$

to write $\gamma_0 W$ as

$$\gamma_0 W = \frac{1}{2} + K \tag{2.113}$$

The fact that the operator W results in a distribution in $H^{\frac{1}{2}}$ agrees with Equation (2.86), where the term $\phi(\mathbf{r})/2$ also resides in $H^{\frac{1}{2}}$ because it is evaluated on the boundary, and the distribution $\phi \in H^{\frac{1}{2}}$.

We can rewrite Equation (2.86) using the operator notation just introduced, as follows

$$\left[\frac{\mathbb{I}}{2} + K_L^{\mathbf{r}\Gamma}\right](\phi_\Gamma) = V_L^{\mathbf{r}\Gamma}\left(\frac{\partial}{\partial \mathbf{n}}\phi_\Gamma\right), \tag{2.114}$$

where $\mathbb{I}$ is the identity operator. In general, we can formulate the system in Equation (2.114) as

$$Ax = f. \tag{2.115}$$

If Neumann boundary conditions are enforced, $A = \left[\frac{\mathbb{I}}{2} + K_L^{\mathbf{r}\Gamma}\right]$, $x = \phi_\Gamma$, and $f = V_L^{\mathbf{r}\Gamma}\left(\frac{\partial}{\partial \mathbf{n}}\phi_\Gamma\right)$, whereas if Dirichlet boundary conditions are enforced, $A = V_L^{\mathbf{r}\Gamma}$, $x = \frac{\partial}{\partial \mathbf{n}}\phi_\Gamma$, and $f = \left[\frac{\mathbb{I}}{2} + K_L^{\mathbf{r}\Gamma}\right](\phi_\Gamma)$.

Considering that all operators involved in Equation (2.114) are elliptic, which makes them coercive, and bounded, the Lax-Milgram lemma ensures the solvability of the system in Equation (2.115). This lemma says that if A is bounded, in bilinear form, and coercive, for any f there exists a unique and bounded solution x.

2.2.3 Discretization and linear system

To obtain a discrete version of the layer operators in Equation (2.107) and Equation (2.113), we discretize the surface into flat triangular panels, and assume a constant value of ϕ and $\partial\phi/\partial\mathbf{n}$ within each panel. The discretized version of the single- and double-layer operators is

$$\begin{aligned} V_{L,\text{disc}}^{\mathbf{r}_i}\left(\frac{\partial}{\partial\mathbf{n}}\phi(\mathbf{r}_\Gamma)\right) &= \sum_{j=1}^{N_p}\frac{\partial}{\partial\mathbf{n}}\phi(\mathbf{r}_{\Gamma_j})\int\limits_{\Gamma_j} G_L(\mathbf{r}_i,\mathbf{r}_{\Gamma_j})\mathrm{d}\Gamma_j \\ K_{L,\text{disc}}^{\mathbf{r}_i}(\phi(\mathbf{r}_\Gamma)) &= \sum_{j=1}^{N_p}\phi(\mathbf{r}_{\Gamma_j})\int\limits_{\Gamma_j}\frac{\partial}{\partial\mathbf{n}}\left[G_L(\mathbf{r}_i,\mathbf{r}_{\Gamma_j})\right]\mathrm{d}\Gamma_j \end{aligned} \tag{2.116}$$

where N_p is the number of discretization elements of Γ, and $\phi(\mathbf{r}_{\Gamma_j})$ and $\frac{\partial}{\partial\mathbf{n}}\phi(\mathbf{r}_{\Gamma_j})$ are the values of ϕ and $\frac{\partial\phi}{\partial\mathbf{n}}$ on panel Γ_j.

In this work, we obtain a linear system by collocating the equation at the centroid of each panel Γ_i. For example, in the Laplace problem, the system would be

$$\left[\frac{\mathbb{I}}{2}+K_{L,\text{disc}}^{\Gamma}\right][\phi] = \left[V_{L,\text{disc}}^{\Gamma}\right]\left[\frac{\partial}{\partial\mathbf{n}}\phi\right] \tag{2.117}$$

where the elements of the matrices are

$$\begin{aligned} V_{L,ij}^{\Gamma} &= \int\limits_{\Gamma_j} G_L(\mathbf{r}_{\Gamma_i},\mathbf{r}_{\Gamma_j})\mathrm{d}\Gamma_j \\ K_{L,ij}^{\Gamma} &= \int\limits_{\Gamma_j}\frac{\partial}{\partial\mathbf{n}}\left[G_L(\mathbf{r}_{\Gamma_i},\mathbf{r}_{\Gamma_j})\right]\mathrm{d}\Gamma_j \end{aligned} \tag{2.118}$$

with $\mathbf{r}_{\Gamma_i}$ at the center of Γ_i.

2.2.4 Acceleration strategies

One drawback of BEM is that the matrix obtained after discretization is dense. This means that solving the linear system with Gaussian elimination requires $\mathcal{O}(N^3)$ computations and $\mathcal{O}(N^2)$ storage. Using an iterative solver, such as the Generalized minimal residual method (GMRES), computations are dominated by a matrix-vector product, and the scaling drops to $\mathcal{O}(N^2)$ per iteration . This scaling in computations and memory makes BEM feasible for only a few thousand boundary elements, which is far from the mesh sizes required for real applications. Fortunately, there are acceleration methods that make computations scale as $\mathcal{O}(N \log N)$, or even $\mathcal{O}(N)$, and memory requirements as $\mathcal{O}(N)$, that allows us to use BEM in larger problems.

All acceleration strategies use the decaying nature of the kernel to compute a low-rank approximation of the dense BEM matrix. The most commonly used ones, namely multipole methods, FFT based methods and adaptive-cross approximations (ACA), accelerate the matrix-vector product and rely on iterative solvers like GMRES. We briefly review these methods next. There are also accelerated direct solvers that have been successfully applied to BEM (Ho and Greengard, 2012).

Multipole methods

Multipole methods were originally developed to accelerate N-body calculations of $\mathcal{O}(N^2)$ complexity in particle simulations (Barnes and Hut, 1986; Greengard and Rokhlin, 1987), where the potential due to point-sources of mass is calculated on target locations. These methods approximate far-away source-target interactions with a multipole expansion using Taylor expansions (Lindsay and Krasny, 2001; Li et al., 2009) or spherical harmonics (Greengard and Rokhlin, 1987; Greengard and Huang, 2002; Greengard et al., 1998). In the context of BEM, multipole algorithms are useful because we usually approximate integrals with Gauss quadrature rules, which converts a matrix-vector product of the BEM matrix into an N-body calculation. The two main algorithms of this kind are the treecode (Barnes and Hut, 1986) and the fast-multipole method (FMM) (Greengard and Rokhlin, 1987).

Treecode The treecode was developed by Barnes and Hut in 1986 (Barnes and Hut, 1986) for astrophysical applications. This algorithm reduces the computational complexity to $\mathcal{O}(N \log N)$ by aggregating far-away sources to a center of expansion that interacts with the target location. In this work, we use the treecode to accelerate the BEM solver, and it will be further discussed in Section 2.3.

Fast multipole method (FMM) The FMM was developed by Greengard and Rokhlin in 1987 (Greengard and Rokhlin, 1987). This algorithm not only performs the multipole expansions at the point-sources of mass, but also expands the target locations, in the so-called local expansions, and then computes interactions between multipole and local expansions. The FMM has an algorithmic complexity of $\mathcal{O}(N)$.

Fast-fourier transform (FFT) based methods

FFT-based methods take advantage of the fast evaluation of convolutions using the FFT, an $\mathcal{O}(N \log N)$ algorithm. In the context of BEM, the matrix-vector product in the Krylov solver can be viewed as a discrete convolution of the elements of the vector, and the source-target interaction kernel, given by the matrix. Two popular FFT based acceleration methods in BEM are the Precorrected FFT (PFFT) (Phillips and White, 1997) and the FFTSVD (Altman et al., 2006).

Precorrected FFT The PFFT algorithm was developed by Phillips and White in 1997 (Phillips and White, 1997) for electrostatic applications. In it, the elements of the vector in the matrix-vector product are projected onto a regular mesh, and the convolution of faraway mass-target interactions is done on this mesh via the FFT. Then, the resulting values are interpolated to the target locations.

FFT with singular value decomposition (FFTSVD) The FFTSVD algorithm was developed by Altman, Bardhan, White and Tidor in 2006 (Altman et al., 2006) for biomolecular electrostatic simulations. This method groups boundary elements into cubes and computes low-rank approximations of the matrix that represents the effect of a cube on a target location by truncating the singular value decomposition

(SVD). Then, they use the FFT to compute long-range interactions in a similar fashion to PFFT, by projecting onto a mesh, applying the FFT, and interpolating to the target locations.

Adaptive cross approximation

Perhaps the most widely used acceleration method in the BEM community is the Adaptive cross approximation (ACA), developed by Bebendorf and Rjasanow (Bebendorf, 2000; Bebendorf and Rjasanow, 2003). This method is agnostic of the free-space Green's function of the original differential equation, which makes it a good black-box type solver, and has been implemented in different libraries, such as AHMED and HLIB.

The ACA method uses geometrical information to cluster the boundary elements in groups, and splits the BEM matrix hierarchically in blocks of various sizes. The diagonal blocks contain group self-interactions, and require their full rank to be well represented. Off-diagonal blocks correspond to longer-range interactions, and we can compute them using a low-rank approximation which can be compressed. It is possible to use ACA to accelerate a direct LU-decomposition solver, however, iterative solvers are more efficient, and the LU-decomposition can be used as a preconditioner (Bebendorf, 2005).

2.3 Treecode

We chose the treecode algorithm to accelerate our BEM solver, and this section reviews it in more detail. The treecode is a fast-summation algorithm capable of performing in $\mathcal{O}(N \log N)$ operations the following $\mathcal{O}(N^2)$ computational pattern:

$$V(\mathbf{x}_i) = \sum_{j=1}^{N} q_j \psi(\mathbf{x}_i, \mathbf{y}_j) \tag{2.119}$$

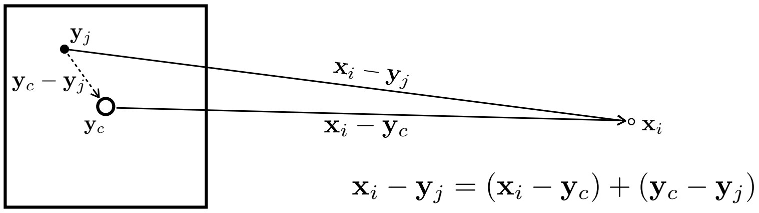

$\mathbf{y}_j$
$\mathbf{y}_c - \mathbf{y}_j$
$\mathbf{y}_c$
$\mathbf{x}_i - \mathbf{y}_j$
$\mathbf{x}_i - \mathbf{y}_c$
$\mathbf{x}_i$
$\mathbf{x}_i - \mathbf{y}_j = (\mathbf{x}_i - \mathbf{y}_c) + (\mathbf{y}_c - \mathbf{y}_j)$

Figure 2·6: Multipole expansion in the treecode

where q_j is the weight, ψ the kernel, $\mathbf{y}_j$ the location of sources and $\mathbf{x}_i$ the location of targets.

In the treecode, we group the sources $\mathbf{y}_j$ according to their spatial positions into cells or boxes of an octree. Having this, we calculate nearby contributions directly, and approximate distant contributions by writing series expansions representing the sources around the box centers $\mathbf{y}_c$, as sketched by Figure 2·6. Expansions are usually based either on Taylor series or spherical harmonics. In this work, we use the simpler Taylor expansions, such as the following approximation for ψ:

$$\psi(\mathbf{x}_i, \mathbf{y}_j) = \sum_{||\mathbf{k}||=0}^{P} \frac{1}{\mathbf{k}!} D_{\mathbf{y}}^{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_j)(\mathbf{y}_c - \mathbf{y}_j)^{\mathbf{k}} \tag{2.120}$$

where P is the order of the expansion, $D_{\mathbf{y}}^{\mathbf{k}} = D_{y_1}^{k_1} D_{y_2}^{k_2} D_{y_3}^{k_3}$ the derivative operator, and $\mathbf{k} = (k_1, k_2, k_3)$. We define the operations on $\mathbf{k}$ as $\mathbf{k}! = k_1!k_2!k_3!$ and $\mathbf{y}^{\mathbf{k}} = y_1^{k_1} y_2^{k_2} y_3^{k_3}$. Rearranging the terms in (2.120), we find the following expression:

$$V(\mathbf{x}_i) = \sum_{||\mathbf{k}||=0}^{P} \frac{1}{\mathbf{k}!} D_{\mathbf{y}}^{\mathbf{k}} \psi(\mathbf{x}_i, \mathbf{y}_c) \sum_j q_j (\mathbf{y}_c - \mathbf{y}_j)^{\mathbf{k}}, \tag{2.121}$$

where

$$a^{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_c) = \frac{1}{\mathbf{k}!} D_{\mathbf{y}}^{\mathbf{k}} \psi(\mathbf{x}_i, \mathbf{y}_c) \tag{2.122}$$

are the coefficients of the Taylor expansion and

$$m_c^{\mathbf{k}} = \sum_j q_j (\mathbf{y}_c - \mathbf{y}_j)^{\mathbf{k}} \tag{2.123}$$

are the multipoles. In Equation (2.121), the calculation of multipoles can be done independent of $\mathbf{x}_i$, thus eliminating the $\mathcal{O}(N^2)$ scaling.

For each target point, we decide if a box of sources is distant enough for the Taylor expansion approximation to be acceptable. If it is, we use Equation (2.121); if not, we ask the same question for the child boxes. We perform this process recursively until a lowest-level box is reached, and compute the influence of those sources directly. This decision is based on the multipole-acceptance criterion θ (MAC), which must be larger than the ratio of the box's size to the distance between the target and the box's center, $\frac{r_b}{r} < \theta$. Common values for θ are $\frac{1}{2}$ and $\frac{2}{3}$

The treecode algorithm can be divided into five parts: tree generation, particle-to-multipole, multipole-to-multipole, multipole-to-particle and particle-to-particle. Figure 2·7 sketches these steps for a group of sources $\mathbf{y}_j$, represented by solid black dots, interacting with two target points $\mathbf{x}_i$ at the locations of the crosses.

Tree generation. Sources are grouped into boxes of an octree structure, making sure that the lower level boxes have fewer than some critical number of particles (NCRIT). In the example of Figure 2·7 the sources $\mathbf{y}_j$ are grouped in a two-level tree: the lowest level has four boxes that are contained inside a bigger box from the top level, centered at $\mathbf{y}_c$.

Particle-to-multipole (P2M). Multipoles are calculated for the lowest level boxes using Equation (2.123). The sources are expanded about the center of the corresponding box.

Multipole-to-multipole (M2M). Multipoles are calculated for the remaining boxes. In the case of Figure 2·7, only the large box that contains all sources needs its multipole to be calculated. For efficiency, we calculate these multipoles by shifting the Taylor expansion from the child boxes, which results in

$$m_p^{\mathbf{k}} = \sum_{\text{children}} \sum_{k_1', k_2', k_3'=0}^{k_1' \leq k_1, k_2' \leq k_2, k_3' \leq k_3} m_c^{\mathbf{k}'} \binom{k_1}{k_1'} \binom{k_2}{k_2'} \binom{k_3}{k_3'} (\mathbf{y}_{cp} - \mathbf{y}_{cc})^{\mathbf{k}-\mathbf{k}'} \tag{2.124}$$

where $m_p^{\mathbf{k}}$ is the order $\mathbf{k}$ multipole of the parent box, $\mathbf{k}' = (k_1', k_2', k_3')$, $\mathbf{y}_{cp}$ is the location of the parent box, and $\mathbf{y}_{cc}$ is the location of the child box.

Multipole-to-particle (M2P). Equation (2.121) is calculated for the particle-box interactions that satisfy the θ criterion. The representation in Figure 2·7 shows that the rightmost target location is far enough that it can interact with the large box, whereas the target location closest to the sources does not meet the θ criterion with the bottom-right box.

In our implementation, we use recursive relations from Krasny and co-workers to compute the coefficients of the Taylor expansion on Equation (2.122) (Lindsay and Krasny, 2001; Li et al., 2009). Using the short-hand notation $\Delta \mathbf{x} = (\Delta x_1, \Delta x_2, \Delta x_3) = \mathbf{x}_i - \mathbf{y}_c$, $\mathbf{e}_1 = (1,0,0)$, $\mathbf{e}_2 = (0,1,0)$ $\mathbf{e}_3 = (0,0,1)$ and $||\mathbf{k}|| = k_1 + k_2 + k_3$, the recursive relation for the Laplace potential is

$$a^{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_c) = \frac{1}{||\mathbf{k}|| \cdot |\Delta \mathbf{x}|^2} \left[-(2||\mathbf{k}||-1) \sum_{j=1}^{3} \Delta x_j a^{\mathbf{k}-\mathbf{e}_j} - (||\mathbf{k}||-1) \sum_{j=1}^{3} a^{\mathbf{k}-2\mathbf{e}_j} \right]. \tag{2.125}$$

For the Yukawa potential, the recursive relation is,

$$\begin{aligned} a^{\mathbf{k}}(\mathbf{x}_i, \mathbf{y}_c) = & \frac{1}{||\mathbf{k}|| \cdot |\Delta \mathbf{x}|^2} \Bigg[-(2||\mathbf{k}||-1) \sum_{j=1}^{3} \Delta x_j a^{\mathbf{k}-\mathbf{e}_j} - \\ & (||\mathbf{k}||-1) \sum_{j=1}^{3} a^{\mathbf{k}-2\mathbf{e}_j} - \kappa \left(\sum_{j=1}^{3} \Delta x_j b^{\mathbf{k}-\mathbf{e}_j} + \sum_{j=1}^{3} b^{\mathbf{k}-2\mathbf{e}_j} \right) \Bigg] \end{aligned} \tag{2.126}$$

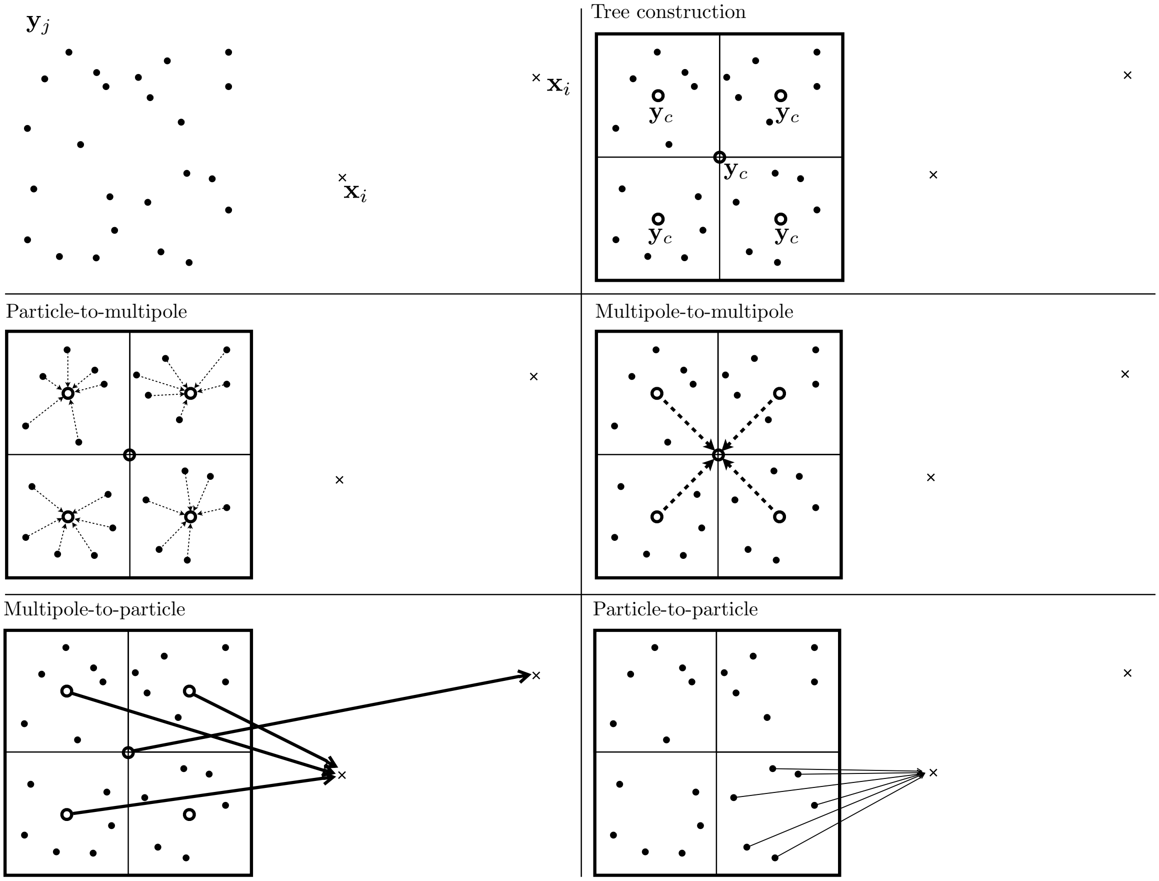


Figure 2·7: Steps of the treecode

where κ is the coefficient in the exponent of the Yukawa potential and,

$$b^{\mathbf{k}} = -\frac{\kappa}{||\mathbf{k}||}\left[\sum_{j=1}^{3}\Delta x_j a^{\mathbf{k}-\mathbf{e}_j} + \sum_{j=1}^{3} a^{\mathbf{k}-2\mathbf{e}_j}\right]. \tag{2.127}$$

Particle-to-particle (P2P). Direct pairwise interactions are computed between particles that are too close together to satisfy the θ criterion. In Figure 2·7 this happens for the interaction of the sources in the bottom-right box with the target location closest to it.

2.4 Treecode in BEM

There are several options for standard linear solvers to compute the linear system generated by BEM. We used the generalized minimal residual method (GMRES), a Krylov subspace iterative solver that accepts non-symmetric matrices. In GMRES, computation time is dominated by matrix vector products —an $\mathcal{O}(N^2)$ computational pattern—, which are performed n times (usually $n << N$).

In real applications, we evaluate most integrals of the BEM stiffness matrix with Gauss quadrature rules, which transforms each integral into a sum. For example, for the single-layer operator in Equation (2.117), each element of the matrix becomes

$$V^{\Gamma}_{L,ij} = \int\limits_{\Gamma_j} G_L(\mathbf{r}_{\Gamma_i}, \mathbf{r}_{\Gamma_j}) \mathrm{d}\Gamma_j = \sum_{k=1}^{N_k} w_k G_L(\mathbf{r}_{\Gamma_i}, \mathbf{r}_{k\Gamma_j}) \tag{2.128}$$

where N_k is the number of Gauss points, located at $\mathbf{r}_{k\Gamma_j}$ and with weight w_k. If we multiply matrix $\left[V^{\Gamma}_{L,\text{disc}}\right]$ with vector $\mathbf{m}$, each element of the resulting vector is computed as

$$V^{\Gamma}_{L,ij} m_j = \sum_{j=1}^{N_p} m_j \sum_{k=1}^{N_k} w_k G_L(\mathbf{r}_{\Gamma_i}, \mathbf{r}_{k\Gamma_j}) = \sum_{l=1}^{N_p \cdot N_k} M_l G_L(\mathbf{r}_{\Gamma_i}, \mathbf{r}_l), \tag{2.129}$$

which is an N-body problem of the form of Equation (2.119), where M_l has contributions from vector $\mathbf{m}$ and the Gauss integration weights w_k, and $\mathbf{r}_l$ is the location of the Gauss points $\mathbf{r}_{k\Gamma_j}$ of all panels Γ_j. We can use the treecode algorithm to accelerate it, which means that there is no need to explicitly store the matrix and computations scale as $\mathcal{O}(N \log N)$ instead of $\mathcal{O}(N^2)$.

In the context of BEM, the sources $\mathbf{y}_j$ in Equation (2.119) are the Gauss points, targets $\mathbf{x}_i$ correspond to the collocation points and the kernel ψ is the free-space Green's function.

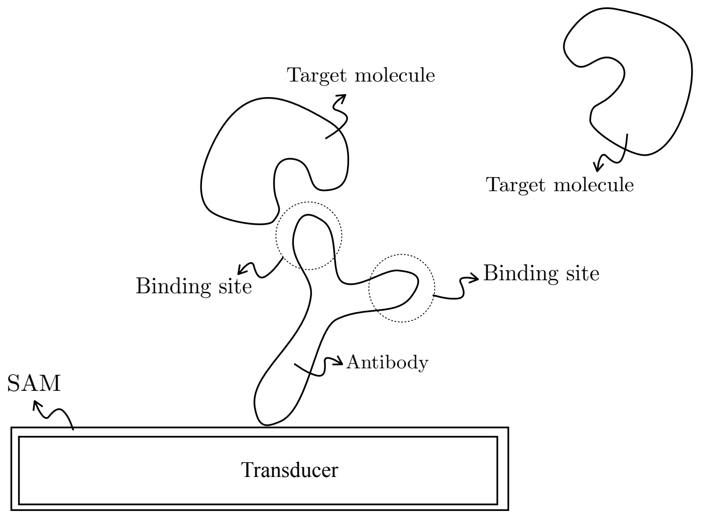


Figure 2·8: Sketch of a biosensor.

2.5 Biosensors

Two applications that motivate this work, presented in Chapters 6 and 7, are related to biosensors, making it useful to briefly review this technology. A biosensor is a device used for the detection of biomolecules, combining a biological component and a transducer. We focus on label-free nanoscale biosensors, where the biological component is a ligand molecule with high affinity to the target or analyte, and the transducer can be, for example, a semiconductor or a nanoparticle. A natural choice for a ligand molecule is an antibody, which is attached to the transducer through a self-assembled monolayer (SAM) of a small charged group. Figure 2·8 is a sketch of such a situation, with an immunoglobulin G-like ligand molecule.

The transducer also differs depending on the technology. For example, field-effect transistor (FET) biosensors use a semiconductor, like a nanowire, where the current density or voltage changes when a target molecule binds to the ligand (Brinkman et al., 1997). This technology has been around for decades, and is actively used in biosensing applications (Palacios, 2012; Patolsky et al., 2006).

Another class of biosensors are based on localized surface plasmon resonance

(LSPR), where the transducer is a metallic nanoparticle. LSPR is an optical phenomenon generated by light shining on conductive nanoparticles that are smaller than the wavelength of light. The surface electrons in the conduction band of the nanoparticle are excited by the incoming electric field, and oscillate with it, forming plasmons. The resonance of plasmons with the incoming wave produces high levels of extinction of the field (scattering plus absorption), creating a large shadow. LSPR biosensors take advantage of the fact that this effect is largely dependent on the local dielectric environment, usually reporting a shift in the resonance frequency as biomolecules approach the nanoparticle. There are several good reviews of this technology (Willets and Van Dyune, 2007; Petryayeva and Krull, 2011), and its application to biosensors is an active area of research (Hutter and Fendler, 2004; Piliarik et al., 2011; Raphael et al., 2013).

2.5.1 Biosensor models with implicit-solvent

The implicit-solvent model has been used in the past to model biosensors. For example, Nair and Alam (Nair and Alam, 2007) modeled the semiconductor in FET biosensors using the drift-diffusion equation, and coupled it to the Poisson-Boltzmann equation in the solvent, using a finite-difference approach. They used this model to study the effect of the solvent permittivity, salt concentration and semiconductor doping concentration on the biosensor, and created a biosensor modeling suite called BiosensorLab, available through nanoHUB.[3] Work led by Heitzinger (Heitzinger et al., 2010b) intends to model FET biosensors for DNA hybridization, which have a brush of single-stranded DNA (ssDNA) as ligand molecules; they present a finite-element approach with a homogenization technique to replace the ssDNA layer with special interface conditions between the solvent and semiconductor regions (Baumgartner and Heitzinger, 2012; Baumgartner and Heitzinger, 2013). This same group also

---

[3] `https://nanohub.org/resources/senstran`

developed a stochastic Poisson-Boltzmann method to account for variations in the orientation of the ssDNA attached to the nanowire (Heitzinger et al., 2010a).

Even though LSPR is an optical phenomenon, electrostatics is a good approximation in the long-wavelength limit (Bohren and Huffman, 1983; Mayergoyz and Zhang, 2007; Fredkin and Mayergoyz, 2003). This approximation, combined with a boundary-element method, allows us to study plasmon modes in single and multiple nanoparticles (Mayergoyz et al., 2005; Mayergoyz and Zhang, 2006; Davis et al., 2009c; Zhang et al., 2007; Davis et al., 2009b; Davis et al., 2009a; Mäkitalo and Kauranen, 2014; Jung et al., 2010; Hohenester and Trugler, 2012), and how nanoparticles interact with electron beams (García de Abajo and Aizpurua, 1997; García de Abajo and Howie, 1997; de Abajo and Howie, 2002), substrates (Mortazavi et al., 2012; Chou et al., 2012), and even molecules (Davis et al., 2010; Antosiewicz et al., 2011). Recent developments in computational modeling of nanoplasmonics using BEM and other approaches were reviewed by Solis and co-workers (Solis et al., 2014).

In the work by Davis and co-workers (Davis et al., 2010), the authors calculate an effective solvent dielectric constant due to the presence of an analyte molecule, represented as a point dipole. Unfortunately, the simplifications in this approach do not account for possible changes due to the analyte geometry, permittivity, dipole moment or charge. Another similar effort is presented by Antosiewicz and co-workers (Antosiewicz et al., 2011), where they describe the shift in resonance frequency of a nanoparticle when a very small analyte adheres to it. This approach requires a proper estimation of the analyte polarizability, and it is also difficult to include analyte details. Haes and Van Duyne (Haes and Van Duyne, 2002) model the LSPR sensor sensitivity using the ideas of Jung and co-workers (Jung et al., 1998), also based on finding an effective permittivity. Their results of the sensitivity relation with the SAM thickness, *i.e.* the analyte-sensor distance, match experimental results, however, this approach precludes investigation of structural details and single analyte response, because it uses the permittivity of the analyte solution.

Chapter 3

Methods

3.1 Integral formulation of molecular systems

3.1.1 Single-surface models

Following the work by Yoon and Lenhoff (Yoon and Lenhoff, 1990), we can obtain an integral formulation of the system of PDEs in Equation (2.62) by applying Green's second identity, which gives

$$\begin{aligned} \phi_{1,\Omega_1} + K_L^{\Omega_1}(\phi_{1,\Gamma}) - V_L^{\Omega_1}\left(\frac{\partial}{\partial \mathbf{n}}\phi_{1,\Gamma}\right) &= \frac{1}{\epsilon_1}\sum_{k=0}^{N_q}\frac{q_k}{4\pi|\mathbf{r}_{\Omega_1}-\mathbf{r}_k|} \quad \text{on } \Omega_1 \\ \phi_{2,\Omega_2} - K_Y^{\Omega_2}(\phi_{2,\Gamma}) + V_Y^{\Omega_2}\left(\frac{\partial}{\partial \mathbf{n}}\phi_{2,\Gamma}\right) &= 0 \quad \text{on } \Omega_2, \end{aligned} \tag{3.1}$$

where $\phi_{1,\Omega_1} = \phi_1(\mathbf{r}_{\Omega_1})$ is the potential ϕ_1 evaluated anywhere in the region Ω_1 and $\phi_{1,\Gamma} = \phi_1(\mathbf{r}_\Gamma)$ is the potential ϕ_1 evaluated at the boundary Γ. Also, K and V are the single- and double-layer potentials defined in Section 2.2. The limits $\mathbf{r}_{\Omega_1} \to \mathbf{r}_\Gamma$ and $\mathbf{r}_{\Omega_2} \to \mathbf{r}_\Gamma$ on Equation (3.1) results in an integral formulation of the system of PDEs

$$\begin{aligned} \frac{\phi_{1,\Gamma}}{2} + K_L^{\Gamma_1}(\phi_{1,\Gamma}) - V_L^{\Gamma}\left(\frac{\partial}{\partial \mathbf{n}}\phi_{1,\Gamma}\right) &= \frac{1}{\epsilon_1}\sum_{k=0}^{N_q}\frac{q_k}{4\pi|\mathbf{r}_{\Gamma}-\mathbf{r}_k|} \quad \text{on } \Gamma \\ \frac{\phi_{2,\Gamma}}{2} - K_Y^{\Gamma}(\phi_{2,\Gamma_1}) + V_Y^{\Gamma}\left(\frac{\partial}{\partial \mathbf{n}}\phi_{2,\Gamma}\right) &= 0 \quad \text{on } \Gamma. \end{aligned} \tag{3.2}$$

Then, we can leave everything in terms of quantities from the interior side only by enforcing the corresponding interface conditions (Equation (2.62)),

$$\begin{aligned}
\frac{\phi_{1,\Gamma}}{2} + K_L^\Gamma(\phi_{1,\Gamma}) - V_L^\Gamma\left(\frac{\partial}{\partial \mathbf{n}}\phi_{1,\Gamma}\right) &= \frac{1}{\epsilon_1}\sum_{k=0}^{N_q}\frac{q_k}{4\pi|\mathbf{r}_\Gamma - \mathbf{r}_k|} \quad \text{on } \Gamma \\
\frac{\phi_{1,\Gamma}}{2} - K_Y^\Gamma(\phi_{1,\Gamma}) + \frac{\epsilon_1}{\epsilon_2}V_Y^\Gamma\left(\frac{\partial}{\partial \mathbf{n}}\phi_{1,\Gamma}\right) &= 0 \quad \text{on } \Gamma.
\end{aligned} \tag{3.3}$$

which can be re-written in matrix form, as

$$\begin{bmatrix} \frac{1}{2} + K_L^\Gamma & -V_L^\Gamma \\ \frac{1}{2} - K_Y^\Gamma & \frac{\epsilon_1}{\epsilon_2}V_Y^\Gamma \end{bmatrix} \begin{bmatrix} \phi_{1,\Gamma} \\ \frac{\partial}{\partial \mathbf{n}}\phi_{1,\Gamma} \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{N_q}\frac{q_k}{4\pi|\mathbf{r}_\Gamma - \mathbf{r}_k|} \\ 0 \end{bmatrix} \tag{3.4}$$

3.1.2 Multi-surface models

A model with only one surface representing the SES is not enough to consider solvent-filled cavities and Stern layers. For example, the protein represented in Figure 3·1 contains two solvent-filled cavities and the external Stern layer, and requires 4 surfaces to be modeled. This system is modeled by Equation (3.5).

$$\begin{aligned}
&\nabla^2\phi_1(\mathbf{r}) = \kappa^2\phi_2(\mathbf{r}) && \text{in } \Omega_1 \\
&\nabla^2\phi_2(\mathbf{r}) = 0 && \text{in } \Omega_2 \\
&\nabla^2\phi_3(\mathbf{r}) = -\sum_k \frac{q_k}{\epsilon_3}\delta(\mathbf{r},\mathbf{r}_k) && \text{in } \Omega_3 \\
&\nabla^2\phi_4(\mathbf{r}) = \kappa^2\phi_4(\mathbf{r}) && \text{in } \Omega_4 \\
&\nabla^2\phi_5(\mathbf{r}) = \kappa^2\phi_5(\mathbf{r}) && \text{in } \Omega_5 \\
&\phi_1 = \phi_2, \quad \epsilon_1\frac{\partial\phi_1}{\partial \mathbf{n}} = \epsilon_2\frac{\partial\phi_2}{\partial \mathbf{n}} && \text{on interface } \Gamma_1 \\
&\phi_2 = \phi_3, \quad \epsilon_2\frac{\partial\phi_2}{\partial \mathbf{n}} = \epsilon_3\frac{\partial\phi_3}{\partial \mathbf{n}} && \text{on interface } \Gamma_2 \\
&\phi_3 = \phi_4, \quad \epsilon_3\frac{\partial\phi_3}{\partial \mathbf{n}} = \epsilon_4\frac{\partial\phi_4}{\partial \mathbf{n}} && \text{on interface } \Gamma_3 \\
&\phi_3 = \phi_5, \quad \epsilon_3\frac{\partial\phi_3}{\partial \mathbf{n}} = \epsilon_5\frac{\partial\phi_5}{\partial \mathbf{n}} && \text{on interface } \Gamma_4
\end{aligned} \tag{3.5}$$

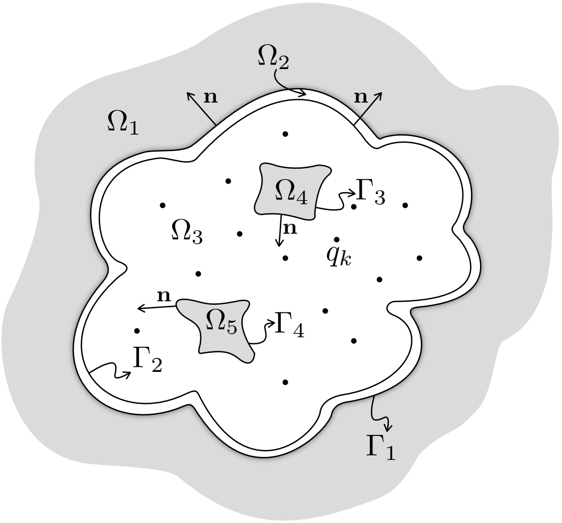


Figure 3·1: Representation of a protein with a Stern layer and two solvent-filled cavities. Gray regions contain salt, white regions don't.

We can extend the derivation for the integral formulation in Equation (3.3) to account for multiple surfaces. Starting from Equation (3.5) and considering all normal vectors pointing out of the region enclosed by the corresponding surface, we obtain a system of integral equations represented by the matrix in Equation (3.6).

The matrix in Equation (3.6) becomes increasingly complicated with the number of surfaces, and is difficult even for a small system like the one in Figure 3·1. Each surface generates two equations: one when the limiting procedure of Equation (3.2) is done from the region enclosed by the surface (the interior equation), and another one when the limit is taken from the region that surrounds the surface (exterior equation). The structure of this matrix reveals a 2×2 block diagonal corresponding to the double- and single-layer operators of ϕ_{i,Γ_j} and $\partial\phi_{i,\Gamma_j}/\partial\mathbf{n}$, respectively, evaluated on the same surface Γ_j. Blocks that are outside the diagonal are evaluations performed on surfaces different than Γ_j.

$$\begin{bmatrix} \frac{1}{2}+K_L^{\Gamma_1} & -V_L^{\Gamma_1} & -K_L^{\Gamma_1} & \frac{\epsilon_3}{\epsilon_2}V_L^{\Gamma_1} & & & & \\ \frac{1}{2}-K_Y^{\Gamma_1} & \frac{\epsilon_2}{\epsilon_1}V_Y^{\Gamma_1} & & & & & & \\ & & \frac{1}{2}+K_L^{\Gamma_2} & -V_L^{\Gamma_2} & -K_L^{\Gamma_2} & \frac{\epsilon_4}{\epsilon_3}V_L^{\Gamma_2} & -K_L^{\Gamma_2} & \frac{\epsilon_5}{\epsilon_3}V_L^{\Gamma_2} \\ K_L^{\Gamma_2} & -V_L^{\Gamma_2} & \frac{1}{2}-K_L^{\Gamma_2} & \frac{\epsilon_3}{\epsilon_2}V_L^{\Gamma_2} & & & & \\ & & & & \frac{1}{2}+K_Y^{\Gamma_3} & -V_Y^{\Gamma_3} & & \\ & & K_L^{\Gamma_3} & -V_L^{\Gamma_3} & \frac{1}{2}-K_L^{\Gamma_3} & \frac{\epsilon_4}{\epsilon_3}V_L^{\Gamma_3} & -K_L^{\Gamma_3} & \frac{\epsilon_5}{\epsilon_3}V_L^{\Gamma_3} \\ & & & & & & \frac{1}{2}+K_Y^{\Gamma_4} & -V_Y^{\Gamma_4} \\ & & K_L^{\Gamma_4} & -V_L^{\Gamma_4} & -K_L^{\Gamma_4} & \frac{\epsilon_4}{\epsilon_3}V_L^{\Gamma_4} & \frac{1}{2}-K_L^{\Gamma_4} & \frac{\epsilon_5}{\epsilon_3}V_L^{\Gamma_3} \end{bmatrix} \cdot \begin{bmatrix} \phi_{2,\Gamma_1} \\ \frac{\partial}{\partial \mathbf{n}}\phi_{2,\Gamma_1} \\ \phi_{3,\Gamma_2} \\ \frac{\partial}{\partial \mathbf{n}}\phi_{3,\Gamma_2} \\ \phi_{4,\Gamma_3} \\ \frac{\partial}{\partial \mathbf{n}}\phi_{4,\Gamma_3} \\ \phi_{5,\Gamma_4} \\ \frac{\partial}{\partial \mathbf{n}}\phi_{5,\Gamma_4} \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ \sum_{k=0}^{N_q} \frac{q_k}{4\pi|\mathbf{r}_{\Gamma_2}-\mathbf{r}_k|} \\ 0 \\ 0 \\ \sum_{k=0}^{N_q} \frac{q_k}{4\pi|\mathbf{r}_{\Gamma_3}-\mathbf{r}_k|} \\ 0 \\ \sum_{k=0}^{N_q} \frac{q_k}{4\pi|\mathbf{r}_{\Gamma_4}-\mathbf{r}_k|} \end{bmatrix} \tag{3.6}$$

In our implementation, we use the approach presented by Altman and co-workers (Altman et al., 2009) to deal with multi-surface models. In that work, they classify interactions in four categories, depending on the position of the surface with the evaluation point relative to the surface that contains the strengths of the single- and double-layer potentials, $\partial\phi_{i,\Gamma}/\partial\mathbf{n}$ and $\phi_{i,\Gamma}$:

> *Self-surface interior operator:* evaluation point and strengths of the single- and double layer potentials are on the same surface. In the limiting procedure, the evaluation point approaches the surface from the inside, hence, the result is added to the interior equation. For example, if the interior region is modeled

by the Laplace or Poisson equation, the operator is

$$\frac{\phi_{i,\Gamma}}{2} + K_L^{\Gamma}(\phi_{i,\Gamma}) - V_L^{\Gamma}\left(\frac{\partial}{\partial \mathbf{n}}\phi_{i,\Gamma}\right). \tag{3.7}$$

Self-surface exterior operator: similar to the previous case, the evaluation point and strengths are on the same surface. However, this time, the limiting procedure moves the evaluation point to the surface from the outside, hence, the result is added to the exterior equation. For example, if the exterior region is modeled by the Laplace or Poisson equation, the operator is

$$\frac{\phi_{i,\Gamma}}{2} - K_L^{\Gamma}(\phi_{i,\Gamma}) + \frac{\epsilon_{\text{in}}}{\epsilon_{\text{out}}} V_L^{\Gamma}\left(\frac{\partial}{\partial \mathbf{n}}\phi_{i,\Gamma}\right). \tag{3.8}$$

Non-self exterior operator: the evaluation point resides in a surface that is outside the region enclosed by the surface with the strengths of the single- and double layer potentials. The result has to be added to the exterior equation if both surfaces share a common external region, and to the interior equation if they don't. For example, in Figure 3·1, Γ_4 and Γ_3 interact in this way and the result is added to the corresponding external equation, whereas if the evaluation point is in Γ_2 and the strength of the potentials in Γ_3 or Γ_4, the result would be added to the interior equation of Γ_2. When the region outside the surface with the strengths is modeled by the Laplace or Poisson equation, the operator is

$$-K_L^{\Gamma_j}(\phi_{i,\Gamma_k}) + \frac{\epsilon_{\text{in}}}{\epsilon_{\text{out}}} V_L^{\Gamma_j}\left(\frac{\partial}{\partial \mathbf{n}}\phi_{i,\Gamma_k}\right). \tag{3.9}$$

Non-self interior operator: evaluation point resides in a surface that is inside the region enclosed by the surface with the strengths of the single- and double layer potentials, and the result has to be added to the exterior equation. For example, in Figure 3·1, this operator will be used when the evaluation point is in Γ_1 and the strengths of the potentials are on Γ_2, and the result will be added to the interior equation of Γ_1. If the region enclosed by the surface with the strengths is modeled by the Laplace or Poisson equation, the operator would be

$$K_L^{\Gamma_j}(\phi_{i,\Gamma_k}) - V_L^{\Gamma_j}\left(\frac{\partial}{\partial \mathbf{n}}\phi_{i,\Gamma_k}\right). \tag{3.10}$$

3.2 Integral formulation for protein surface interaction

We can include a surface of prescribed potential or charge to the system of partial differential equations in Equation (2.62) to simulate the interaction of proteins with charged surfaces. For example, Figure 3·2 is modeled by

$$\begin{aligned}
\nabla^2\phi_1(\mathbf{r}) &= -\sum_k \frac{q_k}{\epsilon_1}\delta(\mathbf{r},\mathbf{r}_k) && \text{in solute } (\Omega_2)\\
\nabla^2\phi_2(\mathbf{r}) &= \kappa^2\phi_2(\mathbf{r}) && \text{in solvent } (\Omega_1)\\
\phi_1 &= \phi_2 && \text{on interface, } \Gamma_1\\
\epsilon_1\frac{\partial\phi_1}{\partial\mathbf{n}} &= \epsilon_2\frac{\partial\phi_2}{\partial\mathbf{n}} &&\\
\phi_2 = \phi_0 \text{ or } -\epsilon_2\frac{\partial\phi_2}{\partial\mathbf{n}} &= \sigma_0 && \text{on surface, } \Gamma_2
\end{aligned} \tag{3.11}$$

We can also write an integral formulation of Equation (3.11), doing a similar derivation that led to Equation (3.3). For a surface with prescribed potential, the final system in matrix form would be

$$\begin{bmatrix} \frac{1}{2}+K_L^{\Gamma_1} & -V_L^{\Gamma_1} & 0 \\ \frac{1}{2}-K_Y^{\Gamma_1} & \frac{\epsilon_1}{\epsilon_2}V_Y^{\Gamma_1} & V_Y^{\Gamma_1} \\ -K_Y^{\Gamma_2} & \frac{\epsilon_1}{\epsilon_2}V_Y^{\Gamma_2} & V_Y^{\Gamma_2} \end{bmatrix} \begin{bmatrix} \phi_{1,\Gamma_1} \\ \frac{\partial}{\partial\mathbf{n}}\phi_{1,\Gamma_1} \\ \frac{\partial}{\partial\mathbf{n}}\phi_{2,\Gamma_2} \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{N_q}\frac{q_k}{4\pi|\mathbf{r}_{\Gamma_1}-\mathbf{r}_k|} \\ K_Y^{\Gamma_1}(\phi_0) \\ -\left(\frac{1}{2}-K_Y^{\Gamma_2}\right)(\phi_0) \end{bmatrix} \tag{3.12}$$

Like in Equation (3.6), the matrix in Equation (3.12) gets increasingly complicated with the number of surfaces in the model. For these problems, we also used the approach suggested by Altman and co-workers (Altman et al., 2009), described in

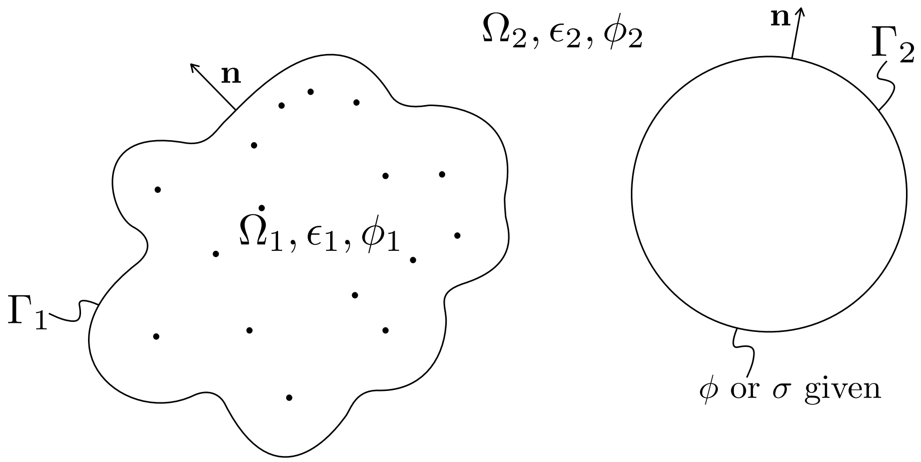

$\Omega_2, \epsilon_2, \phi_2$
n
n
Γ_2
$\Omega_1, \epsilon_1, \phi_1$
Γ_1
ϕ or σ given

Figure 3·2: Representation of a protein interacting with a surface with prescribed charge or potential.

Section 3.1.2. However, in this case, we need to account for surfaces with prescribed charge or potential that generate only the external equation.

3.3 Numerical solution through the boundary element method

3.3.1 Discretization of the equations

The elements of the matrix systems presented in Section 3.1 are the single- and double-layer operators defined in Equation (2.107) and (2.113). To solve the system numerically, we discretize the surface into N_p panels, and use the discrete version of the layer operators from Equation (2.116). Then, we obtain a linear system using center collocation, which means evaluating the equation placing $\mathbf{r}_i$ in the center of each panel. This way, the matrix systems in Section 3.1 become linear systems from which we can solve for the discrete ϕ and $\partial\phi/\partial\mathbf{n}$. The discretized matrix looks exactly like Equation (3.4) or any other matrix equation in Section 3.1, but now the elements of the matrix are sub-matrices of size $N_p \times N_p$ rather than integral operators. Each element of a sub-matrix is an integral over one panel Γ_j, with $\mathbf{r}_i$ located at the center of the collocation panel Γ_i. Also, the terms on the right-hand side and the

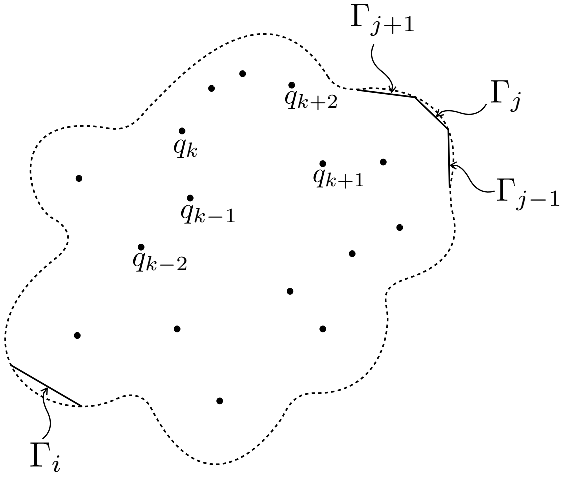


Figure 3·3: Discretization of a molecular surface. Γ_i is the panel where the collocation point resides and Γ_j the panel being integrated.

unknown vector in Equation (3.4) are sub-vectors of size N_p, where each element is the evaluation on a panel Γ_i. Element ij of a sub-matrix looks like Equation (2.118) and, in discretized form, element i of ϕ_{1,Γ_1}, $\frac{\partial}{\partial \mathbf{n}}\phi_{1,\Gamma_1}$ and $\sum_{k=0}^{N_q} \frac{q}{4\pi|\mathbf{r}_{\Gamma_1}-\mathbf{r}_k|}$ in Equation (3.12) are

$$\begin{aligned} \phi_{1,\Gamma_1} &= \phi_1(\mathbf{r}_i) \\ \frac{\partial}{\partial \mathbf{n}}\phi_{1,\Gamma_1} &= \frac{\partial}{\partial \mathbf{n}}\phi_1(\mathbf{r}_i) \\ \sum_{k=0}^{N_q} \frac{q}{4\pi|\mathbf{r}_{\Gamma_1}-\mathbf{r}_k|} &= \sum_{k=0}^{N_q} \frac{q}{4\pi|\mathbf{r}_i-\mathbf{r}_k|}. \end{aligned} \tag{3.13}$$

In this work, we use the freely available software `MSMS`[1] (Sanner et al., 1995) to compute the discretized SES. Figure 3·4 shows an example of a mesh generated with `MSMS`.

---

[1] `http://mgl.scripps.edu/people/sanner/html/msms_home.html`

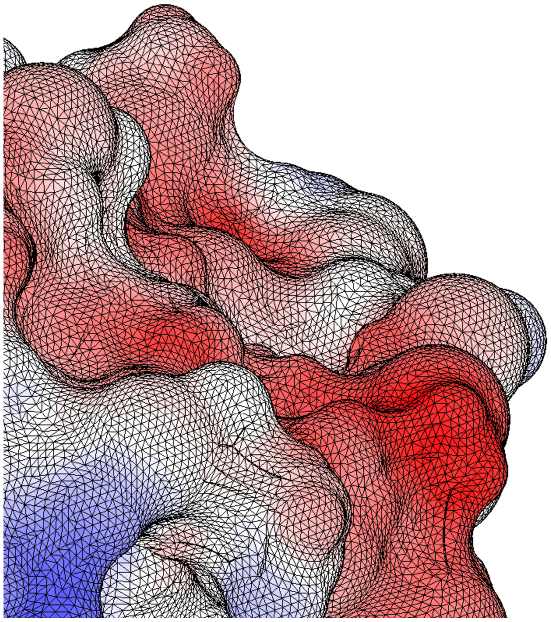


Figure 3·4: Close-up view of the SES of Lysozyme discretized in flat triangles using `MSMS`. Colors represent potential on the surface.

3.3.2 Energy calculation with the boundary element method

Figure 3·2 shows an arrangement with three sources of electrostatic energy: Coulombic energy from the point charges, surface energy due to Γ_2, and solvation energy caused by the influence of the solvent outside the protein region. Next, we detail how to compute the surface and salvation free energy with BEM.

Surface free energy. To compute the surface free energy described in Section 2.1.5, we can plug in the solution of the system in Equation (3.12) to the discretized version of Equation (2.66). In this case, we consider constant values of ϕ and $\frac{\partial \phi}{\partial \mathbf{n}}$ per panel, and the equation takes the form

$$\begin{aligned} F_{\text{surface}} &= \frac{1}{2}\sum_{j=1}^{N_p} \phi(\mathbf{r}_j)\sigma_{0j}A_j, \text{ and} \\ F_{\text{surface}} &= -\frac{1}{2}\sum_{j=1}^{N_p} \phi_{0j}\sigma(\mathbf{r}_j)A_j. \end{aligned} \tag{3.14}$$

where A_j is the area of panel j, and $\sigma = \epsilon\frac{\partial\phi}{\partial\mathbf{n}}$.

Solvation free energy. In the context of boundary element methods, we can compute the reaction potential in Equation (2.65) by subtracting the Coulomb effect from Equation (3.1) in the protein region, which yields

$$\phi_{\text{reac},\mathbf{r}_k} = -K_L^{\mathbf{r}_k}(\phi_{1,\Gamma_1}) + V_L^{\mathbf{r}_k}\left(\frac{\partial}{\partial\mathbf{n}}\phi_{1,\Gamma_1}\right). \tag{3.15}$$

Equation (2.64) requires evaluating ϕ_{reac} for each point charge location $\mathbf{r}_k$. We obtain this by discretizing Equation (3.15) and using the solution of the linear system in Equation (3.12) as input.

3.3.3 Integral evaluation

We use Gauss quadrature rules to evaluate the integrals in Equation (2.118). However, the free space Green's function of both the Laplace and the linearized Poisson-Boltzmann equations contain a $1/r$ singularity, which makes it difficult to obtain good accuracy when the integral is singular or near-singular. This is why we define three different integration regions:

> *Far-away integrals:* When the collocation point and the integration panel are far away, a low order integration scheme is enough to obtain good accuracy, for example, with 1, 3 or 4 Gauss quadrature points per boundary element. We use a representative length of the integrated triangle, $L = \sqrt{2 \cdot \text{Area}}$, to

define a threshold from which collocation points are considered to be far-away. A common threshold value used in this work is $2L$.

Near-singular integrals: In cases where the collocation point is closer than the far-away region threshold, but it is not in the same integration panel, we use higher order Gauss quadrature rules. Common numbers of Gauss points in this region are 19, 25 or 37 points per triangular element.

Singular integrals: The singularity that appears from having the collocation point in the integrated panel makes it impossible to compute the integral with standard Gauss integration schemes. For singular integrals, we use a semi-analytical technique from Zhu and co-workers (Zhu et al., 2001) which can perform accurate integration of the Laplace and Yukawa potentials. This method is briefly discussed next.

Semi analytical integration The projection of the collocation point on the panels' plane with the vertices creates three triangles, denoted by Δ_{blue}, Δ_{red} and Δ_{green} in Figure 3·5. The integration of any function over the panel can be written as the sum of the integral of the same function over those three triangles. Using polar coordinates centered at the projection of the collocation point, we can integrate the radial component analytically and the angular component with one-dimensional Gauss quadrature rules. Since the limits of the integral over the radial component are $r = 0$ and the edge of the triangle, the integrand is effectively being evaluated on the edge of the triangle, where the function varies more smoothly and Gauss quadrature rules can resolve the integration.

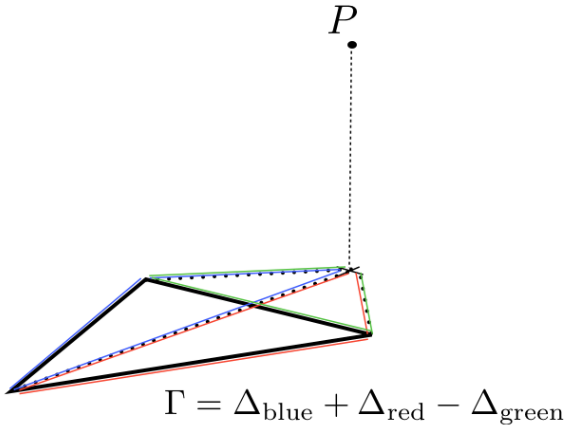

P
$\Gamma = \Delta_{\text{blue}} + \Delta_{\text{red}} - \Delta_{\text{green}}$

Figure 3·5: Sketch of semi analytical integration scheme from (Zhu et al., 2001)

Then, the expressions for semi-analytical integration are

$$\int\limits_{\Gamma} G_L \mathrm{d}\Gamma = \sum_{\text{edges}} \int\limits_{\theta_{V_1}}^{\theta_{V_2}} \left[|\mathbf{r}(\theta)| - h \right] \mathrm{d}\theta \tag{3.16}$$

$$\int\limits_{\Gamma} \frac{\partial G_L}{\partial \mathbf{n}} \mathrm{d}\Gamma = \sum_{\text{edges}} \int\limits_{\theta_{V_1}}^{\theta_{V_2}} \left[\frac{h}{|\mathbf{r}(\theta)|} - \operatorname{sign}(h) \right] \mathrm{d}\theta$$

$$\int\limits_{\Gamma} G_Y \mathrm{d}\Gamma = \sum_{\text{edges}} - \int\limits_{\theta_{V_1}}^{\theta_{V_2}} \left[\frac{\exp(-\kappa |\mathbf{r}(\theta)|) - \exp(-\kappa |h|)}{\kappa} \right] \mathrm{d}\theta$$

$$\int\limits_{\Gamma} \frac{\partial G_Y}{\partial \mathbf{n}} \mathrm{d}\Gamma = \sum_{\text{edges}} - \int\limits_{\theta_{V_1}}^{\theta_{V_2}} \left[\exp(-\kappa |h|) \operatorname{sign}(h) - \frac{h}{|\mathbf{r}(\theta)|} \exp(-\kappa |\mathbf{r}(\theta)|) \right] \mathrm{d}\theta$$

where h is the normal distance between the panel plane and the collocation point, θ_{V_i} is the angle of vertex V_i in the local polar coordinate system, and $|\mathbf{r}(\theta)|$ the distance between the projection of the collocation point on the triangle panel and the vertex, for angle θ.

3.3.4 Preconditioning

The matrix generated by Yoon and Lenhoff's formulation (Yoon and Lenhoff, 1990) has a condition number that depends on the mesh size (Liang and Subramaniam, 1997). The condition number gives an idea of the response of the system to variations in the right-hand side, and it is directly related to the number of iterations that an iterative solver requires to achieve the desired tolerance. Then, in our implementation, the amount of iterations grows with the number of discretization elements, affecting the time to solution of the GMRES. This issue can be alleviated with preconditioners.

A preconditioner is an operator, in this case a matrix ($\mathbb{P}$), that decreases the condition number of matrix $\mathbf{A}$ in the linear system $\mathbf{Ax} = \mathbf{b}$. The most common preconditioning approach is the left preconditioner, where $\mathbb{P}$ is multiplied from the left on both sides of the linear system:

$$\mathbb{P}\mathbf{Ax} = \mathbb{P}\mathbf{b}. \tag{3.17}$$

The perfect preconditioner is the inverse of the matrix, and preconditioners are more effective the closer they are to it ($\mathbb{P} \approx \mathbf{A}^{-1}$). In `PyGBe`, we use a block-diagonal preconditioner described by Altman and co-workers (Altman et al., 2009). This preconditioner is the inverse of the matrix that contains the diagonals of the blocks corresponding to self-surface operators (as defined in Section 3.1.2). Looking at Equation (3.12), surfaces with set charge or potential contribute only one self-surface operator block to the matrix, and the inverse of its diagonal is trivial to calculate. Each molecular surfaces provides four self-surface operator blocks to the matrix, arranged in a 2×2 block matrix

$$\mathbf{A}_s = \begin{bmatrix} \frac{1}{2} + K_L^\Gamma & -V_L^\Gamma \\ \frac{1}{2} - K_Y^\Gamma & \frac{\epsilon_1}{\epsilon_2} V_Y^\Gamma \end{bmatrix}, \tag{3.18}$$

which matches the matrix in Equation (3.4), where we had a system with only one surface. Using LU factorization, we can show that for any 2 × 2 block matrix $\mathbf{M}$, its inverse is

$$\mathbf{M} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \quad \mathbf{M}^{-1} = \begin{bmatrix} M_{11,\text{inv}} & M_{12,\text{inv}} \\ M_{21,\text{inv}} & M_{22,\text{inv}} \end{bmatrix}, \tag{3.19}$$

where,

$$\begin{aligned} M_{11,\text{inv}} &= M_{11}^{-1} + M_{11}^{-1} M_{12} \left(M_{22} - M_{21} M_{11}^{-1} M_{12} \right)^{-1} M_{21} M_{11}^{-1}, \\ M_{12,\text{inv}} &= -M_{11}^{-1} M_{12} \left(M_{22} - M_{21} M_{11}^{-1} M_{12} \right)^{-1}, \\ M_{21,\text{inv}} &= -\left(M_{22} - M_{21} M_{11}^{-1} M_{12} \right)^{-1} M_{21} M_{11}^{-1}, \text{ and} \\ M_{22,\text{inv}} &= \left(M_{22} - M_{21} M_{11}^{-1} M_{12} \right)^{-1}. \end{aligned} \tag{3.20}$$

If $\mathbf{A}_D$ is the tridiagonal matrix with the diagonal elements of the blocks of $\mathbf{A}_s$ in Equation (3.18), then the calculation of the preconditioner $\mathbb{P} = \mathbf{A}_D^{-1}$ using Equation (3.20) is trivial, as all blocks are diagonals.

In cases like Equation (3.6), where there are multiple surfaces, the preconditioner $\mathbb{P}$ is the collection of $\mathbf{A}_D^{-1}$ for each individual surface, placed in the corresponding location of the matrix. This happens because the self-surface operators are independent of each other.

Matrix $\mathbb{P}$ consists of diagonal blocks, then, we can store it in a sparse format for efficient memory usage and matrix-vector multiplication. In our implementation, we apply the preconditioner by multiplying it to the right-hand-side of the linear system, and then to the vector resulting from the matrix-vector product in each GMRES iteration.

3.4 Implementation details of the treecode

3.4.1 Computation of the normal derivative

The evaluation of a matrix-vector product involving the double-layer potential operator in Equation (2.116) requires the computation of the normal derivative of G_L where the normal vector is in the integration panel, with the source points of the treecode. For a target $\mathbf{x} = (x_1, x_2, x_3)$ interacting with a source $\mathbf{y} = (y_1, y_2, y_3)$, and a normal vector to the source panel $\mathbf{n} = (n_1, n_2, n_3)$, the normal derivative is

$$\frac{\partial}{\partial \mathbf{n}} G_L(\mathbf{x}, \mathbf{y}) = \nabla_{\mathbf{y}} G_L(\mathbf{x}, \mathbf{y}) \cdot \mathbf{n} = \frac{\partial G_L}{\partial y_1} n_1 + \frac{\partial G_L}{\partial y_2} n_2 + \frac{\partial G_L}{\partial y_3} n_3. \tag{3.21}$$

The derivative of Equation (2.121) with respect to y_l,

$$\frac{\partial}{\partial y_l} V(\mathbf{x}_i) = -\sum_{||\mathbf{k}||=0}^{P} \frac{1}{\mathbf{k}!} D_{\mathbf{y}}^{\mathbf{k}} \psi(\mathbf{x}_i, \mathbf{y}_c) \sum_j q_j k_l (\mathbf{y}_c - \mathbf{y}_j)^{\mathbf{k}-\mathbf{e}_l}, \tag{3.22}$$

affects only the multipole ($m_c^{\mathbf{k}}$ in Equation (2.123)), and not the coefficients of the Taylor expansion ($a^{\mathbf{k}}$ in Equation (2.122)). Hence, we can use the treecode to compute the derivative in the l^{th} component by modifying the particle-to-multipole calculation of Equation (2.123), and computing instead

$$m_{c,l}^{\mathbf{k}} = -\sum_j q_j k_l (\mathbf{y}_c - \mathbf{y}_j)^{\mathbf{k}-\mathbf{e}_l} = -m_c^{\mathbf{k}} \frac{k_l}{y_{c,l} - y_{j,l}}. \tag{3.23}$$

Meanwhile, the rest of the steps of the treecode remain unchanged.

Equation (3.21) shows that BEM requires a weighted sum of the derivatives. The order of the multipole is not altered by the derivative —$m_{c,1}^{\mathbf{k}}$, $m_{c,2}^{\mathbf{k}}$ and $m_{c,3}^{\mathbf{k}}$ are multiplied by the same $a^{\mathbf{k}}$—, which allows us to factorize the multipole expansions of the three derivatives into one

$$m_c'^{\mathbf{k}} = \sum_{l=1}^{3} m_{c,l}^{\mathbf{k}} = -\sum_{l=1}^{3} \sum_j q_{j,l} k_l (\mathbf{y}_c - \mathbf{y}_j)^{\mathbf{k}-\mathbf{e}_l}. \tag{3.24}$$

In our implementation, we include the effect of $\mathbf{n}$ in $q_{j,l}$. The weight $q_{j,l}$ is different in each direction because it is multiplied by the corresponding component n_l, and $q_{j,l} = q_j \cdot n_l$.

3.4.2 Efficiency considerations in BEM

Tree construction. BEM uses the treecode once every GMRES iteration, but the sources and targets remain at the same location throughout the entire linear solver. Because of this, we only need to construct the tree once, and we do it in terms of boundary elements rather than Gauss quadrature points.

Tree vectorization. Two target points that are close to each other will interact with almost the same boxes in a multipole-to-particle and particle-to-particle fashion. We can identify close-by targets by grouping them in a tree structure (similar to the tree of source points), and considering targets from the same low-level box as being close. Then, instead of traversing the tree for each target point, we can do it once per box of targets, and all targets of a box will interact with the same boxes as multipole-to-particle and particle-to-particle.

Pre-computation of interaction list. For each box of target points we pre-compute the list of source point boxes with which it interacts, creating one list for multipole-to-particle, and one for particle-to-particle. This makes the tree traversal more efficient since we need to evaluate the MAC criterion only once, and for the rest of the treecode evaluations, we just loop over the interaction list.

3.4.3 GPU implementation

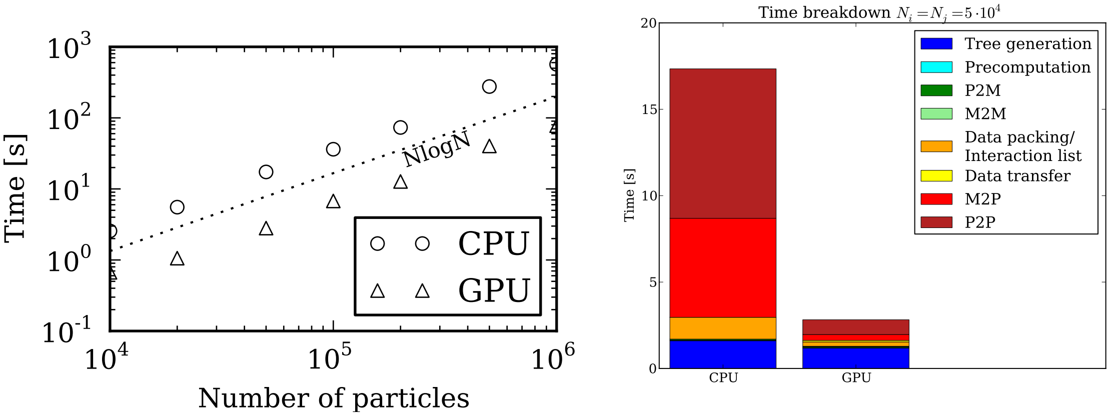
The most time consuming parts of the treecode are the calculation of the multipole-to-particle and the particle-to-particle interactions. Fortunately, those two computations map very well to the GPU architecture, which allows us to get interesting speedups by offloading them to the graphic card. We interface CUDA with Python using PYCUDA (Klöckner et al., 2012).

To obtain good performance out of the GPU, we need to manage the data such that parallelism is maximized, and data transfers between global and shared memory are reduced. We do this by packing the data of the source and target points such that all points of the same twig (or low-level) box are contiguous in memory, and we generate an array of pointers that points to the location where the first value of each twig is in the packed arrays. Likewise, we pack the values of the multipoles in a 1D array, and keep track of where the information of each box begins in this array.

Each box of target points has two interaction lists: one for multipole-to-particle and another for particle-to-particle. The interaction list stores pointers to the source-tree boxes that interact in the corresponding way with the targets inside the box that owns the interaction list. Since we know where the information is in the packed arrays, to evaluate, we just need to loop through one contiguous chunk of memory. This is extremely beneficial for the GPU since it maximizes the use of global-to-shared memory bandwidth.

In our implementation, we have two levels of parallelism. The finest level is at the target-point level: each CUDA thread performs all the operations for one target point. At a coarser level, targets of one same twig box will be computed by threads of the same thread block. This way, the CUDA threads know in which twig they are operating on by looking at their thread block ID.

We compare the CPU and GPU implementations of the treecode in Figure 3·6. This figure shows the time to compute the Yukawa interaction of the same number of target and source points, spread in a cubic box of size 1, using the treecode. In these calculations, $\kappa = 1$, $P = 6$ and $\theta = 0.5$. The tree structure for CPU runs had no more than 200 particles in the lowest-level (or twig) boxes, whereas in the GPU runs, this number was 1000. Figure 3·6a shows how the time to compute the treecode scales as $\mathcal{O}(N \log N)$, rather than $\mathcal{O}(N^2)$. For the case with $5 \cdot 10^4$ sources and targets, the GPU implementation of the multipole-to-particle (M2P) and particle-to-particle (P2P) speeds-up the code $\sim 5\times$. Since our intention is to use this in a BEM code, where the tree construction and data packing is done only once, there is no motivation to implement the rest of the code on the GPU.



(a) Time scaling.

(b) TIme breakdown.

Figure 3·6: GPU versus CPU implementations of the treecode.

3.5 Richardson extrapolation

Richardson extrapolation is a fundamental technique used in numerical computing to obtain an estimation of the exact solution from a series of computations using consecutive resolutions from coarser to finer. To be correctly applied, all of the computations used need to be converging to the exact value at a constant rate, meaning, they are in the asymptotic range. Under these conditions, an estimate of the exact solution can be written as:

$$f_{\text{exact}} \approx f_1 + \frac{f_1 - f_2}{r^p - 1}, \tag{3.25}$$

where f_1 and f_2 are the fine-grid and coarse-grid solutions, respectively; r is the mesh refinement ratio (h_2/h_1, where h can be spacing, area or volume) and p is the order of convergence.

From the calculations alone, we cannot ensure that the chosen grids are in the asymptotic range. Therefore, before using Equation (3.25), one should check that f_1 and f_2 are in that range. We can extract the *observed* order of convergence from three grid resolutions refined with constant ratio r, that is: $r = h_2/h_1 = h_3/h_2$. In that case:

$$p = \frac{\log\left(\frac{f_3-f_2}{f_2-f_1}\right)}{\log(r)}. \tag{3.26}$$

If the result from (3.26) matches the *expected* order of convergence of the method, it indicates that numerical computations for f_1, f_2 and f_3 are in the asymptotic range. For this reason, we need in general three calculations in the asymptotic convergence range to obtain an extrapolated value with Richardson extrapolation.

3.6 `PyGBe`

We implemented a code that computes the electrostatic potential and energy for systems like the ones in Equation (2.62) or Equation (3.11) with the integral formulation in Equation (3.4) and Equation (3.12). We call this implementation `PyGBe`—Python GPU for boundary elements—, and is freely available online[2] under MIT license (Cooper and Barba, 2013).

`PyGBe` is mainly a Python code, with the most computationally intensive parts written in C++ and CUDA—interfacing with PYCUDA (Klöckner et al., 2012)—, supporting graphic processing units (GPU). It solves Equation (3.4) and Equation (3.12) with a boundary element method (BEM) that assumes a piecewise constant distribution of the unknown values on the boundary, and accelerates calculations with the treecode algorithm.

---

[2] `https://github.com/barbagroup/pygbe`

In this work, all results were obtained using `PyGBe`, unless otherwise noted. Meshes were computed using the `MSMS` software[3] (Sanner et al., 1995), and for the van der Waals radii and charges of the atoms, we used the `pdb2pqr` (Dolinsky et al., 2004), with the built-in `amber` force field, unless otherwise noted.

---

[3] `http://mgl.scripps.edu/people/sanner/html/msms_home.html`

Chapter 4

Calculation of solvation and binding energies

The implicit-solvent model presented in Section 2.1 is regularly used for solvation and binding energy calculations, and is applicable in areas like protein engineering and drug design. This has motivated the development of several Poisson-Boltzmann solvers, such as APBS (Baker et al., 2001), `Delphi` (Gilson et al., 1987), MEAD (Bashford and Gerwert, 1992; Bashford, 1997), MIBPB (Geng et al., 2007), AFMPB (Lu et al., 2006), TABI (Geng and Krasny, 2013), among others. The most widely used are volumetric-based solvers, meaning that they create a discretization of the volume in which the equations apply. On the other hand, boundary-element methods (BEM), which discretize only surfaces, have become a popular alternative in biomolecular applications (Yoon and Lenhoff, 1990; Juffer et al., 1991; Zhou, 1993) and several BEM codes have been developed in recent years (Lu et al., 2006; Geng and Krasny, 2013; Altman et al., 2005; Bajaj et al., 2011).

In this Chapter, we show results using `PyGBe`. First, verifying the code against analytical solutions and the well-established APBS (Baker et al., 2001), and then, discussing the effect of features like solvent-filled cavities and Stern layers on the solvation and binding energies. All `PyGBe` runs were made on one core of an Intel Xeon X5650 CPU and one NVIDIA Tesla C2075 GPU, whereas all APBS runs were done on six Intel Xeon X5650 CPU cores.

The contents of this chapter have been previously published (Cooper et al., 2013c;

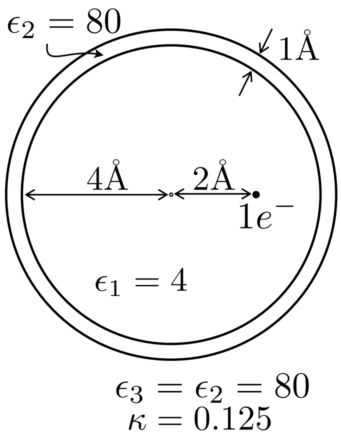


Figure 4·1: Sketch of the Kirkwood sphere with a Stern layer and an off center charge.

Cooper and Barba, 2013).

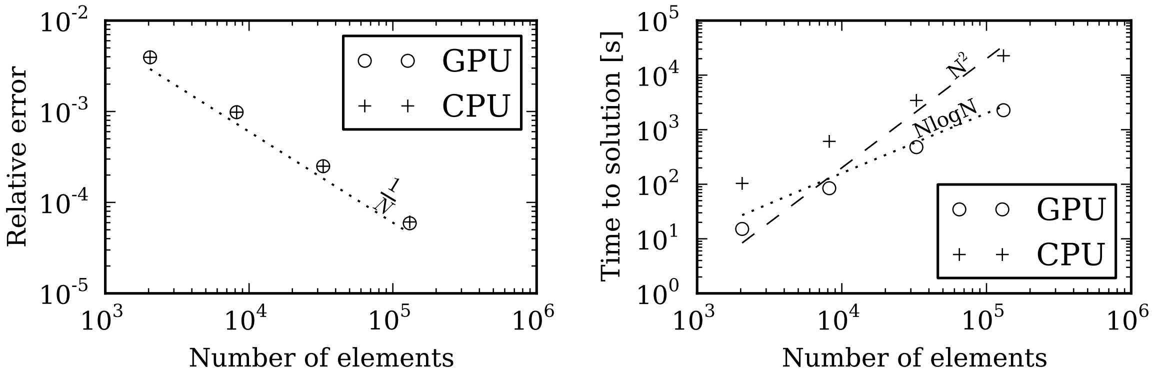
4.1 Verification of `PyGBe`

4.1.1 Comparison with analytical solutions

Kirkwood sphere

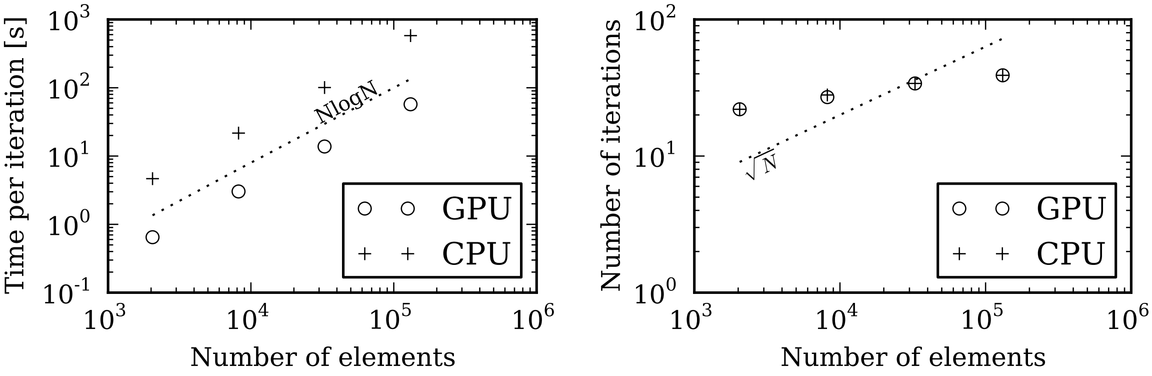
In 1934, Kirkwood (Kirkwood, 1934) derived a closed expression of the electrostatic potential in Equation (2.62) for a spherical molecule with a random distribution of point charges inside, and a Stern layer. Using this expression, we computed the solvation free energy for the setting in Figure 4·1, and compared it to `PyGBe` in Figure 4·2, for different mesh sizes. The 4Å radius spherical molecule in Figure 4·1 has a relative permittivity of $\epsilon_1 = 4$, an off centered charge at 2Å, and a 1Å-thick Stern layer; also, it is dissolved in water ($\epsilon_3 = \epsilon_2 = 80$) with salt at 150mM ($\kappa = 1/8\text{Å}^{-1}$).

For these runs, `PyGBe` used 3 Gauss points per far-away element, 19 Gauss points per element for near-singular integrals, and 9 Gauss points per triangle side for the semi-analytical integrals. The threshold of far-away elements was $2L$ where L is the characteristic length of the integrated triangle ($L = \sqrt{2 \cdot \text{Area}}$), and the GMRES tolerance was 10^{-6}. The treecode used a Taylor expansion of order P=12, and a multipole-acceptance criterion of $\theta = 0.5$.



(a) Relative error convergence.

(b) Time to solution.



(c) Average time per iteration.

(d) Number of iterations.

Figure 4·2: Results for for Kirkwood sphere with Stern layer using 2048, 8192, 32768 and 131072 boundary elements per surface.

The error plot in Figure 4·2a shows that the simulation is converging with the expected $\frac{1}{N}$ rate. The time to solution plot in Figure 4·2b, shows that the runtime scales slightly worse than $\mathcal{O}(N \log N)$; this is due to the fact that in this formulation the condition number depends on the size of the system; thus, as the number of elements increases, the GMRES solver needs more iterations to converge as Figure 4·2d shows. An important result is that the time per iteration does scale as $\mathcal{O}(N \log N)$, showing that it is the treecode which dominates the calculation runtime. The plots in Figure 4·2 include results for CPU-only runs, in order to compare timings and to check that both CPU and GPU codes converge in the expected way.

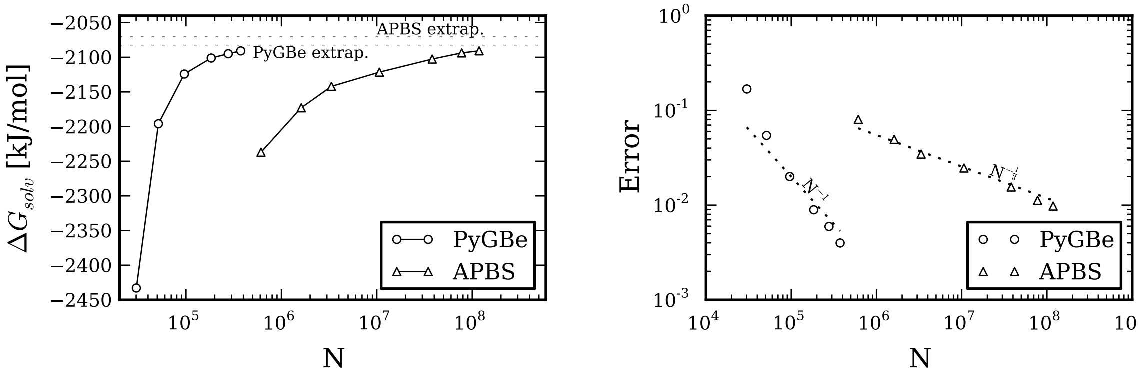
4.1.2 Comparison with existing codes

Here, we compare `PyGBe` with the finite-difference version of APBS. The test case is the solvation energy of the Lysozyme molecule (PDB code 1HEL), but this time taking the `pqr` input file from another publication (Yokota et al., 2011), rather than from `pdb2pqr`. The Lysozyme structure contains three solvent-filled cavities and we placed a Stern layer around the SES, all of which were considered when using `PyGBe` and APBS, for a fair comparison.

The relative permittivity in the protein region was $\epsilon_{\text{prot}} = 4$ and in the solvent region $\epsilon_{\text{solv}} = 80$, with $\kappa = 1/8\text{Å}^{-1}$, equivalent to 150mM of salt. `PyGBe` used 1 Guass point per far-away element —1.25L or more away from the collocation point—, 19 Gauss points per element for near-singular integrals and 5 Gauss points per triangle side for the semi-analytical integration. The GMRES tolerance was set to 10^{-4}, and the treecode contained only 2 terms of the Taylor expansion, with $\theta = 0.6$.

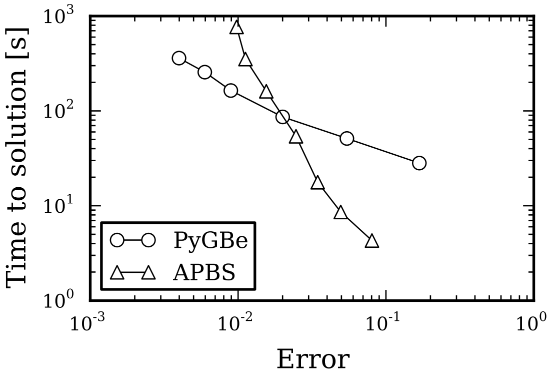
Figure 4·3a shows the computed solvation energy for different mesh sizes. As the mesh density increases, the discretization elements become smaller and the solution tends towards an asymptote, represented by the dotted line, which we calculated with Richardson extrapolation (Equation (3.25)). This value is -2070.47 [kJ/mol] for APBS and -2082.5 [kJ/mol] for `PyGBe`. This $\sim 0.5\%$ difference is expected because APBS is a volumetric, Cartesian-mesh solver that constrains the molecular-surface definition to a staircase representation, quite different from the surface-mesh representation used in `PyGBe`. Also, for volumetric approaches, point charges inside the molecule in general do not coincide with the mesh, and the required interpolation introduces smoothening of the point charges.

The errors plotted in Figure 4·3b were calculated using the corresponding extrapolated values as the base solution. The fact that the error is scaling with area for `PyGBe` and mesh spacing for APBS proves that those calculations are in the regime



(a) Solvation energy convergence with mesh refinement.

(b) Error convergence with mesh refinement.



(c) Runtime vs. estimated error.

Figure 4·3: Results for the lysozyme test, using `PyGBe` and **APBS**. Data sets, figure files and plotting scripts available under CC-BY (Cooper et al., 2013d).

with expected convergence, and that the error is dominated by the discretization. It is important to mention that mesh density for **APBS** corresponds to number of cells per unit volume, whereas for `PyGBe` it is number of boundary elements per unit area, thus we cannot directly compare the discretizations, and we simply used these metrics for graphical purposes to have both cases in one plot.

We measured the time to solution and plotted the comparison between **APBS** and `PyGBe` for different errors in Figure 4·3c, using the errors from Figure 4·3b. For low-accuracy calculations, **APBS** is faster than `PyGBe`; however, **APBS** has worse scaling, and they cross-over near the 2%-error mark. This indicates that a boundary element approach is more suitable when there is a need for more accuracy.

4.2 Effect of solvent-filled cavities and Stern layers

Equation (3.6) shows that the multi-surface models required to consider features like solvent-filled cavities and Stern layers can be problematic in BEM, whereas they are trivially considered in volumetric-based solvers. This section studies the effect of these features on solvation and binding energies, to understand when it is safe to neglect them, which is desirable in BEM approaches, this section investigates their effect.

Equation (2.69) yields the energy of interaction when there are many entities in the solvent. If the entities are bound, the energy of interaction is the binding energy ($F_{\text{interaction}} = F_{\text{bind}}$). The binding energy is a quantity of interest, for example, for protein docking calculations, relevant in drug design applications.

4.2.1 Solvation energy

Using the same simulation parameters that led to the results in Figure 4·3, Figure 4·4 examines the importance of considering cavities and Stern layers. These results correspond to a mesh-refinement study of solvation energy with Lysozyme modeled in four different ways: considering only the dielectric interface (“Single”), the dielectric interface plus cavities (“Cavity”), dielectric interface with Stern layer and no cavities (“Stern”) and including cavities and Stern layer (“Full”). For the most refined mesh, the difference between the “Single” result and the “Full” results is 35 [kJ/mol]. The Richardson extrapolated value using the single-surface model is -2047.16 [kJ/mol], which is also around 35 [kJ/mol] away from the extrapolated value for the “Full” simulation, which is a $\sim 2\%$ difference. This suggests that, for calculating solvation energy, if the required accuracy allows errors larger than 2%, then the simpler, single-surface model can be used, requiring less discretization elements and less runtimes.

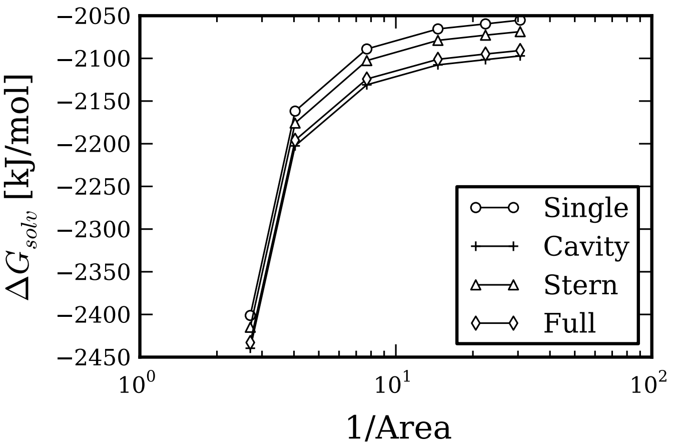


Figure 4·4: Results for the solvation energy of lysozyme for various `PyGBe` models. Data sets, figure files and plotting scripts available under CC-BY (Cooper et al., 2013d).

4.2.2 Binding energy

Using the method outlined by Equation (2.69), we obtained the binding energy of the trypsin-BPTI complex (code 3BTK), and a peptide-RNA complex using both `PyGBe` and **APBS** (BPTI stands for Bovine Pancreatic Trypsin Inhibitor). We first computed the solvation energy for each case (complex, isolated ligand and isolated receptor) and then added the Coulombic energy separately. The Coulombic energies for the trypsin-BPTI complex, receptor and ligand are shown in Table 4.1.

Coulombic Energy [kJ/mol]		
Complex	Trypsin	BPTI
-79763.68	-62956.98	-17046.18

Table 4.1: Coulombic energies for the trypsin-BPTI complex.

The `PyGBe` runs were done with two models: a multi-surface model, where the protein is modeled considering cavities and a 2Å-thick Stern layer, and a single-surface model, where only the dielectric interface or SES is considered. The permittivity in the protein region was $\epsilon_{\text{prot}} = 4$ and in the solvent region $\epsilon_{\text{solv}} = 80$, with $\kappa = 1/8\text{Å}^{-1}$, equivalent to 150mM of salt. `PyGBe` used 1 Gauss point per far-away element —2L or more away from the collocation point—, 19 Gauss points per element for near-singular integrals and 5 Gauss points per triangle side for the semi-analytical integration. The GMRES tolerance was set to 10^{-4}, and the treecode contained only 2 terms of the Taylor expansion, with $\theta = 0.5$.. The data sets, figure files and plotting scripts for the results in this section are made available under a CC-BY license (Cooper et al., 2013b).

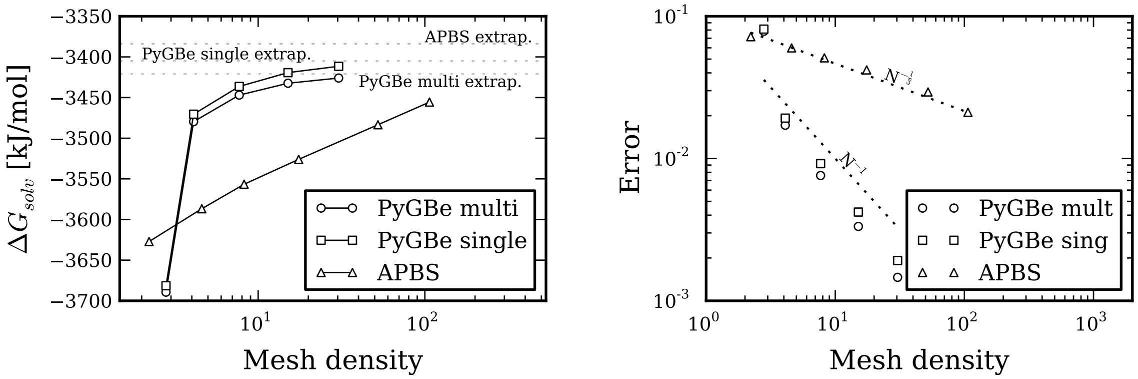
Trypsin-bpti

Trypsin-bpti complex Figure 4·5 shows the results of computing solvation energy for the trypsin-BPTI complex, which we later use for obtaining binding energy. We modeled the trypsin-BPTI complex using 4074 point charges with an `amber` force field, and it contains 7 cavities.

The solvation-energy values obtained with different mesh densities are plotted in Figure 4·5a. The dotted lines represent the extrapolated values obtained using Richardson extrapolation, and shown in Table 4.2 for each case: `APBS`, multi-surface `PyGBe` and single-surface `PyGBe`. We calculated the errors shown in Figure 4·5b using the extrapolated values in Table 4.2 as the base solution. The errors seem to be scaling with mesh spacing for `APBS` and triangle area for `PyGBe`, at the expected rates.

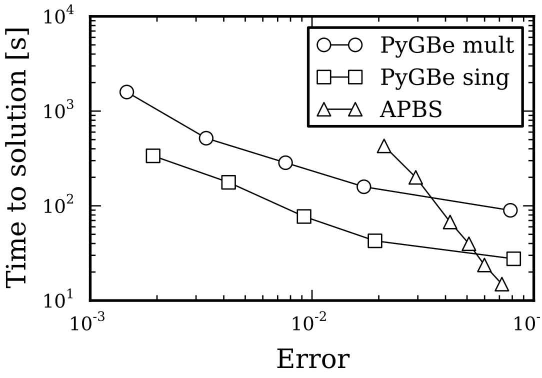
Figure 4·5c shows a plot of time to solution with respect to estimated error. It displays the same behavior as the lysozyme molecule in Figure 4·3c, where for low-accuracy calculations a volumetric approach is faster, but scales worse than the boundary integral approach, crossing over at a level of about 3% error. Of course, the single-surface model requires less elements and takes less time to solve compared to the multi-surface model.

Trypsin Figure 4·6 shows results of solvation energy for trypsin, which we later use for calculating binding energy. We model the trypsin using 3220 point charges with



(a) Solvation energy convergence with mesh refinement.

(b) Error convergence with mesh refinement.



(c) Runtime vs. estimated error.

Figure 4·5: `APBS` and `PyGBe` results for trypsin-BPTI complex. Data sets, figure files and plotting scripts available under CC-BY (Cooper et al., 2013b).

an `amber` force field, and it contains 6 cavities.

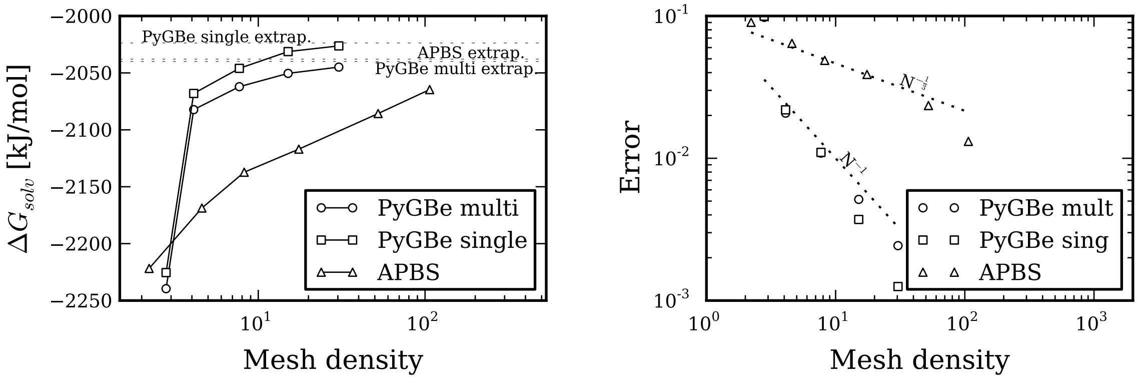
Convergence of the solvation energy with mesh density is plotted in Figure 4·6a. The dotted lines represent the extrapolated values using Richardson extrapolation, listed in Table 4.2. We calculated the errors shown in Figure 4·6b using the extrapolated values in Table 4.2 as the base solution. The errors seem to be scaling with mesh spacing for `APBS` and with area for `PyGBe`, at the expected rates.

The plot of time to solution versus error in Figure 4·6c shows the same behavior as the lysozyme molecule in Figure 4·3c and the trypsin-BPTI complex in Figure 4·5c, where time to solution for the boundary element method has better scaling with error.

Bovine pancreatic trypsin inhibitor (bpti) Figure 4·7 shows results of solvation energy calculations for bovine pancreatic trypsin inhibitor (BPTI), which is modeled using 854 point charges with a Amber force field, and no cavities.

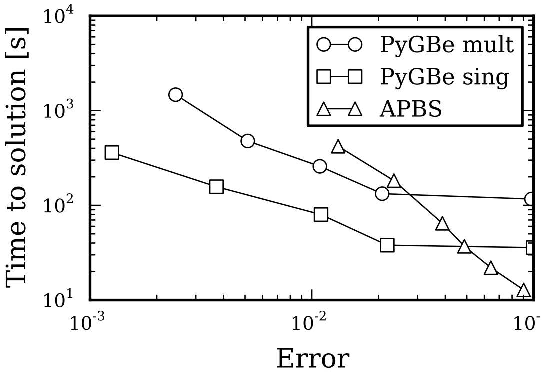
The plot of solvation energy for different values of mesh density in on Figure 4·7a, where the dotted lines represent the extrapolated value using Richardson extrapolation, shown in Table 4.2. We calculated the errors shown in Figure 4·7b using the extrapolated values in Table 4.2 as the base solution. The errors seem to be scaling with mesh spacing for `APBS` and area for `PyGBe`, in the expected rates. Figure 4·7c shows the plot of time to solution vs. error. As before, time to solution for the boundary element method has better scaling with error.

Trypsin-bpti binding energy Binding energies can be computed from the solvation energies as detailed in Equations (2.69) and (2.68), for $F_{\text{bind}} = F_{\text{interaction}}$. Figure 4·8 shows the computed binding energy for different values of mesh density. Using the extrapolated values of solvation energy for `PyGBe` with multi- and single-surface models, shown in Table 4.2, we calculated binding energies. These were 74.42 [kJ/mol] for the multi-surface model and 78.15 [kJ/mol] for the single-surface model, and



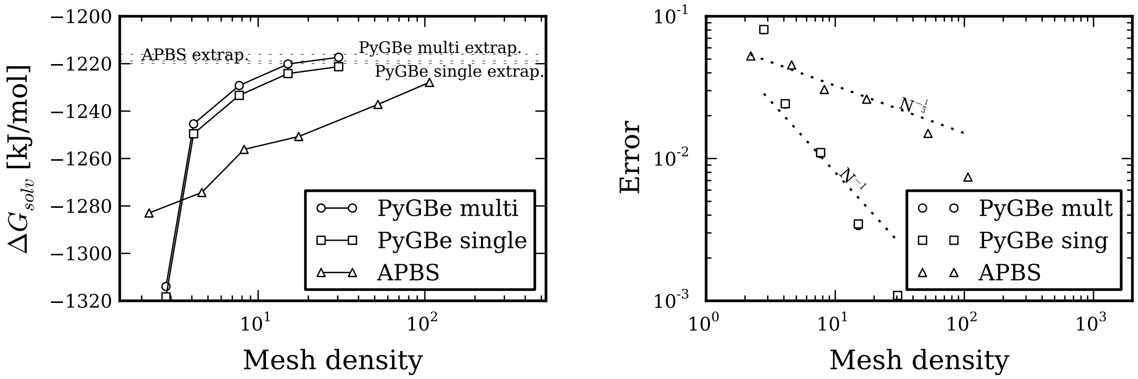
(a) Solvation energy convergence with mesh refinement.

(b) Error convergence with mesh refinement.



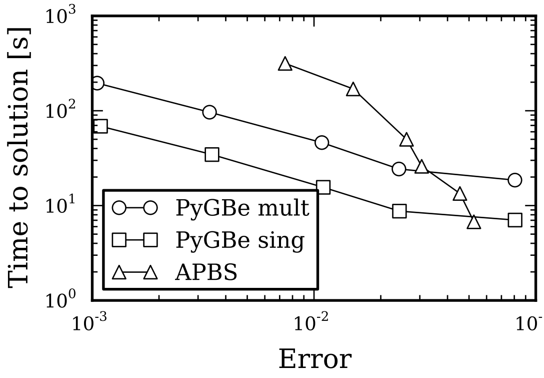
(c) Runtime vs. estimated error.

Figure 4·6: `APBS` and `PyGBe` results for trypsin as receptor of trypsin-BPTI complex. Data sets, figure files and plotting scripts available under CC-BY (Cooper et al., 2013b).



(a) Solvation energy convergence with mesh refinement.

(b) Error convergence with mesh refinement.



(c) Runtime vs. estimated error.

Figure 4·7: `APBS` and `PyGBe` results for BPTI as ligand of trypsin-BPTI complex. Data sets, figure files and plotting scripts available under CC-BY (Cooper et al., 2013b).

are represented in Figure 4·8 by the dotted lines.

Code	Solvation energy [kJ/mol] Complex	 Trypsin	 BPTI
`APBS`	-3384.08	-2037.88	-1218.86
`PyGBe` multi	-3421.0	-2039.92	-1216.02
`PyGBe` single	-3405.0	-2023.74	-1219.93

Table 4.2: Extrapolated solvation energies.

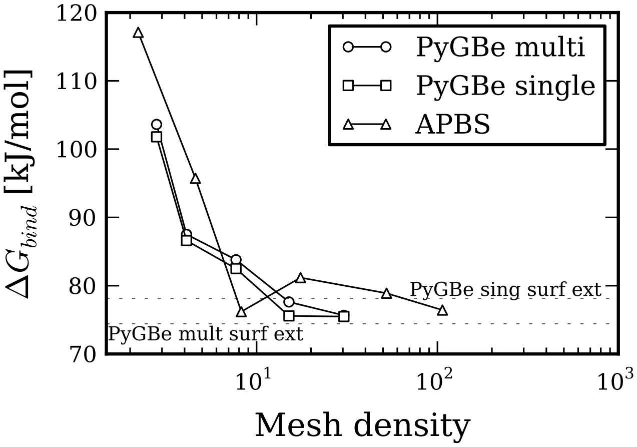


Figure 4·8: `APBS` and `PyGBe` results for binding energies of the trypsin-BPTI complex. The multi-surface model of `PyGBe` considers cavities and Stern layer, and the single-surface model considers neither. Data sets, figure files and plotting scripts available under CC-BY (Cooper et al., 2013b).

4.2.3 Peptide-rna complex

We performed the same set of tests than in Section 4.2.2 but for a 22-residue long α-helical peptide of protein λ bound the "box B" RNA hairpin structure (García-García and Draper, 2003). In this case, the complex has 998 atoms, the RNA 619 atoms and the peptide 379 atoms, and the structures are available online.[1] Neither molecule has cavities and `PyGBe` runs used the multi-surface model with a 2Å-thick Stern layer. Figure 4·9 shows the binding energy of this complex for different mesh densities using

---

[1] `http://www.poissonboltzmann.org/apbs/examples/`

`APBS` and `PyGBe` with a multi-surface and single-surface model. The dotted lines correspond to binding energies calculated using the extrapolated values of solvation energy for the multi-surface model (93.7 [kJ/mol]) and single-surface model (101.29 [kJ/mol]). The data sets, figure files and plotting scripts for the results in this section are made available under a CC-BY license (Cooper et al., 2013a).

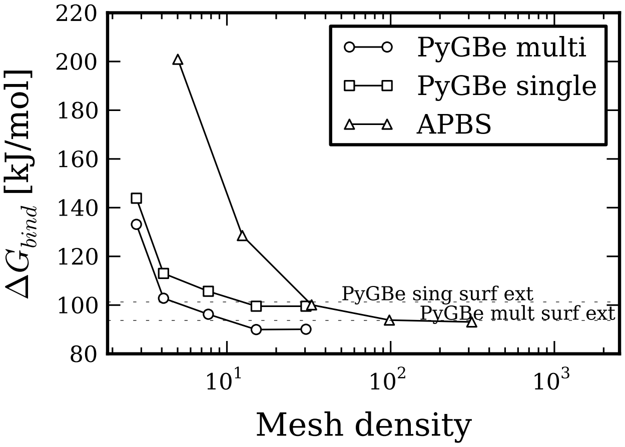


Figure 4·9: `APBS` and `PyGBe` results for binding energies of a 22-residue long α helix peptide with "box B" hairpin structure of RNA. `PyGBe` runs used a multi-surface model (considers Stern layer), and a single-surface model. Data sets, figure file and plotting script available under CC-BY (Cooper et al., 2013a).

Discussion

Although 35 [kJ/mol] may be a small amount in the context of solvation energies, it may become important for derived quantities, for example, binding energies. Binding energy results from the difference between two solvation energies, which may be small. In the trypsin-BPTI complex, the binding energy difference between multi- and single-surface models was only 4 [kJ/mol], or $1.5RT$, with R being the gas constant and RT represents thermal fluctuations. This value is very close to thermal fluctuations, and corresponds to only $\sim$ 5% of the total binding energy, indicating that for this case a single-surface model is good enough. On the other hand, the peptide-RNA complex of Section 4.2.3 shows a larger difference in binding energy when a Stern layer is considered (the components here have no cavities) of around 8 [kJ/mol] or $3RT$, which corresponds to $\sim 10\%$ of the total binding energy. Our results suggest that if the size difference between ligand and receptor is large, like in the trypsin-BPTI complex, the resulting binding energy is less sensitive to Stern layers or cavities. However, for more comparable sizes in the molecules, like the peptide-RNA complex, using all features becomes important. We think this happens because the receptor looks almost identical to the complex when it is much larger than the ligand, making the error introduced by using a single-surface model in the solvation energy calculation very similar for the receptor and complex, and being subtracted out in the binding energy calculation of Equation (2.69).

Our results also include time to solution. Timings for `PyGBe` do not include the mesh generation time with `MSMS`, this time is negligible compared to the total time to solution. The trend of the time to solution plots is very similar in all cases, where lower-accuracy calculations are faster using a volumetric approach, but the scaling of the boundary integral technique with accuracy is better and they cross over at around 1% error. This indicates that the choice of solver should be made considering the required accuracy of the application.

Chapter 5

Protein-surface interaction calculations

The interaction between proteins and surfaces, more specifically protein adsorption, is a widely seen phenomena in biology; however, there is still debate on specific mechanisms. (Rabe et al., 2011; Gray, 2004). Part of the interest in understanding protein adsorption comes from biotechnological applications, sometimes to control it, like biosensors or ion-exchange chromatography, and others to avoid it, like biofouling in biomedical implants.

The implicit-solvent model using the Poisson-Boltzmann equation is broadly used to compute solvation energies in protein systems (Roux and Simonson, 1999; Bardhan, 2012), but little work includes the effect of surfaces. One example is the group led by Lenhoff, who used boundary-element and finite-difference methods to study the interaction of surfaces and proteins (Roth and Lenhoff, 1993; Asthagiri and Lenhoff, 1997), realizing that for realistic molecular geometries, the van der Waals effect usually becomes negligible (Roth et al., 1996), and that this approach is appropriate as long as there are no large conformational changes in the protein (Yao and Lenhoff, 2004; Yao and Lenhoff, 2005). Other groups have used coarser grained variations of this method, by considering the protein as a set of spheres (Sheng et al., 2002; Zhou et al., 2004), or assigning an effective charge at the residue level (Freed and Cramer, 2011; Zhou et al., 2003). Further approximations have been made to account for pH effects (Biesheuvel et al., 2005; Hartvig et al., 2011).

In this chapter, we show verification results for implementation of the scheme detailed in Section 3.2 —that extends `PyGBe` to account for charged surfaces— and then compare `PyGBe` results with experimental observations and molecular dynamics simulations for protein G B1 D4′ near a charged surface. We will later use this solver to study the interaction of proteins and surfaces in biosensing applications. The material of this chapter is part of a manuscript to be submitted for publication in the Journal of Chemical Physics.

All results using `PyGBe` in this chapter were obtained using one core of an Intel Xeon X5650 CPU workstation with one NVIDIA Tesla C2075 GPU card.

5.1 Analytical solution for spheres

It is possible to derive a closed-form expression for the free energy of interaction between a spherical molecule with a centered charge and a spherical surface with imposed potential or charge, like the one sketched in Figure 5·1. Having such an analytical solution is of great utility in the development of a computational model for protein-surface interaction, because it will allow for the proper code verification.

5.1.1 Expansion in Legendre polynomials

The system of partial differential equations from Equation (3.11) models the electrostatic potential field in the setting of Figure 5·1. Using the guidelines from Carnie and co-workers, (Carnie et al., 1994) the axial symmetry allows us to formulate the solution of Equation (3.11) as an expansion in Legendre polynomials:

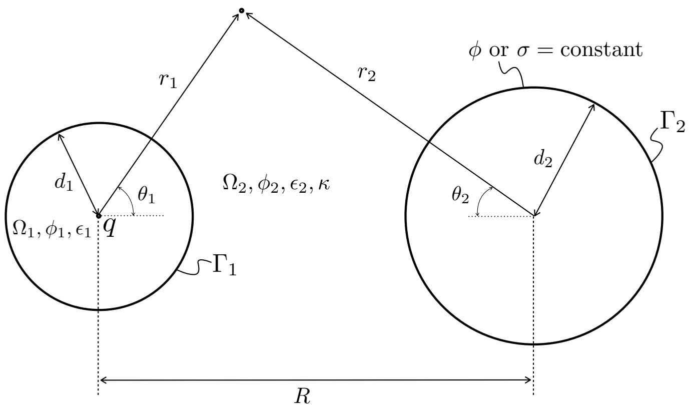

ϕ or σ = constant
r_1
r_2
Γ_2
d_1
d_2
θ_1
θ_2
$\Omega_2, \phi_2, \epsilon_2, \kappa$
$\Omega_1, \phi_1, \epsilon_1$
q
Γ_1
R

Figure 5·1: Sketch of system solved with Legendre polynomials expansions.

$$\begin{aligned}\phi_1 &= \sum_{n=0}^{\infty} c_n r_1^n P_n(\cos\theta_1) + \frac{q}{4\pi\epsilon_1 r_1} \quad \text{on } \Omega_1,\\ \phi_2 &= \sum_{n=0}^{\infty} a_n k_n(\kappa r_1) P_n(\cos\theta_1)\\ &\qquad + \sum_{n=0}^{\infty} b_n k_n(\kappa r_2) P_n(\cos\theta_2) \quad \text{on } \Omega_2, \end{aligned} \tag{5.1}$$

being P_n the n^{th}-degree Legendre polynomial and k_n the modified spherical Bessel function of the second kind.

We make use of the following addition formula (Marcelja et al., 1977),

$$k_n(\kappa r_2) P_n(\cos\theta_2) = \sum_{m=0}^{\infty} (2m+1) B_{nm} i_m(\kappa r_1) P_m(\cos\theta_1), \tag{5.2}$$

to reformulate the expression for ϕ_2 in Equation (5.1) as

$$\begin{aligned}\phi_2 = &\sum_{n=0}^{\infty} a_n k_n(\kappa r_1) P_n(\cos\theta_1) \\ &+ \sum_{n=0}^{\infty} b_n \sum_{m=0}^{\infty} (2m+1) B_{nm} i_m(\kappa r_1) P_m(\cos\theta_1) \\ \phi_2 = &\sum_{n=0}^{\infty} b_n k_n(\kappa r_2) P_n(\cos\theta_2) \\ &+ \sum_{n=0}^{\infty} a_n \sum_{m=0}^{\infty} (2m+1) B_{nm} i_m(\kappa r_2) P_m(\cos\theta_2). \end{aligned} \tag{5.3}$$

Here, i_m is the modified spherical Bessel function of the first kind, B_{nm} is defined by

$$B_{nm} = \sum_{\nu=0}^{\infty} A_{nm}^{\nu} k_{n+m-2\nu}(\kappa R) \tag{5.4}$$

where R is the center-to-center distance, and

$$A_{nm}^{\nu} = \frac{\mathbf{\Gamma}(n-\nu+0.5)\mathbf{\Gamma}(m-\nu+0.5)\mathbf{\Gamma}(\nu+0.5)(n+m-\nu)!(n+m-2\nu+0.5)}{\pi\mathbf{\Gamma}(m+n-\nu+1.5)(n-\nu)!(m-\nu)!\nu!}, \tag{5.5}$$

being $\mathbf{\Gamma}$, in this case only, the gamma function.

Legendre polynomials are orthogonal to each other, and $\frac{q}{4\pi\epsilon_1 r_1}$ is independent of θ. Thus, taking the inner product of the expressions in Equations (5.1) and (5.3) with a generic $P_j(\cos\theta_i)$, yields

$$\begin{aligned} &c_j r_1^j + \frac{q}{4\pi\epsilon_1 r_1}\delta_{0j} \text{ for first expression of Equation (5.1), and} \\ &a_j k_j(\kappa r_1) + \sum_{n=0}^{\infty} b_n (2j+1) B_{nj} i_j(\kappa\theta_1), \\ &b_j k_j(\kappa r_2) + \sum_{n=0}^{\infty} a_n (2j+1) B_{nj} i_j(\kappa\theta_2) \quad \text{for Equation (5.3).} \end{aligned} \tag{5.6}$$

The application of the interface conditions for Γ_1 on the first two expressions of Equation (5.6), produces

$$\begin{aligned} &\sum_{n=0}^{\infty} a_n \left(\kappa k_n'(\kappa d_1) - \frac{\epsilon_1}{\epsilon_2} \frac{n}{d_1} k_n(\kappa d_1) \right) \delta_{nj} + \\ &b_n (2j+1) B_{nj} \left(\kappa i_j'(\kappa d_1) - \frac{\epsilon_1}{\epsilon_2} \frac{j}{d_1} i_j(\kappa d_1) \right) = \\ &\qquad\qquad - \frac{\epsilon_1}{\epsilon_2} \frac{q}{4\pi\epsilon_1 d_1^2} \delta_{0j}(j+1), \end{aligned} \tag{5.7}$$

where d_i is the radius of surface i.

Constant potential ϕ on Γ_2.

The application of the boundary condition on Γ_2, $\phi(\Gamma_2) = \phi_0$, where ϕ_0 is independent on θ_2, gives

$$\sum_{n=0}^{\infty} a_n (2j+1) B_{nj} i_j(\kappa d_2) + b_n k_n(\kappa d_2) \delta_{nj} = \phi_0 \delta_{0j} \tag{5.8}$$

The combination of Equations (5.7) and (5.8) yields the following system of equations for the coefficients a_n and b_n

$$\begin{aligned} \mathbf{IA} + \mathbf{LB} &= -\frac{\epsilon_1}{\epsilon_2} \frac{q}{4\pi\epsilon_1 d_1^2} \mathbf{e} \\ \mathbf{MA} + \mathbf{IB} &= \phi_0 \mathbf{e} \end{aligned} \tag{5.9}$$

where

$$
\begin{aligned}
I_{jn} &= \delta_{jn} \\
e_j &= \delta_{0j} \\
A_n &= a_n \left(\kappa k_n'(\kappa d_1) - \frac{\epsilon_1}{\epsilon_2}\frac{n}{d_1} k_n(\kappa d_1) \right) \\
B_n &= b_n k_n(\kappa d_2) \\
L_{jn} &= (2j+1) B_{nj} \left(\kappa \frac{i_j'(\kappa d_1)}{k_n(\kappa d_2)} - \frac{\epsilon_1}{\epsilon_2}\frac{j}{d_1}\frac{i_j(\kappa d_1)}{k_n(\kappa d_2)} \right) \\
M_{jn} &= (2j+1) B_{nj} i_j(\kappa d_2) \frac{1}{\left(\kappa k_n'(\kappa d_1) - \frac{\epsilon_1}{\epsilon_2}\frac{n}{d_1} k_n(\kappa d_1) \right)}. \qquad (5.10)
\end{aligned}
$$

Constant surface charge σ on Γ_2.

In this case, the application of the boundary condition on Γ_2, $\sigma(\Gamma_2) = -\epsilon_2 \frac{\partial \phi}{\partial \mathbf{n}}|_{\Gamma_2} = \sigma_0$, where σ_0 is independent on θ_2, gives

$$
\sum_{n=0}^{\infty} a_n (2j+1) B_{nj} \kappa i_j'(\kappa d_2) + b_n \kappa k_n'(\kappa d_2) \delta_{nj} = -\frac{\sigma_0}{\epsilon_2} \delta_{0j} \qquad (5.11)
$$

The combination of Equations (5.7) and (5.8) produces a system of equations for the coefficients a_n and b_n

$$
\begin{aligned}
\mathbf{IA} + \mathbf{LB} &= -\frac{\epsilon_1}{\epsilon_2}\frac{q}{4\pi\epsilon_1 d_1^2}\mathbf{e} \\
\mathbf{MA} + \mathbf{IB} &= -\frac{\sigma_0}{\epsilon_2}\mathbf{e} \qquad (5.12)
\end{aligned}
$$

where

$$\begin{aligned}
I_{jn} &= \delta_{jn} \\
e_j &= \delta_{0j} \\
A_n &= a_n \left(\kappa k_n'(\kappa d_1) - \frac{\epsilon_1}{\epsilon_2} \frac{n}{d_1} k_n(\kappa d_1) \right) \\
B_n &= b_n \kappa k_n'(\kappa d_2) \\
L_{jn} &= (2j+1) B_{nj} \left(\frac{i_j'(\kappa d_1)}{k_n'(\kappa d_2)} - \frac{\epsilon_1}{\epsilon_2} \frac{j}{d_1} \frac{i_j(\kappa d_1)}{\kappa k_n'(\kappa d_2)} \right) \\
M_{jn} &= (2j+1) B_{nj} \kappa i_j'(\kappa d_2) \frac{1}{\left(\kappa k_n'(\kappa d_1) - \frac{\epsilon_1}{\epsilon_2} \frac{n}{d_1} k_n(\kappa d_1) \right)}.
\end{aligned} \tag{5.13}$$

5.1.2 Energy calculation

Solvation free energy of the molecule.

According to Equation (2.64), the solvation free energy of a molecule with a centered charge is given by

$$F_{\text{solv}} = \frac{1}{2} q \phi_{\text{reac}}(r_1 = 0), \tag{5.14}$$

and using Equation (5.1), the reaction potential from Equation (2.65) is:

$$\phi_{\text{reac}} = \phi - \frac{q}{4\pi\epsilon_1 r} = \sum_{n=0}^{\infty} c_n r^n P_n(\cos\theta_1). \tag{5.15}$$

Applying the boundary conditions at Γ_1 on Equation (5.6), we can rewrite c_j in terms of the already computed a_j and b_j:

$$\begin{aligned}
c_j = \frac{1}{d_1^j} \Big(& a_j k_j(\kappa d_1) + \\
& \sum_{m=0}^{\infty} b_m (2j+1) B_{mj} i_j(\kappa d_1) - \frac{q}{4\pi\epsilon_1 d_1} \delta_{0j} \Big)
\end{aligned} \tag{5.16}$$

Because the charge is located at $r = 0$, only the $n = 0$ terms of Equation (5.15) will survive, and the potential at this location is:

$$\begin{aligned}\phi_{\text{reac}}(r_1 = 0) =&a_0 k_0(\kappa d_1)+\\ &\sum_{m=0}^{\infty} b_m B_{m0} i_0(\kappa d_1) - \frac{q}{4\pi\epsilon_1 d_1}\end{aligned} \tag{5.17}$$

The result from Equation (5.17) in Equation (5.14) yields the solvation free energy.

For the isolated molecule, $R \to \infty$ makes $B_{nm} \to 0$, which nullifies the sum in Equation (5.17) and a_0 for $R \to \infty$, from the system in Equation (5.9), is

$$a_0^{\infty} = -\frac{q}{d_1^2}\frac{\epsilon_1}{\epsilon_2}\frac{1}{4\pi\kappa k_0'(\kappa d_1)\epsilon_1} \tag{5.18}$$

Free energy of surface.

Surface with set potential ϕ_0. We can expand G_p from Equation (2.66) in Legendre polynomials as

$$\begin{aligned}G_p = -\frac{\epsilon_2\kappa}{\phi_0}\Bigg[&\sum_{n=0}^{\infty} b_n k_n'(\kappa d_2) P_n(\cos\theta_2)\\ &+\sum_{n=0}^{\infty} a_n \sum_{m=0}^{\infty}(2m+1) B_{nm} i_m'(\kappa d_2) P_m(\cos\theta_2)\Bigg].\end{aligned} \tag{5.19}$$

Applying Equation (5.19) in Equation (2.66) gives

$$G_p = 2\pi\kappa\phi_0 d_2^2\epsilon_2\left[b_0 k_0'(\kappa d_2) + \sum_{n=0}^{\infty} a_n B_{n0} i_0'(\kappa d_2)\right] \tag{5.20}$$

If the surface is isolated, $R \to \infty$ makes $B_{n0} \to 0$, and the free energy in this case is

$$G_p = 2\pi\kappa\phi_0 d_2^2 b_0^{\infty} k_0'(\kappa d_2)\epsilon_2 \tag{5.21}$$

where b_0^∞ is taken from the system in (5.9) considering $B_{nm} \to 0$, which results in

$$b_0^\infty = \frac{\phi_0}{k_0(\kappa d_2)}. \tag{5.22}$$

Surface with set charge σ_0. We can expand G_c from Equation (2.66) in Legendre polynomials as

$$\begin{aligned} G_c =& \frac{1}{\sigma_0}\left[\sum_{n=0}^{\infty} b_n k_n(\kappa d_2) P_n(\cos\theta_2) + \right. \\ & \left. \sum_{n=0}^{\infty} a_n \sum_{m=0}^{\infty} (2m+1) B_{nm} i_m(\kappa d_2) P_m(\cos\theta_2)\right] \end{aligned} \tag{5.23}$$

Applying Equation (5.23) into Equation (2.66) gives

$$G_c = 2\pi\sigma_0 d_2^2 \left[b_0 k_0(\kappa d_2) + \sum_{n=0}^{\infty} a_n B_{n0} i_0(\kappa d_2)\right] \tag{5.24}$$

For the isolated surface, $R \to \infty$ and $B_{n0} \to 0$, and the free energy is

$$G_c = 2\pi\sigma_0 d_2^2 b_0^\infty k_0(\kappa d_2) \tag{5.25}$$

where b_0^∞ is calculated from the system in (5.12) considering $B_{nm} \to 0$, which results in

$$b_0^\infty = -\frac{\sigma_0}{\epsilon_2 \kappa k_0'(\kappa d_2)}. \tag{5.26}$$

This scheme can aso be used to compute the interaction of two spherical molecules with centered charge.

5.2 Verification of `PyGBe`'s extension

Using the analytical solution detailed in Section 5.1, we carried out a mesh convergence study of `PyGBe`, extended to treat surfaces interacting with biomolecules. In this case, we cannot verify against existing software because there is no available code that considers the presence of a surface. The results shown in Figure 5·3 correspond to a spherical molecule with a 5Å radius and a centered charge of $1e^-$, interacting with a spherical surface of 4Å radius and an imposed potential of $\phi = 1$. The center-to-center distance between the spheres was 12Å, and they are dissolved in water with salt at 150mM, which gives a Debye length of 8 ($\kappa = 0.125$), and permittivity $\epsilon_{\text{sol}} = 80$. The permittivity inside the spherical protein is $\epsilon_{\text{mol}} = 4$. Figure 5·2 is a sketch of this system.

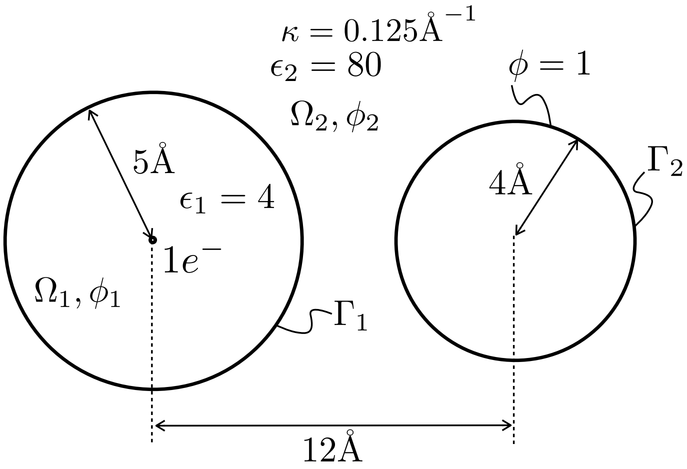


Figure 5·2: Sketch of system used in the convergence study of Figure 5·3.

The calculations for the results in Figure 5·3 used 3 Gauss points per element for far-away integrals, 37 if the element and collocation point were close, and 9 points per element side for the singular integral. The treecode evaluation uses a tree with no more than 300 boundary elements per box, a Taylor expansion with terms up to $P = 15$, and a multipole-acceptance criterion $\theta = 0.5$. The GMRES tolerance in these calculations was 10^{-9}.

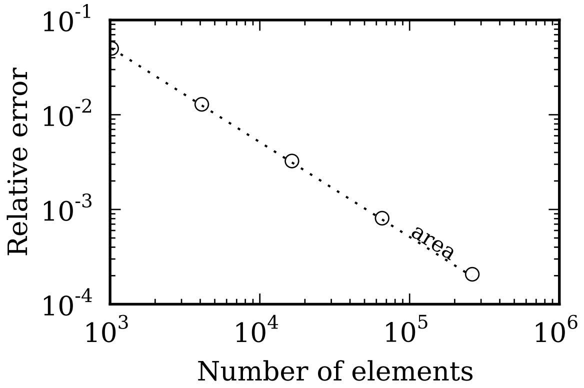


Figure 5·3: Mesh convergence study for the interaction free energy between a spherical molecule with a centered charge and a sphere with potential $\phi = 1$.

Figure 5·3 shows the error decaying with the average area of the boundary elements ($\frac{1}{N}$), which is an expected behavior considering our previous work (Cooper and Barba, 2013). This proves that the extension of `PyGBe` to treat charged surfaces is solving the mathematical model correctly.

5.3 Protein G B1 D4′ near a charged surface

The orientation of protein G B1 D4′ near a COO^- and NH_3^+ self-assembled monolayer (SAM) was explored experimentally by Baio and co-workers (Baio et al., 2012) and computationally through molecular dynamics (MD) simulations by Liu and co-workers (Liu et al., 2013). In the context of the implicit-solvent model, we can consider the SAM as a surface with prescribed charge (σ), and in this section, we use `PyGBe` to validate this model in a protein-surface interaction calculation.

Figure 5·4 is the structure of protein G B1 (PDB code 1PGB), to which we performed mutations E19Q, D22N, D46N and D47N to obtain the D4′ mutant, using the SwissPdb Viewer software [1] (Guex and Peitsch, 1997) . We also used `pdb2pqr` for input files and `MSMS` for meshing.

---

[1] `http://www.expasy.org/spdbv/`

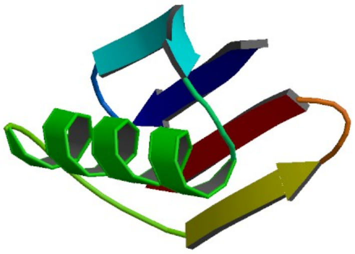


Figure 5·4: Structure of Protein G B1 (PDB code: 1PGB).

5.3.1 Setup of experiments

According to the Boltzmann distribution, the probability of finding the system in micro-state λ depends on the total free energy

$$P(\lambda) = \frac{\int_\lambda \exp\left(-\frac{F_{\text{total}}}{k_B T}\right) \mathrm{d}\lambda}{\int_\Lambda \exp\left(-\frac{F_{\text{total}}}{k_B T}\right) \mathrm{d}\Lambda}, \tag{5.27}$$

where Λ is the ensemble of all micro-states, F_{total} the total free energy, k_B the Boltzmann constant and T the temperature. To obtain a probability distribution, we used Equation (5.27) assumming that electrostatic effects were dominant, and sampled F_{total} for different orientations. We defined the orientation using the angle between the dipole moment and surface normal vectors as a reference (tilt angle), varying from 0° to 180°. Also, for each tilt angle, we rotated the protein about the dipole moment vector in 360° to examine all possible conformations. This process is sketched in Figure 5·5.

In this case, micro-states are defined by the tilt (α_{tilt}) and rotational (α_{rot}) angles, making the integral in Equation (5.27)

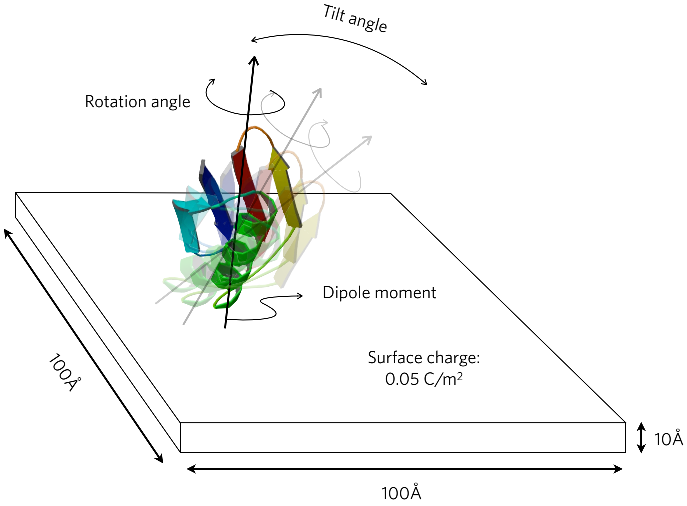


Figure 5·5: Setup of orientation experiment.

$$\int\limits_{\lambda} \exp\left(-\frac{F_{\text{total}}}{k_B T}\right) \mathrm{d}\lambda = \int\int \exp\left(-\frac{F_{\text{total}}}{k_B T}\right) \mathrm{d}\alpha_{\text{rot}} \mathrm{d}\alpha_{\text{tilt}}, \tag{5.28}$$

where micro-state λ is a range of angles α_{rot} and α_{tilt}.

We simulated the electrostatic field of protein G B1 D4′ interacting with a $100 \times 10 \times 100$Å block with surface charge density $\pm 0.05\text{C/m}^2$, to find its preferred orientation. The protein was centered with respect to a 100×100Å face, 2Å above it.

Figure 5·5 shows that α_{tilt} is the angle between the dipole moment of the protein and the normal vector to the 100×100Å surface, and that α_{rot} rotates about the dipole moment. The dipole moment vector placed on the center of mass of the protein generates an axis, and we used the line of shortest distance between the outermost atom and this axis as a reference vector $\mathbf{V}_{\text{ref}}$. The rotation angle α_{rot} is the angle between the normal vector to a 100×10Å face of the block and $\mathbf{V}_{\text{ref}}$.

In these cases, we considered a solvent with no salt, i.e. $\kappa = 0$, and with relative permittivity 80. The region inside the protein had a relative permittivity of 4.

5.3.2 Results for protein G B1 D4′

Mesh refinement study for of Protein G B1 D4′

We performed a mesh refinement study to make sure the geometry was well resolved by the mesh, and to find appropriate simulation parameters for sampling orientations. Then, by means of Richardson extrapolation, we found reference values to help us choose parameters for the sampling runs. This way, we were confident that the sampling runs were both accurate and efficient in computer time.

In these runs, we evaluated the solvation and surface energy of a system containing a surface with charge density 0.05C/m^2, and a protein at $\alpha_{\text{tilt}} = 10^\circ$ and $\alpha_{\text{rot}} = 200^\circ$. Using the runs with mesh density 2, 4, and 8 elements per square Angstrom, we obtained the values in Table 5.1 with Richardson extrapolation, which are the reference values for the error calculations in Figure 5·6.

For the mesh refinement study, we used 7 Gauss points per far-away element, 19 Gauss points per close-by element, and 9 Gauss points per triangle side for the singular integral. The treecode had 15 terms in the Taylor expansion, a multipole-acceptance criterion of 0.5, and no more than 500 boundary elements per box of the lowest level of the tree. Also, the GMRES tolerance was 10^{-8}.

Table 5.1: Extrapolated values of energy for Protein G B1 D4′.

Energy [kcal/mol]	
Solvation	Surface
−222.43	317.98

Figure 5·6 shows errors that are decaying as $1/N$ for both the solvation and surface energies, indicating that the geometry is well resolved.

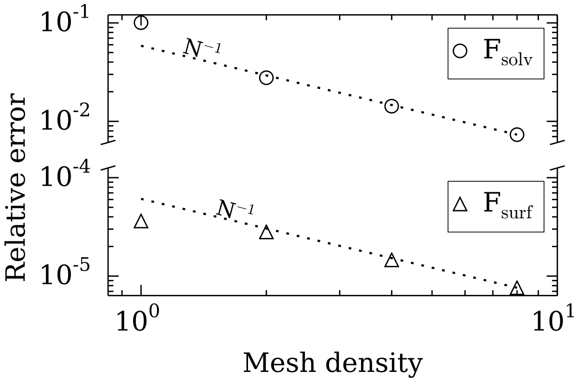


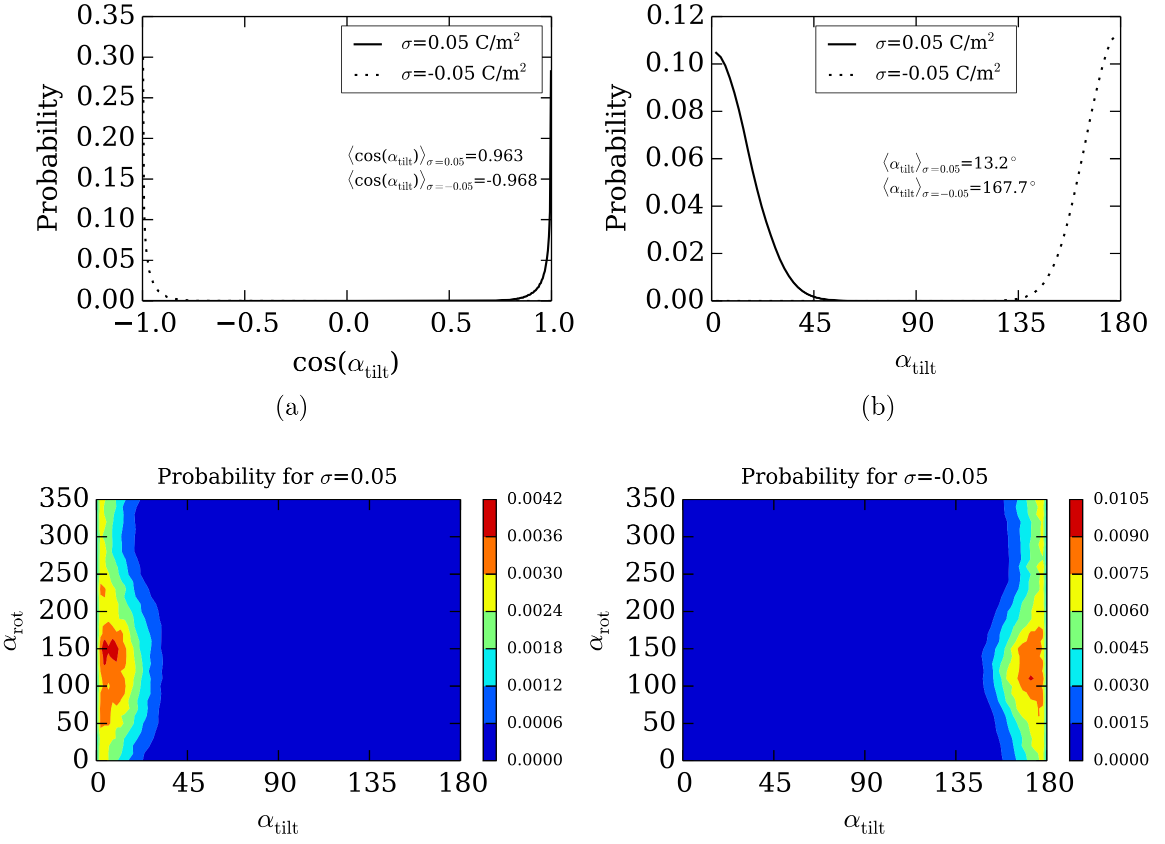
Figure 5·6: Mesh convergence study of the solvation and surface energy for protein G B1 D4′ mutant interacting with a surface with charge density 0.05C/m^2.

Probing orientation of protein G B1 D4′

We sampled the total free energy every $\Delta\alpha_{\text{tilt}} = 2°$ of tilt angle and $\Delta\alpha_{\text{rot}} = 10°$ of rotation angle. In these runs, meshes had 4 triangles per square angstrom on the protein geometry and 2 triangles per square angstrom on the charged surface. We used 1 Gauss point per element further away than 2L of the collocation point, 19 Gauss points for close-by elements, and 9 Gauss points per side of the singular element. The treecode used 4 terms in the Taylor expansion, a multipole-acceptance criterion of 0.5 and no more than 300 elements per lowest level box. The GMRES tolerance was 10^{-5}.

The total free energy was the input for numerically computing the integrals in Equation (5.28) with the trapezoidal rule. Figure 5·7 shows the probability of the protein orientation in terms of $\cos(\alpha_{\text{tilt}})$ and α_{tilt}, in intervals of $\Delta\cos(\alpha_{\text{tilt}}) = 0.005$ for Figure 5·7a, and $\Delta\alpha_{\text{tilt}}$=2° for Figure 5·7b, respectively. Table 5.2 contains the average orientation ($<\cos(\alpha_{\text{tilt}})>$) for the surface with positive and negative charge density, and Figure 5·8 shows the electrostatic potential for the preferred orientation in each case.

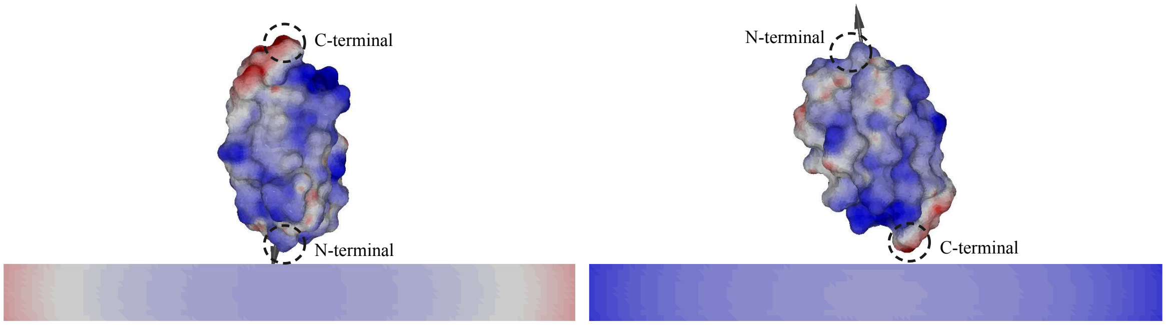
Figure 5·7 shows that for the most likely orientations, the dipole moment vector



(a) (b)

(c) (d)

Figure 5·7: Orientation distribution of protein G B1 D4′. Figures 5·7a and 5·7b are the probability with respect to the tilt angle and its cosine, respectively. Figures 5·7c and 5·7d are the orientation with respect to both the tilt and rotation angle.



(a) Negative surface charge ($\alpha_{\text{tilt}} = 172°$, $\alpha_{\text{rot}} = 110°$)

(b) Positive surface charge ($\alpha_{\text{tilt}} = 8°$, $\alpha_{\text{rot}} = 150°$)

Figure 5·8: Electrostatic potential of protein G B1 D4′ for the preferred orientations according to Figure 5·7. Black arrow indicates direction of dipole moment vector.

Table 5.2: Average orientation.

$< \cos(\alpha_{\text{tilt}}) >$	
Negative	Positive
-0.968	0.963

is aligned with the vector normal to the surface. This indicates that the dipole moment is the dominant force that defines its orientation, over local protein-surface interactions. This is somewhat an expected result, as Protein G B1 D4′ is a small protein.

Moreover, Figure 5·7 reveals that Protein G B1 D4′ behaves like a point dipole, as the most likely orientations shift 180° when the sign of the surface charge changes. This is also explained by the dipole moment that dominates the orientation.

The dipolar behavior agrees with experiments done by Baio and co-workers, (Baio et al., 2012) where they observe opposite orientations of Protein G B1 D4′ adsorbed on NH_3^+ and COO^- self-assembled monolayers. For positively charged surfaces, most of the proteins oriented with the N-terminal of the protein pointing away from the surface, while for negatively charged surfaces they saw the opposite, where the C-terminal was pointing out. This agrees with Figure 5·7 because the dipole moment vector of Protein G B1 D4′ points from the C-terminal to the N-terminal.

Also, Liu and co-workers (Liu et al., 2013) used MD simulations to determine $< \cos(\alpha_{\text{tilt}}) >= 0.95$ for $\sigma = 0.05\text{C/m}^2$, and $< \cos(\alpha_{\text{tilt}}) >= -0.85 \pm 0.05$ for $\sigma = -0.05\text{C/m}^2$, agreeing well with Table 5.2. MD simulations consider van der Waals interactions and conformational changes of the protein, whereas in our approach they are not, which explains the differences in $< \cos(\alpha_{\text{tilt}}) >$.

Chapter 6

Ligand orientation in biosensors

In biosensors, the analyte is sensed when it binds to the ligand molecule, hence, the antigen binding sites of the ligand molecule need to be facing the flow so that target molecules have access to them (Tajima et al., 2011). Careful treatment of the ligand molecule orientation can make orders of magnitude difference in the performance of the biosensor (Trilling et al., 2013a; Trilling et al., 2013b), which is evidence of the large effect of proper ligand biofunctionalization on the probability of capturing an analyte. In nanoscale devices, where the small sensor area limits the number of ligand molecules, this issue becomes even more important. The sketch in Figure 2·8 exemplifies this, as there is a clear preferred orientation for immunoglobulin G: with the fragment-antigen binding sites (Fabs) pointing away from the sensor surface.

The most common choice of ligand molecule for biosensors are full antibodies, due to their high specificity, which are convalently attached to the self-assembled monolayer (SAM) through their lysine residues or N-terminal amino groups. Full antibodies have between 60 and 80 potential binding sites to the SAM, many of which yield a poor orientation for analyte detection (Wimalasena and Wilson, 1991). The orientation then will be determined by longer-range interactions, compared to covalent bonds, and electrostatics plays an important role at that level.

For biosensing applications, the antibody immobilization on the surface needs to be done without introducing denaturation or any significant conformational changes to it. Such changes would affect the antigen binding sites, compromising the specificity of the antibody, and the effectiveness of the biosensor; in practice, the affinity of adsorbed antibodies is comparable to their native counterpart (Tajima et al., 2011). This indicates that the implicit-solvent model, with a rigid-structure antibody, is suitable for this application. Moreover, it is evidence that the sensor surface plus SAM must be hydrophilic: entropic effects would make a hydrophobic surface interact with the hydrophobic portion of the antibody, which is buried inside, leading to denaturation (Lu et al., 1991). A hydrophilic surface is best represented by a charged surface.

In this chapter, we use electrostatics and the implicit-solvent model to compute the orientation of an antibody near a charged surface.

6.1 Results for an antibody near a charged surface

With the extension of `PyGBe` verified in Section 5.2, and confirmation from Section 5.3 that the implicit-solvent model is appropriate for protein-surface interaction calculations, we explored the effect of the surface charge and salt concentration on the orientation of the antibody Immunoglobulin G (PDB code 1IGT), shown in Figure 6·1. Antibodies are widely used in biosensors as ligand molecules, due to their affinity and specificity with the target molecule (antigen), and it is vitally important that they are adsorbed on the sensor with the fragment antigen-binding (Fab fragment) pointing away from the sensor, into the flow containing the antigens. The material of this section is part of a manuscript to be submitted for publication in the Journal of Chemical Physics.

6.1.1 Mesh refinement study for immunoglobulin G

Similar to Section 5.3, we performed a mesh refinement study to make sure the geometry was well resolved and to find appropriate simulation parameters for sampling different orientations.

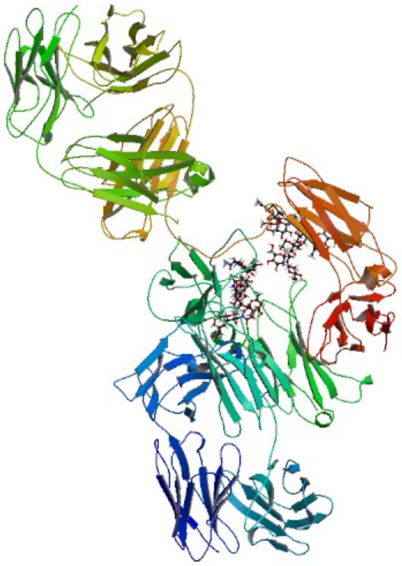


Figure 6·1: Structure of immunoglobulin G (PDB code: 1IGT).

In the mesh refinement runs, we evaluated the solvation energy and surface energy of a system with a surface with charge density $0.05\mathrm{C/m^2}$, and a protein with $\alpha_{\text{tilt}} = 31^\circ$ and $\alpha_{\text{rot}} = 130^\circ$. Using the results from runs with mesh density 2, 4, and 8 elements per square Angstrom, we added the solvation and surface energies, and extrapolated them using Richardson extrapolation to obtain -2798.66 kcal/mol, with an observed order of convergence of 0.995. This value is the reference to calculate the errors in Figure 6·2.

For the mesh refinement study, we used 1 Gauss points per far-away element, 19 Gauss points per close-by element, and 9 Gauss points per triangle side for the singular integral. The treecode had 6 terms in the Taylor expansion, a multipole-acceptance criterion of 0.5, and no more than 1000 boundary elements per box of the lowest level of the tree. Also, the GMRES tolerance was 10^{-4}.

Figure 6·2 show errors that are decaying as $1/N$ for the sum of the solvation and surface energies, indicating that the geometry is well resolved.

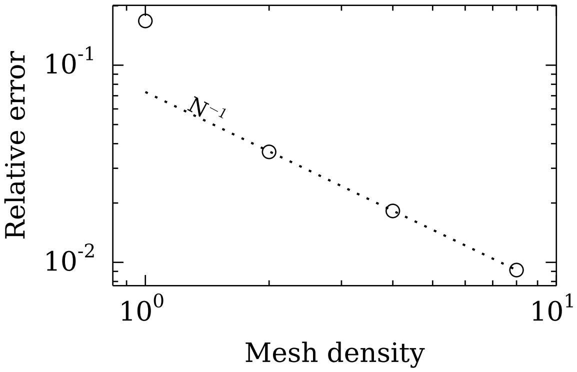


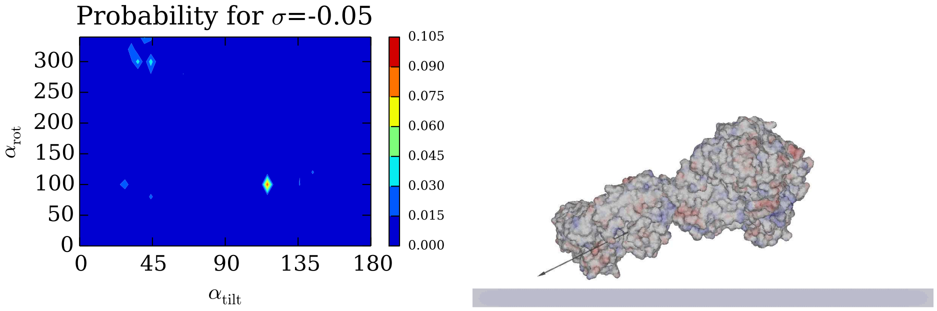
Figure 6·2: Mesh convergence study of the solvation plus surface energy for immunoglobulin G interacting with a surface with charge density 0.05C/m^2.

6.1.2 Probing orientation of immunoglobulin G

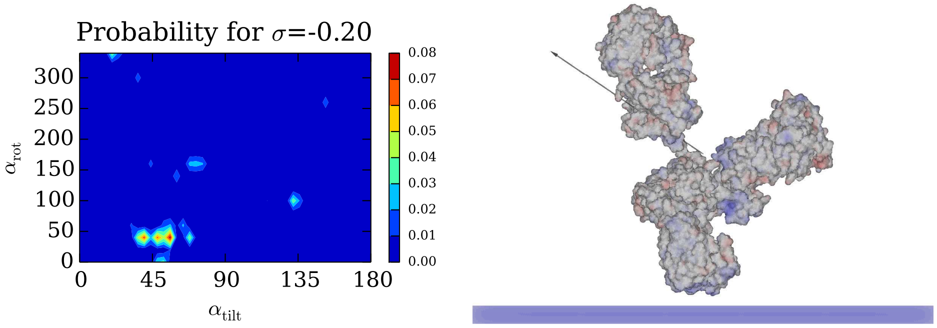
We sampled the total free energy every $\Delta\alpha_{\text{tilt}} = 4°$ of tilt angle and $\Delta\alpha_{\text{rot}} = 20°$ of rotation angle. In these runs, meshes had 2 triangles per square angstrom throughout. We used 1 Gauss point per element further away than 2L of the collocation point, 19 Gauss points for close-by elements, and 9 Gauss points per side of the singular element. The treecode used 2 terms in the Taylor expansion, a multipole-acceptance criterion of 0.5 and no more than 300 elements per lowest level box. The GMRES tolerance was 10^{-4}. We used the total free energy obtained from these simulations to compute the probability of each conformation with Equation (5.28) by means of the trapezoidal rule.

We performed this sampling for all combinations with surface charges of $\sigma = \pm 0.05$C/m^2 and $\sigma = \pm$ 0.2C/m^2 and salt concentrations of 150mM ($\kappa = 0.125$ Å^{-1}) and 9mM ($\kappa = 0.03125$ Å^{-1}). For every combination, Figure 6·3 and Figure 6·4 show a color plot of the probability distribution with the tilt and rotation angles, and a 3D plot of the preferred orientation, where the solvent-excluded surface (SES) is colored by the electrostatic potential.

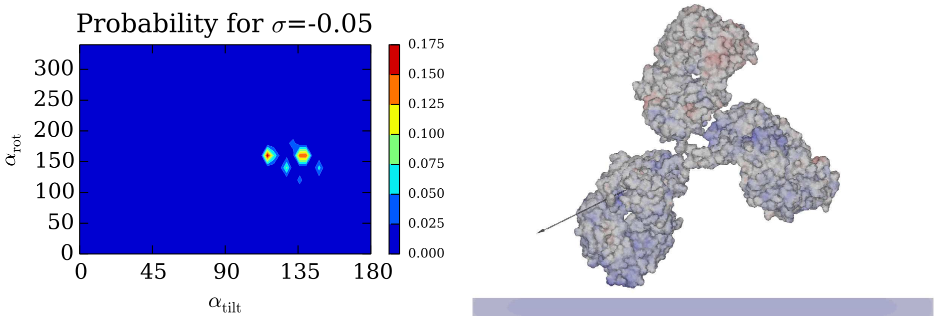
Figure 6·3 and Figure 6·4 are plots of the probability distribution of Immunoglob-



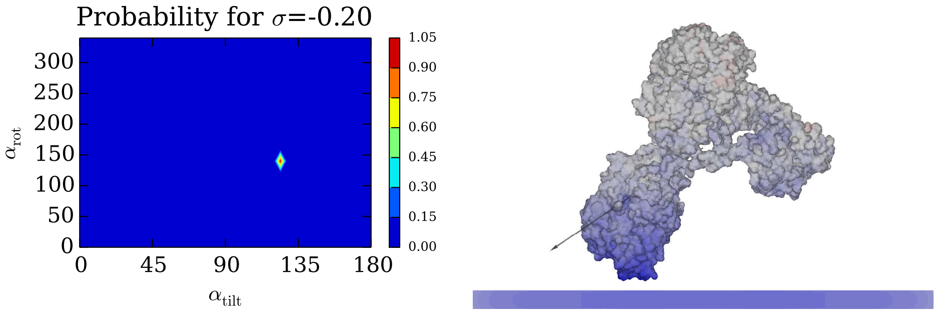
(a) σ=-0.05C/m^2, κ=0.125Å^{-1} (b) $\alpha_{\text{tilt}} = 116°$, $\alpha_{\text{rot}} = 100°$



(c) σ=-0.2C/m^2, κ=0.125Å^{-1} (d) $\alpha_{\text{tilt}} = 56°$, $\alpha_{\text{rot}} = 40°$

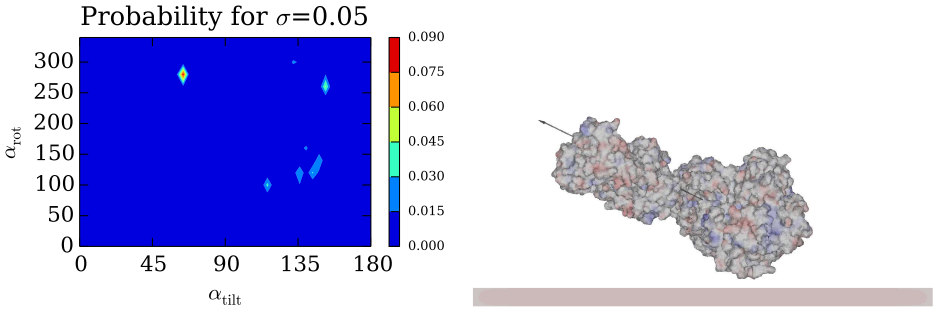


(e) σ=-0.05C/m^2, κ=0.03125Å^{-1} (f) $\alpha_{\text{tilt}} = 116°$, $\alpha_{\text{rot}} = 160°$

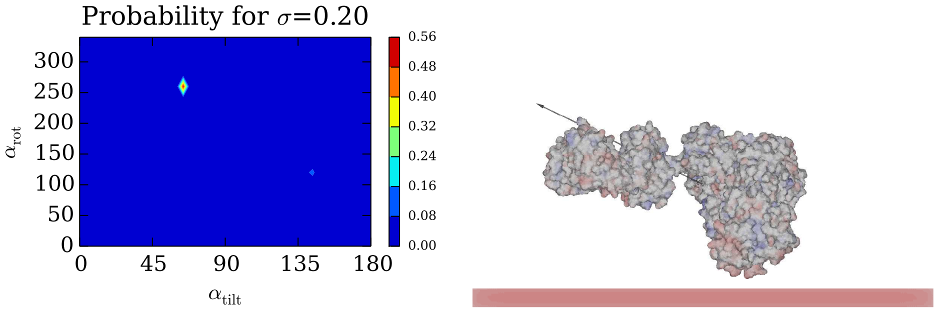


(g) σ=-0.2C/m^2, κ=0.03125Å^{-1} (h) $\alpha_{\text{tilt}} = 124°$, $\alpha_{\text{rot}} = 140°$

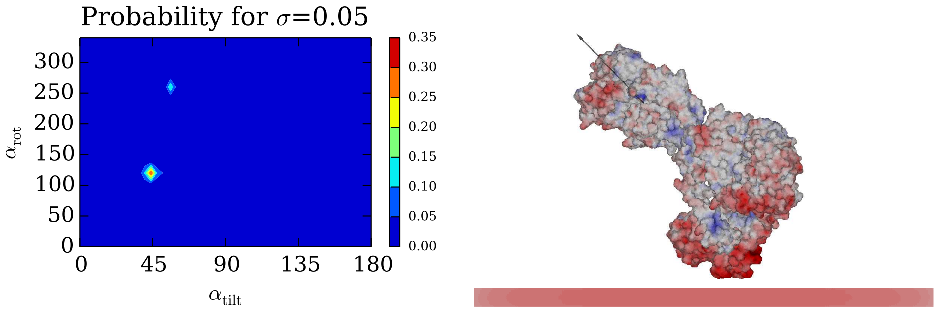
Figure 6·3: Orientation probability and surface potential of immunoglobulin's preferred orientation on a negative surface charge. The black arrow is the dipole vector.


Probability for σ=0.05

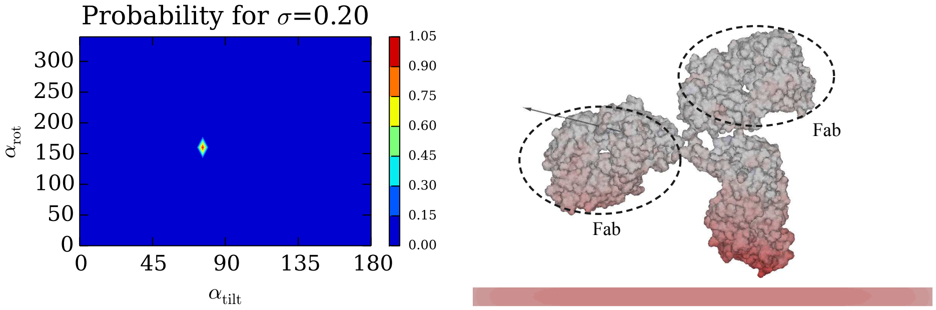
(a) σ=0.05C/m^2, κ=0.125Å^{-1} (b) $\alpha_{\text{tilt}} = 64°$, $\alpha_{\text{rot}} = 280°$


Probability for σ=0.20

(c) σ=0.2C/m^2, κ=0.125Å^{-1} (d) $\alpha_{\text{tilt}} = 64°$, $\alpha_{\text{rot}} = 260°$


Probability for σ=0.05

(e) σ=0.05C/m^2, κ=0.03125Å^{-1} (f) $\alpha_{\text{tilt}} = 44°$, $\alpha_{\text{rot}} = 120°$


Probability for σ=0.20

(g) σ=0.2C/m^2, κ=0.03125Å^{-1} (h) $\alpha_{\text{tilt}} = 76°$, $\alpha_{\text{rot}} = 160°$

Figure 6·4: Orientation probability and surface potential of immunoglobulin's preferred orientation on a positive surface charge. The black arrow is the dipole vector.

ulin G for many conformations, given by α_{tilt} and α_{rot}, varying the surface charge (σ) and salt concentration (κ). Figure 6·3a and Figure 6·4a show that for low surface charge ($\sigma \pm 0.05\mathrm{C/m}^2$) and high salt concentration ($\kappa = 0.125\text{Å}^{-1}$), there is no clear preferred orientation, to the point that the most likely orientation has a probability around 10%. This means that adsorbing the antibodies under these conditions would result in a wide variety of orientations, which is not favorable for biosensor applications.

Effect of surface charge With higher surface charge, in this case $\sigma = \pm 0.2\mathrm{C/m}^2$, the orientation distribution gets narrower for positive surface charge, and is maintained for negative surface charge. Figure 6·4c shows a much clearer preferred orientation, 5× higher for positive surface charge at high salt concentration. For low salt concentration, in Figure 6·3g and Figure 6·4g, this effect is much more significant.

Figure 6·3 and Figure 6·4 also show that increasing the surface charge has no significant effect in the dipole moment orientation. This is evidence that, opposite to the Protein G B1 D4′ case, local interactions dominate over the dipole moment. If the dipole moment were the dominant force, the dipole moment vector would tend to align to the surface normal as the surface charge increases.

Effect of salt concentration We also varied the Debye length (κ^{-1}) four times. In terms of salt concentration, it means decreasing the amount of salt 16 times.

Like increasing the surface charge, lowering the salt concentration narrows the orientation probability distribution. For $\sigma = \pm 0.05\mathrm{C/m}^2$ in Figure 6·3e and Figure 6·4e, the effect on positive or negative surface charge is very similar: the preferred orientation is about 2× more likely. However, for $\sigma = \pm 0.2\mathrm{C/m}^2$, the increase is larger in Figure 6·3g, with negative surface charge, rather than in Figure 6·4g, with positive surface charge. The narrower probability distribution is explained by the lower shielding effect caused by the decrease in salt content, which at the same time increases the electrostatic interaction.

From the results in Figure 6·3 and Figure 6·4, we can conclude that it is easier to control the antibody orientation with low salt concentration and high surface charge, because the orientation probability distribution is the narrowest. In our results, Figure 6·3h and Figure 6·4h show the orientation of the antibody at the lowest salt concentration and higher surface charge, however, only the orientation in Figure 6·4h, for positive surface charge, is favorable for biosensing applications, as the Fab fragments are pointing up.

Favorable orientations for biosensing applications with high positive surface charge and low salt concentration agrees with experimental observations from Chen and coworkers (Chen et al., 2003). Also, the same group developed a coarse-grained method known as the united residue model, (Zhou et al., 2003) which qualitatively aligns with these results.

Chapter 7

Response of a localized surface plasmon resonance biosensor

In this chapter, we perform response calculations of localized surface plasmon resonance (LSPR) biosensors using electrostatics and the implicit-solvent model. Even though LSPR is an optic effect, electrostatics is a good approximation in the long wavelength limit (see Section 7.2), making our BEM approach suitable for this application.

The electric field decays very fast moving away from the nanoparticle, to the point that analytes placed 15nm away of the surface are hard to sense (Haes et al., 2004). This inconvenience makes large ligand molecules, like immunoglobulin G, unsuitable for LSPR biosensors. The most recent developments of LSPR biosensors use single-chain (ScFvs) and single-domain (sdAb) antibodies, which are smaller than full antibodies (Byun et al., 2013; Trilling et al., 2014). Smaller ligand molecules are not only beneficial in terms of sensitivity, but also makes it possible to have a more compact biofunctionalization. Currently, the development and modification of sdAb and ScFvs for better sensitivity in LSPR is an active area of research (George et al., 2014; Turner et al., 2014).

The high dependence of sensitivity on distance in LSPR biosensors is the motivation to investigate this relation using our BEM approach. This chapter presents proof-of-concept calculations that show how BEM can be used to study LSPR response. The main advantage of this approach is that it can account for fine structural details

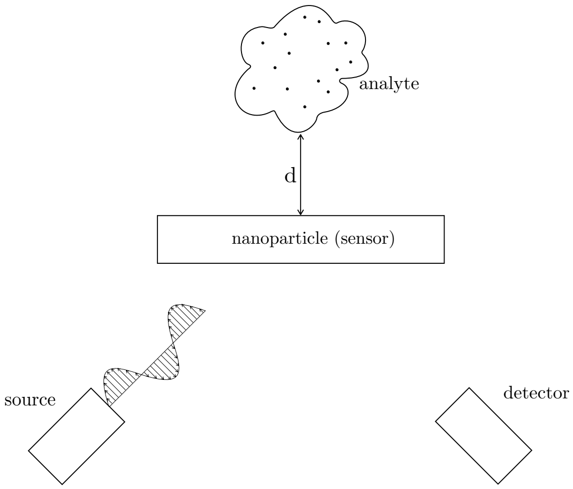


Figure 7·1: Sketch of a LSPR biosensor

of the analyte and nanoparticle, and the relative position between them.

7.1 Scattering in the far-field

7.1.1 The scattered electric field

When light shines on a particle, like in Figure 7·2, the incoming wave ($\mathbf{E}_i$, $\mathbf{B}_i$) is scattered throughout the domain, yielding a total electromagnetic field ($\mathbf{E}$, $\mathbf{B}$) that depends on the incoming wave, the particle's geometry, and material constants. In this section we will derive an expression for the scattered electric field in the long-range limit, *i.e.* far from the particle. This will be relevant in the subsequent sections, as LSPR biosensors measure the scattered electromagnetic field on a detector placed far away from the nanoparticle (see Figure 7·1).

The macroscopic Maxwell's equations model the electromagnetic field in a dielectric medium:

$$\begin{aligned} \nabla \cdot \mathbf{D} =& \rho_f \qquad \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \cdot \mathbf{B} =& 0 \qquad \nabla \times \mathbf{H} = \mathbf{J}_f + \frac{\partial \mathbf{D}}{\partial t}, \end{aligned} \tag{7.1}$$

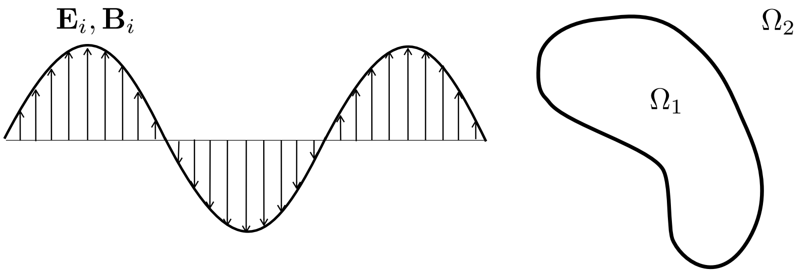


Figure 7·2: Nanoparticle subject to an electromagnetic wave.

where $\mathbf{D}$ is the electric displacement, $\mathbf{E}$ the electric field, $\mathbf{B}$ the magnetic field, $\mathbf{H}$ the magnetizing field, $\mathbf{J}_f$ the free current density, and ρ_f the free charge distribution. Assuming a linear material ($\mathbf{D} = \epsilon \mathbf{E}$, $\mathbf{B} = \mu \mathbf{H}$), Equation (7.1) applied to Figure 7·2, becomes

$$\begin{aligned}
\epsilon_1 \nabla \cdot \mathbf{E}_1 =& \rho_{f1} \qquad \nabla \times \mathbf{E}_1 = -\mu_1 \frac{\partial \mathbf{H}_1}{\partial t} \\
\mu_1 \nabla \cdot \mathbf{H}_1 =& 0 \qquad \nabla \times \mathbf{H}_1 = \mathbf{J}_{f1} + \epsilon_1 \frac{\partial \mathbf{E}_1}{\partial t} \qquad \text{on } \Omega_1, \\
\epsilon_2 \nabla \cdot \mathbf{E}_2 =& \rho_{f2} \qquad \nabla \times \mathbf{E}_2 = -\mu_2 \frac{\partial \mathbf{H}_2}{\partial t} \\
\mu_2 \nabla \cdot \mathbf{H}_2 =& 0 \qquad \nabla \times \mathbf{H}_2 = \mathbf{J}_{f2} + \epsilon_2 \frac{\partial \mathbf{E}_2}{\partial t} \qquad \text{on } \Omega_2, \text{ and} \\
\epsilon_1 \mathbf{E}_1 \cdot \mathbf{n} =& \epsilon_2 \mathbf{E}_2 \cdot \mathbf{n} \qquad \mathbf{H}_1 \cdot \mathbf{n} = \mathbf{H}_2 \cdot \mathbf{n} \qquad \text{on the interface } \Gamma,
\end{aligned} \tag{7.2}$$

where the subscript 1 and 2 correspond to the particle and host medium regions, respectively, being ϵ the permittivity and μ the permeability, and $\mathbf{n}$ is the unit normal to the interface Γ, pointing away from the particle.

If $\epsilon_1 = \epsilon_2$ and $\mu_1 = \mu_2$, the particle has no effect on the incoming wave (*i.e.* no scattering occurs), and the solution to Equation (7.1) is $\mathbf{E}_i$ and $\mathbf{H}_i = \mathbf{B}_i/\mu_2$ everywhere. In that case, Equation (7.2) is

$$\begin{aligned}
\epsilon_2 \nabla \cdot \mathbf{E}_i =& \rho_{f1} \qquad \nabla \times \mathbf{E}_i = -\mu_2 \frac{\partial \mathbf{H}_i}{\partial t} \\
\mu_2 \nabla \cdot \mathbf{H}_i =& 0 \qquad \nabla \times \mathbf{H}_i = \mathbf{J}_{f1} + \epsilon_2 \frac{\partial \mathbf{E}_i}{\partial t} \qquad \text{on } \Omega_1, \\
\epsilon_2 \nabla \cdot \mathbf{E}_i =& \rho_{f2} \qquad \nabla \times \mathbf{E}_i = -\mu_2 \frac{\partial \mathbf{H}_i}{\partial t} \\
\mu_2 \nabla \cdot \mathbf{H}_i =& 0 \qquad \nabla \times \mathbf{H}_i = \mathbf{J}_{f2} + \epsilon_2 \frac{\partial \mathbf{E}_i}{\partial t} \qquad \text{on } \Omega_2, \text{ and} \\
\epsilon_2 \mathbf{E}_i \cdot \mathbf{n} =& \epsilon_2 \mathbf{E}_i \cdot \mathbf{n} \qquad \mathbf{H}_i \cdot \mathbf{n} = \mathbf{H}_i \cdot \mathbf{n} \qquad \text{on the interface } \Gamma.
\end{aligned} \tag{7.3}$$

The electromagnetic field can be decomposed into the incoming and scattered $(\mathbf{E}_s, \mathbf{B}_s)$ field as

$$\begin{aligned}
\mathbf{E} &= \mathbf{E}_i + \mathbf{E}_s \\
\mathbf{B} &= \mathbf{B}_i + \mathbf{B}_s,
\end{aligned} \tag{7.4}$$

Using the definition of Equation (7.4), we can obtain a system of partial-differential equations for the scattered electromagnetic field by subtracting Equation (7.3) from Equation (7.2), which gives

$$\begin{aligned}
\epsilon_1 \nabla \cdot \mathbf{E}_{1s} =& \rho_{f1} \qquad \nabla \times \mathbf{E}_{1s} = -\mu_1 \frac{\partial \mathbf{H}_{1s}}{\partial t} \\
\mu_1 \nabla \cdot \mathbf{H}_{1s} =& 0 \qquad \nabla \times \mathbf{H}_{1s} = \mathbf{J}_{f1} + \epsilon_1 \frac{\partial \mathbf{E}_{1s}}{\partial t} + (\epsilon_1 - \epsilon_2) \frac{\partial \mathbf{E}_i}{\partial t} \qquad \text{on } \Omega_1, \\
\epsilon_2 \nabla \cdot \mathbf{E}_{2s} =& \rho_{f2} \qquad \nabla \times \mathbf{E}_{2s} = -\mu_2 \frac{\partial \mathbf{H}_{2s}}{\partial t} \\
\mu_2 \nabla \cdot \mathbf{H}_{2s} =& 0 \qquad \nabla \times \mathbf{H}_{2s} = \mathbf{J}_{f2} + \epsilon_2 \frac{\partial \mathbf{E}_{2s}}{\partial t} \qquad \text{on } \Omega_2, \text{ and} \\
(\epsilon_1 \mathbf{E}_{1s} - \epsilon_2 \mathbf{E}_{2s}) \cdot \mathbf{n} =& (\epsilon_2 - \epsilon_1) \mathbf{E}_i \cdot \mathbf{n} \\
\mathbf{H}_{1s} \cdot \mathbf{n} =& \mathbf{H}_{2s} \cdot \mathbf{n} \qquad \text{on the interface } \Gamma.
\end{aligned} \tag{7.5}$$

For a time-hamonic wave with frequency ω $(\mathbf{E}(\mathbf{x}, t) = \mathbf{E}(\mathbf{x})e^{i\omega t}$, $\mathbf{B}(\mathbf{x}, t) = \mathbf{B}(\mathbf{x})e^{i\omega t})$,

Equation (7.5) is

$$\begin{aligned}
\epsilon_1\nabla\cdot\mathbf{E}_{1s} =& \rho_{f1} \qquad \nabla\times\mathbf{E}_{1s} = -\mu_1 i\omega\mathbf{H}_{1s} \\
\mu_1\nabla\cdot\mathbf{H}_{1s} =& 0 \qquad \nabla\times\mathbf{H}_{1s} = \mathbf{J}_{f1} + \epsilon_1 i\omega\mathbf{E}_{1s} + (\epsilon_1-\epsilon_2)i\omega\mathbf{E}_i \qquad \text{on } \Omega_1, \\
\epsilon_2\nabla\cdot\mathbf{E}_{2s} =& \rho_{f2} \qquad \nabla\times\mathbf{E}_{2s} = -\mu_2 i\omega\mathbf{H}_{2s} \\
\mu_2\nabla\cdot\mathbf{H}_{2s} =& 0 \qquad \nabla\times\mathbf{H}_{2s} = \mathbf{J}_{f2} + \epsilon_2 i\omega\mathbf{E}_{2s} \qquad \text{on } \Omega_2, \text{ and} \\
(\epsilon_1\mathbf{E}_{1s} - \epsilon_2\mathbf{E}_{2s})\cdot\mathbf{n} =& (\epsilon_2-\epsilon_1)\mathbf{E}_i\cdot\mathbf{n} \\
\mathbf{H}_{1s}\cdot\mathbf{n} =& \mathbf{H}_{2s}\cdot\mathbf{n} \qquad \text{on the interface } \Gamma.
\end{aligned} \tag{7.6}$$

Up to Equation (7.5), **E** and **H** vary in space and time, however, in Equation (7.6), they vary only in space: they are the amplitudes of the time-harmonic quantities. For notation convenience in the notation, from now on, we will refer to the space-varying amplitudes of the time-harmonic electromagnetic field as **E**, **H**, **B** and **D**.

The scattered magnetic field in the host medium can be written in terms of a vector potential **A** such that $\mathbf{B}_{2s} = \nabla\times\mathbf{A}$. For a time-harmonic field, and using Lorentz gauge (Jackson, 1998), **A** is

$$\mathbf{A}(\mathbf{r}) = \frac{\mu}{4\pi}\int\limits_{\Omega_2}\frac{\mathbf{J}_{f2}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}e^{ik|\mathbf{r}-\mathbf{r}'|}\mathrm{d}\Omega' \tag{7.7}$$

where $k = 2\pi/\lambda$ is the wave number and λ the wavelength. The integrand in Equation (7.7) has non-zero values only near the particle, then, when the observation point $\mathbf{r}$ is in the far field, $|\mathbf{r}-\mathbf{r}'| \approx |\mathbf{r}| = r$, and Equation (7.7) becomes:

$$\mathbf{A}(\mathbf{r}) = \frac{\mu}{4\pi}\frac{e^{ikr}}{r}\int\limits_{\Omega_2}\mathbf{J}_{f2}(\mathbf{r}')\mathrm{d}\Omega'. \tag{7.8}$$

Integration by parts of each component in Equation (7.8) leads to

$$\mathbf{A}(\mathbf{r}) = \frac{\mu}{4\pi}\frac{e^{ikr}}{r}\int\limits_{\Omega_2}\mathbf{r}'\nabla\cdot\mathbf{J}_{f2}(\mathbf{r}')\mathrm{d}\Omega', \tag{7.9}$$

and using Ampère's ($\nabla\cdot\mathbf{J}_{2s} = i\omega\epsilon_2\nabla\cdot\mathbf{E}_{2s}$) and Gauss ($\epsilon_2\nabla\cdot\mathbf{E} = \rho_{f2}$) laws from Equation (7.6) , we get

$$\mathbf{A}(\mathbf{r}) = \frac{i\epsilon\omega\mu}{4\pi}\frac{e^{ikr}}{r}\int\limits_{\Omega_2}\mathbf{r}'\nabla\cdot\mathbf{E}_{s2}(\mathbf{r}')\mathrm{d}\Omega = \frac{i\epsilon\omega\mu}{4\pi}\frac{e^{ikr}}{r}\int\limits_{\Omega_2}\mathbf{r}'\rho_{f2}(\mathbf{r}')\mathrm{d}\Omega'. \tag{7.10}$$

The final integral in Equation (7.10) is the definition of the dipole moment $\mathbf{p}$, then

$$\mathbf{A}(\mathbf{r}) = \frac{i\epsilon\omega\mu}{4\pi}\frac{e^{ikr}}{r}\mathbf{p}. \tag{7.11}$$

We know that $\mathbf{B}_{2s} = \mu\mathbf{H}_{2s} = \nabla\times\mathbf{A}$, then, evaluating this expression with Equation (7.11) gives:

$$\mathbf{H}_{2s} = \frac{k\omega}{4\pi}\frac{e^{ikr}}{r}\left(1-\frac{1}{ikr}\right)\hat{\mathbf{r}}\times\mathbf{p}, \tag{7.12}$$

where $\hat{\mathbf{r}}$ is a unit vector in the direction of the observation point.

Ampère's law in Equation (7.6), for $\mathbf{J}_{f2} = 0$ at the observation point, reads

$$\mathbf{E}_{2s} = \frac{1}{\epsilon_2 i\omega}\nabla\times\mathbf{H}_{2s}. \tag{7.13}$$

Applying Equation (7.12) into Equation (7.13), yields an expression for the scattered electric field:

$$\mathbf{E}_{2s} = \frac{1}{4\pi\epsilon}\left[k^2\frac{e^{ikr}}{r}(\hat{\mathbf{r}}\times\mathbf{p})\times\hat{\mathbf{r}} + e^{ikr}\left((3\hat{\mathbf{r}}(\mathbf{p}\cdot\hat{\mathbf{r}})-\mathbf{p})\left(\frac{1}{r^3}-\frac{ik}{r^2}\right)\right)\right] \tag{7.14}$$

but in the far field, $1/r^3$ and $1/r^2$ terms are negligible, leaving

$$\mathbf{E}_{2s} = \frac{1}{4\pi\epsilon_2}k^2\frac{e^{ikr}}{r}(\hat{\mathbf{r}}\times\mathbf{p})\times\hat{\mathbf{r}}. \tag{7.15}$$

7.1.2 The extinction cross-section

Suppose there is a beam of light with intensity I_i in a non-absorbing medium, and a detector is placed downstream. The detector captures the entire beam, and measures a detected energy $W_d = I_i A_D$, where A_D is the area of the detector. Now, we place a particle between the source of light and detector: the energy measured by the detector drops ($W_d < I_i A_D$), and we say that the presence of the particle causes an extinction of the incident beam. The energy that did not reach the detector was either scattered in different directions or absorbed by the particle.

The extinction cross-section is a common way to quantify how much extinction is caused by a particle. It is defined as the ratio between the extinct energy (W_{ext}) and the intensity of the incoming wave:

$$C_{\text{ext}} = \frac{W_{\text{ext}}}{I_i}, \tag{7.16}$$

where the extinct energy is the sum of the scattered and absorbed energies ($W_{\text{ext}} = W_{\text{sca}} + W_{\text{abs}}$). Equivalently, the scattered and absorbed cross-sections are defined as

$$C_{\text{sca}} = \frac{W_{\text{sca}}}{I_i} \text{ and } C_{\text{abs}} = \frac{W_{\text{abs}}}{I_i}, \tag{7.17}$$

In an experimental setting, part of the scattered energy (W_{sca}) will go in the forward direction, towards the detector; however, when the detector is in the far field, this contribution becomes negligible (Bohren and Huffman, 1983). Therefore, the energy on the detector is

$$W_d = I_i[A_D - C_{\text{ext}}]. \tag{7.18}$$

High levels of extinction cross-section is an indication that the plasmons are resonating with the incoming electric field.

7.1.3 The optical theorem

One way to calculate the extinction-cross section is to define a spherical shell around the particle, compute the energy crossing that surface, and use the definition of extinction cross-section in Equation (7.16). The optical theorem is a more straightforward way to compute C_{ext}.

The electric field in the long-range limit ($kr >> 1$) can be written as

$$\mathbf{E}_{2s}(\mathbf{r})_{r\to\infty} = \frac{e^{ikr}}{r}\mathbf{F}(\mathbf{k}, \mathbf{k}_0), \tag{7.19}$$

where $\mathbf{F}$ is the forward-scattering amplitude, $\mathbf{k}$ is the scattered wave vector in the direction of propagation and $\mathbf{k}_0$ the wave vector of the incident field. The wave number k is the magnitude of the wave vector $\mathbf{k}$, which points in the direction of propagation.

The optical theorem relates the extinction cross-section with the forward scattering amplitude (Bohren and Huffman, 1983; Jackson, 1998) as

$$C_{\text{ext}} = \frac{4\pi}{k}\,\text{Im}\left[\frac{\hat{\mathbf{e}}_i}{|\mathbf{E}_i|}\mathbf{F}(\mathbf{k} = \mathbf{k}_0, \mathbf{k}_0)\right], \tag{7.20}$$

where $\hat{\mathbf{e}}_i$ is the incoming polarization vector —a unit vector that points in the direction of $\mathbf{E}_i$— and $|\mathbf{E}_i|$ is the magnitude of the incoming electric field.

The expression for C_{ext} in Equation (7.20) is valid in a non-absorbing host medium. The definition of the extinction cross-section is less clear if the medium is lossy, since extinction occurs even when the particle is not there. For an absorbing medium, the wave number k is complex valued

$$k = k' + ik'' = \frac{2\pi}{\lambda}n, \tag{7.21}$$

where n is the refraction index of the host medium.

To account for the lossy medium, Chylek (Chỳlek, 1977) explicitly computed the energy loss in a control volume, but this formulation depends on the size of the control volume, which is an arbitrary parameter. In 1979, Bohren and Gilra (Bohren and Gilra, 1979) presented an equation for spherical particles which was very similar to Equation (7.20), and later Videen and Sun (Videen and Sun, 2003) arrived to the same expression for a general geometry. More recently, Mishchenko (Mishchenko, 2007) presented a correction to the work of Bohren, Videen and co-workers, which led to

$$C_{\text{ext}} = \frac{4\pi}{k'} \operatorname{Im}\left[\frac{\hat{\mathbf{e}}_i}{|\mathbf{E}_i|}\mathbf{F}(\mathbf{k}=\mathbf{k}_0, \mathbf{k}_0)\right]. \tag{7.22}$$

Equation (7.22) corresponds to the extinction caused by the particle only, and not the host medium. This definition removes any ambiguities in C_{ext}, and is closer to what is observed in experimental settings. In this work, we use Equation (7.22) to compute the extinction cross-section.

7.2 Electrostatic approximation of localized surface plasmon resonance

This section demonstrates that electrostatic theory is a first order approximation of the full Maxwell's equations, and it is valid in the long-wavelength limit. This derivation has been presented in the past (Mayergoyz and Zhang, 2007), but we review it here, as it is key to allow us to use our BEM approach for LSPR modeling.

Figure 7·2 is a sketch of the system that we will analyze for this derivation. Region Ω_1 is a nanoparticle immersed in a host medium Ω_2, and subjected to an incoming electrictromagnetic wave with $\mathbf{E}_i$ and $\mathbf{B}_i$. The following derivation considers only one particle in the domain, however, the analysis extends to cases with many particles. Also, in this analysis, we do not consider the presence of salt ions in the host medium.

Equation (7.6) is a system of partial-differential equations that models a scattered, time-harmonic electromagnetic field. In the setting with no charges or currents of Figure 7·2, Equation (7.6) becomes

$$\begin{aligned} \nabla \cdot \mathbf{E}_{1s} &= 0 \qquad \nabla \times \mathbf{E}_{1s} = -\mu_1 i\omega \mathbf{H}_{1s} \\ \nabla \cdot \mathbf{H}_{1s} &= 0 \qquad \nabla \times \mathbf{H}_{1s} = \epsilon_1 i\omega \mathbf{E}_{1s} + (\epsilon_1 - \epsilon_2) i\omega \mathbf{E}_i \qquad \text{on } \Omega_1 \\ \nabla \cdot \mathbf{E}_{2s} &= 0 \qquad \nabla \times \mathbf{E}_{2s} = -\mu_2 i\omega \mathbf{H}_{2s} \\ \nabla \cdot \mathbf{H}_{2s} &= 0 \qquad \nabla \times \mathbf{H}_{2s} = \epsilon_2 i\omega \mathbf{E}_{2s} \qquad \text{on } \Omega_2 \\ (\epsilon_1 \mathbf{E}_{1s} - \epsilon_2 \mathbf{E}_{2s}) \cdot \mathbf{n} &= (\epsilon_2 - \epsilon_1) \mathbf{E}_i \cdot \mathbf{n} \\ (\mathbf{H}_{1s} - \mathbf{H}_{2s}) \cdot \mathbf{n} &= 0 \qquad \text{on the interface } \Gamma, \end{aligned} \tag{7.23}$$

To study the effect of the size of the particle in the scattered field, we will define the following scaled quantities:

$$\begin{aligned} \mathbf{e}_i &= \sqrt{\epsilon_2}\mathbf{E}_i, \qquad \mathbf{e}_s = \sqrt{\epsilon_2}\mathbf{E}_s, \\ \mathbf{h}_s &= \sqrt{\mu_2}\mathbf{H}_s \qquad \mathbf{x}' = \frac{\mathbf{x}}{d}, \end{aligned} \tag{7.24}$$

for d the particle size, and $\mathbf{x}'$ a scaled domain. If we replace Equation (7.24) into Equation (7.23), we get

$$\begin{aligned} \nabla \cdot \mathbf{e}_{1s} &= 0 \qquad \nabla \times \mathbf{e}_{1s} = -i\beta \mathbf{h}_{1s} \frac{\mu_1}{\mu_2} \\ \nabla \cdot \mathbf{h}_{1s} &= 0 \qquad \nabla \times \mathbf{h}_{1s} = \frac{\epsilon_1 - \epsilon_2}{\epsilon_2} i\beta \mathbf{e}_i + \frac{\epsilon_1}{\epsilon_2} i\beta \mathbf{e}_{1s} \qquad \text{on } \Omega_1 \\ \nabla \cdot \mathbf{e}_{2s} &= 0 \qquad \nabla \times \mathbf{e}_{2s} = -i\beta \mathbf{h}_{2s} \\ \nabla \cdot \mathbf{h}_{2s} &= 0 \qquad \nabla \times \mathbf{h}_{2s} = i\beta \mathbf{e}_{2s} \qquad \text{on } \Omega_2 \\ (\epsilon_1 \mathbf{e}_{1s} - \epsilon_2 \mathbf{e}_{2s}) \cdot \mathbf{n} &= (\epsilon_2 - \epsilon_1) \mathbf{e}_i \cdot \mathbf{n} \\ (\mathbf{h}_{1s} - \mathbf{h}_{2s}) \cdot \mathbf{n} &= 0 \qquad \text{on the interface } \Gamma, \end{aligned} \tag{7.25}$$

with $\beta = \omega d\sqrt{\epsilon_2 \mu_2}$. We can expand $\mathbf{e}$ and $\mathbf{h}$ in terms of β as

$$\begin{aligned} \mathbf{e} &= \mathbf{e}_s^{(0)} + \beta \mathbf{e}_s^{(1)} + \beta^2 \mathbf{e}_s^{(2)} + \ldots \\ \mathbf{h} &= \mathbf{h}_s^{(0)} + \beta \mathbf{h}_s^{(1)} + \beta^2 \mathbf{h}_s^{(2)} + \ldots, \end{aligned} \tag{7.26}$$

and considering only the zeroth order terms in Equation (7.25) yields

$$\begin{aligned} \nabla \cdot \mathbf{e}_{1s}^{(0)} = 0 &\qquad \nabla \times \mathbf{e}_{1s}^{(0)} = 0 \\ \nabla \cdot \mathbf{h}_{1s}^{(0)} = 0 &\qquad \nabla \times \mathbf{h}_{1s}^{(0)} = 0 \\ \nabla \cdot \mathbf{e}_{2s}^{(0)} = 0 &\qquad \nabla \times \mathbf{e}_{2s}^{(0)} = 0 \\ \nabla \cdot \mathbf{h}_{2s}^{(0)} = 0 &\qquad \nabla \times \mathbf{h}_{2s}^{(0)} = 0 \\ (\epsilon_1 \mathbf{e}_{1s}^{(0)} - \epsilon_2 \mathbf{e}_{2s}^{(0)}) \cdot \mathbf{n} &= (\epsilon_2 - \epsilon_1)\mathbf{e}_i \cdot \mathbf{n} \\ (\mathbf{h}_{1s}^{(0)} - \mathbf{h}_{2s}^{(0)}) \cdot \mathbf{n} &= 0. \end{aligned} \tag{7.27}$$

Equation (7.27) has the form of an electrostatic field: the electric and magnetic fields are decoupled, $\mathbf{h}^{(0)} = 0$, and $\mathbf{e}^{(0)}$ can be described by a scalar potential because $\nabla \times \mathbf{e}^{(0)} = 0$.

Equation (7.27) is a good approximation of Equation (7.5), as long as β is small. On the host medium, the wave speed is $c = 1/\sqrt{\epsilon_2 \mu_2}$, then $\beta = \omega d\sqrt{\epsilon_2 \mu_2} = d\omega/c = d/\lambda$. In this chapter, we look at the long-wavelength limit, *i.e.* when the particle is much smaller than the wavelength of the incoming wave, making Equation (7.27) a valid approximation. This result allows us to use electrostatic theory to model LSPR.

7.3 Boundary integral formulation

7.3.1 Electrostatic potential under an incoming electric field

From now on, we will focus only on the electric field, because the zeroth order term of the magnetic field is zero everywhere. Equation (7.27), written in terms of $\mathbf{E}$ and

$\mathbf{x}$ rather than the scaled quantities $\mathbf{e}$ and $\mathbf{x}'$, is

$$\begin{aligned} \nabla \cdot \mathbf{E}_{1s}^{(0)} &= 0 \qquad \nabla \times \mathbf{E}_{1s}^{(0)} = 0 \\ \nabla \cdot \mathbf{E}_{2s}^{(0)} &= 0 \qquad \nabla \times \mathbf{E}_{2s}^{(0)} = 0 \\ (\epsilon_1 \mathbf{E}_{1s}^{(0)} - \epsilon_2 \mathbf{E}_{2s}^{(0)}) \cdot \mathbf{n} &= (\epsilon_2 - \epsilon_1)\mathbf{E}_i \cdot \mathbf{n}. \end{aligned} \tag{7.28}$$

The curl of $\mathbf{E}_{1s}^0$ is zero, hence, there is a scalar potential $-\nabla \phi_s = \mathbf{E}_s^{(0)}$, and Equation (7.28) takes the form

$$\begin{aligned} &\nabla^2 \phi_{1s} = 0 \qquad \nabla^2 \phi_{2s} = 0 \qquad \text{on } \Omega_1,\, \Omega_2 \\ &\epsilon_1 \frac{\partial \phi_{1s}}{\partial \mathbf{n}} - \epsilon_2 \frac{\partial \phi_{2s}}{\partial \mathbf{n}} = (\epsilon_2 - \epsilon_1) \frac{\partial \phi_i}{\partial \mathbf{n}} \qquad \phi_{1s} = \phi_{2s} \qquad \text{on the boundary } \Gamma. \end{aligned} \tag{7.29}$$

Using the same principles that led to Equation (3.2), we can rewrite the system of partial differential equations in Equation (7.29) as

$$\begin{aligned} \frac{\phi_{1s,\Gamma}}{2} + K_L^\Gamma(\phi_{1s,\Gamma}) - V_L^\Gamma\left(\frac{\partial}{\partial \mathbf{n}}\phi_{1s,\Gamma}\right) &= 0 \\ \frac{\phi_{2s,\Gamma}}{2} - K_L^\Gamma(\phi_{2s,\Gamma}) + V_L^\Gamma\left(\frac{\partial}{\partial \mathbf{n}}\phi_{2s,\Gamma}\right) &= 0 \quad \text{on } \Gamma, \end{aligned} \tag{7.30}$$

and applying the interface conditions in Equation (7.29), we obtain

$$\begin{aligned} \frac{\phi_{1s,\Gamma}}{2} + K_L^\Gamma(\phi_{1s,\Gamma}) - V_L^\Gamma\left(\frac{\partial}{\partial \mathbf{n}}\phi_{1s,\Gamma}\right) &= 0 \\ \frac{\phi_{1s,\Gamma}}{2} - K_L^\Gamma(\phi_{1s,\Gamma}) + \frac{1}{\epsilon_2} V_L^\Gamma\left(\epsilon_1 \frac{\partial}{\partial \mathbf{n}}\phi_{1s,\Gamma} - (\epsilon_2 - \epsilon_1)\frac{\partial}{\partial \mathbf{n}}\phi_{i,\Gamma}\right) &= 0 \quad \text{on } \Gamma. \end{aligned} \tag{7.31}$$

In matrix form, Equation (7.31) is

$$\begin{bmatrix} \frac{1}{2} + K_L^\Gamma & -V_L^\Gamma \\ \frac{1}{2} - K_L^\Gamma & \frac{\epsilon_1}{\epsilon_2} V_L^\Gamma \end{bmatrix} \begin{bmatrix} \phi_{1s,\Gamma} \\ \frac{\partial}{\partial \mathbf{n}}\phi_{1s,\Gamma} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\epsilon_2 - \epsilon_1}{\epsilon_2} \frac{\partial \phi_i}{\partial \mathbf{n}} \end{bmatrix} \tag{7.32}$$

To solve Equation (7.32), we use the discretization scheme described in Section 3.3.

7.3.2 Boundary integral expression of the dipole moment

According to Equation (7.14), the scattered electric field far away depends on the dipole moment of the scatterer. The dipole moment is defined as

$$\mathbf{p} = \int\limits_{\Omega} \mathbf{r}\rho \mathrm{d}\Omega, \tag{7.33}$$

but we can rewrite this using Gauss's law

$$\mathbf{p} = -\epsilon_2 \int\limits_{\Omega} \mathbf{r}\nabla^2 \phi_{2s} \mathrm{d}\Omega. \tag{7.34}$$

Using index notation, Equation (7.34) is

$$p_i = -\epsilon_2 \int\limits_{\Omega} r_i \frac{\partial}{\partial r_j} \phi_{2s} \mathrm{d}\Omega, \tag{7.35}$$

and integrating by parts twice, we get

$$p_i = -\epsilon_2 \left[-\oint\limits_{\Gamma} r_i \frac{\partial}{\partial n_j} \phi_{2s} \mathrm{d}\Gamma + \oint\limits_{\Gamma} \frac{\partial r_i}{\partial n_j} \phi_s \mathrm{d}\Gamma + \int\limits_{\Omega} \frac{\partial}{\partial r_j} \frac{\partial r_i}{\partial r_j} \phi \mathrm{d}\Omega \right]. \tag{7.36}$$

We know that $\frac{\partial}{\partial r_j}\frac{\partial r_i}{\partial r_j} = 0$ and that $\frac{\partial r_i}{\partial n_j} = \frac{\partial r_i}{\partial r_j} n_j = \delta_{ij} n_j = n_i$, then, Equation (7.36) becomes

$$p_i = \epsilon_2 \left[\oint\limits_{\Gamma} r_i \frac{\partial}{\partial n_j} \phi_{2s} \mathrm{d}\Gamma - \oint\limits_{\Gamma} n_i \phi \mathrm{d}\Gamma \right]. \tag{7.37}$$

With BEM, we compute the electrostatic potential on the surface of the scatterer, to then obtain the dipole moment with Equation (7.37) and the scattered electric field with Equation (7.14). Having the scattered electric field, Equation (7.19) gives us the forward scattering amplitude, which we feed into Equation (7.22) to get the extinction cross-section.

7.4 Results for an isolated nanoparticle

This section presents a verification exercise that compares numerical calculations of the extinction cross-section using BEM, with analytical expressions, available for spherical geometries. We did not use `PyGBe` to obtain the results of this section, but a simpler version of the code which explicitly creates the BEM stiffness matrix, and does not use any acceleration technique. This version is also available in `PyGBe`'s Github website.[1]

The analytical solution of the extinction cross-section for a spherical particle in vacuum, using full electromagnetic theory, is available from Mie theory (Mie, 1908). However, in the long-wavelength limit, electrostatics yields the same expression (Bohren and Huffman, 1983), which is

$$C_{\text{ext}} = 4\pi a^3 k \, \text{Im}\left(\frac{\epsilon_p/\epsilon_m - 1}{\epsilon_p/\epsilon_m - 2}\right) \tag{7.38}$$

where a is the radius of the sphere, k the wave number, ϵ_p the particle's dielectric constant, and ϵ_m the dielectric constant of the host medium, in this case, vacuum permittivity ($\epsilon_m = \epsilon_0$). For a lossy medium (Mishchenko, 2007), Equation (7.38) is

$$C_{\text{ext}} = \frac{4\pi a^3}{k'} \, \text{Im}\left(k^2 \frac{\epsilon_p/\epsilon_m - 1}{\epsilon_p/\epsilon_m - 2}\right) \tag{7.39}$$

with $k = k' + k''i$. If the host medium is non-absorbing, k is real-valued, and Equation (7.39) converges to Equation (7.38).

The electrostatic approximation detailed in Section 7.2, reduces the simulation of electromagnetic wave scattering of a spherical nanoparticle to a sphere under a constant electric field, like the one in Figure 7·3. The results in this section are numerical solutions of the setting described by Figure 7·3.

Figure 7·4 is a mesh refinement study of the extinction cross-section of a sil-

---

[1] `https://github.com/barbagroup/pygbe`

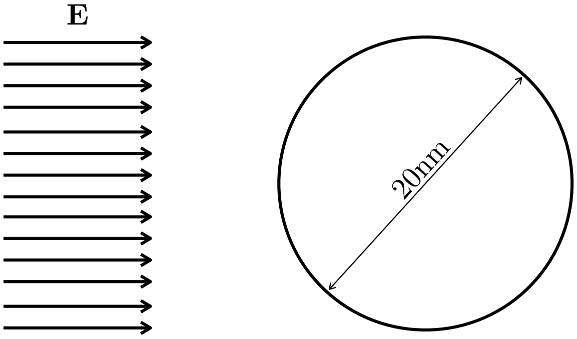


Figure 7·3: Spherical nanoparticle subject to a constant electric field.

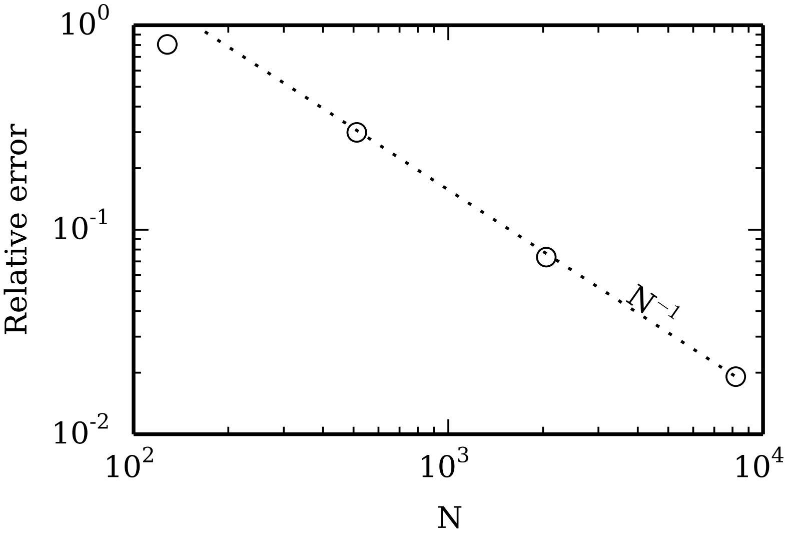


Figure 7·4: Convergence of extinction cross-section of single silver nanoparticle.

ver sphere with diameter 20nm immersed in water, subject to a z-polarized electric field with wavelength 380nm and magnitude 1. Under these conditions, water has a dielectric constant of $1.796 + 8.5 \cdot 10^{-9}i$ (Johnson and Christy, 1972) and silver of $-3.383 + 0.1914i$ (Hale and Querry, 1972). In these simulations, we used 3 Gauss quadrature points per far-away element, 37 Gauss quadrature points for near singular integrals —closer than $\sqrt{2 \cdot \text{Area}}$—, and 9 Gauss quadrature points per side for the semi-analytical integration in the singular elements. The GMRES tolerance was 10^{-6}, and meshes had 128, 512, 2048 and 8192 elements. We computed the errors using the analytical solution as a reference, which is 3587.25nm^2.

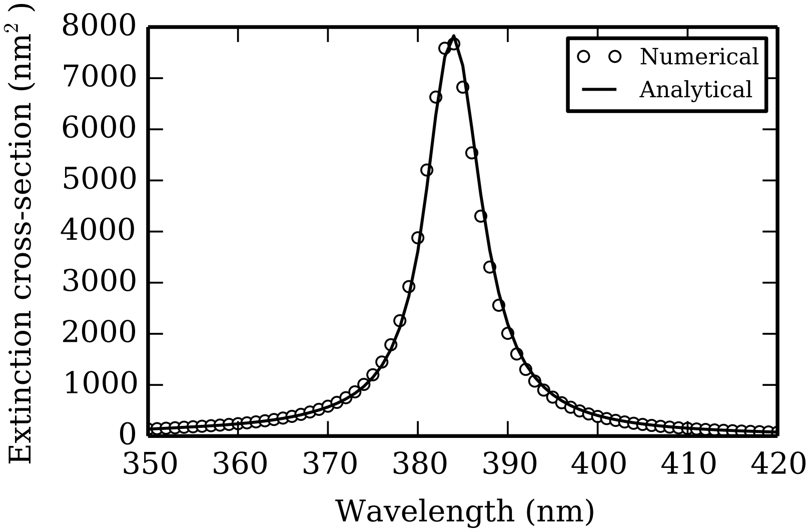


Figure 7·5: Extinction cross-section with wavelength for a 20nm silver sphere immersed in water.

The 1/N convergence in Figure 7·4, proves that the numerical solution is correctly resolved by the meshes.

Figure 7·5 and Figure 7·6 show the relation between extinction cross-section and wavelength, for a silver and a gold sphere immersed in water. The diameter of the spheres was 20nm, and the mesh contained 2048 elements. These simulations use the same parameters that led to the results in Figure 7·4, taking the wavelength-dependent complex-valued diffraction index (n) from Johnson and Christy for gold and silver (Johnson and Christy, 1972), and Hale and Querry for water (Hale and Querry, 1972). We calculated the dielectric constant using the relation $\epsilon = n^2$. The peak in extinction cross-section is an indication that the plasmons of the metallic sphere are resonating with the incoming electric field.

7.5 LSPR sensor response calculations

LSPR biosensors detect a target molecule because the plasmon resonance frequency changes when the target is near the metallic nanoparticle. There are variations of

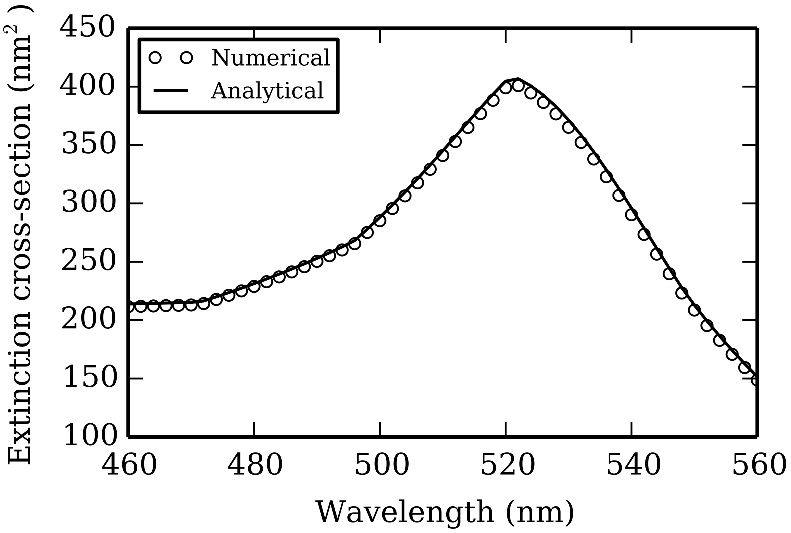


Figure 7·6: Extinction cross-section with wavelength for a 20nm gold sphere immersed in water.

this technology, such as LSPR imaging biosensors, where the change in intensity on the detector is also taken in consideration. In this section, we will perform proof-of-concept calculations that show how our BEM approach can be used to model the response in LSPR biosensors. We did not use `PyGBe` to obtain the results of this section, but a simpler version of the code which explicitly creates the BEM stiffness matrix, and does not use any acceleration technique. This version is also available in `PyGBe`'s Github website.[2]

Setup. We placed a 20nm silver sphere in water, with six 4nm spherical proteins around it, as sketched by Figure 7·7. We took the complex-valued, frequency-dependent refraction index (n) from Johnson and Christy (Johnson and Christy, 1972) for silver, Hale and Querry (Hale and Querry, 1972) for water, and Paliwal, Tomar and Gupta (Paliwal et al., 2014) for the proteins, and computed the corresponding dielectric constant using the relation $\epsilon = n^2$. This section presents four tests:

---

[2] `https://github.com/barbagroup/pygbe`

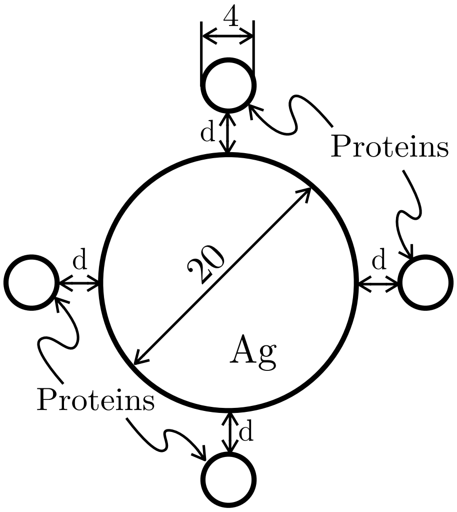


Figure 7·7: Setup for LSPR response calculations. Two spherical proteins are not represented in this figure: they are also placed a distance d from the silver surface, on the axis going into the paper.

- Isolated silver sphere,
- Silver sphere with proteins 1nm away,
- Silver sphere with proteins 2nm away, and
- Silver sphere with proteins 4nm away.

Mesh convergence study. Figure 7·8 shows the result of a mesh convergence study of the silver sphere with the proteins 1nm away, subjected to an incoming electric field with wavelength $\lambda =$380nm. Under these conditions, the dielectric constants are $\epsilon_{\text{Ag}} = -3.383+0.191i$ for silver (Johnson and Christy, 1972), $\epsilon_{\text{w}} = 1.796+8.5\cdot 19^{-9}i$ for water (Hale and Querry, 1972) and $\epsilon_{\text{prot}} = 9.61+0.149i$ for the protein (Paliwal et al., 2014). The mesh of the silver sphere has 0.1, 0.4 and 1.6 elements per square nanometer, respectively, whereas the protein spheres have 0.65, 2.6 and 10.2 elements per square nanometer, respectively. The error calculation used the Richardson extrapolated

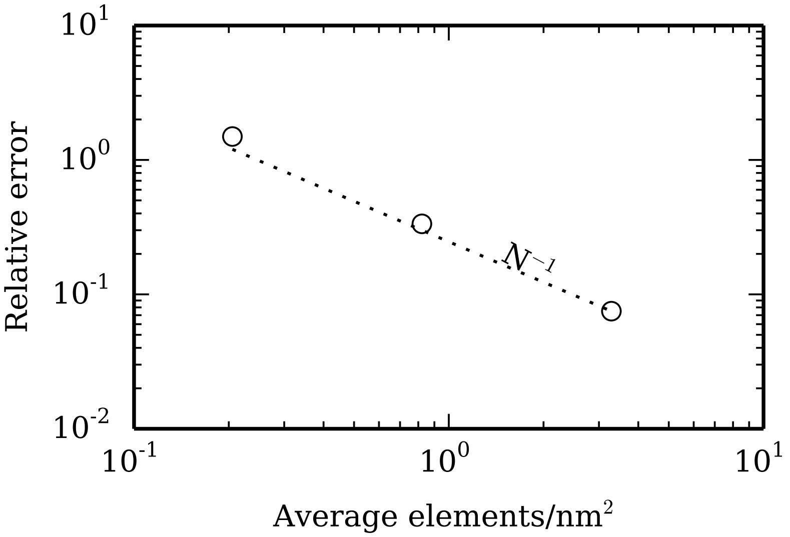


Figure 7·8: Convergence of the extinction cross-section of a 20nm silver sphere with six spherical proteins placed 1nm away from the silver surface, versus the average number of elements per square nanometer. This setup is sketched in Figure 7·7.

value of extinction cross-section as a reference, which was $C_{\text{ext}} = 2538.5\text{nm}^2$, and the observed order of convergence was 1.08.

The error decaying as 1/N in Figure 7·8 implies that the mesh is resolving the geometry, and hence, the extinction cross-section, correctly.

LSPR response calculations. To investigate the relation between the LSPR response and the sensor-analyte distance using our BEM approach, we meshed the silver nanosphere with 1.6 elements per square nanometer, and the proteins with 10.2 elements per square nanometer. The mesh refinement study shows that this mesh is appropriate, and, for $\lambda = 380\text{nm}$, yields a result that is $\sim 7.5\%$ away from the extrapolated value.

Figure 7·9 shows the variation of the extinction cross-section with wavelength, for each of the four cases mentioned above. The red shift in resonance frequency in the presence of an analyte agrees with experimental observations, moreover, the increase in extinction cross section is also observed in LSPR imaging-type biosensors

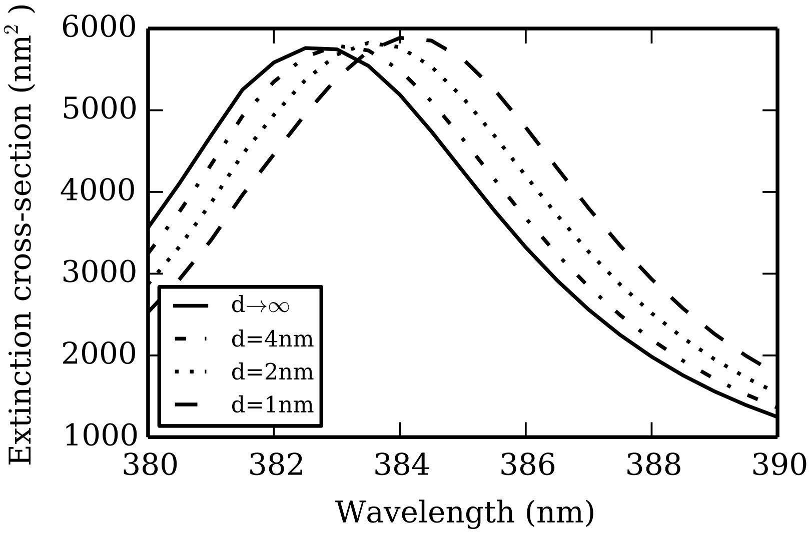


Figure 7·9: Extinction-cross section with wavelength for a 20nm silver sphere with six spherical proteins placed 1, 2 and 4nm away from the silver surface, and at infinity. This setup is sketched in Figure 7·7.

(Raphael et al., 2013). This indicates that the electrostatic approximation is capable of reproducing the characteristic resonance frequency shift in LSPR biosensors.

The sensitivity of the biosensor is the relation between the size of the shift and the number of analytes bound to the ligand molecule. Figure 7·9 shows that sensitivity is affected by the distance between the analyte and the sensor: the closest they are, the larger the shift. For this case with six spherical proteins, as the protein-sensor distance doubles, the shift drops by a constant amount of 0.5nm.

Experiments suggest that the distance between the analyte and the nanoparticle has a large impact on sensitivity (Haes et al., 2004), and these proof-of-concept calculations show the potential to build a sensitivity versus distance map, using electrostatic theory. Moreover, with this approach we could explore structural effects of the analyte and the nanoparticle. Simplifications in existing methods to model LSPR biosensor response make it impossible to study such structural and positioning details (Davis et al., 2010; Antosiewicz et al., 2011; Haes and Van Duyne, 2002).

Chapter 8

Conclusions

In this work we used the implicit-solvent model to study biomolecular processes that are dominated by electrostatics, like protein solvation, binding and adsorption. We also applied this model to simulations of localized surface plasmon resonance of metallic nanoparticles, and computed the shift in resonance frequency when a protein is nearby. The implicit-solvent model is based on dividing the domain into a solvent and protein region, and solving the Poisson-Boltzmann equation in the solvent, and the Poisson equation in the protein. Those partial-differential equations are coupled through interface conditions in the solvent-excluded surface.

Contributions of the work

We implemented a numerical solution of the coupled system of partial-differential equations with the boundary-element method (BEM), in a code called `PyGBe`.[1] `PyGBe` is written in Python and interfaces with GPUs in the most computationally intensive parts, using PYCUDA. Compared to volumetric methods, BEM has the advantage that the molecular surface, or solvent-excluded surface, is accurately represented, and the interface conditions are exactly enforced at those points. Results in Chapter 4 suggest that a BEM approach is better suited for high accuracy calculations, with errors under $\sim 1\%$.

Chapter 4 discusses the importance of Stern layers and solvent-filled cavities in

---

[1] `https://github.com/barbagroup/pygbe`

the implicit-solvent model. These features are easily considered in volumetric numerical methods, but in BEM approaches they involve the generation of a new surface, greatly affecting the structure of the linear system and the computational time. As a consequence, most BEM implementations do not consider Stern layers or solvent-filled cavities, but `PyGBe`'s design makes it easy include or remove them. Tests indicate that for solvation energy calculations, Stern layers and solvent-filled cavities have a small impact, however, they become relevant for binding energy calculations, especially when the proteins involved are similar in size.

The extension of `PyGBe` to consider charged surfaces detailed in Chapter 5 is another contribution of this work. We used this extension to study how proteins interact with charged surfaces, which is relevant in protein adsorption. To verify our extension, we developed a closed solution for a spherical protein near a sphere with prescribed charge or potential, which compared as expected with the numerical solution. Then, to validate the model, we computed the orientation of protein G B1 adsorbed on a charged surface with `PyGBe`, and compared it with molecular dynamics simulations and experimental observations, leading to coincident results. This proves that the implicit-solvent model is appropriate to study protein adsorption on charged surfaces.

Using the extension of `PyGBe`, we investigated the interaction of immunoglobulin G with a charged surface. This setting is relevant in biosensing applications, as the biosensor is functionalized with a self-assembled monolayer of a small charged molecule, where the ligand molecules —usually antibodies— are adsorbed. In the context of the implicit-solvent model, we considered the self-assembled monolayer as a surface with prescribed charge, and then probed the energy for different orientations of the antibody. Results in Chapter 6 indicate that low salt concentration and high positive surface charge leads to better orientations of the antibody for biosensing applications.

The final chapter uses the implicit-solvent model with electrostatics to study plasmons in metallic nanoparticles. BEM has been used in the past to study this phenomenon, but our work adds the effect of proteins that could be in the solvent. This is important for localized surface plasmon resonance biosensors, which relies on the shift of the plasmon resonance frequency as proteins come close to the sensor. The proof-of-concept results in Chapter 7 suggest that an electrostatic model is capable of reproducing the plasmon resonance frequency shift observed in experiments.

Future work

The system of integral equations that `PyGBe` solves exhibits the behavior of a first kind integral equation: the condition number is affected by the size of the matrix. As a consequence, the number of iterations of the linear solver increases with the number of elements. The system can be formulated entirely in terms of a second kind integral equation (Juffer et al., 1991), where the condition number is independent of the number of elements. This formulation is more involved because it contains hypersingular operators and the integrands have more terms, but the extra time taken per iteration, compared to its first-kind counterpart, is quickly balanced because it requires less GMRES iterations. In fact, most BEM implementation use this formulation (Lu et al., 2006; Geng and Krasny, 2013). We plan to implement the second kind formulation in `PyGBe`'s framework, to decrease the computer time.

Also related to decreasing the computer time, we are interested in developing a version of `PyGBe` for distributed systems. This will also allow us to do larger tests, that include many interacting molecules, to study crowding effects. Luckily, the treecode —`PyGBe`'s numerical engine— is an algorithm that maps very well to distributed systems, and has been successfully parallelized in the past.

In Chapter 6, we studied the orientation of immunoglobulin G adsorbed on a charged surface. Even though immunoglobulin G is a widely used ligand molecule, they are too big for, for example, localized surface plasmon resonance biosensors. There is ongoing work to find mutations of single-domain or single-chain antibodies that would improve the orientation, and hence, the sensitivity of localized surface plasmon resonance biosensors. In the near future, we will experiment with single-domain antibodies, to then look at the effect of mutations on the preferred orientation of the protein. This will be of great help to guide experimentalists of this area, in their search for better ligand orientation.

The calculations from Chapter 7 were done with a simpler version of `PyGBe`, that constructs the BEM stiffness matrix explicitly and uses a built-in linear solver in `scipy`, a Python library, to solve the linear system. We resort to existing libraries because `PyGBe`'s GMRES cannot handle complex-valued entries, as it is required for plasmonic simulations. We plan to add this capability to the main code. This will allow us to perform larger simulations and study the effects of the protein's and nanoparticle's structure, and how the relative positioning between them impacts sensitivity.

Chapter 7 discusses the use of the electrostatic approximation in plasmonic simulations. Even though we saw that this approximation is capable of reproducing the plasmon resonance shift seen in experiments, it is limited by size. We plan to extend this for larger problems by solving the full Maxwells' equations. This would allow us to study bigger sensors, and cases when there are many target molecules and the whole system is not within the limits for the electrostatic approximation.

List of Journal Abbreviations

Adv. Colloid Interface Sci.	Advances in Colloid and Interface Science
Adv. Mater.	Advanced Materials
Anal. Chem.	Analytical Chemistry
Anal. Chim. Acta	Analytica Chimica Acta
Annu. Rev. Phys. Chem.	Annual Review of Physical Chemistry
Ann. Rev. Biophys. Biophys. Chem.	Annual Review of Biophysics and Biophysical Chemistry
Appl. Opt.	Applied Optics
Biophys. Chem.	Biophysical Chemistry
Biophys. J.	Biophysical Journal
Biosens. Bioeletron.	Biosensors and Bioelectronics
Chem. Commun.	Chemical Communications
Chem. Rev.	Chemical Reviews
Comput. Phys. Commun.	Computer Physics Communications
Comput. Sci. Disc.	Computational Science and Discovery
Commun. Math. Sci.	Communications in Mathematical Sciences
Curr. Opin. Struct. Biol.	Current Opinion in Structural Biology
IEEE J. Solid-State Circuits	IEEE Journal of Solid-State Circuits
IEEE Trans. Comput. Aid. D.	IEEE Transactions on Computer-Aided Design of Integrated Circuits
IEEE Trans. on Elec. Dev.	IEEE Transactions on Electron Devices
IEEE Trans. Magn.	IEEE Transactions on Magnetics
Int. J. Solid Struct.	International Journal of Solids and Structures
J. Am. Chem. Soc.	Journal of the American Chemical Society
J. Appl. Phys.	Journal of Applied Physics
J. Biomater. Sci., Plym. Ed.	Journal of Biomaterial Science, Polymer Edition
J. Chem. Phys.	The Journal of Chemical Physics
J. Chem. Soc. Faraday II	Journal of the Chemical Society, Faraday Transactions 2
J. Comput. Chem.	Journal of Computational Chemistry
J. Comp. Phys.	Journal of Computational Physics
J. Comput. Theor. Nanos.	Journal of Computational and Theoretical Nanoscience

J. Mol. Biol.	Journal of Molecular Biology
J. Phys. Chem. B	The Journal of Physical Chemistry B
J. Phys. Chem. C	The Journal of Physical Chemistry C
J. of Coll. and Inter. Science	Journal of Colloid and Interface Science
Meth. Enzymol.	Methods in Enzymology
Numer. Math.	Numerische Mathematik
Opt. Express	Optics Express
Phys. Rev. A	Physical Review A
Phys. Rev. B	Physical Review B
Phys. Rev. E	Physical Review E
Phys. Rev. Lett.	Physical Review Letters
Proc. R. Soc. A	Proceedings of the Royal Society A: Mathematical, Physical & Engineering Sciences
Protein Enf. Des. Sel.	Protein Engineering, Design and Selection
Proteins: Struct., Funct., Bioinf.	Proteins: Structure, Function, and Bioinformatics
P. Natl. Acad. Sci. USA	Proceedings of the National Academy of Sciences of the United States of America
Rep. Prog. Phys.	Reports on Progress in Physics
SIAM J. Appl. Math.	SIAM Journal on Applied Mathematics
SIAM J. Sci. Comput.	SIAM Journal on Scientific Computing
Z. Elektrochem.	Zeitschrift für Elektrochemie

References

Altman, M. D., Bardhan, J. P., Tidor, B., and White, J. K. (2006). FFTSVD: A fast multiscale boundary-element method solver suitable for BioMEMS and biomolecule simulation. *IEEE Trans. Comput. Aid. D.*, 25:274–284.

Altman, M. D., Bardhan, J. P., White, J. K., and Tidor, B. (2005). An accurate surface formulation for biomolecule electrostatics in non-ionic solutions. In *Proceedings of the 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2005).*

Altman, M. D., Bardhan, J. P., White, J. K., and Tidor, B. (2009). Accurate solution of multi-region continuum electrostatic problems using the linearized Poisson–Boltzmann equation and curved boundary elements. *J. Comput. Chem.*, 30:132–153.

Antosiewicz, T. J., Apell, S. P., Claudio, V., and Käll, M. (2011). A simple model for the resonance shift of localized plasmons due to dielectric particle adhesion. *Opt. Express*, 20(1):524–533.

Asthagiri, D. and Lenhoff, A. M. (1997). Influence of structural details in modeling electrostatically driven protein adsorption. *Langmuir*, 13:6761–6768.

Atkinson, K. E. (1997). *The Numerical Solution of Integral Equations of the Second Kind.* Cambridge Monographs on Applied and Computational Mechanics.

Baio, J. E., Weidner, T., Baugh, L., Gamble, L. J., Stayton, P. S., and Castner, D. G. (2012). Probing the orientation of electrostatically immobilized protein g b1 by time-of-flight secondary ion spectrometry, sum frequency generation, and near-edge x-ray adsorption fine structure spectroscopy. *Langmuir*, 28:2107–2112.

Bajaj, C., Chen, S. C., and Rand, A. (2011). An efficient higher-order fast multipol boundary element solution for poisson-boltzmann-based molecular electrostatics. *SIAM Journal of Scientific Computing*, 33(2).

Baker, N. A. (2004). Poisson–Boltzmann methods for biomolecular electrostatics. *Meth. Enzymol.*, 383:94–118.

Baker, N. A., Sept, D., Holst, M. J., and McCammon, J. A. (2001). Electrostatics of nanoysystems: Application to microtubules and the ribosome. *P. Natl. Acad. Sci. USA*, 98:10037–10041.

Bardhan, J. P. (2012). Biomolecular electrostatics — I want your solvation (model). *Comput. Sci. Disc.*, 5(013001).

Barnes, J. and Hut, P. (1986). A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324:446–449.

Bashford, D. (1997). An object-oriented programming suite for electrostatic effects in biological molecules. an experience report on the MEAD project. In Ishikawa, Y., Oldehoeft, R., Reynders, J., and Tholburn, M., editors, *Scientific Computing in Object-Oriented Parallel Environments*, volume 1343 of *Lecture Notes in Computational Science*, pages 233–240. Springer Berlin Heidelberg.

Bashford, D. and Gerwert, K. (1992). Electrostatic calculations of the pK_a values of ionizable groups in bacteriorhodopsin. *J. Mol. Biol.*, 224(2):473–486.

Baumgartner, S. and Heitzinger, C. (2012). Existence and local uniqueness for 3d self-consistent multiscale models of field-effect sensors. *Commun. Math. Sci.*, 10(2):693–716.

Baumgartner, S. and Heitzinger, C. (2013). A one-level FETI method for the drift diffusion-poisson system with discontibuities at an interface. *J. Comp. Phys.*, 243:74–86.

Bebendorf, M. (2000). Approximation of boundary element matrices. *Numer. Math.*, 86(4):565–589.

Bebendorf, M. (2005). Hierarchical LU decomposition-based preconditioners for BEM. *Computing*, 74:225–247.

Bebendorf, M. and Rjasanow, S. (2003). Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1):1–24.

Betti, E. (1872). Theoria dell' elasticita. *Il Nuovo Cimento*, 7-8(1):69–97.

Biesheuvel, P. M., van der Veen, M., and Norde, W. (2005). A modified poisson-boltzmann model including charge regulation for the adsorption of ionizable polyelectrolytes to charged interfaces, applied to lysozyme adsorption on silica. *J. Phys. Chem. B*, 109:4172–4180.

Bohren, C. F. and Gilra, D. P. (1979). Extinction by a spherical particle in an absorbing medium. *Journal of Colloid and Interface Science*, 72(2):215–221.

Bohren, C. F. and Huffman, D. R. (1983). *Absorption and scattering of light by small particles.* John Wiley & Sons.

Bonthuis, D. J. and Netz, O. (2013). Beyond the continuum: How molecular solvent structure affects electrostatics and hydrodynamics at solid-electrolyte interfaces. *J. Phys. Chem. B*, 117:11397–11413.

Brebbia, C. and Dominguez, I. (1992). *Boundary Elements An Introductory Course*. WIT Press, 2nd edition.

Brinkman, W. F., Haggan, D. E., and Troutman, W. W. (1997). A history of the invention of the transistor and where it will lead us. *IEEE J. Solid-State Circuits*, 32(12):1858–1865.

Byun, J.-Y., Shin, Y.-B., Li, T., Park, J.-H., Kim, D.-M., Choi, D.-H., and Kim, M.-G. (2013). The use of an engineered single chain variable fragment in a localized surface plasmon resonance method for analysis of the c-reactive protein. *Chem. Commun.*, 49(82):9497–9499.

Carnie, S. L. and Chan, D. Y. (1993). Interaction free energy between identical spherical colloidal particles: The linearized poisson-boltzmann theory. *J. of Coll. and Inter. Science*, 155:297–312.

Carnie, S. L., Chan, D. Y., and Gunning, J. S. (1994). Electrical double layer interaction between dissimilar spherical colloidal particles and betwen a sphere and a plate: The linearized poisson-boltzmann theory. *Langmuir*, 10:2993–3009.

Chan, D. Y. and Mitchell, D. J. (1983). The free energy of an electrical double layer. *J. of Coll. and Inter. Science*, 95(1):193–197.

Chen, S., Liu, L., Zhou, J., and Jiang, S. (2003). Controlling antibody orientation on charged self assembled-monolayers. *Langmuir*, 19:2859–2864.

Chou, A., Vernon, K. C., Piro, L., Radi, B., Jaatinen, E. A., and Davis, T. J. (2012). Predicting the localized surface plasmon resonances of spherical nanoparticles on a substrate: Electrostatic eigenmode method. *J. Phys. Chem. C*, 116:26517–26522.

Chỳlek, P. (1977). Light scattering by small particles in an absorbing medium. *JOSA*, 67(4):561–563.

Cooper, C. D. and Barba, L. A. (2013). Validation of the `PyGBe` code for Poisson-Boltzmann equation with boundary element methods. Technical Report on **figshare**, CC-BY license, doi:10.6084/m9.figshare.154331.

Cooper, C. D., Bardhan, J. P., and Barba, L. A. (2013a). Binding energy of peptide-rna complex with pygbe and apbs. Data, figures and plottings script on **figshare**, CC-BY license, http://dx.doi.org/10.6084/m9.figshare. 799704.

Cooper, C. D., Bardhan, J. P., and Barba, L. A. (2013b). Binding energy of trypsin-bpti complex with pygbe and apbs. Data, figures and plottings script on **figshare**, CC-BY license, http://dx.doi.org/10.6084/m9.figshare. 799703.

Cooper, C. D., Bardhan, J. P., and Barba, L. A. (2013c). A biomolecular electrostatics solver using python, GPUs and boundary elements that can handle solvent-filled cavities and Stern layers. *Comput. Phys. Commun.* Published online. Preprint on arXiv:/1309.4018.

Cooper, C. D., Bardhan, J. P., and Barba, L. A. (2013d). Convergence and time to solution of pygbe with lysozyme molecule. Data, figures and plottings script on **figshare**, CC-BY license, http://dx.doi.org/10.6084/m9.figshare.799702.

Cruse, T. (1969). Numerical solutions in three dimensional elastostatics. *Int. J. Solids Struct.*, 5:1259–1274.

Davis, T. J., Gomez, D. E., and Vernon, K. C. (2010). Interaction of molecules with localized surface plasmons in metallic nanoparticles. *Phys. Rev. B*, 81(4).

Davis, T. J., Vernon, K. C., and Gómez, D. E. (2009a). Designing plasmonic systems: applications to dark modes in nanoparticle pairs and triplets. In Stockman, M. I., editor, *Proceedings of SPIE 7394. Plasmonics: Metallic Nanostructures and Their Optical Properties VII.*

Davis, T. J., Vernon, K. C., and Gomez, D. E. (2009b). Designing plasmonic systems using optical coupling between nanoparticles. *Phys. Rev. B*, 79(15):155423.

Davis, T. J., Vernon, K. C., and Gómez, D. E. (2009c). Effect of retardation on localized surface plasmon resonances in a metallic nanorod. *Opt. Express*, 17(26):23655–23663.

de Abajo, F. J. G. and Howie, A. (2002). Retarded field calculation of electron energy loss in inhommogeneous dielectrics. *Phys. Rev. B*, 65(11):115418.

Dolinsky, T. J., Nielsen, J. E., McCammon, J. A., and Baker, N. A. (2004). PDB2PQR: an automated pipeline for the setup of Poisson–Boltzmann electrostatics calculations. *Nucleic Acids Research*, 32:W665–W667.

Fredholm, I. (1903). Sur une classe d'équations functionnelles. *Acta Mathematica*, 27(1):365–390.

Fredkin, D. R. and Mayergoyz, I. D. (2003). Resonant behavior of dielectric objects (electrostatic resonances). *Phys. Rev. Lett.*, 91(25):253902(1) – 253902(4).

Freed, A. S. and Cramer, S. M. (2011). Protein-surface interaction maps for ion-exchange chromatography. *Langmuir*, 27:3561–3568.

García de Abajo, F. J. and Aizpurua, J. (1997). Numerical simulation of electron energy loss near inhomogeneous dielectrics. *Phys. Rev. B*, 56(24):15873(12).

García de Abajo, F. J. and Howie, A. (1997). Relativistic electron energy loss and electron-induced photon emission in inhomogenoeous dielectrics. *Phys. Rev. Lett.*, 80(23):5180(4).

García-García, C. and Draper, D. E. (2003). Electrostatic interactions in peptide-rna complex. *J. Mol. Biol.*, 331(1):75–88.

Geng, W. H. and Krasny, R. (2013). A treecode-accelerated boundary integral poisson-boltzmann solver for solvated biomolecules. *J. Comp. Phys.*, 247:62–78.

Geng, W. H., Yu, S. N., and Wei, G. W. (2007). Treatment of charge singularities in implicit solvent models. *J. Chem. Phys.*, 127:114106.

George, J., Compton, J. R., Leary, D. H., Olson, M. A., and Legler, P. M. (2014). Structural and mutational analysis of a monomeric and dimeric form of a single domain antibody with implications for protein misfolding. *Proteins: Struct., Funct., Bioinf.*, 82(11):3101–3116.

Gilson, M. K. and Honig, B. H. (1986). The dielectric constant of a folded protein. *Biopolymers*, 25(11):2097–2119.

Gilson, M. K. and Honig, B. H. (1987). Calculation of electrostatic potentials in an enzyme active site. *Nature*, 330(6143):84–86.

Gilson, M. K., Rashin, A., Fine, R., and Honig, B. (1985). On the calculation of electrostatic interactions in proteins. *J. Mol. Biol.*, 184:503–516.

Gilson, M. K., Sharp, K. A., and Honig, B. (1987). Calculating the electrostatic potential of molecules in solution: method and error assessment. *J. Comput. Chem.*, 9(4):327–335.

Gray, J. J. (2004). The interaction of proteins with solid surfaces. *Curr. Opin. Struct. Biol.*, 14:110–115.

Green, G. (1828). An essay on the application of mathematical analysis to the theories of electricity and magnetism. Nottingham.

Greengard, L., Huang, J., and Wandzura, S. (1998). Acclelerating fast multipole methods for the helmholtz equation at low frequencies. *IEEE Computational Science and Engineering*, 5(3):32–38.

Greengard, L. and Rokhlin, V. (1987). A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348.

Greengard, L. F. and Huang, J. (2002). A new version of the fast multipole method for screened Coulomb interactions in three dimensions. *J. Comp. Phys.*, 180(2):642–658.

Griffiths, D. J. (1999). *Introduction to Electrodynamics.* Benjamin Cummings, 3rd edition.

Guex, N. and Peitsch, M. C. (1997). SWISS-MODEL and the Swiss-PdbViewer: An envoronment for comparative protein modeling. *Electrophoresis*, 18:2714–2723.

Haes, A. J. and Van Duyne, R. P. (2002). A nanoscale optical biosensor: Sensitivity and selectivity of an approach based on the localized surface plasmon resonance spectroscopy of triangular silver nanoparticles. *J. Am. Chem. Soc.*, 124(35):10596–10604.

Haes, A. J., Zou, S., Schantz, G. C., and Van Duyne, R. P. (2004). A nanoscale optical biosensor: The long range distance dependence of the localized surface plasmon resonance of noble metal nanoparticles. *J. Phys. Chem B*, 108:109–116.

Hale, G. M. and Querry, M. R. (1972). Optical constants of water in the 200-nm to 200-μm wavelength region. *Appl. Opt.*, 12(3):555–563.

Hartvig, R. A., van de Weert, M., Ostergaard, J., Jorgensen, L., and Jensen, H. (2011). Protein adsorption at charged surfaces: The role of electrostatic interactions and interfacial charge regulation. *Langmuir*, 27:2634–2643.

Heitzinger, C., Liu, Y., Mauser, N. J., Ringhofer, C., and Dutton, R. W. (2010a). Calculation of fluctuations in boundar layers of nanowire field-effect biosensors. *J. Comput. Theor. Nanos.*, 7:1–7.

Heitzinger, C., Mauser, N. J., and Ringhofer, C. (2010b). Multiscale modeling of planar and nanowire field-effect biosensors. *SIAM J. Appl. Math.*, 70(5):1634–1654.

Hill, T. L. (1960). *An introduction to statistical thermodynamics.* Addison-Wesley Publishing Company, 1st edition.

Ho, K. L. and Greengard, L. (2012). A fast direct solver for structured linear systems by recursive skeletonization. *SIAM J. Sci. Comput.*, 34(5):A2507–A2532.

Hohenester, U. and Trugler, A. (2012). MNPBEM — A Matlab toolbox for the simylation of plasmonic nanoparticles. *Comput. Phys. Commun.*, 183:370–381.

Hutter, E. and Fendler, J. H. (2004). Exploitation of localized surface plasmon resonance. *Adv. Mater.*, 16(19):1685–1706.

Jackson, J. D. (1998). *Classical Electrodynamics.* Wiley, 3^{rd} edition.

Jawson, M. A. (1963). Integral equation methods in potential theory I. *Proc. R. Soc. A*, 275:23–32.

Johnson, P. B. and Christy, R. W. (1972). Optical constants of nobble metals. *Phys. Rev. B*, 12(6):4370–4379.

Juffer, A. H., Botta, E. F. F., van Keulen, B. A. M., van der Ploeg, A., and Berendsen, H. J. C. (1991). The electric potential of a macromolecule in a solvent: A fundamental approach. *J. Comp. Phys.*, 97(1):144–171.

Jung, J., Pedersen, T. G., Sondergaard, T., Pedersen, K., Larsen, A. N., and Nielsen, B. B. (2010). Electrostatic plasmon resonances of metal nanospheres in layered geometries. *Phys. Rev. B*, 81(12).

Jung, L. S., Campbell, C. T., Chinowsky, T. M., Mar, M. N., and Yee, S. S. (1998). Quantitative interpretation of the response of surface plasmon resonance sensors to adsorbed fields. *Langmuir*, 14:5636–5648.

Katsikadelis, J. T. (2002). *Boundary Elements Theory and Applications.* Elsevier Science.

Kirkwood, J. G. (1934). Theory of solutions of molecules containing widely separated charges with special application to zwitterions. *J. Chem. Phys.*, 2(7):351–361.

Klöckner, A., Pinto, N., Lee, Y., Catanzaro, B., Ivanov, P., and Fasih, A. (2012). PyCUDA and PyOpenCL: A scripting-based approach to GPU run-time code generation. *Parallel Computing*, 38(3):157–174.

Li, P., Johnston, H., and Krasny, R. (2009). A Cartesian treecode for screened Coulomb interactions. *J. Comp. Phys.*, 228:3858–3868.

Liang, J. and Subramaniam, S. (1997). Computation of molecular electrostatics with boundary element methods. *Biophys. J.*, 73:1830–1841.

Lindsay, K. and Krasny, R. (2001). A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow. *J. Comput. Phys.*, 172:879–907.

Liu, J., Liao, C., and Zhou, J. (2013). Multiscale simulations of protein G B1 adsorbed on charged self-assembled monolayers. *Langmuir*, 29:11366–11374.

Lu, B., Cheng, X., Huang, J., and McCammon, J. A. (2006). Order N algorithm for computation of electrostatic interactions in biomolecular systems. *P. Natl. Acad. Sci. USA*, 103(51):19314–19319.

Lu, B. Z., Zhou, Y. C., Holst, M. J., and McCammon, J. A. (2008). Recent progress in numerical methods for the poisson-boltzmann equation in biophysical applications. *Communications in Computational Physics*, 3(5):973–1009.

Lu, D., Lee, S. J., and Park, K. (1991). Calculation of solvation interaction energies for protein adsorption on polymer surfaces. *J. Biomater. Sci., Polym. Ed.*, 3(2):127–147.

Mäkitalo, J. and Kauranen, M. (2014). Modes and resonances of plasmonic scatterers. *Phys. Rev. B*, 89:165429.

Marcelja, S., Mitchell, D. J., Ninham, B. W., and Sculley, M. J. (1977). Role of solvent structure in solution theory. *J. Chem. Soc. Faraday II*, 73:630–648.

Mayergoyz, I. D., Fredkin, D. R., and Zhang, Z. (2005). Electrostatic (plasmon) resonances in nanoparticles. *Phys. Rev. B*, 72(15):155412.

Mayergoyz, I. D. and Zhang, Z. (2006). Numerical analysis of plasmon resonances in nanoparticles. *IEEE Trans. Magn.*, 42(4):759–762.

Mayergoyz, I. D. and Zhang, Z. (2007). The comptation of extinction cross sections of resonant metallic nanoparticles subject to optical radiation. *IEEE Trans. Magn.*, 43(4):1681–1684.

McLean, W. (2000). *Strongly Elliptic Systems and Boundary Integral Equations.* Cambridge University Press.

Mie, G. (1908). Beiträge zur optik trüber medien, speziell kolloidaler metallösungen. *Annalen der Physik*, 330(3):377–445.

Mishchenko, M. I. (2007). Electromagnetic scattering by a fixed finite object embedded in an absorbing medium. *Opt. Express*, 20(15):13188–13202.

Mortazavi, D., Kouzani, A. Z., and Kaynak, A. (2012). Investigating nanoparticle-substrate interaction in lspr biosensing using image-charge theory. In *34th Annual International Conference of the IEEE EMBS.*

Nair, P. R. and Alam, M. A. (2007). Design considerations of silicon nanowire biosensors. *IEEE Trans. on Elec. Dev.*, 54(12):3400–3408.

Orozco, M. and Luque, F. J. (2000). Theoretical methods for the description of the solvent effect in biomolecular systems. *Chem. Rev.*, 100:4187–4225.

Palacios, T. (2012). Nanowire electronics comes of age. *Nature*, 481:152–153.

Paliwal, A., Tomar, M., and Gupta, V. (2014). Complex dielectric constant of various biomolecules as a function of wavelength using surface plasmon resonance. *J. Appl. Phys.*, 116:023109.

Patolsky, F., Zheng, G., and Lieber, C. M. (2006). Nanowire sensors for medicine and life sciences. *Nanomedicine*, 1(1):51–65.

Petryayeva, E. and Krull, U. J. (2011). Localized surface plasmon resonance: Nanostructures, bioassays and biosensing — a review. *Anal. Chim. Acta*, 706:8–24.

Phillips, J. R. and White, J. K. (1997). A precorrected-FFT method for electrostatic analysis of complicated 3-D structures. *IEEE Trans. Comput. Aid. D.*, 16(10):1059–1072.

Piliarik, M., Sipova, H., Kvasnicka, P., Galler, N., Frenn, J. R., and Homola, J. (2011). High-resolution biosensor based on localized surface plasmons. *Opt. Express*, 20(1):672–680.

Rabe, M., Verdes, D., and Seeger, S. (2011). Understanding protein adsorption phenomena at solid surfaces. *Adv. Colloid Interface Sci.*, 162:87–106.

Raphael, M. P., Christodoulides, J. A., Delehanty, J. B., Long, J. P., Pehrsson, P. E., and Byers, J. M. (2013). Quantitative LSPR imaging for biosensing with single nanostructure resolution. *Biophys. J.*, 104:30–36.

Rizzo, F. J. (1967). An integral equation approach to baundary value problems of classical elastostatics. *Quarterly of Applied Mathematics*, 25:83–95.

Roth, C. M. and Lenhoff, A. M. (1993). Electrostatic and van der Waals contributions to protein adsorption: computation of equilibrium constants. *Langmuir*, 9:962–972.

Roth, C. M., Neal, B. L., and Lenhoff, A. M. (1996). Van del Waals interactions involving proteins. *Biophys. J.*, 70:977–987.

Roux, B. and Simonson, T. (1999). Implicit solvent models. *Biophys. Chem.*, 78:1–20.

Sanner, M. F., Olson, A. J., and Spehner, J.-C. (1995). Fast and robust computation of molecular surfaces. In *Proceedings of the eleventh annual symposium on Computational geometry*, pages 406–407. ACM.

Sharp, K. A. and Honig, B. (1990). Electrostatic interactions in macromolecules: Theory and applications. *Ann. Rev. Biophys. Biophys. Chem.*, 19:301–332.

Shaw, P. B. (1985). Theory of the Poisson Green's-function for discontinuous dielectric media with an application to protein biophysics. *Phys. Rev. A*, 32(4):2476–2487.

Sheng, Y.-J., Tsao, H.-K., Zhou, J., and Jiang, S. (2002). Orientation of a Y-shaped biomolecule adsorbed on a charged surface. *Phys. Rev. E*, 66(1):011911.

Simonson, T. (2003). Electrostatics and dynamics of proteins. *Rep. Prog. Phys.*, 66:737–787.

Solis, D. M., Taboada, J. M., Obelleiro, F., Liz-Marzán, L. M., and García de Abajo, F. J. (2014). Toward ultimate nanoplasmonics modeling. *ACS Nano*, 8(8):7559–7570.

Somigliana, C. (1885). Sopra l' equilibrio di' un corpo elastico isotropo. *Il Nuovo Cimento*, 3:17–20.

Steinbach, O. (2008). *Numerical Approximation Methods for Elliptic Boundary Value Problems. Finite and Boundary Elements.* Springer.

Stern, O. (1924). Zur theorie der elektrolytischen doppelschicht. *Z. Elektrochem.*, 30:508–516.

Still, W., Tempczyk, A., Hawley, R. C., and Hendrickson, T. F. (1990). Semianalytical treatment of solvation for molecular mechanics and dynamics. *J. Am. Chem. Soc.*, 112(16):6127–6129.

Symm, G. T. (1963). Integral equation methods in potential theory II. *Proc. R. Soc. A*, 275(33-46).

Tajima, N., Takai, M., and Ishihara, K. (2011). Significance of antibody orientation unraveled: Well-oriented antibodies recorded high binding affinity. *Anal. Chem.*, 83:1969–1976.

Tomasi, J., Menucci, B., and Cammi, R. (2005). Quantum mechanical continuum solvation models. *Chem. Rev.*, 105:2999–3093.

Trilling, A. K., Beekwilder, J., and Zuilhof, H. (2013a). Antibody orientation on biosensor surfaces: a minireview. *Analyst*, 138:1619–1627.

Trilling, A. K., Harmsen, M. M., Ruigrok, V. J. B., Zuilhof, H., and Beekwilder, J. (2013b). The effect of uniform capture molecule orientation on biosensor sensitivity: Dependence on analyte properties. *Biosens. Bioelectron.*, 40:219–226.

Trilling, A. K., Hesselink, T., van Houwelingen, A., Cordewener, J. H. G., Jongsma, M. A., Schoffelen, S., van Hest, J. C. M., Zuilhof, H., and Beekwilder, J. (2014). Orientation of llama antibodies strongly increases sensitivity of biosensors. *Biosen. Bioelectron.*, 60:130–136.

Turner, K. B., Zabetakis, D., Goldman, E. R., and Anderson, G. P. (2014). Enhanced stabilization of a stable single domain antibody for seb toxin by random mutagenesis and stringent selection. *Protein Eng. Des. Sel.*, 27(3):89–95.

Videen, G. and Sun, W. (2003). Yet another look at light scattering from particles in absorbing media. *Applied optics*, 42(33):6724–6727.

Wagoner, J. A. and Baker, N. A. (2006). Assesing implicit models for nonpolar mean solvation forces: The importance of dispersion and voume terms. *P. Natl. Acad. Sci. USA*, 103(22):8331–8336.

Warwicker, J. and Watson, H. C. (1982). Calculation of the electric potential in the active site cleft due to alpha-helix dipoles. *J. Mol. Biol.*, 157:671–679.

Willets, K. A. and Van Dyune, R. P. (2007). Localized surface plasmon resonance spectroscopy and sensing. *Annu. Rev. Phys. Chem.*, 58:267–297.

Wimalasena, R. L. and Wilson, G. S. (1991). Factors affecting the specific activity of immobilized antibodies and their biologically active fragments. *Journal of Chromatography B: Biomedical Sciences and Applications*, 572(1-2):85–102.

Xu, Z. and Cai, W. (2011). Fast analytical methods for macroscopic electrostatic models in biomolecular simulations. *SIAM Review*, 53(4):683–720.

Yao, Y. and Lenhoff, A. M. (2004). Electrostatic contributions to protein retention in ion-exchange chromatography. 1. cytochrome c variants. *Anal. Chem.*, 76:6743–6752.

Yao, Y. and Lenhoff, A. M. (2005). Electrostatic contributions to protein retention in ion-exchange chromatography. 2. proteins with vaious degrees of structural differences. *Anal. Chem.*, 77:2157–2165.

Yokota, R., Bardhan, J. P., Knepley, M. G., Barba, L. A., and Hamada, T. (2011). Biomolecular electrostatics using a fast multipole BEM on up to 512 GPUs and a billion unknowns. *Comput. Phys. Comm.*, 182(6):1271–1283.

Yoon, B. J. and Lenhoff, A. M. (1990). A boundary element method for molecular electrostatics with electrolyte effects. *J. Comput. Chem.*, 11(9):1080–1086.

Zhang, Z., Mayergoyz, I. D., Gumerov, N. A., and Duraiswami, R. (2007). Numerical analysis of plasmon resonances in nanoparticles based on fast multipole method. *IEEE Trans. Magn.*, 43(4):1465–1468.

Zhou, H. X. (1993). Boundary-element solution of macromolecular electrostatics—interaction energy between 2 proteins. *Biophys. J.*, 65:955–963.

Zhou, J., Chen, S., and Jiang, S. (2003). Orientation of adsorbed antibodies on charged surfaces by computer simulation based on a united-residue model. *Langmuir*, 19:3472–3478.

Zhou, J., Tsao, H.-K., Sheng, Y.-J., and Jiang, S. (2004). Monte Carlo simulations of antibody adsorption and orientation on charged surfaces. *J. Chem. Phys.*, 121:1050–1057.

Zhu, Z., Huang, J., Song, B., and White, J. (2001). Improving the robustness of a surface integral formulation for wideband impedance extraction of 3D structures. In *Proceedings of the 2001 IEEE/ACM Int. Conf. on Computer-Aided Design*, pages 592–597.

Curriculum Vitae

Personal information

Full name

Christopher David Cooper Villagrán

Place of birth

Viña del Mar, Chile (1984)

Nationality

Chilean/Spanish

Contact details

Address

Department of Mechanical Engineering, Boston University.
110 Cummington Street
Boston, MA 02215

Phone number

(617) 480 - 4879
(617) 353 - 2814

Email address

cdcooper@bu.edu
christophercooperv@gmail.com

Personal website

https://sites.google.com/site/cdcooperbu/

Education

2010 - present

PhD. Mechanical Engineering (Expected 2014). Boston University, Boston, MA.

2010 - 2012

MSc. Mechanical Engineering. Boston University, Boston, MA.

2003 - 2009

Professional Mechanical Engineer. Universidad Técnica Federico Santa María, Valparaíso, Chile.

2003 - 2007

BSc. Mechanical Engineering. Universidad Técnica Federico Santa María, Valparaíso, Chile.

Experience

Research

2011-2014

Research assistant. Boston University, Boston, MA.
Worked on Boundary Element Methods for continuum electrostatics in biomolecular systems using cartesian treecodes and GPUs.
Supervisor: Prof. Lorena Barba.

2010

Research assistant for summer term. Boston University, Boston, MA.
Worked on meshfree methods for PDEs and studied basics of CUDA programming.
Supervisor: Prof. Lorena Barba.

2008

Nine month research exchange program funded by SCAT-ALFA project (`scat-alfa.eu`). University of Bristol, Bristol, UK.
Worked on vortex methods for Computational Fluid Dynamics and Radial Basis Functions for Partial Differential Equations.
Supervisor: Prof. Lorena Barba.

2007

Research assistant for FONDEF D05110098 project (Dependent of the chilean government). Universidad Técnica Federico Santa María, Valparaíso, Chile. Worked on modeling the heat transfer to certain fluid in the copper mining process.
Supervisor: Prof. Fernando Carvajal.

Teaching

Fall 2014

Teaching assistant for Practical Numerical Methods in Python (MOOC developed at GWU)
Department of Mechanical and Aerospace Engineering, George Washington University, Washington, DC.

Spring/Summer 2014

Graduate teaching assistant for Mechanics I course.
Mechanical Engineering Department, Boston University, Boston, MA.

August 2011

Designed and taught a one month course on GPU computing at Universidad Técnica Federico Santa María, Valparaíso, Chile.

Summer 2011

Graduate teaching assistant for Mechanics I course.
Mechanical Engineering Department, Boston University, Boston, MA.

2010

Graduate teaching assistant for Fluid Mechanics course.
Mechanical Engineering Department, Boston University, Boston, MA.

2006 - 2007 and 2009

Teaching assistant for General Fluid Mechanics and Fluid Mechanics II courses.
Mechanical Engineering Department, Universidad Técnica Federico Santa María, Valparaíso, Chile.

2004

Teaching assistant for Mathematics I course.
Mathematics Department, Universidad Técnica Federico Santa María, Valparaíso, Chile.

Publications

Journal publications

C. D. Cooper, J. P. Bardhan, L. A. Barba. A biomolecular electrostatics solver using Python, GPUs and boundary elements that can handle solvent-filled cavities and Stern layers. *Comput. Phys. Commun.* (2014)

Conference proceedings

C. D. Cooper and L. A. Barba. Implicit solvent model using Python and GPUs for proteins interacting with charged surfaces. *XXVI IUPAP Conference on Computational Physics, CCP 2014.* Boston, MA (August, 2014).

C. D. Cooper, J. P. Bardhan and L. A. Barba. Efficient boundary element methods for molecular electrostatics using Python and GPUs. *SIAM Conference on Computational Science and Engineering.* Boston, MA (February, 2013).

C. D. Cooper and L. A. Barba. Fast tree methods and GPUs for protein electrostatics. *Symposium of the International Association for Boundary Element Method, IABEM 2013.* Santiago, Chile (January, 2013).

C. D. Cooper and L. A. Barba. Efficient Boundary Element Methods in Python with GPUs. *10th World Congress on Computational Mechanics.* Sao Paulo, Brazil (July, 2012).

F. A. Cruz, C. D. Cooper, R. Yokota, L. A. Barba. Parallel Implementation of the Panel-free Boundary Conditions for the Viscous Vortex Method. *21st International Conference on Parallel Computational Fluid Dynamics.* Moffet Field, CA (2009).

C. D. Cooper, L. A. Barba. Panel-free Boundary Conditions for the Viscous Vortex Method. *19th AIAA Computational Fluid Dynamics Conference.* San Antonio, TX (2009).

Workshops

International Summer School on HPC Challenges in Computational Sciences. Organized by Xsede, Prace and Riken. New York, NY (July, 2013).

Pan-American Advanced Studies Institute: Scientific computing in the Americas, the challenges of massively parallel. Organized by L. Barba with NSF funding. Valparaíso, Chile (January, 2011).