Mathematical methods for implicit solvation models in quantum chemistry
Chaoyu Quan

To cite this version:

HAL Id: tel-01919793
https://theses.hal.science/tel-01919793
Submitted on 12 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Thèse de Doctorat de
l’Université Pierre et Marie Curie

Présentée et soutenue publiquement le 21 Novembre 2017

pour l’obtention du grade de

Docteur de l’Université Pierre et Marie Curie
Spécialité : Mathématiques Appliquées

par

Chaoyu QUAN
sous la direction de

Yvon MADAY et Benjamin STAMM

Mathematical Methods for Implicit Solvation Models in Quantum Chemistry

après avis des rapporteurs

M. Martin J. GANDER & M. Aihui ZHOU

devant le jury composé de

M. Eric CANCES Examinateur
M. Pascal FREY Examinateur
M. Martin J. GANDER Rapporteur
Mme. Laura GRIGORI Examinateur
M. Yvon MADAY Directeur de thèse
M. Jean-Philip PIQUEMAL Examinateur
M. Benjamin STAMMM Directeur de thèse
M. Aihui ZHOU Rapporteur
Chaoyu QUAN :


Adresse électronique: quan@ann.jussieu.fr, quanchaoyu@gmail.com
Remerciements

First and foremost I would like to thank my advisors Yvon Maday and Benjamin Stamm. It has been a great honor to be their Ph.D. student. The first time that I met Yvon was in his course on the variational discretization of the elliptic partial differential equations, which was attractive and inspired me to start my researches with him. Obviously, this is one of the best decisions that I have ever made in my life. Yvon is very kind, experienced in academic researches, full of new ideas and good at seeking funding etc. To me, he himself is the best model of what a good researcher should be, which is important to encourage a beginner in the academic world like me. Benjamin is both an advisor and a friend to me. As the advisor, he has taught me many things associated with researches, such as how to think over questions in a strict way, how to use Matlab efficiently, how to write a professional article etc. As a friend, he cares about my life in Paris, including the visa, the housing, the food etc. I feel really happy to be his first Ph.D. student and appreciate all his contributions of time, ideas and patience to me.

Then, I want to thank Pascal Frey and Eric Cancès for their fruitful discussions with us on the molecular surfaces and the implicit solvation models. Pascal has provided me with some good advices on meshing the molecular surfaces. Furthermore, I do appreciate his efforts on the funding of my postdoc in the CalSimLab, Université Pierre et Marie Curie. Eric has discussed with us about the implicit solvation models. In fact, the second part of my thesis is based on his work on the integral equation formulation of the polarizable continuum models and on the domain “spherical” decomposition method for the conductor-like screening models. Moreover, I would thank Eric for the recommendation letter for my postdoc applications.

In regards to the academic discussions, I also thank Filippo Lipparini, Louis La-
gardère, Benedetta Mennucci and Jean-Philip Piquemal. They have helped me to know more about the quantum chemistry.

Besides, I want to thank the two reviewers of my thesis for their positive opinions. In particular, thank Martin J. Gander for changing his schedule in order to join my defense in Paris. Thank Aihui Zhou for taking a long journey from Beijing to Paris and for spending much time on the visa.

In my lab, Laboratoire Jacques-Louis Lions, there are many persons that I would like to thank for their help as well as their accompany, including Carlo Marcati, Geneviève Dusson, Étienne Polack, Amaury Hayat, Rim El Dbaissy, Gabriela Lopez Ruiz, Philippe Ung, Florian Omnes, Shuyang Xiang, Long Hu, Jiamin Zhu, Yashan Xu, Can Zhang, Haisen Zhang, Chen-Yu Chiang, Helin Gong, Yangyang Cao, Shijie Dong, Hongjun Ji, Yuqing Wu etc. In addition, I want to thank my good friends Wen Sun, Xianglong Duan, Qilong Weng, Bingxiao Liu and Zicheng Qian who do mathematical researches in Paris and come from the same Chinese university as me, University of Science and Technology of China. We have spent a pleasant time together in Paris.

Finally, I would like to thank my wife and my parents who always support me to do researches, and I also want to express my best wishes to my upcoming baby who motivates me to work hard.
Contents

Introduction

1 The larger context ...................................................... 9
2 Implicit solvation models ............................................ 11
3 Domain decomposition methods for implicit solvation models .... 20

I Molecular Surfaces ..................................................... 31

1 Mathematical analysis and calculation of molecular surfaces .... 33

1.1 Introduction .......................................................... 34
1.2 Introduction to implicit surfaces .................................... 36
1.3 Implicit molecular surfaces ......................................... 37
1.4 Solvent accessible surface .......................................... 39
1.5 Solvent excluded surface ........................................... 50
1.6 Construction of molecular surfaces ............................... 59
1.7 Numerical results .................................................. 63
1.8 Conclusion ............................................................ 67

2 Meshing molecular surfaces based on analytical implicit representa-
tion ............................................................................ 69

2.1 Introduction .......................................................... 70
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2</td>
<td>Molecular surfaces</td>
<td>72</td>
</tr>
<tr>
<td>2.3</td>
<td>Construction of molecular surfaces</td>
<td>77</td>
</tr>
<tr>
<td>2.4</td>
<td>Molecular inner holes</td>
<td>80</td>
</tr>
<tr>
<td>2.5</td>
<td>Meshing</td>
<td>82</td>
</tr>
<tr>
<td>2.6</td>
<td>Conclusion</td>
<td>92</td>
</tr>
</tbody>
</table>

## II Domain Decomposition Method for Implicit Solvation Models

### 3 Domain Decomposition Method for the Polarizable Continuum Model based on the Solvent Excluded Surface

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>98</td>
</tr>
<tr>
<td>3.2</td>
<td>Solute-solvent boundary</td>
<td>103</td>
</tr>
<tr>
<td>3.3</td>
<td>Dielectric permittivity function</td>
<td>105</td>
</tr>
<tr>
<td>3.4</td>
<td>Problem formulation and global strategy</td>
<td>107</td>
</tr>
<tr>
<td>3.5</td>
<td>Domain decomposition strategy</td>
<td>111</td>
</tr>
<tr>
<td>3.6</td>
<td>Single-domain solvers</td>
<td>112</td>
</tr>
<tr>
<td>3.7</td>
<td>Numerical results</td>
<td>119</td>
</tr>
<tr>
<td>3.8</td>
<td>Conclusion</td>
<td>130</td>
</tr>
</tbody>
</table>

### 4 Domain Decomposition Method for the Poisson-Boltzmann Solvation Model

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>132</td>
</tr>
<tr>
<td>4.2</td>
<td>PB solvation model</td>
<td>137</td>
</tr>
<tr>
<td>4.3</td>
<td>Problem transformation</td>
<td>139</td>
</tr>
<tr>
<td>4.4</td>
<td>Strategy</td>
<td>142</td>
</tr>
<tr>
<td>4.5</td>
<td>Single-domain solvers</td>
<td>145</td>
</tr>
</tbody>
</table>
Introduction

1 The larger context

Quantum chemistry aims at understanding the properties (such as spectroscopic observables, equilibrium geometry of the ground state or reactivity) of matter through the modeling of its behavior at a molecular scale \[23, 24\], where matter is described as an assembly of nuclei and electrons. This problem is known as the many-body problem and its solution, the wave function \( \Psi \), is described by the Schrödinger equation in its time-dependent

\[ i \frac{\partial}{\partial t} \Psi(t) = \mathcal{H} \Psi(t), \]

or time-independent form

\[ \mathcal{H} \Psi = \mathcal{E} \Psi, \]

where \( \mathcal{H} \) denotes the Hamiltonian of the molecular system under consideration and the constant \( \mathcal{E} \) is the energy of the stationary state \( \Psi \). The above two equations are very high dimensional differential equations whose solution can not even be approximated for a small molecule directly. To make the problem more tractable, the nuclei structure computation and the electronic structure computation are usually considered separately, as the mass of a nuclei is several magnitudes heavier than the one of an electron (known as the Born-Oppenheimer approximation).

In fact, most physical and chemical phenomena of interest in chemistry and biology take place in the liquid phase, and it is relevant and crucial to model the solvent in these processes. The typical situation is that a solute (biological protein for example) is surrounded by the solvent. To describe the solvent effects on the solute, two approaches are commonly-used. The first one is to use an explicit solvation model, in which the simulated chemical system is composed of the solute molecule and a large number of explicit solvent molecules. The second one is to use an implicit solvation model (or continuum solvation model), in which the solute molecule is embedded in a cavity surrounded by a continuous medium representing the solvent, i.e., the average response of the solvent molecules over the phase-space of the solvent molecules in the sense of statistical mechanics. Comparing to the explicit solvation model, the computation of implicit solvation model is usually much less expensive in computational time, see \[21, 109, 82\] for an overview of the implicit solvation model.
In the implicit solvation model, the solvent is usually treated as a polarizable continuum with a specific dielectric permittivity. Embedding the solute molecule in the continuous solvent, the solute’s charge distribution interacts with the continuous dielectric field and polarizes the surrounding medium, which in turn causes a change in the polarization on the solute. This defines the reaction potential, a response to the presence of the continuous environment. In quantum chemistry, where charge distributions come from ab initio methods, such as the Hartree-Fock (HF) electronic functionals or the Density Functional Theory (DFT), the implicit solvent models represent the solvent as a perturbation to the solute Hamiltonian in the following way [83, 109]:

$$\mathcal{H} = \mathcal{H}^M + \mathcal{H}^{MS},$$

(1.3)

where $H^M$ is the Hamiltonian of the solute molecule $M$ and $H^{MS}$ is the interaction between the solute $M$ and the solvent $S$.

Actually, $\mathcal{H}^{MS}$ is a sum of different interaction operators, each of which is related to an interaction with a different physical origin. In the standard implicit solvation model, four interaction operators are usually used to describe the solute-solvent interaction [31, 83]. This gives thus four contribution terms to the solvation energy. Supplemented by a fifth describing contribution due to thermal motions of the molecular framework, the solvation energy $G$ of $M$ is written in the following form

$$G = G_{cav} + G_{el} + G_{dis} + G_{rep} + G_{tm},$$

(1.4)

where the five terms on the right side represent respectively the cavitation, the electrostatic contribution, the dispersion, the repulsion and the thermal motion [83].

One main topic of this thesis is the computation of the electrostatic contribution $G_{el}$, a crucial issue in the calculation of solvation free energy, which involves solving a partial differential equation (PDE). To be precise, the electrostatic potential $\psi$ of an implicit solvation model is characterized as follows

$$-\nabla \cdot \varepsilon(x) \nabla \psi(x) = 4\pi \rho(x), \text{ in } \mathbb{R}^3,$$

(1.5)

where $\psi(x) \sim \frac{1}{|x|}$ as $|x| \to \infty$. Here, $\varepsilon(x)$ represents the space-dependent dielectric constant and $\rho(x)$ represents the charge distribution of the solvation system. The electrostatic contribution $G_{el}$ to the solvation energy, also denoted by $E^s$ in this thesis, is given by

$$G_{el} = E^s = \frac{1}{2} \int_{\mathbb{R}^3} \rho(x) \left( \psi(x) - \psi_0(x) \right) dx,$$

(1.6)

where

$$\psi_0(x) = \int_{\mathbb{R}^3} \frac{\rho(x')}{|x - x'|} dx'$$

(1.7)

is the electrostatic potential generated by $\rho$ in vacuo.
For the sake of simplicity, it is usually assumed that the solute’s charge distribution function $\rho_M$ (part of the whole charge distribution $\rho$) is supported in the solute cavity $\Omega$ and is presented by the sum of $M$ point charges in the form of

$$\rho_M(x) = \sum_{i=1}^{M} q_i \delta(x - x_i),$$

(1.8)

where $M$ is the number of solute atoms, $q_i$ represents the charge carried on the $i$th atom with center $x_i$, $\delta$ is the Dirac delta function. As a consequence, when the solvent does not contain any ion, we have $\rho = \rho_M$ and therefore, $\psi_0$ can be derived easily.

In this thesis, we will develop two Schwarz domain decomposition methods for solving the PDEs of the form (1.5) for two implicit solvation models. But before that, we first see some of our achievements on the characterization of the solute-solvent interface, especially the so-called “smooth” molecular surface (or the solvent excluded surface). The solute-solvent interface, which determines both the solute cavity and the solvent region, plays a fundamental role in an implicit solvation model.

For the sake of completeness, the following part of introduction might be repeated latter in Chapter 1–4. If the reader wants to get a general idea of this thesis as well as its contribution, it is helpful to read this introduction. If the reader is only interested in a particular part, each chapter can be read directly.

## 2 Implicit solvation models

In this section, we focus on introducing some basic notations of implicit solvation models. We first introduce the solute cavity determined by the solute-solvent boundary, which is a fundamental concept of an implicit solvation model. Then, we introduce different kinds of implicit solvation models, classified by different physical laws describing the solute-solvent interaction. Briefly speaking, an implicit solvation model consists of a suitable solute cavity and a specific physical law.

### 2.1 Solute-solvent boundary: solute cavities

In an implicit solvation model, the solute molecule is embedded in a cavity (the solute cavity), denoted by $\Omega$, surrounded by a continuous medium representing the solvent on a macroscopic scale. The definition of the solute cavity is not an intrinsic property of the solute molecule. As a consequence, determining a suitable solute-solvent boundary is important. It builds an interface between the solute and the solvent respectively between the atomistic and the continuum description of the physical model. Physically speaking, the solute cavity, i.e., the region enclosed by the solute-solvent boundary, occupies the space of the solute molecule where the sol-
vent molecules have no access. Therefore, a precise understanding and modeling of the nature of the solute-solvent boundary is essential. In fact, there are several well-established molecular surfaces that are usually chosen as the solute-solvent boundary: the van der Waals (VdW) surface, the Solvent Accessible Surface (SAS) and the Solvent Excluded Surface (SES).

2.1.1 Molecular surfaces

In the simplest model, atoms of a molecule are represented by VdW-balls with VdW-radii which are experimentally fitted, given the underlying chemical element, for example, the UFF radii [97]. As a consequence, the VdW-surface is defined as the topological boundary of the union of all VdW-balls.

In addition, the SAS and the SES were first introduced by Lee & Richards in the 1970s [67, 99], where the solvent molecules surrounding a solute molecule are reduced to spherical probes [109]. The SAS of a solute molecule is defined by rolling the center of an idealized spherical probe over the solute molecule, that is, the surface enclosing the region in which the center of a spherical probe can not enter. The SES is also called “the smooth molecular surface” or “the Connolly surface”, due to Connolly’s fundamental work [30]. It is defined by the same spherical probe rolling over the molecule, but now one considers the surface enclosing the region in which a spherical probe can not access. In other words, the SES is the boundary of the union of all possible probes that do not intersect the VdW-balls of the solute molecule, see Figure 1 for a 2D schematic diagram of different molecular surfaces. Indeed, the SES can be considered to be the prototype for the computational study of molecular surfaces.

The definition of VdW-surface is based on the model that each atom has a specific radius around the atom center. However, the definition of the VdW-surface has ignored the size and shape of the surrounding solvent molecules in solvation models. The definition of SAS has taken this into account by modeling them by idealized
Introduction 13

Figure 2: 3D schematic of the SES illustrating the convex spherical patches (red), the toroidal patches (yellow) and the concave spherical patches (blue).

spherical probes with a certain probe radius. The definition of the SES is different from the SAS in the sense that not the probe center traces out the desired surface, but the surface of the probe. In the application of docking [76], the SES will not lead to the overlapping of neighboring surfaces since the SES does not inflate the atom radii but the SAS will.

Sometimes, the SAS can be non-connected: it can be composed of several separate surfaces. We call the outmost surface as the exterior Solvent Accessible Surface (eSAS) and the union of all separated surfaces as the complete Solvent Accessible Surface (cSAS), see Chapter 11 for details. Correspondingly, we will also propose the concept of the complete Solvent Excluded Surface (cSES) and the exterior Solvent Excluded Surface (eSES).

Both the VdW-surface and the SAS are composed of three parts: open spherical patches, open circular arcs (or circles) and intersection points (formed by the intersection of three or more spheres). Their geometric features are therefore easier to understand. The SES can be divided into three corresponding types of patches as presented in Connolly’s work [30]: convex spherical patches, toroidal patches and concave spherical patches, see Figure 2 for an illustration. Any point on a convex spherical patch of the SES has a closest point to the SAS on a spherical patch. Similarly, any point on a toroidal patch of the SES has a closest point to the SAS on a circular arc, and any point on a concave patch has a closest point to the SAS which is an intersection point.

Despite that the whole SES is smooth almost everywhere, self-intersections among different SES-patches can occur and often cause SES-singularities. This singularity problem has led to difficulty in associated researches on the SES, for example, the failure of SES meshing algorithms and the imprecise calculation of molecular areas or volumes, or has been circumvented by approximate techniques [28]. In 1996, Sanner treated some special cases of SES-singularity in his MSMS (Michel Sanner’s Molecular Surface) package for the analytical calculation of molecular areas and volumes, and the triangulation of molecular surfaces [102]. Nevertheless, to our knowledge, the complete characterization of the SES-singularities remains unsolved despite of a large
In this thesis, we will present a computable method to represent implicitly the molecular surfaces. We suppose that the solute molecule is composed of $M$ atoms and the $j$th atom has a center $c_j$ and a VdW radius $r_j$. The solvent probe radius is denoted by $r_p$. Thus, we can denote by $f_{\text{sas}}$ the signed distance function to the SAS (negative inside the SAS and positive outside the SAS) as follows

$$f_{\text{sas}}(p) = \begin{cases} -\|p - x^p_{\text{sas}}\| & \text{if } p \text{ lies inside the SAS}, \\ \|p - x^p_{\text{sas}}\| & \text{if } p \text{ lies outside the SAS}, \end{cases}$$

(2.1)

where $x^p_{\text{sas}}$ denotes a closest point on the SAS to $p$. Note that there might exist more than one closest point on the SAS and in this case $x^p_{\text{sas}}$ is chosen as one of them. The implicit functions of the SAS (denoted by $\Gamma_{\text{sas}}$) and the SES (denoted by $\Gamma_{\text{ses}}$) are consequently deduced as follows

$$\Gamma_{\text{sas}} = f_{\text{sas}}^{-1}(0) \quad \text{and} \quad \Gamma_{\text{ses}} = f_{\text{sas}}^{-1}(-r_p),$$

which characterize mathematically these two molecular surfaces. However, the challenge is that $x^p_{\text{sas}}$ is not easy to compute because the shape of the molecule can be as complicated as possible.

In Chapter II of Part II, an efficient method for computing analytically the function value of $f_{\text{sas}}$ will be proposed, which therefore gives a complete characterization of the SES. In fact, it is based on three equivalence statements, which induces a nonoverlapping partition of $\mathbb{R}^3$. In addition, we redefine different types of SES-patches mathematically so that the SES-singularities will be characterized explicitly and no self-interaction will occur. By applying the Gauss-Bonnet theorem [35] and the Gauss-Green theorem [38], we will give an explicit formula of calculating analytically all the molecular areas and volumes, in particular for the SES. These quantities are useful in many protein models, such as describing the hydration effects [100, 19]. Furthermore, the complete characterization of different molecular surfaces will allow us to visualize them more precisely.

### 2.1.2 Molecular visualization

Molecular visualization is helpful for researchers to understand the geometrical structure of a molecule and to illustrate charge distributions on molecular surfaces. This topic is of course intrinsically linked to the notion of a solute cavity.

Since the VdW surface and the SAS are topologically simple, it is more challenging to visualize the SES (the so-called “smooth” molecular surface). As mentioned in the above section, Sanner proposed the reduced surfaces and then the MSMS algorithm [102] for computing an (in fact approximately) analytical representation of the SES. In addition, this algorithm also provides a triangulation of the SES with a user-
specified density of vertices and therefore the molecular visualization based on this triangulation is feasible. Despite of the self-intersection problem among different SES-patches (see Figure 3 for an example where self-intersection occurs), the MSMS is still one of the most widely-used packages for molecular visualization. Besides, there are many other contributions on the molecular visualization [64, 89, 59], the visualization of molecular dynamics [55, 61], the Eulerian representation of SESs [73], the high-quality mesh generation of molecular surfaces [65] and so on.

Benefitting from the complete characterization of the SES for the first time (see Chapter 1), we are now able to develop a piecewise meshing algorithm for molecular surfaces which avoids the self-intersection between SES-patches. In Chapter 2 of Part I, we will first give a detailed strategy for constructing the data structures of molecular surfaces, thanks to this analytical characterization. Then, a meshing algorithm for molecular surfaces, especially the SES, is developed, combining an advancing-front algorithm with the pre-computed data structures. The explicit characterization of all singularities resolves the issue of self-intersection that is experienced due to singularities as they can be computed prior to the meshing of the surface. This, in turn, allows the possibilities of meshing the SES exactly, in the sense that each vertex of the mesh lies exactly on the surface. Here, we emphasize that this is only possible due to the above-developed analysis of the molecular surfaces and it is not the case for the existing meshing algorithms. In addition, we propose an algorithm for filling molecular inner holes with virtual atoms for the reason that the appearance of these inner holes is not always justified in the solute cavity of the implicit solvation models, as this would mean that the solvent is present in these inner holes.

So far, we have introduced our achievements on the solute-solvent interface, including the complete characterization of molecular surfaces and the molecular visualization. Next, we will introduce different physical laws that describe the electrostatic potential of the implicit solvation model in different ways.
2.2 Solute-solvent interaction: physical laws

Based on a suitable choice of solute-solvent boundary (interface), the implicit solvation model divides the whole space into the solute cavity and the solvent region. Further, the solvent is characterized in terms of its macroscopic physical properties, such as the dielectric permittivity and the ionic strength. Different types of implicit solvation model are consequently constructed as follows, based on different physical laws to model the electrostatic potential of the solute-solvent interaction: the Polarizable Continuum Model (PCM), the COnductor-like Screen MOdel (COSMO) and the Poisson-Boltzmann (PB) solvation model.

2.2.1 Polarizable continuum model

The polarizable continuum model (PCM) \([26,109]\) is a widely-used type of implicit solvation model in computational chemistry to model solvation effects, in which the solvent is represented by a polarizable continuum. This implies that there is no charge in the solvent region and the charge distribution of the solvation system is the solute charge distribution \(\rho_M\) supported in the solute cavity. In the classical PCM with a solute cavity \(\Omega\), the space-dependent dielectric permittivity \(\varepsilon(x)\) is defined as (see the right of Figure 4)

\[
\varepsilon(x) = \begin{cases} 
1 & x \in \Omega, \\
\varepsilon_s & x \in \Omega^c := \mathbb{R}^3 \setminus \overline{\Omega},
\end{cases}
\] (2.2)

where \(\varepsilon_s\) is the (bulk) solvent dielectric constant and \(\varepsilon(x)\) has a jump on the solute-solvent boundary \(\Gamma := \partial \Omega\).

With the above dielectric permittivity (2.2), the PDE (1.5) of the electrostatic potential \(\psi\) can be rewritten as

\[
\begin{aligned}
-\Delta \psi &= 4\pi \rho_M, & \text{in } \Omega, \\
-\Delta \psi &= 0, & \text{in } \Omega^c, \\
[\psi] &= 0, & \text{on } \Gamma, \\
[\partial_n(\varepsilon \psi)] &= 0, & \text{on } \Gamma,
\end{aligned}
\] (2.3)

where \(n\) denotes the unit normal vector pointing outwards with respect to \(\Omega\), \([\psi]\) and \([\partial_n(\varepsilon \psi)]\) respectively denote the jump (inside \(\Omega\) minus outside) of \(\psi\) and the jump of the normal derivative of \(\varepsilon \psi\) on \(\Gamma\). An integral equation formalism (IEF) \([22,51,26]\) of this equation was proposed by E. Cancès, B. Mennucci and J. Tomasi, which has been the default PCM formulation in Gaussian \([14]\). This formalism transforms the original problem defined in the 3D space equivalently to an integral equation on the dielectric boundary and therefore the computational cost can be greatly reduced.
2.2.2 Conductor-like screening model

A reduced version of PCM is also popular in quantum chemistry, that is, the COnductor-like Screen MOdel (COSMO) \[109\], in which the solvent continuum is assumed to be conductor-like (see the right of Figure 4), i.e., one takes $\varepsilon_s = \infty$ in Eq. \(\text{(2.4)}\). That means that the dielectric permittivity function $\varepsilon(\mathbf{x})$ is taken as

$$
\varepsilon(\mathbf{x}) = \begin{cases} 
1 & \mathbf{x} \in \Omega, \\
\infty & \mathbf{x} \in \Omega^c.
\end{cases}
$$

(2.4)

This reduced model is usually employed to approximate the PCM when the solvent dielectric permittivity $\varepsilon_s$ is relatively large, for example, the (relative) dielectric permittivity of water is $\varepsilon_s = 78.4$ at room temperature (25°C).

In this case, the original PDE \(\text{(1.5)}\) of the electrostatic potential is simply a Poisson equation defined on a bounded domain with a homogeneous Dirichlet boundary condition

$$
\begin{aligned}
-\Delta \psi(\mathbf{x}) &= 4\pi \rho_M(\mathbf{x}), & \text{in } \Omega, \\
\psi(\mathbf{x}) &= 0, & \text{on } \Gamma.
\end{aligned}
$$

(2.5)

Here, the electrostatic potential $\psi$ vanishes on the solute-solvent boundary $\Gamma$ because the solvent is idealized as perfect conductor and consequently there is no electric field in the solvent region.

Comparing to the PCM defined in $\mathbb{R}^3$, the COSMO is only a problem defined on the bounded domain $\Omega$, which is consequently easier. As assumed, the charge distribution $\rho_M$ of the solute molecule is already known. For calculating the solvent effects with a finite dielectric constant $\varepsilon_s$, the electrostatic contribution to the solvation energy $E^s$ is usually approximated by

$$
E^s = f(\varepsilon_s)E^s_{\infty},
$$
where $E_s^\infty$ is the electrostatic contribution to the solvation energy computed from the COSMO by solving the PDE (2.3) and the factor $f(\varepsilon_s)$ is empirically given by

$$f(\varepsilon_s) = \frac{\varepsilon_s - 1}{\varepsilon_s + x},$$

with $x$ usually set to 0.5 based on theoretical arguments.

### 2.2.3 PB solvation model

The properties of numerous charged bio-molecules and their complexes with other molecules are dependent on not only the polarizable effects of the environment, but also on the ionic effects. In this case, the Poisson-Boltzmann (PB) solvation model [116, 86] takes into account both the solvent dielectric permittivity and the ionic strength, which is now widely-used. In such a model, the solvent is seen as a polarizable continuum containing ions. The solvent dielectric permittivity is defined by (2.2) in the PCM and the movement of ions in solution is accounted for by Boltzmann statistics. That is to say, the Boltzmann equation is used to calculate the local density $c_i$ of the $i$-th type of ion as follows

$$c_i = c_i^\infty e^{-\frac{W_i}{k_B T}},$$

(2.6)

where $c_i^\infty$ is the bulk ion concentration at an infinite distance from the solute molecule, $W_i$ is the work required to move the ion to the position from an infinitely far distance, $k_B$ is the Boltzmann constant, $T$ is the temperature in Kelvins (K). Combining (1.5), (1.8) and (2.6), we derive the Poisson-Boltzmann equation as follows (see [40])

$$\nabla \cdot [\varepsilon(x) \nabla \psi(x)] = 4\pi \rho_M(x) + \sum z_i e c_i^\infty e^{-\frac{z_i e \psi(x)}{k_B T}} \chi_{\Omega^c}(x),$$

(2.7)

where $z_i e$ is the charge of the $i$-th type of ion, $e$ is the elementary charge and $\chi_{\Omega^c}$ is the characteristic function of the solvent region $\Omega^c$.

In the PB solvation model with 1 : 1 electrolyte, there are two types of ions respectively with charge $+e$ and $-e$ (see Figure 5). With the assumption that $\psi$ satisfies the low potential condition, i.e., $\left|\frac{e\psi}{k_B T}\right| \ll 1$, the above Poisson-Boltzmann equation can be linearized to (see [86] for this form)

$$\nabla \cdot [\varepsilon(x) \nabla \psi(x)] + \tilde{\kappa}(x) \psi(x) = 4\pi \rho_M(x),$$

(2.8)

which $\psi$ is determined by the spatial dielectric permittivity function $\varepsilon(x)$, the modified Debye-Hückel parameter $\tilde{\kappa}(x)$ and the solute’s charge distribution function $\rho_M(x)$.

**Remark 2.1.** If the solvent contains more than two types of ions, the associated
Poisson-Boltzmann equation can still be linearized to the same form (2.8).

The dielectric permittivity function is defined in the classical way

\[ \varepsilon(x) = \begin{cases} 
1 & x \in \Omega, \\
\varepsilon_\infty & x \in \Omega^c, 
\end{cases} \]  

(2.9)

where \( \varepsilon_\infty \) is the solvent dielectric constant as previous. The modified Debye-Hückel parameter is taken as

\[ \tilde{\kappa}(x) = \begin{cases} 
0 & \text{in } \Omega, \\
\sqrt{\varepsilon_\infty} \kappa & \text{in } \Omega^c, 
\end{cases} \]  

(2.10)

where \( \kappa \) is the Debye-Hückel screening constant representing the attenuation of interactions due to the presence of ions in the solvent region, which is related to the ionic strength \( I \) of the aqueous salt solution according to

\[ \kappa^2 = \frac{8\pi e^2 N_A I}{1000\varepsilon_\infty k_B T}, \]  

(2.11)

where \( N_A \) is the Avogadro constant.

In summary, we have presented in this part three types of physical law for the implicit solvation model, in three different cases of solvent: the polarizable continuum (PCM), the conductor-like continuum (COSMO) and the polarizable continuum containing ions (PB solvation model). In the following part, we introduce a particular kind of domain decomposition method for solving the electrostatic problem of these implicit solvation models.
3 Domain decomposition methods for implicit solvation models

Implicit solvation models are widely-used in the chemistry community, but little interaction with applied mathematics can be observed despite the fact that solutions to partial differential equations need to be approximated. The underlying physical law is provided by electrostatic interaction involving elliptic operators.

In this section, we first introduce two state-of-art Schwarz domain decomposition (dd) methods respectively for solving the COSMO and the PCM: the ddCOSMO and the ddPCM. Then, we introduce two new Schwarz domain decomposition methods respectively for solving an SES-based PCM and the PB solvation model, which are contributions of this thesis. The methods generally consist of two steps:

[1] if the electrostatic problem is defined on an unbounded domain, transform the original problem equivalently to problems that are defined on a bounded domain and might be coupled by a non-local condition;

[2] develop a classical Schwarz domain decomposition method that decomposes the bounded domain into balls and only solve local coupled sub-problems restricted to balls.

For the SES-based PCM, we will first construct the model and then propose the corresponding domain decomposition method for solving it, called the ddPCM-SES method. Further, for the PB solvation model which is already established in Section 2.2.3, we will propose a domain decomposition method for solving it, which is called the ddLPB method.

3.1 ddCOSMO

To solve the electrostatic problem of the COSMO, the finite element method or the finite difference method can be used. However, the computational cost is too expensive for a large realistic molecule. In particular, meshing the solute cavity $\Omega$ of a complicated molecule is already too costly. The state-of-art COSMO solver is the ddCOSMO [25, 72, 69], a Schwarz domain decomposition method [96, 107] developed for solving the COSMO in the past several years, which has attracted much attention due to its impressive efficiency, that is, it performs about two orders of magnitude faster than other equivalent methods [69].

The crucial part consists in decomposing the domain into a union of balls and solving only each sub-problem restricted to a ball. Let $\Omega$ be the VdW-cavity or the
SAS-cavity, meaning that $\Omega$ is a union of overlapping balls in the following form

$$\Omega = \bigcup_{j=1}^{M} \Omega_j, \quad \Omega_j = B_{r_j}(x_j),$$

(3.1)

where each $\Omega_j$ denotes the $j$-th atomic VdW-ball (or SAS-ball) with center $x_j$ and radius $r_j$.

We then homogenize the COSMO equation (2.5) by defining

$$\psi_0 = \sum_{i=1}^{M} \frac{q_i}{|x - x_i|},$$

(3.2)

which satisfies $-\Delta \psi_0 = 4\pi \rho_M(x)$ in $\mathbb{R}^3$. Here, $\rho_M$ is given by Eq. (1.8). The reaction potential $\psi_r := \psi - \psi_0$ satisfies consequently the following Laplace equation with the Dirichlet boundary condition

$$\begin{cases}
-\Delta \psi_r = 0, & \text{in } \Omega, \\
\psi_r = -\psi_0, & \text{on } \Gamma.
\end{cases}$$

(3.3)

The reaction potential is indeed the electrostatic potential that is additionally created by the pressure of the solvent with respect to vacuum.

Using the Schwarz decomposition of $\Omega$, this Laplace equation can be recast as the following group of coupled sub-equations, each restricted to $\Omega_j$:

$$\begin{cases}
-\Delta \psi_r|_{\Omega_j} = 0 & \text{in } \Omega_j, \\
\psi_r|_{\Gamma_j} = \phi_{r,j} & \text{on } \Gamma_j,
\end{cases}$$

(3.4)

where $\Gamma_j = \partial \Omega_j$ and

$$\phi_{r,j} = \begin{cases}
\overline{\psi}_r & \text{on } \Gamma^i_j, \\
-\psi_0 & \text{on } \Gamma^e_j.
\end{cases}$$

(3.5)

Here, $\Gamma^e_j$ is the external part of $\Gamma_j$ not contained in any other ball $\Omega_i$ ($i \neq j$), i.e., $\Gamma^e_j = \Gamma \cap \Gamma_j$; $\Gamma^i_j$ is the internal part of $\Gamma_j$, i.e., $\Gamma^i_j = \Omega \cap \Gamma_j$ (see Figure 6 for the illustration of notations). In addition, $\overline{\psi}_r$ is defined as

$$\overline{\psi}_r(x) = \frac{1}{|\mathcal{N}(j,x)|} \sum_{i \in \mathcal{N}(j,x)} \psi_r|_{\Omega_i}(x), \quad \forall x \in \Gamma^i_j,$$

(3.6)

where $\mathcal{N}(j,x)$ represents the index set of all balls such that $x \in \Omega_i$. In fact, for a fixed point $x \in \Gamma^i_j$, $\overline{\psi}_r(x)$ is the average value of $\psi_r|_{\Omega_i}(x)$ obtained by solving the local Laplace equation in the neighboring $\Omega_i$.

For each sub-equation (3.4)–(3.5) in a ball, the spherical harmonics can consequently be used as basis functions to solve it numerically. An iterative solver is
then applied to solve the coupled sub-equations. For instance, the idea of the Jacobi algorithm is to solve each local problem based on the boundary condition of the neighboring solutions derived from the previous iteration. During this iterative procedure, the computed value of $\psi_j|_{\Gamma_j}$ is updated step by step and finally converges to the exact value.

3.2 ddPCM

Latter on, a similar Schwarz domain decomposition method for solving the classical PCM with the PDE (1.13) of electrostatic potential and the definition (2.2) of $\varepsilon(x)$ was proposed in [108] (called ddPCM, see also website [70]), which is based on the integral equation formalism (IEF) of PCM [26].

Let $\Omega$ be again the VdW-cavity or the SAS-cavity in the form of Eq. (3.1). By defining the same reaction potential $\psi_r$ as in the ddCOSMO, we derive the following PDE according to Eq. (2.3)

$$\begin{cases}
-\Delta \psi_r = 0, & \text{in } \Omega, \\
-\Delta \psi_r = 0, & \text{in } \Omega^c, \\
[\psi_r] = 0, & \text{on } \Gamma, \\
[\partial_n (\varepsilon \psi_r)] = (\varepsilon_0 - 1)\partial_n \psi_0, & \text{on } \Gamma,
\end{cases}$$

The idea of the ddPCM is to first transform the problem defined in the whole space $\mathbb{R}^3$ equivalently to the IEF-PCM equation of an unknown surface charge density $\sigma \in H^{-\frac{1}{2}}(\Gamma)$ as follows

$$\mathcal{R}_\varepsilon \mathcal{S} \sigma = -\mathcal{R}_\infty \psi_0, \quad \text{on } \Gamma,$$

(3.7) where $\mathcal{S}$ is the single-layer operator on $\Gamma$ defined by

$$\mathcal{S} \sigma(x) := \int_{\Gamma} \frac{\sigma(y)}{4\pi|x-y|}, \quad \forall x \in \Gamma, \sigma \in H^{-\frac{1}{2}}(\Gamma),$$

(3.8)
the two operators $R_1$ and $R_{\infty}$ are defined using the double-layer operator $D$ as follows

$$R_\varepsilon := \frac{\varepsilon + 1}{2(\varepsilon - 1)} - D, \quad R_{\infty} := \frac{1}{2} - D,$$

(3.9)

and

$$D\sigma(x) := \int_{\Gamma} \left( \nabla_y \frac{1}{4\pi |x - y|} \cdot n_y \right) \sigma(y), \quad \forall x \in \Gamma, \quad \sigma \in H^{-\frac{1}{2}}(\Gamma).$$

(3.10)

Here, $n_y$ denotes the unit normal vector at $y$ on $\Gamma$. Then, to solve the IEF-PCM equation (3.7), two steps are taken, including solving the following equation of an auxiliary function $\Phi$

$$R_\varepsilon \Phi = -R_{\infty} \psi_0,$$

(3.11)

and then the following equation of $\sigma$

$$S\sigma = \Phi.$$

(3.12)

In the above two equations, the second one is equivalent to the Laplace equation of the COSMO and therefore, can be solved by the ddCOSMO solver. Solving the first equation involves decomposing the solute cavity $\Omega$ into balls and solving a group of sub-equation each restricted to a ball. To do this, a similar Schwarz decomposition method to the ddCOSMO is applied, see details in [108].

Both the ddCOSMO and the ddPCM fully take advantage of the geometrical structure of the VdW-cavity or SAS-cavity, i.e., the union of balls. These two methods give a fast resolution of respectively the COSMO and the classical PCM. Then, we would like to generalize them to an SES-based PCM and the PB solvation model.

### 3.3 ddPCM-SES

As mentioned, the previous two methods (ddCOSMO and ddPCM) are based on the VdW-cavity or the SAS-cavity, due to their simpler geometrical structure and easier computation. However, this might not be physically appropriate and the choice of the cavity is indeed important as pointed out in [110, Section II. C.]: The shape and size of the cavity are critical factors in the elaboration of a method. An ideal cavity should reproduce the shape of the solute $M$, with the inclusion of the whole charge distribution $\rho_M$ and with the exclusion of empty spaces which can be filled by the solvent continuous distribution. If the cavity is too large the solvation effects are damped; if it is too small serious errors may arise in the evaluation of the interaction energy for the portions of $\rho_M$ (atoms or bonds) near the solute-solvent boundary.

The SES-cavity, the region enclosed by the SES, is thought to be a more appropriate choice of solute cavity, since it has a stronger physical meaning in the sense that it represents the region where solvent molecules (represented by idealized spheres) can
not touch. As a consequence, we would like to study the possibility of constructing and then solving an SES-based solvation model, such as an SES-based PCM.

### 3.3.1 SES-based PCM

In some chemical calculations, it has been confirmed that taking the SES-cavity $\Omega_{\text{ses}}$ into account can yield more accurate results, such as in [98, 91]. The shape of the cavity is represented by the dielectric permittivity function which in this case equals to one (the dielectric permittivity of vacuum) within the SES-cavity.

In the classical PCM, the permittivity function $\varepsilon(x)$ in the form of (2.2) is discontinuous and equals to the bulk solvent dielectric permittivity outside the solute cavity. A solver for such an SES-based PCM has been proposed in [50, 17], using the integral equation formulation of PCM and an efficient mesh generator of the SES. However, it has been argued that treating the solvent dielectric permittivity as constant is not sufficient and as a remedy, continuous permittivity functions $\varepsilon_s(x)$ have been proposed [17, 13], based on the VdW-cavity or the SAS-cavity. But a method containing a continuous permittivity function based on the SES-cavity does not exist, to our knowledge. We are convinced that introducing the SES-based continuous permittivity function is the next logical step to further refine the PCM.

In our model, we assume that the dielectric permittivity in the region $\Omega_\infty$ far away from the solute molecule is equal to the bulk solvent dielectric permittivity $\varepsilon_s$, which is a reasonable assumption since at the position far from the solute molecule, the solvent molecules are influenced little by the solute molecule. Between the solute cavity and $\Omega_\infty$, an intermediate dielectric boundary layer (the switching region) $\mathcal{L} := \Omega_\infty^c \cap \Omega_{\text{ses}}^c$ is constructed, to obtain a continuous dielectric permittivity function $\varepsilon(x)$ of the following form (see the left of Figure 4)

$$
\varepsilon(x) = \begin{cases} 
1 & x \in \Omega_{\text{ses}}, \\
\varepsilon_s(x) & x \in \mathcal{L}, \\
\varepsilon & x \in \Omega_\infty.
\end{cases}
$$

(3.13)

See Figure 6 for an example of $\varepsilon(x)$, where $\varepsilon(x)$ is a distance-dependent function. The “distance” here represents the signed distance to the SAS, denoted by $f_{\text{sas}}$, which also characterizes the SES-cavity as follows

$$
\Omega_{\text{ses}} = \{x \in \mathbb{R}^3 : f_{\text{sas}}(x) < -r_p\},
$$

where $r_p$ represents the probe radius. In addition, the bulk solvent region $\Omega_\infty$ is characterized as

$$
\Omega_\infty = \{x \in \mathbb{R}^3 : f_{\text{sas}}(x) > r_0\},
$$

where $r_0$ is a positive constant. We then deduce that $\psi$ of Eq. (1.3) is harmonic in $\Omega_\infty$, i.e. $-\Delta \psi = 0$ in $\Omega_\infty$. 

To solve the electrostatic problem of this SES-based PCM, one shall solve Eq. (1.5) with a space-dependent parameter $\varepsilon(x)$ in the form of (3.13).

### 3.3.2 Domain decomposition method for the SES-based PCM

According to the equation (1.5) together with the definition (3.13) of $\varepsilon(x)$, we consider the PDE for the SES-based model in the following general form

\[
\begin{cases}
-\nabla \cdot \varepsilon \nabla u = f, & \text{in } \Omega_0, \\
-\Delta u = 0, & \text{in } \Omega_\infty, \\
[u] = 0, & \text{on } \Gamma_0, \\
[\partial_n u] = 0, & \text{on } \Gamma_0,
\end{cases}
\]

where the bounded domain $\Omega_0$ and the unbounded domain $\Omega_\infty$ are complementary in $\mathbb{R}^3$, $[u]$ and $[\partial_n u]$ denote the jump of $u$ respectively its normal derivative on $\Gamma_0 := \partial \Omega_0$.

The scheme of a two-step domain decomposition method for the SES-based PCM (called the ddPCM-SES) is illustrated in Figure 7. The unbounded problem defined in $\mathbb{R}^3$ is first transformed into two coupled problems both defined on $\Omega_0$

\[
\begin{cases}
-\nabla \cdot \varepsilon \nabla u = f, & \text{in } \Omega_0, \\
u = g, & \text{on } \Gamma_0,
\end{cases} \quad \text{and} \quad \begin{cases}
-\Delta u_\infty = 0, & \text{in } \Omega_0, \\
u_\infty = g, & \text{on } \Gamma_0,
\end{cases}
\]

where the coupling condition arises through the auxiliary variable $g$ defined as

\[
g = \frac{1}{4\pi} \mathcal{S}_{\Gamma_0}(\partial_n u_\infty - \partial_n u), \quad \text{on } \Gamma_0.
\]

Here, $\mathcal{S}_{\Gamma_0}$ denotes the single-layer potential of the interface $\Gamma_0$, associated with the
3. Domain decomposition methods for implicit solvation models

Figure 8: Schematic diagram of the dielectric permittivity function $\varepsilon(x)$ with respect to $f_{\text{sas}}$. The dielectric boundary layer $\mathcal{L}$ (switching region) is bounded by two dashed lines (red), i.e., the region where $-r_p \leq f_{\text{sas}} \leq r_0$.

The bounded domain $\Omega_0$ is taken as a union of balls, inspired by the geometrical structure of the solute molecule. Since $\Omega_0$ consists of a union of balls, we propose to further apply a classical domain decomposition algorithm in order to solve the two problems (3.14) by only solving local sub-problems restricted to balls.

As a consequence, a Laplace solver and a Generalized Poisson (GP) solver are developed respectively for solving the Laplace equation and the GP equation in each ball, which allows us to use the spherical harmonics as basis functions in the angular direction to propose an efficient spectral method within each ball. It is important to notice that this algorithm does not require any meshing and only involves problems in balls that are coupled to each other within the domain decomposition paradigm.

In Chapter 3 of Part II, we will present the details of this Schwarz domain decomposition method for the SES-based PCM. We will first remind different solute-solvent boundaries including the VdW surface, the SAS and the SES. We will also provide more details on the construction of the continuous dielectric permittivity function $\varepsilon(x)$ of PCM, ensuring that the SES-cavity always has the dielectric constant of vacuum as explained above. Then, we will present the problem formulation of the PCM as well as its equivalent transformation, and therefore, a global strategy for solving the problem. Later, we will introduce the scheme of the domain decomposition method for solving the associated partial differential equations iteratively. This requires to...
develop the Laplace solver and the GP-solver in a ball, which are presented. After that, we will give a series of numerical results of the proposed method.

### 3.4 ddLPB

For the sake of simplicity, we focus on solving the linearized Poisson-Boltzmann equation (2.8) defined in $\mathbb{R}^3$. This problem can be divided into a Poisson equation defined in the bounded solute cavity $\Omega$ and a homogeneous screened Poisson (HSP) equation defined in the unbounded solvent region $\Omega^c$ as follows

\[
\begin{align*}
-\Delta \psi(x) &= 4\pi \rho_M(x) \quad \text{in } \Omega, \\
-\Delta \psi(x) + \kappa^2 \psi(x) &= 0 \quad \text{in } \Omega^c,
\end{align*}
\]

combined with two classical jump-conditions

\[
\begin{align*}
[\psi] &= 0 \quad \text{on } \Gamma, \\
[\partial_n (\varepsilon \psi)] &= 0 \quad \text{on } \Gamma.
\end{align*}
\]

Here, $\kappa$ is the Debye-Hückel screening constant. We remind that the solute cavity $\Omega$ in the context of the PB equation consists of a union of balls (i.e., we take the VdW-cavity or the SAS-cavity due to the simple geometry).
The scheme of a two-step domain decomposition method for the PB solvation model (called the ddLPB method) is illustrated in Figure 10. As in the ddCOSMO, we first homogenize the Poisson equation (3.16), using $\psi_0$ defined by (3.2). The reaction potential $\psi_r := \psi - \psi_0$ satisfies the following Laplace equation

$$
\begin{cases}
-\Delta \psi_r = 0, & \text{in } \Omega, \\
\psi_0 + \psi_r = g, & \text{on } \Gamma.
\end{cases}
$$

(3.18)

Further, we use the integral equation formulation to represent the electrostatic potential $\psi|_{|\Omega|$ in the solvent region, which simultaneously gives the potential $\psi_e$ of an extended screened Poisson equation in the solute cavity (this is an interior Dirichlet problem)

$$
\begin{cases}
-\Delta \psi_e(x) + \kappa^2 \psi_e(x) = 0, & \text{in } \Omega, \\
\psi_e = g, & \text{on } \Gamma,
\end{cases}
$$

(3.19)

where $\psi_e$ satisfies the same HSP equation as $\psi|_{|\Omega|$ with the same Dirichlet boundary condition on $\Gamma$, but defined in $\Omega$. The coupling condition between $\psi_r$ and $\psi_e$ arises through an auxiliary variable $g$ defined as

$$
g = S_\kappa \left( \frac{\partial_n \psi_e - 1}{\varepsilon_s} \frac{\partial_n (\psi_0 + \psi_r)}{\sigma_n} \right), \quad \text{on } \Gamma,
$$

(3.20)

where $S_\kappa$ is another single-layer potential defined as

$$
S_\kappa \sigma(x) := \int_{\Gamma} e^{-\kappa |x-y|} \sigma(y) \, 4\pi |x-y|^{-2}, \quad \forall x \in \Gamma, \quad \sigma \in H^{-\frac{1}{2}}(\Gamma).
$$

(3.21)

At this moment, the initial problem has been transformed equivalently into two coupled problems (3.18)-(3.19) both defined on the bounded domain $\Omega$ with a coupling condition (3.20). Taking advantage of the fact that $\Omega$ is a union of overlapping balls, the Schwarz domain decomposition can be applied to solve these two equations by respectively solving a group of coupled sub-problems each defined on a ball.

Ultimately, a Laplace solver and a HSP solver are developed respectively for solving the Laplace equation and the HSP equation in each ball, which allows us to use the spherical harmonics as basis functions in the spherical direction to propose an efficient spectral method within each ball. Similar to the ddCOSMO, the ddPCM and the ddPCM-SES, the ddLPB does not require any meshing and only involves problems in balls that are coupled to each other.

**Remark 3.1.** The ddPCM-SES method and the ddLPB method are inspired by the previous ddCOSMO method and ddPCM method which run impressively fast. The ddLPB has the same computational complexity as the ddPCM.

In Chapter 4 of Part II, we will present the details about this Schwarz domain decomposition method for the PB solvation model. We will first introduce different
definitions of the solute-solvent boundary as in Chapter 5 and also the Poisson-Boltzmann equation defined in the whole space in the case where the solvent is an ionic solution. Then, we will transform equivalently the original Poisson-Boltzmann defined in $\mathbb{R}^3$ to two coupled equations both defined on the bounded solute cavity. Next, we will present the global strategy for solving these two coupled equations, using the domain decomposition method. The domain decomposition scheme requires to develop two single-domain solvers respectively for the Laplace equation and the HSP equation defined in a ball. After that, we will give a reformulation of the coupling conditions that should be discretized and consequently derive a global linear system to be solved. We will also present a series of numerical tests about the calculation of the electrostatic contribution to the solvation energy using the ddLPB method.
3. Domain decomposition methods for implicit solvation models
Part I

Molecular Surfaces
Chapter 1

Mathematical analysis and calculation of molecular surfaces

Contents

1.1 Introduction .................................................. 34
   1.1.1 Previous Work ........................................ 35
   1.1.2 Contribution .......................................... 35
   1.1.3 Outline ................................................ 36
1.2 Introduction to implicit surfaces ................................. 36
1.3 Implicit molecular surfaces .................................... 37
   1.3.1 Mathematical definitions ................................. 39
   1.3.2 Equivalence statements ................................ 41
   1.3.3 New Voronoi-type diagram ............................... 47
   1.3.4 SAS-area and SAS-volume ............................... 49
1.4 Solvent accessible surface .................................... 50
   1.4.1 Mathematical definitions ................................. 51
   1.4.2 Equivalence statements ................................ 53
   1.4.3 SES-singularities ....................................... 56
   1.4.4 SES-area and SES-volume ............................... 56
1.5 Solvent excluded surface .................................... 59
   1.5.1 Mathematical definitions ................................. 59
   1.5.2 SES-singularities ....................................... 59
   1.5.3 SES-area and SES-volume ............................... 59
1.6 Construction of molecular surfaces ............................. 59
   1.6.1 Construction of the cSAS and the eSAS .................. 60
   1.6.2 Binary tree to construct spherical patches .............. 60
   1.6.3 Interior of a loop ...................................... 62
   1.6.4 Construction of the cSES and the eSES ................. 62
1.7 Numerical results ............................................ 63
1.1 Introduction

As mentioned in Section 1 of the introduction of this thesis, the majority of chemically relevant reactions take place in the liquid phase and the effect of the environment (solvent) is important and should be considered in various chemical computations. The implicit solvation model (or continuum solvation model) is a model in which the effect of the solvent molecules on the solute are described by a continuous model. In continuum solvation continuum models, the notion of molecular cavity and molecular surface is a fundamental part of the model. The molecular cavity occupies the space of the solute molecule where a solvent molecule cannot touch and the molecular surface, the boundary of the corresponding cavity, builds the interface between the solute and the solvent.

A precise understanding of the nature of the surface is essential for the implicit solvation model and as a consequence for running numerical computations. The van der Waals (VdW) surface, the Solvent Accessible Surface (SAS) and the Solvent Excluded Surface (SES) are well-established concepts. The VdW surface is more generally used in chemical calculations, such as in the recent developments for example, of numerical approximations to the COnductor-like Screening MOdel (COSMO) due to the simplicity of the cavity. Since the VdW surface is the topological boundary of the union of spheres, the geometric features are therefore easier to understand. However, the SES, which is considered to be a more precise description of the cavity, is more complicated and its analytical characterization remains unsatisfying despite a large number of contributions in literature.
1.1.1 Previous Work

In quantum chemistry, atoms of a molecule can be represented by VdW-balls with VdW-radii obtained from experiments [97]. The VdW surface of a solute molecule is consequently defined as the topological boundary of the union of all VdW-balls. For a given solute molecule, its SAS and the corresponding SES were first introduced by Lee & Richards in the 1970s [67, 99], where the solvent molecules surrounding a solute molecule are reduced to spherical probes [109]. The SES is also called “the smooth molecular surface” or “the Connolly surface”, due to Connolly’s fundamental work [30]. He has divided the SES into three types of patches: convex spherical patches, saddle-shaped toroidal patches and concave spherical triangles. But the self-intersection among different patches in this division often causes singularities despite that the whole SES is smooth almost everywhere. This singularity problem has led to difficulty in many associated researches on the SES, for example, failure of SES meshing algorithms and imprecise calculation of molecular areas or volumes, or has been circumvented by approximate techniques [28]. In 1996, Sanner treated some special singularity cases in his MSMS (Michel Sanner’s Molecular Surface) software for meshing molecular surfaces [102]. However, to our knowledge, the complete characterization of the singularities of the SES remains unsolved.

1.1.2 Contribution

In this chapter, we will characterize the above molecular surfaces with implicit functions, as well as provide explicit formulas to compute analytically the area of molecular surfaces and the volume of molecular cavities. We first propose a method to compute the signed distance function to the SAS, based on three equivalence statements which also induce a new partition of $\mathbb{R}^3$. As a consequence, a computable implicit function of the corresponding SES is given from the relatively simple relationship between the SES and the SAS. Furthermore, we will redefine different types of SES patches mathematically so that the singularities will be characterized explicitly. Besides, by applying the Gauss-Bonnet theorem [35] and the Gauss-Green theorem [38], we succeed to calculate analytically all the molecular areas and volumes, in particular for the SES. These quantities are thought to be useful in protein modeling, such as describing the hydration effects [100, 19].

In addition, we will refine the notion of SAS and SES by considering the possible inner holes in the solute molecule yielding the notions of the complete SAS (cSAS) and the corresponding complete SES (cSES). To distinguish them, we call respectively the previous SAS and the previous SES as the exterior SAS (eSAS) and the exterior SES (eSES). A method with binary tree to construct all these new molecular surfaces will also be proposed in this chapter in order to provide a computationally efficient method.
1.1.3 Outline

We first introduce the concepts of implicit surfaces in the second section and the implicit functions of molecular surfaces are given in the third section. In the fourth section, we present two more precise definitions about the SAS, either by taking the inner holes of the solute molecule considered into account or not. Then, based on three equivalence statements that are developed, we propose a computable method to calculate the signed distance function from any point to the SAS analytically. In this process, a new Voronoi-type diagram for the SAS-cavity is given which allows us to calculate analytically the area of the SAS and the volume inside the SAS. In the fifth section, a computable implicit function of the SES is deduced directly from the signed distance function to the SAS and according to the new Voronoi-type diagram, all SES-singularities are characterized. Still within this section, the formulas of calculating the area of the SES and the volume inside the SES will be provided. In the sixth section, we explain how to construct the SAS (cSAS and eSAS) and the SES (cSES and eSES) for a given solute molecule considering the possible inner holes. Numerical results are illustrated in the seventh section and finally, we provide some conclusions of this chapter in the last section.

1.2 Introduction to implicit surfaces

We start with presenting the definition of implicit surfaces [112]. In a very general context, a subset \( O \subset \mathbb{R}^n \) is called an implicit object if there exists a real-valued function \( f: U \rightarrow \mathbb{R}^k \) with \( O \subset U \subset \mathbb{R}^n \) and a subset \( V \subset \mathbb{R}^k \), such that \( O = f^{-1}(V) \). That is,

\[
O = \{ p \in U : f(p) \in V \}.
\]

The above definition of an implicit object is broad enough to include a large family of subsets of the space. In this chapter, we consider the simple case where \( U = \mathbb{R}^3 \), \( V = \{0\} \) and \( f: \mathbb{R}^3 \rightarrow \mathbb{R} \) is a real-valued function. As a consequence, an implicit object is represented as a zero-level set \( O = f^{-1}(0) \), which is also called an implicit surface in \( \mathbb{R}^3 \), and the function \( f \) is called an implicit function of the implicit surface. Notice that there are various implicit functions to represent one surface in the form of a zero-level set.

The signed distance function \( f_S \) of a closed bounded oriented surface \( S \) in \( \mathbb{R}^3 \), determines the distance from a given point \( p \in \mathbb{R}^n \) to the surface \( S \), with the sign determined by whether \( p \) lies inside \( S \) or not. That is to say,

\[
f_S(p) = \begin{cases} 
-\inf_{x \in S} \|p - x\| & \text{if } p \text{ lies inside } S, \\
\inf_{x \in S} \|p - x\| & \text{if } p \text{ lies outside } S.
\end{cases}
\]  

(1.2.1)
Figure 1: This is a 2-dimension (2D) schematics of the SAS and the SES, both defined by a spherical probe in orange rolling over the molecule atoms in dark blue.

This is naturally an implicit function of $S$.

### 1.3 Implicit molecular surfaces

In quantum chemistry, atoms of a molecule are represented by VdW-balls with VdW radii which are experimentally fitted, given the underlying chemical element \cite{97}. As a consequence and mathematically speaking, the VdW surface is defined as the topological boundary of the union of all VdW-balls. Besides, the SAS of a solute molecule is defined by the center of an idealized spherical probe rolling over the solute molecule, that is, the surface enclosing the region in which the center of a spherical probe can not enter. Finally, the SES is defined by the same spherical probe rolling over the solute VdW-cavity, that is, the surface enclosing the region in which a spherical probe can access. In other words, the SES is the boundary of the union of all spherical probes that do not intersect the VdW-balls of the solute molecule, see Figure 1 for a graphical illustration.

We denote by $M$ the number of atoms in a solute molecule, by $c_i \in \mathbb{R}^3$ and $r_i \in \mathbb{R}^+$ the center and the radius of the $i$-th VdW atom. The open ball with center $c_i$ and radius $r_i$ is called the $i$-th SAS-ball denoted by $B_i$, where $r_p$ is the radius of the idealized spherical probe. Furthermore, we denote by $S_i$ the $i$-th SAS-sphere corresponding to $B_i$, that is, $S_i = \partial B_i$. Similar to the VdW surface, the SAS can be represented as an implicit surface $f_{\text{sas}}^{-1}(0)$ with the following implicit function:

$$f_{\text{vdw}}(p) = \min_{i=1,\ldots,M} \{ \| p - c_i \|_2 - r_i \}, \quad \forall p \in \mathbb{R}^3. \quad (1.3.1)$$

Similarly, the open ball with center $c_i$ and radius $r_i + r_p$ is called the $i$-th SAS-ball denoted by $B_i$, where $r_p$ is the radius of the idealized spherical probe. Additionally, we denote by $S_i$ the $i$-th SAS-sphere corresponding to $B_i$, that is, $S_i = \partial B_i$. Similar to the VdW surface, the SAS can be represented as an implicit surface $f_{\text{sas}}^{-1}(0)$ with
the following implicit function:

\[ \tilde{f}_{\text{SAS}}(p) = f_{\text{vdw}}(p) - r_p = \min_{i=1,\ldots,M} \{ \|p - c_i\|_2 - r_i - r_p \}, \quad \forall p \in \mathbb{R}^3. \]  

(1.3.2)

We notice that the above implicit function of the SAS is simple to compute. It seems nevertheless hopeless to us to further obtain an implicit function of the SES if constructing upon this simple implicit function \( \tilde{f}_{\text{SAS}}(p) \) which is not a distance function. On the other hand, having the signed distance function, see (1.2.1), at hand would allow the construction of an implicit function for the SES due to the geometrical relationship between the SAS and the SES.

Indeed, according to the fact that any point on the SES has signed distance \(-r_p\) to the SAS, an implicit function of the SES is obtained directly as:

\[ f_{\text{SES}}(p) = f_{\text{SAS}}(p) + r_p, \]  

(1.3.3)

which motivates the choice of using the signed distance function to represent the SAS. From the above formula, the SES can be represented by a level set \( f_{\text{SAS}}^{-1}(-r_p) \), associated with the signed distance function \( f_{\text{SAS}} \) to the SAS. Therefore, the key point becomes how to compute the signed distance \( f_{\text{SAS}}(p) \) from a point \( p \in \mathbb{R}^3 \) to the SAS. Generally speaking, given a general surface \( S \subset \mathbb{R}^3 \) and any arbitrary point \( p \in \mathbb{R}^3 \), it is difficult to compute the signed distance from \( p \) to \( S \). However, considering that the SAS is a special surface formed by the union of SAS-spheres, this computation can be done analytically.

We state a remark about another implicit function to characterize the SES, proposed by Pomelli and Tomasi [90]. In [89], this function can be written as:

\[ \tilde{f}_{\text{SES}}(p) = \min_{1 \leq i < j < k \leq M} f_{ijk}(p), \quad \forall p \in \mathbb{R}^3, \]  

(1.3.4)

where \( f_{ijk} \) represents the signed distance function to the SES of the \( i \)-th, \( j \)-th and \( k \)-th VdW atom. However, this representation might fail sometimes, see two representative 2D examples in Figure 2. Indeed, the formula (1.3.4) for each molecule in Figure 2 can be rewritten as:

\[ \tilde{f}_{\text{SES}}(p) = \min_{1 \leq i < j \leq 3} f_{i,j}(p), \quad \forall p \in \mathbb{R}^2, \]  

(1.3.5)

where \( f_{i,j} \) represents the signed distance function to the SES of the \( i \)-th and the \( j \)-th VdW atom. However, each molecular cavity defined by \( \{ p \in \mathbb{R}^2 : \tilde{f}_{\text{SES}}(p) \leq 0 \} \) has excluded the region in grey inside the real SES.

Further, the region enclosed by the van der Waals surface is called the VdW-cavity, that is, any point \( p \) in the VdW-cavity satisfies \( f_{\text{vdw}}(p) \leq 0 \). More generally, we call the region enclosed by a molecular surface as the corresponding molecular cavity. As a consequence, the region enclosed by the SAS is called the SAS-cavity,
and the region enclosed by the SES is called the SES-cavity. Similarly, any point $p$ in the SAS-cavity satisfies $f_{\text{sas}}(p) \leq 0$, and any point $p$ in the SES-cavity satisfies $f_{\text{ses}}(p) \leq 0$.

### 1.4 Solvent accessible surface

In the framework of continuum solvation models, using the VdW-cavity as the solvent molecular cavity has the characteristic that its definition does not depend in any way on characteristics of the solvent. In other words, the above-mentioned VdW surface has ignored the size and shape of the solvent molecules, while the definition of the SAS includes some of these characteristics. In the following, we first provide two more precise mathematical definitions of the SAS considering possible inner holes of a solute molecule. After that, we will provide a formula for the signed distance function to the SAS, which is indeed based on three equivalence statements, providing explicitly a closest point on the SAS to any point in $\mathbb{R}^3$. In this process, a new Voronoi-type diagram will be proposed to make a partition of the space $\mathbb{R}^3$, which, in turn, will also be used to calculate the exact volume of the SAS-cavity.

### 1.4.1 Mathematical definitions

In [67], the SAS is defined by the set of the centers of the spherical probe when rolling over the VdW surface of the molecule. At first glance, one could think that it can equivalently be seen as the topological boundary of the union of all SAS-balls of the molecule. However, we notice that there might exist some inner holes inside the molecules where a solvent molecule can not be present. As a consequence, the SAS may or may not be composed of several separate surfaces, see Figure 3 for a
40 1.4. Solvent accessible surface

Figure 3: The left figure shows the components of the eSAS of the protein 1B17 (the probe radius $r_p = 1.2\text{Å}$): open spherical patches in blue, open circular arcs in yellow and intersection points in red. The right figure shows the exterior surface (transparent) and the interior surfaces of the eSES of the protein 1B17. The boundary of an exterior spherical patch of the eSAS is composed of circular arcs depicted in yellow, and the boundary of an interior spherical patch of the cSAS is composed of circular arcs depicted in purple.

graphical illustration. This inspires us to propose two more precise surfaces: the complete Solvent Accessible Surface (cSAS) defined simply as the boundary of the union of the SAS-balls, and the exterior Solvent Accessible Surface (eSAS) defined as the outmost surface obtained when a probe rolls over the exterior of the molecule. See Figure 4 for an example, where the cSAS is the union of SAS$_1$ and SAS$_2$, while the eSAS is just SAS$_1$. In addition, the cSES is the union of SES$_1$ and SES$_2$, while the eSES is just SES$_1$. Finally, $x_{iq}$ is a closest point on the SAS to a given point $p$. In the case where there are no interior holes inside the molecule, the cSAS and the eSAS will coincide. We make a convention that the SAS refers to both the cSAS and the eSAS in a general context.

Since both the cSAS and the eSAS are two closed sets, there exists a closest point on the SAS to any given point $p \in \mathbb{R}^3$, which is denoted by $x^p_{\text{sas}}$. Thus, the signed distance function $f_{\text{sas}}$ can be written as:

$$f_{\text{sas}}(p) = \begin{cases} -\|p - x^p_{\text{sas}}\| & \text{if } p \text{ lies inside the SAS,} \\ \|p - x^p_{\text{sas}}\| & \text{if } p \text{ lies outside the SAS}. \end{cases}$$

(1.4.1)

In the above formula, $x^p_{\text{sas}}$ depends on $p$. When $p$ lies on the SAS, $p$ coincides with $x^p_{\text{sas}}$ and $f_{\text{sas}}(p) = 0$. It remains therefore to find a closest point $x^p_{\text{sas}}$ on the SAS to the given point $p \in \mathbb{R}^3$. Note that there might exist more than one closest point on the SAS to the point $p$ and $x^p_{\text{sas}}$ is chosen as one of them. In the context, this is not
Figure 4: This is a 2D schematic of different molecular surfaces, including the VdW surface, the cSAS, the eSAS, the cSES and the eSES. The yellow discs denote the probes, representing solvent molecules, while the blue discs denote the VdW-balls. SAS\textsubscript{1} is the trace of the probe center when a probe rolls over the exterior of the VdW-balls, while SAS\textsubscript{2} is the trace of the probe center when a probe rolls over the inner holes of the VdW-balls. SES\textsubscript{1} and SES\textsubscript{2} respectively denote the corresponding solvent excluded surfaces to SAS\textsubscript{1} and SAS\textsubscript{2}.

an easy task. In the following, for the particular case of the SAS, we propose a way to calculate analytically a closest point on the SAS to a given point \( p \), based on three equivalence statements.

### 1.4.2 Equivalence statements

According to the definitions of the cSAS and the eSAS, these two molecular surfaces are both composed of three types of parts: open spherical patches, open circular arcs and intersection points (formed by the intersection of at least three SAS-spheres), see Figure 3. Note that an SAS intersection point can in theory be formed by the intersection of more than three SAS-spheres. However, these cases can be divided into multiple triplets of SAS-spheres for simplicity as mentioned in [59]. In this spirit, we make an assumption that all SAS intersection points are formed by the intersection of three SAS-spheres. Furthermore, we assume that any SAS-ball is not included by any other (otherwise, the inner SAS-ball can be ignored) In the following analysis.

For an SAS, denote by \( m_1 \) the number of the SAS spherical patches, by \( m_2 \) the number of the SAS circular arcs, and by \( m_3 \) the number of the SAS intersection points. Then, denote by \( P_m \) the \( m \)-th SAS spherical patch on the SAS where \( m = 1, \ldots, m_1 \). Denote by \( l_m \) the \( m \)-th SAS circular arc on the SAS where \( m = 1, \ldots, m_2 \). Denote by \( x_m \) the \( m \)-th SAS intersection point on the SAS where \( m = 1, \ldots, m_3 \). Furthermore,
denote by $I$ the set of all SAS intersection points written as:

$$I = \{x_m : m = 1, \ldots, m_3\} = \{x \in \text{SAS} : \exists 1 \leq i < j < k \leq M, \text{ s.t. } x \in S_i \cap S_j \cap S_k\}.$$ 

With the above notations, we consider to calculate a closest point on the SAS to a given point $p$ in the case when $p$ lies outside the SAS (the cSAS or the eSAS).

**Lemma 1.4.1.** Let the point $p$ lie outside the SAS (the cSAS or the eSAS), i.e. $\|p - c_i\| \geq r_i + r_p, \ \forall 1 \leq i \leq M$. Further, let $i_0 \in \{1, \ldots, M\}$ be such that

$$\|p - c_{i_0}\| - (r_{i_0} + r_p) = \min_{1 \leq i \leq M}\{\|p - c_i\| - (r_i + r_p)\}. \tag{1.4.2}$$

Then, the point

$$x_{i_0} = c_{i_0} + \frac{r_{i_0} + r_p}{\|p - c_{i_0}\|} (p - c_{i_0}), \tag{1.4.3}$$

is the closest point to $p$ on the SAS and

$$f_{\text{sas}}(p) = \|p - x_{i_0}\| = \min_{1 \leq i \leq M}\{\|p - c_i\| - (r_i + r_p)\}. \tag{1.4.4}$$

**Proof.** The proof involves basic geometric manipulations and is left to the reader. \qed

So far, we have discussed the case when $p$ lies outside the SAS (both the cSAS and the eSAS), which is not too difficult to deal with. Next, we need to consider the case where $p$ lies inside the (complete or exterior) SAS-cavity, to obtain the signed distance function $f_{\text{sas}}(p)$ from any point $p \in \mathbb{R}^3$ to the SAS. The following analysis can be applied to both the cSAS and the eSAS. This problem is handled inversely, in the sense that we will determine the region in the molecular cavity for an arbitrary given point $x_{\text{sas}} \in \text{SAS}$, such that $x_{\text{sas}}$ is a closest point to any point in this region. To do this, we first define a mapping $R : X \mapsto Y$, where $X$ is a subset of the SAS, and $Y = R(X)$ is the region in the SAS-cavity, such that there exists a closest point in $X$ on the SAS to any point in $Y$. That is,

$$Y = \{y : y \text{ lies in the SAS-cavity and } \exists x_{\text{sas}}^y \in X \text{ s.t. } x_{\text{sas}}^y \text{ is a closest point to } y\}.$$

In the following, we propose three equivalence statements between a point $x_{\text{sas}}$ on the SAS and the corresponding region $R(x_{\text{sas}})$ for three cases where $x_{\text{sas}}$ lies respectively on the three different types of the SAS. We recall first, however, a useful inequality between two signed distance functions to two surfaces.

**Proposition 1.4.1.** Consider two bounded, closed and oriented surfaces $S \subset \mathbb{R}^3$ and $S' \subset \mathbb{R}^3$ with two corresponding signed distance functions $f_S(p)$ and $f_{S'}(p)$. If the cavity inside $S$ is contained in the cavity inside $S'$, then we have $f_{S'}(p) \leq f_S(p), \ \forall p \in \mathbb{R}^3$. 

With the above proposition, we propose first a result which connects a point on an SAS spherical patch with a closed line segment in the SAS-cavity.

**Theorem 1.4.1.** Assume that \( p \in \mathbb{R}^3 \) is a point in the SAS-cavity and \( x_{\text{sas}} \) is a point on the SAS. If \( x_{\text{sas}} \) is on an SAS spherical patch \( P_m \) on the \( i \)-th SAS-ball \( S_i \), then \( x_{\text{sas}} \) is a closest point on the SAS to \( p \) if and only if \( p \) lies on the closed line segment \( [c_i, x_{\text{sas}}] \). That is, \( R(x_{\text{sas}}) = [c_i, x_{\text{sas}}] \). Further, the closest point \( x_{\text{sas}} \) on the SAS is unique if and only if \( p \neq c_i \). If \( p = c_i \), then any point on \( P_m \) is a closest point to \( p \).

*Proof.* First suppose that \( x_{\text{sas}} \) is a closest point to \( p \) with \( x_{\text{sas}} \in P_m \subset S_i \). Since \( P_m \) is an open set, take a small enough neighborhood \( V \) of \( x_{\text{sas}} \) such that \( V \subset P_m \), see Figure 1, and since \( x_{\text{sas}} \) is a closest point on the SAS to \( p \), we have

\[
\|p - x_{\text{sas}}\| \leq \|p - x\|, \quad \forall x \in V, \tag{1.4.5}
\]

which yields that the vector from \( x_{\text{sas}} \) to \( p \) is perpendicular to any vector in the tangent plane of \( S_i \) at \( x_{\text{sas}} \), thus the vector from \( p \) to \( x_{\text{sas}} \) is the normal vector at \( x_{\text{sas}} \) of \( S_i \). As a consequence, \( p \) must lie on the line passing through \( x_{\text{sas}} \) and the center \( c_i \) of \( S_i \). Furthermore, from the convexity of \( P_m \), \( p \) has to lie on the closed line segment \( [c_i, x_{\text{sas}}] \).

On the other hand, suppose that \( x_{\text{sas}} \in P_m \subset S_i \), and \( p \in [c_i, x_{\text{sas}}] \). In consequence, \( x_{\text{sas}} \) is obviously a closest point on the sphere \( S_i \) to \( p \), see Figure 2. The signed distance function \( f_{S_i}(p) \) is equal to \(-\|p - x_{\text{sas}}\|\). Notice that the cavity inside \( S_i \), i.e. \( B_i \), is contained in the SAS-cavity. We can then use Proposition 1.4.1 by taking \( S \) as \( S_i \) and \( S' \) as the SAS, to obtain \( f_{\text{sas}}(p) \leq f_{S_i}(p) = -\|p - x_{\text{sas}}\| \). Therefore, we have

\[
-\|p - x\| \leq f_{\text{sas}}(p) \leq -\|p - x_{\text{sas}}\|, \quad \forall x \in \text{SAS}. \tag{1.4.6}
\]

That is, \( \|p - x\| \geq \|p - x_{\text{sas}}\| \), \( \forall x \in \text{SAS} \), which means that \( x_{\text{sas}} \) is a closest point on the SAS to \( p \).

Finally, assume that \( p \in [c_i, x_{\text{sas}}] \). If \( p = c_i \), then any point on \( P_m \) is a closest point to \( p \) because the distance is uniformly \( \|c_i - x_{\text{sas}}\| = r_i + r_p \). If \( p \neq c_i \), then the open ball \( B_r(p) \) with \( r = \|p - x_{\text{sas}}\| < r_i + r_p \) is included in \( B_i \) and \( \partial B_r(p) \cap S_i = \{x_{\text{sas}}\} \), which implies that \( x_{\text{sas}} \) is the unique closest point to \( p \).

Next, we propose another equivalence statement, which connects a point on an SAS circular arc with a closed triangle in the SAS-cavity.

**Theorem 1.4.2.** Assume that \( p \in \mathbb{R}^3 \) is a point in the SAS-cavity and \( x_{\text{sas}} \) is a point on the SAS. If \( x_{\text{sas}} \) is on an SAS circular arc \( l_m \) associated with \( S_i \) and \( S_j \), then \( x_{\text{sas}} \) is a closest point on the SAS to \( p \) if and only if \( p \) lies in the closed triangle \( \triangle x_{\text{sas}}c_ic_j \) defined by three vertices \( x_{\text{sas}}, c_i \) and \( c_j \). That is, \( R(x_{\text{sas}}) = \triangle x_{\text{sas}}c_ic_j \).
Further, the closest point \( x_{\text{sas}} \) on the SAS is unique if and only if \( p \) does not belong to the edge \([c_i, c_j]\). If \( p \in [c_i, c_j]\), any point on \( l_m \) is a closest point to \( p \).

**Proof.** First suppose that \( x_{\text{sas}} \) is a closest point to \( p \) with \( x_{\text{sas}} \in l_m \subset S_i \cap S_j \). Since \( l_m \) is an open circle arc on the circle \( S_i \cap S_j \), take a small enough neighboring curve \( \gamma_0 \) of \( x_{\text{sas}} \) with \( \gamma_0 \subset l_m \), see Figure 6. Since \( x_{\text{sas}} \) is a closest point to \( p \), we have

\[
\|p - x_{\text{sas}}\| \leq \|p - x\|, \quad \forall x \in \gamma_0. \tag{1.4.7}
\]

Denote by \( p' \) the projection of \( p \) onto the plane where \( l_m \) lies. By substituting \( \|p - x_{\text{sas}}\|^2 = \|p - p'\|^2 + \|p' - x_{\text{sas}}\|^2 \) and \( \|p - x\|^2 = \|p - p'\|^2 + \|p' - x\|^2 \) into the inequality (1.4.7), we obtain that

\[
\|p' - x_{\text{sas}}\| \leq \|p' - x\|, \quad \forall x \in \gamma_0, \tag{1.4.8}
\]

which yields that the vector from \( x_{\text{sas}} \) to \( p' \) is perpendicular to the tangent vector of \( \gamma_0 \) at \( x_{\text{sas}} \). As a consequence, \( p' \) has to lie on the ray \( O x_{\text{sas}} \) starting from \( O \) and passing through \( x_{\text{sas}} \), where \( O \) is the center of the circular arc \( l_m \). Thus, \( p \) must lie on the closed half plane \( \Pi \) defined by the three points \( x_{\text{sas}}, c_i \) and \( c_j \) and whose borderer is the line passing through \( c_i \) and \( c_j \). The closed half plane \( \Pi \) contains the ray \( O x_{\text{sas}} \) and is perpendicular to the plane where \( l_m \) lies. We then take another two neighboring curves \( \gamma_1 \) and \( \gamma_2 \) of \( x_{\text{sas}} \), with \( \gamma_1 \subset \Pi \cap S_i \cap \text{SAS} \) and \( \gamma_2 \subset \Pi \cap S_j \cap \text{SAS} \), see Figure 6. In this case, \( \gamma_1 \) has one closest endpoint \( x_{\text{sas}} \) and is open at the other end, which is the same for \( \gamma_2 \). From the assumption that \( x_{\text{sas}} \) is a closest point on the SAS to \( p \), we have the following inequality

\[
\|p - x_{\text{sas}}\| \leq \|p - x\|, \quad \forall x \in \gamma_1 \cup \gamma_2. \tag{1.4.9}
\]

This yields that \( p \in \Delta x_{\text{sas}} c_i c_j \), where \( \Delta x_{\text{sas}} c_i c_j \) is the closed triangle on \( \Pi \) with three vertices \( x_{\text{sas}}, c_i \) and \( c_j \). Otherwise, we can find a point \( x \in (\gamma_1 \cup \gamma_2) \setminus x_{\text{sas}} \) strictly
closer to \( p \) than \( x_{\text{sas}} \), which contradicts the assumption.

On the other hand, suppose that \( p \in \Delta x_{\text{sas}}c_ic_j \). As a consequence, it is not difficult to obtain that \( x_{\text{sas}} \) is the closest point to \( p \) on \( \partial(B_i \cup B_j) \), where \( B_i \) and \( B_j \) are the corresponding SAS-balls corresponding to \( S_i \) and \( S_j \) as mentioned above. Similarly, we know that the signed distance function \( f_{\partial (B_i \cup B_j)}(p) \) is equal to \(-\|p - x_{\text{sas}}\|\), and notice that \( B_i \cup B_j \) is contained in the SAS-cavity. We can use again Proposition \[ \text{(1.4.1)} \] by taking \( S \) as \( \partial(B_i \cup B_j) \) and \( S' \) as the SAS, to obtain \( f_{\text{sas}}(p) \leq f_{\partial (B_i \cup B_j)}(p) = -\|p - x_{\text{sas}}\| \). Therefore, we have

\[
-\|p - x\| \leq f_{\text{sas}}(p) \leq -\|p - x_{\text{sas}}\|, \quad \forall x \in \text{SAS}. \tag{1.4.10}
\]

That is, \( \|p - x\| \geq \|p - x_{\text{sas}}\|, \forall x \in \text{SAS} \), which means that \( x_{\text{sas}} \) is the closest point on the SAS to \( p \).

If \( p \in [c_i, c_j] \), then any point on \( l_m \) is a closest point to \( p \) since the distance from any point on \( l_m \) to \( p \) is constant. If \( p \in \Delta x_{\text{sas}}c ICC_j \), then the open ball \( B_r(p) \) with \( r = \|p - x_{\text{sas}}\| < r_i + r_p \) is included in \( B_i \cup B_j \) and \( \partial B_r(p) \cap \partial (B_i \cup B_j) = \{x_{\text{sas}}\} \), which implies that \( x_{\text{sas}} \) is the unique closest point to \( p \).

By mapping with \( R \) a whole SAS spherical patch \( P_m \) to the SAS-cavity, we obtain a spherical sector \( R(P_m) \) in the SAS-cavity with cap \( P_m \), center \( c_i \) and radius \( r_i + r_p \), see Figure 5. Similarly, by mapping with \( R \) a whole SAS circular arc \( l_m \) to the SAS-cavity, we obtain a double-cone region \( R(l_m) \) in the SAS-cavity with the circular sector corresponding to \( l_m \) as base, \( c_i \) and \( c_j \) as vertices, see Figure 6.
Figure 7: This figure illustrates the closed region $T$ of three SAS-spheres with the centers $(c_i, c_j, c_k)$, where the tetrahedron $T$ has five vertices $(x_{sas}, c_i, c_j, c_k, x_I)$. Here, $x_{sas}$ is one SAS intersection point in $I$, and $x_I$ is the other SAS intersection point.

Removing now the above-mentioned spherical sectors and double-cone regions from the SAS-cavity, the closed hull of the remaining region, denoted by $T$, is consequently a collection of closed separate polyhedrons, see an example of three atoms in Figure [1]. Each polyhedron is denoted by $T_n$ with index $n$ and thus $T = \bigcup T_n$. Considering now the last case where $x_{sas}$ is an SAS intersection point $x_m$, we have a third equivalence statement as a corollary of the previous two equivalence statements.

**Corollary 1.4.1.** If $x_{sas}$ is an SAS intersection point $x_m \in I$ associated with $S_i$, $S_j$ and $S_k$, then $x_{sas}$ is a closest point on the SAS to $p$ if and only if $p$ lies in the closed region $T$ and $x_{sas}$ is a closest point in $I$ to $p$.

In other words, for an arbitrary intersection point $x_m \in I$, we have the formula of its corresponding region in the SAS-cavity as follows:

$$R(x_m) = T \cap \{p : \|p - x_m\| \leq \|p - x\|, \forall x \neq x_m, x \in I\}.$$  \hfill (1.4.11)

It is not difficult to find that $R(x_m)$ is a closed polyhedron. At the same time, mapping $I$ into the SAS-cavity with $R$, we obtain $R(I) = T$.

With the three equivalence statements as well as the above-defined map $R$, we obtain in fact a non-overlapping decomposition of the SAS-cavity, including spherical sectors, double-cone regions and polyhedrons. It should be emphasized that given a point $p \in \mathbb{R}^3$ contained in the SAS-cavity and knowing the region where $p$ lies, we can then calculate a closest point $x^p_{sas}$ on the SAS to $p$ according to this decomposition. The signed distance function $f_{sas}(p)$ can therefore be calculated analytically by the formula (1.4.11), which, in turn, will ultimately provide an implicit function of the SES. In the next subsection, we will present a partition of the SAS-cavity based on the above equivalence statements, in order to compute efficiently the area of the SAS and the volume of the SAS-cavity.
1.4.3 New Voronoi-type diagram

The above non-overlapping decomposition of the SAS-cavity can also be seen as a new Voronoi-type diagram, which will be presented comparing with the well-known Voronoi Diagram and the Power Diagram recalled in the following.

1.4.3.1 Voronoi diagram

The Voronoi diagram was initially a partition of a 2D plane into regions based on the distance to points in a specific subset of the plane. In the general case, for an Euclidean subspace $X \subset \mathbb{R}^n$ endowed with a distance function $d$ and a tuple of nonempty subsets $(A_i)_{i \in K}$ in $X$, the Voronoi region $R_i$ associated with $A_i$ is the set of all points in $X$ whose distance to $A_i$ is not greater than their distance to any other set $A_j$ for $j \neq i$. In other words, with the distance function between a point $x$ and a set $A_i$ defined as

$$d_V(x, A_i) = \inf_{y \in X} \{d(x, y) \mid y \in A_i\},$$

the formula of the Voronoi region $R_i$ is then given by:

$$R_i = \{x \in X \mid d_V(x, A_i) \leq d_V(x, A_j), \forall j \neq i\}.$$  \hfill (1.4.12)

Most commonly, each subset $A_i$ is taken as a point and its corresponding Voronoi region $R_i$ is consequently a polyhedron, see an example of three points in $\mathbb{R}^2$ in Figure 8.

1.4.3.2 Power diagram

In computational geometry, the power diagram, also called the Laguerre-Voronoi diagram, is another partition of a 2D plane into polygonal cells defined from a set of circles in $\mathbb{R}^2$. In the general case, for a set of circles $(C_i)_{i \in K}$ (or spheres) in $\mathbb{R}^n$ with $n \geq 2$, the power region $R_i$ associated with $C_i$ consists of all points whose power distance to $C_i$ is not greater than their power distance to any other circle $C_j$, for $j \neq i$. The power distance from a point $x$ to a circle $C_i$ with center $c_i$ and radius $r_i$ is defined as

$$d_P(x, C_k) = \|x - c_k\|^2 - r_k^2,$$

the formula of the Power region $R_i$ is then given by:

$$R_i = \{x \in \mathbb{R}^n \mid d_P(x, C_i) \leq d_P(x, C_j), \forall j \neq i\}.$$  

The power diagram is a form of generalized Voronoi diagram, in the sense that you can take the circles $C_i$ instead of the centers $c_i$ and simply replace the distance function $d_V$ in the Voronoi diagram with the power distance function $d_P$, to obtain the power diagram, see an example of three circles in $\mathbb{R}^2$ in Figure 8. Notice that the power
distance is not a real distance function. By summing up the volume of each power region inside the SAS-cavity, one can calculate the exact volume of the cSAS-cavity, which is equivalent to calculate the volume of the union of balls, see [27, 27].

1.4.3.3 New Voronoi-type diagram

In this section, we propose a new Voronoi-type diagram for a set of spheres in $\mathbb{R}^3$ (or circles in $\mathbb{R}^2$), which is inspired by the non-overlapping decomposition of the SAS-cavity. We will see that this new diagram allows us to calculate exactly not only the volume of the (complete or exterior) SAS-cavity, but also the (complete or exterior) SES-cavity which will be defined in the next section.

We first look at the new Voronoi-type diagram for a set of discs in the 2D case.

Notice that the boundary $\gamma$ of the union of these discs can be classified into two types: open circular arcs $\{l_1, l_2, \ldots, l_n\}$ and intersection points $\{x_1, x_2, \ldots, x_n\}$, with the number of circular arcs equal to the number of intersection points. Take $A_1 = l_1, \ldots, A_n = l_n$, $A_{n+1} = \{x_1, x_2, \ldots, x_n\}$ in the Voronoi diagram. As a consequence, we obtain $n + 1$ corresponding new Voronoi-type regions $\{R_1, \ldots, R_n, R_{n+1}\}$, where $R_i$ is given by (1.4.12).

From a similar mapping $R$ and similar equivalence theorems, we know that the part of $R_i$ inside $\gamma$ is a circular sector when $1 \leq i \leq n$, and $R_{n+1}$ is the remaining region composed of polygons, see an example of three circles in Figure 8. For any point $x \in \mathbb{R}^2$, we have $x \in R_i$ if and only if there exists a point in $A_i$ such that it is a closest point to $x$ on $\gamma$.

In the 3D case, the SAS consists of three types of geometrical quantities: open spherical patches, open circular arcs and intersection points. Similarly to the case in 2D, we take $A_1 = P_1, \ldots, A_{m_1} = P_{m_1}$, $A_{m_1+1} = l_1, \ldots, A_{m_1+m_2} = l_{m_2}$ and $A_{m_1+m_2+1} = \{x_1, \ldots, x_{m_3}\}$ with the above-mentioned notations. As a consequence, we obtain $m_1 + m_2 + 1$ corresponding new Voronoi-type regions $\{R_1, \ldots, R_{m_1}, R_{m_1+1}, \ldots, R_{m_1+m_2}, R_{m_1+m_2+1}\}$.

With the mapping $R$ defined in the last section, we know that the spherical sector $R(P_m)$ corresponds to $R_m$ in the SAS-cavity, the double-cone region $R(l_m)$ corresponds to $R_{m_1}+m$ in the SAS-cavity, and the closed region $R(I)$ coincides with $R_{m_1+m_2+1}$. For any point $x \in \mathbb{R}^3$, we have $x \in R_i$ if and only if there exists a point in $A_i$ such that it is a closest point on the SAS to $x$. The new Voronoi-type diagram is a powerful tool for calculating analytically the molecular area and volume, which is a direct consequence of the three equivalence statements. Given the components of the SAS, the new Voronoi-type diagram can be obtained directly according to the three equivalence statements, while the power diagram needs more complicated computations.
1.4.4 SAS-area and SAS-volume

In this section, the area of the SAS is called the SAS-area and the volume of the SAS-cavity is called the SAS-volume. Similarly, the area of the SES is called the SES-area and the volume of the SES-cavity is called the SES-volume. Next, we will use the Gauss-Bonnet theorem of differential geometry to calculate the SAS-area, and then will use the new Voronoi-type diagram to calculate the SAS-volume.

1.4.4.1 SAS-Area

To calculate analytically the solvent accessible area, the Gauss-Bonnet theorem has already been applied in the 1980s [30], which allows one to calculate the area of each SAS spherical patch with the information along its boundary, and then sum them up. We introduce briefly the key formula used in this calculation to keep the chapter complete. The area of a spherical patch $P$ can be obtained from the Gauss-Bonnet theorem as follows (see Figure 5):

$$\sum \alpha_i + \sum k_{e_i}|e_i| + \frac{1}{r^2}A_P = 2\pi \chi, \quad (1.4.13)$$

where $\alpha_i$ is the angle at the vertex $v_i$ on the boundary between two neighboring circular arcs $e_{i-1}$ and $e_i$, $k_{e_i}$ is the geodesic curvature of the edge $e_i$, $|e_i|$ is the length of this edge $e_i$ and $A_P$ is the area of the patch $P$. Finally, $\chi$ is the Euler characteristic of $P$, which is equal to 2 minus the number of loops forming the boundary of $P$. Through the above formula, we calculate the area of each SAS spherical patch $A_{P_m}$.
and sum them up to get the SAS-area $A_{\text{sas}}$:

$$A_{\text{sas}} = \sum_{m=1}^{m_1} A_{P_m},$$  \hspace{1cm} (1.4.14)

### 1.4.4.2 SAS-volume

In the new Voronoi-type diagram, the SAS-cavity is decomposed into three types of regions: spherical sectors, double-cone regions and polyhedrons. For a spherical patch $P_m$ on $S_i$ with center $c_i$ and radius $r_i + r_p$, the volume of the corresponding spherical sector $R(P_m)$, denoted by $V_{P_m}$, can be calculated as follows:

$$V_{P_m} = \frac{1}{3} A_{P_m} (r_i + r_p).$$

For a circular arc $l_m$ associated with $S_i$ and $S_j$ (see Figure 1), the volume of the corresponding double-cone region $R(l_m)$, denoted by $V_{l_m}$, can be calculated as follows:

$$V_{l_m} = \frac{1}{6} r_{l_m} |l_m| \|c_i - c_j\|,$$

where $r_{l_m}$ is the radius of this circular arc and $|l_m|$ is the length of $l_m$. Finally, the volume of the closed region $T = R(I)$ can be calculated according the Gauss-Green theorem [38]:

$$V_T = \frac{1}{3} \sum_{t} A_t \ (n_t \cdot x_t),$$

where $t$ denotes each triangle on the boundary of $T$, $A_t$ is the area of the triangle $t$, $n_t$ is the outward pointing normal vector of $t$ and $x_t$ is an arbitrary point on the triangle $t$, for example, its vertex.

From the above three formulas, we sum up the volume of each spherical sector, each double-cone region and the polyhedron $T$, to get the solvent accessible volume $V_{\text{sas}}$ as below:

$$V_{\text{sas}} = \frac{1}{3} \sum_{m=1}^{m_1} A_{P_m}(r_i + r_p) + \frac{1}{6} \sum_{m=1}^{m_2} r_{l_m} |l_m| \|c_i - c_j\| + \frac{1}{3} \sum_{t} A_t \ (n_t \cdot x_t).$$  \hspace{1cm} (1.4.15)

### 1.5 Solvent excluded surface

The Solvent Excluded Surface was first proposed by Lee & Richards in the 1970s [99]. Although the SES is believed to give a more accurate description of the molecular cavity, it is more complicated than the other two molecular surfaces. We first give two more precise definitions of the SES considering inner holes of a molecule. After
that, we define mathematically different types of patches on the SES, which helps us to characterize and calculate analytically singularities on the SES. Finally, combining the above-proposed new Voronoi-type diagram and the upcoming singularity analysis, we calculate the exact SES-area and the SES-volume.

1.5.1 Mathematical definitions

In previous work [67, 99], the SES is defined as the topological boundary of the union of all possible spherical probes that do not intersect any VdW atom of the molecule. In other words, the SES is the boundary of the cavity where a spherical probe can never be present. However, there might exist inner holes inside the solute molecule. Similar to the definitions of the cSAS and the eSAS, the complete Solvent Excluded Surface (cSES) is defined as the set of all points with signed distance \(-r_p\) to the cSAS, while the exterior Solvent Excluded Surface (eSES) is defined as the set of all points with signed distance \(-r_p\) to the eSAS. We make a similar convention as before that the SES refers to both the cSES and the eSES. From the geometrical relationship between the SAS and the SES, i.e. \(\text{SES} = f^{-1}(r_p)\), we propose an implicit function of the SES:

\[
f_{\text{ses}}(p) = \begin{cases} 
-\|p - x_p^{\text{sas}}\| + r_p & \text{if } f_{\text{sas}}(p) \leq 0, \\
\|p - x_p^{\text{sas}}\| + r_p & \text{if } f_{\text{sas}}(p) \geq 0, 
\end{cases}
\]

where \(x_p^{\text{sas}}\) is a closest point on the SAS to \(p\), which depends on \(p\) and can be obtained directly from the equivalence statements.

In Connolly’s work [30], the SES is divided into three types of patches: convex spherical patches, saddle-shaped toroidal patches and concave spherical triangles, see Figure 9. The convex spherical patches are the remainders of VdW-spheres, which occur when the probe is rolling over the surface of an atom and touches no other atom. The toroidal patches are formed when the probe is in contact with two atoms at the same time and rotates around the axis connecting the centers of these two atoms. While rolling, the probe traces out small circular arcs on each of the two VdW-spheres, which build the boundaries between the convex spherical patches and the toroidal patches. The concave spherical triangles occur if the probe is simultaneously in contact with more than or equal to three VdW-spheres. Here, the probe is in a fixed position, meaning that it is centered at an SAS intersection point and cannot roll without losing contact to at least one of the atoms.

The singularities might appear on the SES, referred to as SES-singularities, see Figure 10. Despite some particular cases which have been studied by Sanner [102], a characterization of these singularities is not known. We will provide a complete characterization, using the above equivalence statements as well as the following analysis of the SES-singularities. To start with, we give three new definitions of different SES patches from the mathematical point of view:
1.5. Solvent excluded surface

Figure 9: The SAS and the SES of caffeine with probe radius $r_p = 1.5\text{"Å"}$. On the left, the SAS is composed of spherical patches in blue, circular arcs in yellow and intersection points in red. On the right, the patches in red (resp. in yellow or in blue) are convex spherical patches (resp. toroidal patches or concave spherical patches) on the SES, each corresponding to a spherical patch (resp. a circular arc or an intersection point) on the SAS.

1) **Convex Spherical Patch**: A convex spherical patch on the SES, denoted by $P_+$, is defined as the set of the points on the SES such that there exists a closest point on the SAS belonging to a common SAS spherical patch $P_m$, where $1 \leq m \leq m_1$.

2) **Toroidal Patch**: A toroidal patch on the SES, denoted by $P_t$, is defined as the set of the points on the SES such that there exists a closest point on the SAS belonging to a common SAS circular arc $l_m$, where $1 \leq m \leq m_2$.

3) **Concave Spherical Patch**: A concave spherical patch on the SES, denoted by $P_-$, is defined as the set of the points on the SES such that there exists a common SAS intersection point $x_m$ which is a closest point on the SAS to each point on $P_-$, where $1 \leq m \leq m_3$.

According to these new definitions, the three types of patches can be rewritten mathematically as follows:

\[
\begin{align*}
P_+ &= \{ p : f_{ses}(p) = 0, \ p \in R(P_m) \}, \ 1 \leq m \leq m_1 \\
P_t &= \{ p : f_{ses}(p) = 0, \ p \in R(l_m) \}, \ 1 \leq m \leq m_2 \\
P_- &= \{ p : f_{ses}(p) = 0, \ p \in R(x_m) \}, \ 1 \leq m \leq m_3
\end{align*}
\]

where $f_{ses}$ is the implicit function of the SES and $R$ is the mapping defined in the previous section. Actually, a convex spherical patch and a toroidal patch are defined in the same way as Connolly, while a new-defined concave patch might not be triangle-shaped because its new definition takes into account the intersection among different SES patches. The above new definitions of the SES patches ensure that different
Figure 10: Schematics of a rectangle-shaped toroidal patch (left, yellow) and a double-triangle-shaped toroidal patch (right, yellow) corresponding to two circular SAS circles respectively with radius larger than and smaller than \( r_p \).

SES patches will not intersect with each other, for the reason that different patches belong to different new Voronoi-type regions in the new Voronoi-type diagram.

1.5.2 SES-singularities

Before characterizing the singularities on the SES, it is necessary to recall the properties of the signed distance function \( f_S \) to a surface \( S \) in \( \mathbb{R}^n \) (\( n \geq 1 \)) as below:

1) \( f_S \) is differentiable almost everywhere, and it satisfies the Eikonal Equation:
\[ |\nabla f_S| = 1. \]
2) If \( f_S \) is differentiable at a point \( p \in \mathbb{R}^n \), then there exists a small neighborhood \( V \) of \( p \) such that \( f_S \) is differentiable in \( V \).
3) For any point \( p \in \mathbb{R}^n \), \( f_S \) is non-differentiable at \( p \) if and only if the number of the closest points on \( S \) to \( p \) is greater than or equal to 2.

We call a point \( x_{ses} \in \text{SES} \) a singularity if the SES is not smooth at \( x_{ses} \), which means that its implicit function \( f_{ses}(p) = f_{sas}(p) + r_p \) is non-differentiable at \( x_{ses} \) in \( \mathbb{R}^3 \). If \( x_{ses} \in \text{SES} \) is a singularity, we obtain consequently that \( f_{sas} \) is non-differentiable at \( x_{ses} \). From the last property above, we therefore can characterize the SES-singularities by the following equivalence.

**Corollary 1.5.1.** A point \( x_{ses} \in \text{SES} \) is a singularity if and only if the number of the closest points on the SAS to \( x_{ses} \) is greater than or equal to 2.

We now investigate the different types of singularities that can appear on each of the three patch types. From the definitions of different SES patches, it is relatively easy to calculate the convex spherical patches and the toroidal patches given the components of the SAS. In the following, we illustrate how to calculate \( P_- \) exactly, which provides therefore a complete characterization of the SES-singularities later. Denote by \( P_0 \) the concave spherical triangle \((P_- \subset P_0)\) corresponding to an SAS
intersection point $x_m$ showed in Figure 11 and by $K$ the set of all intersection points in $I$ with distances less than $2r_p$ to $x_m$, that is,

$$K = \{ x \in I : \| x - x_m \| < 2r_p, \ x \neq x_m \},$$

so that $K$ collects all SAS intersection points "near" $x_m$. According to the definition of $P_-$ corresponding to $x_m$ and the equation (1.5.1), we have the following formula:

$$P_- = P_0 \cap R(x_m) = P_0 \cap T \cap \{ p : \| p - x_m \| \leq \| p - x \|, \ \forall x \neq x_m, \ x \in I \}. \quad (1.5.3)$$

The above formula can be used to calculate $P_-$ directly, in which $R(x_m)$ is a polyhedron. However, this formula is not convenient to calculate, since one has to calculate $T$ as well as the union of $P_0$ and $R(x_m)$. This motivates us to look deep into the relationship between $P_-$ and $P_0$. In the following theorem, we will present a simpler formula of $P_-$, which allows us to calculate it analytically and more efficiently.

**Theorem 1.5.1.** $P_- = P_0 \setminus \bigcup_{x \in K} B_{r_p}(x)$ where $B_{r_p}(x)$ denotes the open ball (or disc in $\mathbb{R}^2$) centered at $x$ with radius $r_p$.

The proof of Theorem 1.5.1 is given in the Appendix A. From the definition of $P_-$, it is not surprising to have the inclusion $P_- \subset P_0 \setminus \bigcup_{x \in K} B_{r_p}(x)$. However, the above theorem gives a stronger result that $P_-$ and $P_0 \setminus \bigcup_{x \in K} B_{r_p}(x)$ are identical. This means that the concave spherical patch can be obtained from $P_0$ by removing the parts intersecting other "nearby" spherical probes centered at SAS intersection points. This theorem also allows us to characterize the singular circular arcs on the concave spherical patch $P_-$. We propose a theorem of the SES-singularities as follows.

**Theorem 1.5.2.** The following statements hold:
(1) There can not exist any singularity on a convex spherical SES patch \( P_+ \).

(2) On a toroidal patch \( P_t \), two point-singularities occur when the corresponding SAS circular arc has a radius smaller than \( r_p \) and they can be computed following the sketch in Figure 13.

(3) On a concave spherical SES patch \( P_- \), singular arcs occur when \( P_- \) does not coincide with the corresponding concave spherical triangle \( P_0 \). Further, these singular arcs form the boundary of \( P_- \) that does not belong to the boundary of \( P_0 \).

Proof. First, if \( x_{ses} \) is a point on a convex spherical SES patch, then it has a closest point \( x_{sas} \) on an spherical SAS patch. Moreover, \( x_{sas} \) is the unique closest point to \( x_{ses} \) from Theorem 1.4.1. This, in turn, implies by Corollary 1.5.1 that \( x_{ses} \) can not be a singularity. As a consequence, there can not exist any singularity on a convex spherical SES patch \( P_+ \).

Second, if \( x_{ses} \) is a point on a toroidal SES patch, then it has a closest point \( x_{sas} \) on an SAS circular arc \( l_m \) associated with two SAS-spheres \( S_i \) and \( S_j \). Moreover, \( x_{sas} \) is not unique if and only if \( x_{ses} \) belongs to \( [c_i, c_j] \) by Theorem 1.4.2, which happens only when the radius of \( l_m \) is smaller than \( r_p \) and \( x_{ses} \) is one of two cusps on the toroidal SES patch as showed in Figure 13. By Corollary 1.5.1, two point-singularities on the toroidal patch \( P_t \) can only occur when the corresponding SAS circular arc has a radius smaller than \( r_p \).

Third, if \( x_{ses} \) is a point on a concave spherical SES patch \( P_- \) corresponding to an SAS intersection point \( x_m \), then \( x_{sas} = x_m \) is a closest point to \( x_{ses} \). If \( x_{ses} \) belongs to the boundary of \( P_- \) but not to the boundary of \( P_0 \), then by the formula characterizing \( P_- \) in Theorem 1.5.1 we know that \( x_{ses} \) lies on another nearby probe \( \partial B_{r_p}(x) \) where \( x \in K \) (and thus \( x \neq x_m \)). As a consequence, \( x \) is another closest point to \( x_{ses} \) implying that \( x_{ses} \) is singularity by Corollary 1.4.1. On the other hand, if \( x_{ses} \in P_- \) but does not belong to \( \bigcup_{x \in K} B_{r_p}(x) \), then \( x_{ses} \) does not lie on any nearby probe and \( x_{sas} \) is the unique closest point among all SAS intersection points. Assume by contradiction that there exist another closest point \( x \) to \( x_{ses} \) on some SAS spherical patch \( P_m \). Since now by Theorem 1.4.1 the closest point to any point on \( (c_i, x] \) on the SAS is unique, this implies that \( x_{ses} = c_i \) (because \( x_m \) is another closest point). This is however a contradiction since \( x_{ses} \) can not coincide with the center of any SAS-sphere as all the VdW-radii are assumed to be positive. Further, assume by contradiction that there exist a closest point \( x \) to \( x_{ses} \) on some SAS circular arc \( l_m \). Consequently, \( x_{ses} \) lies on the corresponding toroidal patch. According to Theorem 1.4.2, \( x_{ses} \) belongs to \( [c_i, c_j] \). If \( r_p < r_{l_m} \), then \( x_{ses} \) can not belong to the SES which is a contradiction. If \( r_p \geq r_{l_m} \), then \( x_{ses} \) has to be one of the two cusps. In this case, the two ending points of \( l_m \) are both closest points to \( x_{ses} \) which are also SAS intersection points. This conflicts with the fact that \( x_{sas} = x_m \) is the unique closest point among all SAS intersection points. \( \blacksquare \)
Remark 1.5.1. In the third case of Theorem 1.5.2, \( P_- \) can be calculated according to Theorem 1.5.1. It is obvious that \( P_- \) does not coincide with \( P_0 \) if and only if \( P_0 \cap \left( \bigcup_{x \in \mathcal{K}} B_{r_p}(x) \right) \) is nonempty.

Finally, we state a corollary about classifying all points on the SES into four classes according to the number of its closest points on the SAS.

**Corollary 1.5.2.** For any point \( x_{\text{ses}} \in \text{SES} \), assume that the number of its closest points on the SAS is \( N \) (denoting \( N = \infty \) for an infinite number of closest points). Then, there exists four cases:

1. \( N = 1 \): \( x_{\text{ses}} \) is not a singularity on the SES, and \( x_{\text{ses}} \) has an unique closest point \( \{ x_{\text{sas}}^1 \} \) on the SAS.
2. \( 2 \leq N < \infty \): \( x_{\text{ses}} \) is a singularity on a concave spherical SES patch, and its closest points \( \{ x_{\text{sas}}^1, \ldots, x_{\text{sas}}^N \} \) are among the SAS intersection points.
3. \( N = \infty \): \( x_{\text{ses}} \) is a singularity on a toroidal patch corresponding to an SAS circular arc (or a complete circle) on which any point is a closest point on the SAS.

### 1.5.3 SES-area and SES-volume

With the new Voronoi-type diagram as well as the above singularity analysis we can calculate the SES-area and the SES-volume analytically, which shall be explained in the following.

#### 1.5.3.1 SES-area

In Connolly’s paper [30], the presence of singularities on the SES concave spherical patches made it infeasible to calculate the SES-area exactly. The characterization of the singularities carried out earlier in this chapter allows us to calculate the area of each SES patch and sum them up to obtain the exact area of the whole SES. To calculate the area of a convex spherical patch \( P_+ \), we use a similar Gauss-Bonnet formula as (1.4.13):

\[
\sum_v \alpha_v + \sum_e k_e |e| + \frac{1}{r_+^2} A_{P_+} = 2\pi \chi, \tag{1.5.4}
\]

where \( \alpha_v \) denotes the angle at a vertex \( v \) between neighboring circular arcs on the boundary of \( P_+ \), \( k_e \) the geodesic curvature of an edge \( e \) on the boundary of \( P_+ \), \( A_{P_+} \) is the area of \( P_+ \), and \( \chi \) is the Euler characteristic of \( P_+ \).

To calculate the area of a toroidal patch \( P_t \) analytically, we consider two cases, and suppose that \( P_t \) corresponds to an SAS circular arc \( l_m \) with radius \( r_{l_m} \). In the
Figure 12: The yellow patch is a toroidal patch $P_t$ on the SES corresponding to an SAS circular arc $l_m$ with the radius $r_m > r_p$ and the radian $\beta_m$. $\theta_1$ denotes the angle between the line connecting $c_i$ with a point on $l_m$ and the disc where $l_m$ lies. Similarly, $\theta_2$ denotes the angle between the line connecting $c_j$ with the same point on $l_m$ and the disc where $l_m$ lies.

Figure 13: The two yellow parts form a toroidal patch $P_t$ on the SES corresponding to an SAS circular arc $l_m$ with the radius $r_m < r_p$ and the radian $\beta_m$. $\theta_0$ denotes the angle between the line connecting a singularity on $P_t$ with a point on $l_m$ and the disc where $l_m$ lies and $\theta_1, \theta_2$ are defined as in Fig. 12.

In the case where $r_m > r_p$, there will be no singularity on $P_t$. With the notations introduced in Figure 12, we therefore have the following formula for calculating the area of $P_t$:

$$A_{P_t} = r_p \beta_m \{ r_m (\theta_1 + \theta_2) - r_p (\sin \theta_1 + \sin \theta_2) \}. \quad (1.5.5)$$

In the case where $r_m \leq r_p$, there are two singular points on $P_t$. With the notations introduced in Figure 13, we have the following formula for calculating the area of $P_t$:

$$A_{P_t} = r_p \beta_m \{ r_m (\theta_1 + \theta_2 - 2\theta_0) - r_p (\sin \theta_1 + \sin \theta_2 - 2\sin \theta_0) \}. \quad (1.5.6)$$

To calculate analytically the area of a concave spherical patch $P_-$, we use the Gauss-Bonnet theorem on the spherical probe again, see Figure 14. In the previous section, we have obtained that $P_-=P_0 \setminus \bigcup_{x \in K} B_{r_p}(x)$, which implies that all informa-
tion about the boundary of $P_-$ is known. It allows us to apply the Gauss-Bonnet theorem on the spherical probe, to obtain:

$$\sum_v \alpha_v + \sum_e k_e |e| + \frac{1}{r_p^2} A_{P_-} = 2\pi \chi,$$

(1.5.7)

where $\alpha_v$ denotes the angle at a vertex $v$ on the boundary of $P_-$, $k_e$ the geodesic curvature of an edge $e$ on the boundary of $P_-$, $A_{P_-}$ is the area of $P_-$ and $\chi$ is the Euler characteristic of $P_-$. 

In summary, the area of each SES patch can be calculated independently and analytically, and we can then sum them up to obtain the exact SES-area $A_{\text{ses}}$.

### 1.5.3.2 SES-volume

According to the new Voronoi-type diagram, we can calculate the exact SES-volume. We propose to subtract the volume of the region between the SAS and the SES from the SAS-volume to obtain the SES-volume. This region that needs to be subtracted and which is denoted by $R_s$ can be characterized as below

$$R_s = \{ p : -r_p \leq f_{\text{sas}}(p) \leq 0 \}.$$ 

From the new Voronoi-type diagram, we decompose $R_s$ into small regions, each of which corresponds to a spherical patch $P_m$, a circular arc $l_m$ or an intersection point $x_m$: $R_s \cap R(P_m)$, $R_s \cap R(l_m)$ and $R_s \cap R(x_m)$. The volume of the region $R_s \cap R(P_m)$ is given by

$$V_{R_s \cap R(P_m)} = \frac{1}{3} A_{P_m} (r_i + r_p) \left( 1 - \frac{r_i^3}{(r_i + r_p)^3} \right),$$

(1.5.8)

where $A_{P_m}$ is the area of the SAS spherical patch $P_m$, $r_i + r_p$ is the radius of the corresponding SAS-sphere $S_i$ on which $P_m$ lies. For $R_s \cap R(l_m)$, denote by $r_{lm}$ the radius of $l_m$ and by $\beta_m$ the radian of $l_m$. In the case where $r_{lm} > r_p$, using the notations of Figure 12, the volume of the region $R_s \cap R(l_m)$ is given by

$$V_{R_s \cap R(l_m)} = \beta_m r_p^2 \left[ \frac{r_{lm}}{2} (\theta_1 + \theta_2) - \frac{r_p}{3} (\sin \theta_1 + \sin \theta_2) \right].$$

(1.5.9)

In the case where $r_{lm} \leq r_p$, using the notations of Figure 13, it is given by

$$V_{R_s \cap R(l_m)} = \beta_m r_p^2 \left[ \frac{r_{lm}}{2} (\theta_1 + \theta_2 - 2\theta_0) - \frac{r_p}{3} (\sin \theta_1 + \sin \theta_2 - 2 \sin \theta_0) \right] + \frac{1}{3} \beta_m r_{lm}^2 \sqrt{r_p^2 - r_{lm}^2}.$$ 

(1.5.10)

Consider now $R_s \cap R(x_m)$ corresponding to a concave spherical patch $P_-$. Notice that there might be some flat regions on the boundary of $R_s \cap R(x_m)$, caused by the intersection of the probe $B_{r_p}(x_m)$ and its nearby probes, see Figure 14 (right). Denote by $D_i$ the $i$-th flat region with the boundary composed of line segments and
Chapter 1: Mathematical analysis and calculation of molecular surfaces

Figure 14: On the left, the concave spherical triangle $P_0$ with vertices $(a_1, a_2, a_3)$ corresponds to an intersection point $x_m$. In the case where $P_-$ coincide with $P_0$, there will be no singularity on the concave spherical patch. On the right, the concave spherical patch $P_-$ does not coincide with $P_0$ and there are singular circular arcs as parts of its boundary. The vertices of $P_-$ are $(a_1, a_2, a_3, a_4, b_1, b_2, a_5)$. The two flat grey regions $D_1$ and $D_2$ are formed by the intersection of $P_0$ with two other nearby spherical probes. These two regions have the boundaries composed of line segments and circular arcs.

circular arcs. Furthermore, denote by $d_i$ the distance from $x_m$ to the plane where $D_i$ lies, and by $A_{D_i}$ the area of $D_i$. Then, the volume of the region $R_s \cap R(x_m)$ can be formulated as:

$$V_{R_s \cap R(x_m)} = \frac{1}{3} A_{P_-} r_p + \sum_i \frac{1}{3} A_{D_i} d_i,$$

where, $A_{P_-}$ is the area of the concave spherical SES patch $P_-$. Finally, by summing up the volume of each subtracted region, we obtain the subtracted volume as follows:

$$V_{R_s} = \sum_{\xi=P_m, l_m, x_m} V_{R_s \cap R(\xi)}.$$ 

Therefore, the SES-volume denoted by $V_{ses}$ is equal to $V_{sas} - V_{R_s}$.

1.6 Construction of molecular surfaces

In the above sections, we have defined and analyzed the cSAS and the eSAS, as well as the cSES and the eSES. However, all work is based on the explicit knowledge of the components of these molecular surfaces. In this section, we will present a method of constructing the cSAS and the eSAS using a binary tree to construct its spherical patches. After that, we will give a brief construction strategy of the cSES and eSES based on the construction of the cSAS and the eSAS. The construction of molecular surfaces ensures that our previous analysis is feasible.
1.6.1 Construction of the cSAS and the eSAS

To start the construction, we need some quantities for representing different components of the SAS. First, an intersection point will be represented by its coordinate in $\mathbb{R}^3$ and an identifier. Second, to represent a circular arc, we use its starting and ending intersection points (resp. the identifiers), its radius, center and radian as well as an identifier. Finally, to represent a spherical patch, we use the SAS-sphere on which it lies on and the identifiers of all circular arcs forming its boundary. Then, we propose to construct the SAS in five basic steps as below:

1) Compute the set of all intersection points $\{x_1, \cdots, x_m\}$, denoted by $I$.

2) Calculate each SAS circular arc or circle $l_m$ associated with some $S_i$ and $S_j$. For each SAS circular arc or circle, we record the information, including the center, the radius, the radian, the corresponding two SAS spheres, the identifiers of the starting and ending intersection points. Notice that each circular arc connects two intersection points.

3) Construct all loops on each SAS sphere $S_i$, which also form the boundaries of SAS spherical patches on $S_i$. Notice that each loop is composed of circular arcs, or a complete circle. We start from a circular arc on $S_i$, find another arc connecting this arc, add it into the loop, and repeat this until we finally obtain a complete loop. The $k$-th loop on the $i$-th SAS sphere is denoted by $L^i_k$.

4) Construct all spherical patches on each SAS sphere $S_i$. Since the boundary of a spherical patch on $S_i$ is composed by one or several loops on $S_i$, we can use the identifiers of these loops to represent a spherical patch. Denote by $P^i_k$ the $k$-th spherical patch on $S_i$. The difficulty lies on determining whether two loops on $S_i$ belong to the boundary of a common spherical patch or not. This final question will be discussed in the next section.

5) Finally, we should distinguish the cSAS and the eSAS. The cSAS is just the set of all above-constructed SAS spherical patches. To construct the eSAS, we say that two spherical patches $P^i_k$ and $P^j_l$ are neighbors if they have a common circular arc or circle on their boundaries. Then, we start our construction of the eSAS, by mapping a faraway point $p_\infty$ onto an SAS spherical patch, which is the initial patch on the eSAS. Then, we add the neighboring spherical patches into the eSAS one by one, to finally obtain the whole eSAS.

1.6.2 Binary tree to construct spherical patches

In the above construction process, there remains a problem of classifying the loops on an SAS-sphere into several parts, such that each part forms the boundary
Chapter 1: Mathematical analysis and calculation of molecular surfaces

Figure 15: On the left is a brief schematic of an SAS-sphere and the loops on it. There are six loops on the SAS-sphere \( \{L_1, L_2, L_3, L_5, L_6, L_6^b\} \), and three spherical patches with the boundaries formed by two loops in green \( \{L_4, L_5\} \), three loops in red \( \{L_1, L_3, L_6\} \) and one loop (circle) in blue \( \{L_2\} \) respectively. The tree on the right illustrates the corresponding binary tree whose leaves identify the boundaries of three different spherical patches.

of a spherical patch. To do this, we need to determine whether two loops on the SAS-sphere belong to the boundary of a same spherical patch or not. Note that two different loops won’t cross each other but can have common vertices. We propose to construct a binary tree whose leaves are the different spherical patches.

Given a loop \( L \) on a fixed SAS-sphere \( S_i \), \( L \) divides the sphere into two open parts and it is composed of circular arcs formed by the intersection of \( S_i \) and other SAS-spheres. We call the open part of \( S_i \) which is not hidden by those other SAS-spheres as the interior of \( L \), denoted by \( L^o \), while the other open part as the exterior of \( L \), denoted by \( L^c \). We say that another loop \( L' \) on \( S_i \) is inside \( L \) if \( L' \in \overline{L^o} \), where \( \overline{L^o} = L \cup L^o \) is the closed hull of \( L^o \) on \( S_i \). Notice that each loop \( L \) being part of the boundary of a spherical patch contains all the other loops that belong to the boundary of the same patch. By testing if a loop \( L' \) on \( S_i \) is inside the loop \( L \), we can classify all loops on \( S_i \) into two parts: the loops inside \( L \) (including \( L \) itself) and the remaining loops outside \( L \). We do this division repeatedly until each loop is tested which results in a binary tree.

To better understand this process, let’s see the example of Figure 15. On an SAS-sphere \( S_i \), there are six loops \( \{L_1, L_2, L_3, L_5, L_6, L_6^b\} \) forming three spherical patches \( \{P_1, P_2, P_3\} \) with green, red and blue boundaries. The leaves of the binary tree in Figure 15 represent the boundary of the three spherical patches. To construct the tree, consider all loops \( \{L_1, L_2, L_3, L_5, L_6, L_6^b\} \) and test with \( L_1 \), to divide into the set of loops inside \( L_1 \) (\( \{L_1, L_2, L_3, L_5, L_5^b\} \) and the set of the remaining loops outside \( L \) (\( \{L_4, L_5\} \)). Then, we test with \( L_2 \), to find that \( L_2 \) is inside itself, while \( \{L_1, L_3, L_5\} \) are outside, which implies that \( L_2 \) itself forms the boundary of a spherical patch. Afterwards, we find that \( \{L_1, L_6\} \) are both inside \( L_3 \) by testing with \( L_3 \) and that
\{L_1, L_3\} are inside \(L_6\) by testing with \(L_6\). This implies that \(\{L_1, L_3, L_6\}\) form the boundary of the second spherical patch, as these three loops are inside each other. Finally, we test respectively with \(L_4\) and \(L_5\) to find that they are inside each other and they form the boundary of the last spherical patch.

### 1.6.3 Interior of a loop

Let \(L'\) and \(L\) be two loops on \(S_i\). It is left to explain how to test whether \(L'\) is inside \(L\) or not. We assume that \(L\) is composed by \(n\) circular arcs which are formed by the intersection of \(S_i\) and other SAS-spheres \(S'_1, \ldots, S'_n\). Denote all intersection circles by \(C_1, \ldots, C_n\), where \(C_j = S_j \cap S'_i\). The corresponding SAS-balls to \(S'_1, \ldots, S'_n\) are denoted by \(B'_1, \ldots, B'_n\). Since \(L'\) and \(L\) do not cross each other, we can determine if the loop \(L'\) is inside \(L\) or not by testing whether a particular point \(x \in L'\) is inside \(L\) or not. We denote by \(x_k (1 \leq k \leq n)\) the closest point on \(C_k\) to \(x\), which can be analytically given. Notice that \(x_k\) has the smallest Euclidean distance in \(\mathbb{R}^3\) from \(x\) to any point on \(C_k\), which is equivalent to that the shortest path on \(S_i\) from \(x\) to \(C_k\) which ends at \(x_k\). For all circles \(C_k, k = 1, \ldots, n\), we can then find the circle \(C_{k_0}\) with the minimum path length on \(S_i\) from \(x\) to \(C_k\). The closest point on \(C_{k_0}\) to \(x\) is thus denoted by \(x_{k_0}\). With these notations, the following lemma is proposed to test whether \(x\) is inside \(L\) or not, note that \(L' \cap \bigcup_{j=1}^n B'_j = \emptyset\).

**Lemma 1.6.1.** Given an arbitrary point \(x \in S_i \setminus \bigcup_{j=1}^n B_j\), \(x\) is inside \(L\) if and only if \(x_{k_0} \in L\).

**Proof.** We first prove the sufficiency. Assume that \(x\) is inside \(L\), i.e. \(x \in \overline{L^c} = \overline{L^c} \cup L\). Assume by contradicition that \(x_{k_0} \in C_{k_0}\) does not belong to \(L\). Then, since \(C_{k_0} \subset L \cup L^c\) it follows that \(x_{k_0} \in L^c\). Then, the shortest path on \(S_i\) from \(x \in L^c\) to \(x_{k_0} \in L^c\) must cross \(L\). As a consequence, the intersection point between \(L\) and the shortest path on \(S_i\) has a smaller path length to \(x\) than \(x_{k_0}\) does, which is a contradiction. Therefore there holds that \(x_{k_0} \in \mathcal{L}\).

Second, we prove the necessity. Assume that \(x_{k_0} \in L\). By contradiction again, we assume that \(x\) is not inside \(L\), i.e. \(x \in L^c\), which yields that \(x \in \Omega := L^c \setminus \bigcup_{j=1}^n B_j\) from the lemma’s condition. Then, the shortest path on \(S_i\) between \(x \in \Omega\) and \(x_{k_0} \notin \Omega\) must cross \(\Gamma = \partial \Omega\). As a consequence, the intersection point between \(\Gamma\) and the shortest path on \(S_i\) has a smaller path length to \(x\) than \(x_{k_0}\) does, which is a contradiction. Therefore, \(x\) lies inside \(L\).\hfill\square

### 1.6.4 Construction of the cSES and the eSES

After the construction of the cSAS and the eSAS, patches of the cSES and the eSES can consequently be distinguished since each patch corresponds to one component of the SAS. An SES patch belongs to the cSES if and only if its corresponding
SAS patch belongs to the cSAS. In analogy, an SES patch belongs to the eSES if and only if its corresponding SAS patch belongs to the eSAS.

To construct the whole SES, one can construct each SES patch one by one. A convex spherical patch $P_+$ on the SES can be obtained directly by shrinking its corresponding SAS-patch $P_m$ from the SAS-sphere to the VdW-sphere. A toroidal patch $P_t$ corresponding to an SAS circular arc $l_m$ can also be computed without difficulty given the radius, center, starting and ending points of $l_m$, considering two cases respectively in Figure 12 and Figure 13 (see also [30]). To calculate a concave spherical patch corresponding to an SAS intersection point $x_m$, we can use the formula presented in Theorem 1.5.1:

$$P_+ = P_0 \setminus \bigcup_{x \in K} B_{r_p}(x),$$

which is one of the main results of this chapter. Notice that $P_+$ lies on the spherical probe centered at $x_m$ and its boundary is constituted by circular arcs, which can be computed analytically. To construct $P_-$, we can consequently use the same tree-structure (see also Figure 14 on the right) as was used in the previous subsection for constructing SAS spherical patches to classify loops into different sets on the spherical probe (instead of on an SAS-sphere).

1.7 Numerical results

In this section, we illustrate the proposed method with some numerical results based on the previous theoretical results.

1.7.1 A system of two atoms

The proposed method gives the analytic expression of surface areas and cavity volumes for both the SAS and the SES. Here, we test the simplest case of two atoms for the comparison of molecular areas and volumes between the closed form expressions based on the properties of a spherical cap (see website [114]) and the proposed method that relies on the Gauss-Bonnet theorem.

With Matlab, we compare the closed form expressions of the surface areas and cavity volumes with the proposed method. We considered two hydrogen atoms (having radius 1.2Å) with distance 2Å and the probe radius is also set to 1.2Å. Table 1.1 reports the molecular areas and molecular volumes obtained. We confirm that the results coincide up to a small error due to round-off errors.


### 1.7. Numerical results

<table>
<thead>
<tr>
<th>Name</th>
<th>$A_{\text{cses}}$ ($\text{Å}^2$)</th>
<th>$V_{\text{cses}}$ ($\text{Å}^3$)</th>
<th>$A_{\text{ses}}$ ($\text{Å}^2$)</th>
<th>$V_{\text{ses}}$ ($\text{Å}^3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Closed form</strong></td>
<td>1.025416e+02</td>
<td>9.200259e+01</td>
<td>3.223514e+01</td>
<td>1.475567e+01</td>
</tr>
<tr>
<td><strong>Proposed method</strong></td>
<td>1.025416e+02</td>
<td>9.200259e+01</td>
<td>3.223514e+01</td>
<td>1.475567e+01</td>
</tr>
<tr>
<td><strong>Difference</strong></td>
<td>0</td>
<td>-2.842171e-14</td>
<td>7.105427e-15</td>
<td>-3.552714e-15</td>
</tr>
</tbody>
</table>

Table 1.1: Surface areas and cavity volumes for a system of two intersecting atoms obtained respectively from the closed form expression and the proposed method in the chapter. The last row records the difference between these two results.

<table>
<thead>
<tr>
<th>Name</th>
<th>$A_{\text{cses}}$</th>
<th>$A_{\text{ses}}$</th>
<th>$\Delta A_{\text{ses}}$</th>
<th>$A_{\text{csas}}$</th>
<th>$A_{\text{esas}}$</th>
<th>$\Delta A_{\text{esas}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1ETN</td>
<td>956.636</td>
<td>956.636</td>
<td>0</td>
<td>1397.664</td>
<td>1397.664</td>
<td>0</td>
</tr>
<tr>
<td>1B17</td>
<td>3372.577</td>
<td>3372.577</td>
<td>0</td>
<td>4297.098</td>
<td>4297.098</td>
<td>0</td>
</tr>
<tr>
<td>101M</td>
<td>7355.438</td>
<td>7096.387</td>
<td>-0.034</td>
<td>8424.749</td>
<td>8413.275</td>
<td>-0.001</td>
</tr>
<tr>
<td>3WPE</td>
<td>28575.829</td>
<td>27390.001</td>
<td>-0.043</td>
<td>29747.737</td>
<td>29644.146</td>
<td>-0.003</td>
</tr>
<tr>
<td>1A0C</td>
<td>61066.132</td>
<td>54231.268</td>
<td>-0.125</td>
<td>53107.185</td>
<td>51933.798</td>
<td>-0.023</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>$V_{\text{cses}}$</th>
<th>$V_{\text{ses}}$</th>
<th>$\Delta V_{\text{ses}}$</th>
<th>$V_{\text{csas}}$</th>
<th>$V_{\text{esas}}$</th>
<th>$\Delta V_{\text{esas}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1ETN</td>
<td>1567.728</td>
<td>1567.728</td>
<td>0</td>
<td>3333.077</td>
<td>3333.077</td>
<td>0</td>
</tr>
<tr>
<td>1B17</td>
<td>8630.389</td>
<td>8630.389</td>
<td>0</td>
<td>14435.224</td>
<td>14435.224</td>
<td>0</td>
</tr>
<tr>
<td>101M</td>
<td>25029.270</td>
<td>25195.072</td>
<td>0.007</td>
<td>36942.276</td>
<td>36943.700</td>
<td>0.000</td>
</tr>
<tr>
<td>3WPE</td>
<td>9651.803</td>
<td>99642.385</td>
<td>0.001</td>
<td>143181.258</td>
<td>143201.169</td>
<td>0.000</td>
</tr>
<tr>
<td>1A0C</td>
<td>238282.734</td>
<td>244765.065</td>
<td>0.020</td>
<td>326348.079</td>
<td>326601.929</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Table 1.2: Different molecular areas (in $\text{Å}^2$, top) and volumes (in $\text{Å}^3$, bottom) for five molecules (1ETN, 1B17, 101M, 3WPE, 1A0C from the protein data bank, see website [http://www.rcsb.org/pdb/home/home.do](http://www.rcsb.org/pdb/home/home.do)) where we use the notations $\Delta A_{\text{ses}} = A_{\text{ses}} - A_{\text{cses}}$, $\Delta A_{\text{esas}} = A_{\text{esas}} - A_{\text{csas}}$, $\Delta V_{\text{ses}} = V_{\text{ses}} - V_{\text{cses}}$, and $\Delta V_{\text{esas}} = V_{\text{esas}} - V_{\text{csas}}$. The probe radius $r_p$ is fixed to be 1.5Å.

#### 1.7.2 Molecular areas and volumes

The complete and exterior surface of a molecule can have different areas and volumes. Table 1.2 reports the surface areas and cavity volumes of different molecules and also the relative errors between the complete results and the exterior results. We observe that the relative difference between the complete and exterior SES-area and SES-volume is up to 12.5% resp. 2.7% in this test cases.

#### 1.7.3 Comparison with the MSMS-algorithm

We have run the MSMS-algorithm to compare it with the proposed method. It has been observed that if there is no SES-singularity, the exterior molecular areas and volumes from the MSMS coincide with the results of the exterior molecular surface from the proposed method in general. However, in the case where SES-singularities occur on concave spherical patches (which is often the case for complex molecules),
Figure 16: The SES of three balls with a singular circle on the concave spherical patches (left) and the SES with the intersection of two concave spherical patches (right, triangulation provided by the MSMS-algorithm). The probe radius \( r_p = 1 \text{Å} \).

<table>
<thead>
<tr>
<th></th>
<th>( A_{\text{ses}} (r_p = 2 \text{Å}) )</th>
<th>( A_{\text{sas}} (r_p = 2 \text{Å}) )</th>
<th>( A_{\text{ses}} (r_p = 1 \text{Å}) )</th>
<th>( A_{\text{sas}} (r_p = 1 \text{Å}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MSMS</strong></td>
<td>63.712</td>
<td>253.122</td>
<td>64.616</td>
<td>153.545</td>
</tr>
<tr>
<td><strong>Proposed method</strong></td>
<td>63.712</td>
<td>253.122</td>
<td>58.857</td>
<td>153.545</td>
</tr>
</tbody>
</table>

Table 1.3: Molecular areas (Å\(^2\)) for three balls with different probe radii \( (r_p = 2 \text{Å}, 1 \text{Å}) \), obtained from the MSMS and the proposed method.

the MSMS might fail to give the right SES-area (or SES-volume) due to an incomplete characterization of all SES-singularities.

This can be illustrated by a simple example of three atoms, where we calculate the molecular areas respectively obtained from the MSMS and the proposed method. Consider that there are three atoms with the same radius 1.2Å. In the case of a probe radius of \( r_p = 2 \text{Å} \), there is no SES-singularity and the molecular areas obtained from the two methods coincide (see Table 1.3). In the case of a probe radius of \( r_p = 1 \text{Å} \) (see the left of Figure 16), there is a singular circle on the concave spherical patches. The SAS-areas obtained from the two methods still coincide, but the SES-areas are different (see Table 1.3) because the MSMS has ignored the "self-intersection" of the surface by simply treating the concave spherical patches as two triangle-shaped spherical patches (see the right of Figure 16).

Furthermore, we have tested the proposed method for sequences of an increasing number of triangles in the mesh for the case of three atoms with \( r_p = 1 \text{Å} \), see Figure 17. Indeed, based on the provided analysis one can construct meshes at various sizes, see Chapter 2. Like this, we obtain the analytical SES-area as well as a numerical (approximate) value derived from the mesh. We observe that this numerical value converges to the analytical one if using finer and finer meshes. In the same manner, the MSMS-algorithm provides an analytical and a numerical value for the SES-area. However, the “analytical” value does not account for “self-intersection” and therefore this value is too large. As a consequence, the numerical value also converges to this wrong limit.

In addition, we have computed the exterior SAS-areas and the exterior SES-areas of molecules at various sizes by both the MSMS and the proposed method for comparison, see Table 1.4.
Figure 17: The numerical SES-areas of the three atoms reported in Å² \((r_p = 1\text{Å})\) with respect to the number of triangles in the mesh. For each method, the analytical and numerical area is computed.

<table>
<thead>
<tr>
<th>Name</th>
<th>Number of atoms</th>
<th>MSMS</th>
<th>Proposed method</th>
<th>∆(A_{\text{esas}})</th>
<th>∆(A_{\text{eses}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1YJO</td>
<td>67</td>
<td>1325.395</td>
<td>822.629</td>
<td>1325.395</td>
<td>822.628</td>
</tr>
<tr>
<td>1E1N</td>
<td>160</td>
<td>1396.191</td>
<td>955.899</td>
<td>1397.664</td>
<td>956.636</td>
</tr>
<tr>
<td>1B17</td>
<td>483</td>
<td>4297.099</td>
<td>3372.577</td>
<td>4297.098</td>
<td>3372.577</td>
</tr>
<tr>
<td>101M</td>
<td>1413</td>
<td>8413.273</td>
<td>7096.387</td>
<td>8413.275</td>
<td>7096.387</td>
</tr>
<tr>
<td>2K4C</td>
<td>2443</td>
<td>13825.406</td>
<td>11404.567</td>
<td>13826.835</td>
<td>11409.523</td>
</tr>
<tr>
<td>3WPE</td>
<td>5783</td>
<td>29656.324</td>
<td>27387.289</td>
<td>29644.146</td>
<td>27390.001</td>
</tr>
<tr>
<td>1KJU</td>
<td>7671</td>
<td>34551.902</td>
<td>33172.387</td>
<td>34630.252</td>
<td>33148.676</td>
</tr>
<tr>
<td>1A0T</td>
<td>9996</td>
<td>40821.195</td>
<td>38470.543</td>
<td>40757.735</td>
<td>38486.055</td>
</tr>
<tr>
<td>1A0C</td>
<td>13992</td>
<td>51231.840</td>
<td>53987.203</td>
<td>51934.798</td>
<td>54231.268</td>
</tr>
<tr>
<td>4XBG</td>
<td>19608</td>
<td>108054.734</td>
<td>101242.242</td>
<td>108158.716</td>
<td>101437.41</td>
</tr>
</tbody>
</table>

Table 1.4: Exterior molecular areas (Å²) of molecules at different sizes respectively obtained from the MSMS and the proposed method \((r_p = 1.5\text{Å})\), where \(A_{\text{esas}}^*\) and \(A_{\text{eses}}^*\) are results obtained from the MSMS, \(A_{\text{esas}}\) and \(A_{\text{eses}}\) are results obtained from the proposed method, \(\Delta A_{\text{esas}} = A_{\text{esas}}^* - A_{\text{esas}}\) and \(\Delta A_{\text{eses}} = A_{\text{eses}}^* - A_{\text{eses}}\).
1.7.4 Computational cost

We have implemented the proposed method of computing the areas and volumes with Matlab for molecules at various sizes. Figure 18 demonstrates the relationship between the total CPU-time and the size of the molecule, where we observe almost a linear relationship. The program was run on a laptop with 2.5GHz quad-core Intel Core i7 processor. This first implementation in Matlab is to be understood as a proof of concept. The important information to retain is that it can be implemented in a linearly scaling fashion. Of course, in order to lower the pre-constant of the linear scaling, one has to refer to a proper and more professional implementation in a better performing language. We therefore expect a better performance of the proposed method in fortran or C++.

1.8 Conclusion

In this chapter, based on three equivalence statements and a new Voronoi-type diagram, an analytical and also computable implicit function of the SES has been provided. Furthermore, all SES patches including SES-singularities can be characterized and computed according to Theorem 1.5.1 and Theorem 1.5.2. Consequently, different molecular areas and volumes can be calculated analytically with explicit formulas. In addition, we have also proposed the concept of the cSAS and the eSAS (as
well as the cSES and the eSES), considering the possible inner holes inside a solute molecule or not. Finally, a corresponding method to construct (or distinguish) the cSAS and the eSAS has been given, with the use of a binary tree. In summary, the molecular surfaces have been analytically characterized (especially the SES), which allows one to do more accurate calculation in solvation models associated with different molecular cavities than the state-of-the-art methods.
Chapter 2

Meshing molecular surfaces based on analytical implicit representation

Contents

2.1 Introduction ............................................. 70
  2.1.1 Previous work ..................................... 70
  2.1.2 Contribution ..................................... 71
  2.1.3 Outline ........................................... 71

2.2 Molecular surfaces ................................. 72
  2.2.1 Definitions ...................................... 72
  2.2.2 Implicit representations ......................... 73
  2.2.3 Complete characterization of the SES ............ 75

2.3 Construction of molecular surfaces ............... 77
  2.3.1 Data structures of the SAS ...................... 77
  2.3.2 Assembling spherical patches ..................... 78
  2.3.3 Assembling surfaces ............................. 79

2.4 Molecular inner holes .............................. 80

2.5 Meshing ............................................. 82
  2.5.1 Boundary division ................................ 83
  2.5.2 Spherical patches ................................ 83
  2.5.3 Toroidal patches ................................ 89
  2.5.4 Mesh refinement ................................ 90
  2.5.5 Remeshing ....................................... 90
  2.5.6 Illustrations .................................... 90
  2.5.7 Computational cost .............................. 92
2.1 Introduction

As emphasized in the introduction of the thesis as well as in Chapter 1, many fields of research like chemistry, biochemistry, physics, and biomedicine work with molecular surfaces. For example, the majority of (bio-)chemically relevant reactions take place in the liquid phase and the effect of the environment (solvent) is important and should be considered in one way or the other in the model. As an alternative to the way of taking solvent molecules explicitly into account, various implicit solvation models have been proposed in which the molecular surface of the solute is a part of the model and constitutes the interface of the atomistic and the continuum model. A second field where the notion of molecular surface is important is simply the visualization of molecules.

In the simplest model of a molecular surface of the solute or a molecule in general, each constituting atom is idealized by a simple sphere with its van der Waals (VdW) radius. The boundary of the union of these VdW spheres is the so-called VdW-surface.

Besides the VdW-surface, two other kinds of molecular surfaces are commonly used in solvation models or in molecular visualization: the Solvent Accessible Surface (SAS) and the Solvent Excluded Surface (SES). Since the VdW-surface and the SAS are both the topological boundary of the union of spheres, their geometric features are therefore easier to understand. However, the SES, which sometimes performs better in a chemical calculation or is more suitable for applications like docking, is more complicated.

2.1.1 Previous work

The definitions of the SAS and the SES were first introduced by Lee & Richards in the 1970s. The SES is also called the "smooth molecular surfaces" or "Connolly’s surfaces" due to Connolly’s fundamental calculation on it. Indeed, the
SES can be considered to be the prototype for the computational study of molecular surfaces. This surface model has been applied to a very large variety of problems and has also been used to compute solvation energies with continuum solvation models [109, 108, 57].

Latter, Sanner developed the reduced surfaces and proposed the MSMS (Michel Sanner’s Molecular Surface) algorithm for computing an (in fact approximately) analytical representation of the SES [102]. The MSMS algorithm can also provide a triangulation of the SES with a user-specified density of vertices. Although it can not deal with all self-intersections between different patches of the SES (see Figure 17 of Chapter 1 for an example where self-intersection occurs), the MSMS algorithm is now one of the most widely-used packages of molecular surfaces.

Besides, there are many other contributions on the molecular visualization [55, 51, 61, 64, 89], high-quality meshing [65] or the calculation of molecular areas and volumes [111, 113, 18]. However, a computable analytical implicit representation of the SES and a complete characterization of all SES-singularities remained unsolved until a recently-published paper by us [93].

2.1.2 Contribution

We give a detailed strategy for constructing the data structures of molecular surfaces, based on the analytical characterization of the SES including its singularities presented in [93]. Then, a meshing algorithm for molecular surfaces, especially the SES, is developed, combining an advancing-front algorithm with the pre-computed data structures. The explicit characterization of all singularities resolves the issue of self-intersection that is experienced due to singularities as they can be computed prior to the meshing of the surface. This, in turn, allows the possibilities of meshing the SES exactly, in the sense that each vertex of the mesh lies exactly on the surface. We want to emphasize once again that this is only possible due to the newly developed analysis of the molecular surfaces and it is not the case for the existing meshing algorithms. In addition, we propose an algorithm for filling molecular inner holes with virtual atoms for the reason that the appearance of these inner holes is not always justified in the solute molecular cavity of the continuum solvation models, as this would mean that the solvent is present in these inner holes.

2.1.3 Outline

In the next section, we give the definitions and the implicit representations of different molecular surfaces. In the third section, the construction of molecular surfaces, which is defined by different patches and their connectivity, is proposed. Based on this pre-computed data, an algorithm for filling molecular inner holes is proposed in the fourth section. Further, a meshing algorithm for molecular surfaces including
two sub-algorithms for meshing respectively a (convex or concave) spherical patch and a toroidal patch is developed in the fifth section where we also present some illustrations of artificial as well as realistic molecular surfaces. Finally, a conclusion is presented in the last section.

2.2 Molecular surfaces

A mathematical analysis and calculation of the SAS and the SES has been presented in our recent work [93]. In this section, we recall some results including a mathematical definition of the surfaces, their implicit representations and the complete characterization of the SES.

2.2.1 Definitions

As already emphasized, atoms of a molecule can be represented by VdW-balls with VdW-radii which are experimentally fitted, given the underlying chemical element [97]. As a consequence, the VdW-surface is defined as the topological boundary of the union of all VdW-balls. Further, the SAS of a solute molecule is defined by rolling the center of an idealized spherical probe over the solute molecule, that is, the surface enclosing the region in which the center of a spherical probe can not enter. Finally, the SES is defined by the same spherical probe rolling over the molecule, but now we consider the surface enclosing the region in which a spherical probe can not access. In other words, the SES is the boundary of the union of all spherical probes that do not intersect the VdW-balls of the solute molecule.

The definition of VdW-surface is based on the model that each atom has a specific radius around the atom center. However, the definition of the VdW-surface has ignored the size and shape of the surrounding solvent molecules in solvation models. The definition of SAS has taken this into account by modeling them by idealized spherical probes with a certain probe radius. The definition of the SES is different from the SAS in the sense that not the probe center traces out the desired surface, but the surface of the probe. In the application of the above-mentioned docking [76], the SES will not lead to the overlapping of neighboring surfaces since the SES does not inflate the atom radii but the SAS will.

Sometimes, the SAS can be non-connected: it can be composed of several separate surfaces. We call the outmost surface as the exterior Solvent Accessible Surface (eSAS) and the union of all separated surfaces as the complete Solvent Accessible Surface (cSAS), see [93] for details. Correspondingly, we also propose the concept of the complete Solvent Excluded Surface (cSES) and the exterior Solvent Excluded Surface (eSES). We make a convention that the SAS refers to both the cSAS and the eSAS in a general context, and the SES refers in the same spirit to both the cSES
Chapter 2: Meshing molecular surfaces based on analytical implicit representation

Figure 1: 3D schematic of the SES illustrating the convex spherical patches (red), the toroidal patches (yellow) and the concave spherical patches (blue).

Both the VdW-surface and the SAS are composed of three parts: open spherical patches, open circular arcs (or circles) and intersection points (formed by the intersection of three or more spheres). The SES can be divided into three corresponding types of patches [30]: convex spherical patches, toroidal patches and concave spherical patches, see Figure 1 for a 3D illustration. As showed in [93], any point on a convex spherical patch of the SES has a closest point to the SAS on a spherical patch. Similarly, any point on a toroidal patch of the SES has a closest point to the SAS on a circular arc, and any point on a concave patch has a closest point to the SAS which is an intersection point.

2.2.2 Implicit representations

We denote by $M$ the number of atoms in a solute molecule, by $c_i \in \mathbb{R}^3$ and $r_i \in \mathbb{R}^+$ the center and the radius of the $i$-th VdW atom. The open ball with center $c_i$ and radius $r_i$ is called the $i$-th VdW-ball. The VdW-surface can consequently be represented as an implicit surface $\tilde{f}_{vdw}^{-1}(0) = \{p \in \mathbb{R}^3 | \tilde{f}_{vdw}(p) = 0\}$ with the following implicit function:

$$\tilde{f}_{vdw}(p) = \min_{i=1,\ldots,M} \{\|p - c_i\|_2 - r_i\}, \quad \forall p \in \mathbb{R}^3. \quad (2.2.1)$$

Similarly, the open ball with center $c_i$ and radius $r_i + r_p$ is called the $i$-th SAS-ball denoted by $B_i$, where $r_p$ is the radius of the idealized spherical probe. Furthermore, we denote by $S_i$ the $i$-th SAS-sphere corresponding to $B_i$, that is, $S_i = \partial B_i$. Similar to the VdW-surface, the SAS can be represented as an implicit surface $\tilde{f}_{sas}^{-1}(0)$ with the following implicit function:

$$\tilde{f}_{sas}(p) = f_{vdw}(p) - r_p = \min_{i=1,\ldots,M} \{\|p - c_i\|_2 - r_i - r_p\}, \quad \forall p \in \mathbb{R}^3. \quad (2.2.2)$$

We notice that the above implicit function of the SAS is simple to compute. It seems
nevertheless hopeless for us to further obtain an implicit function of the SES for the reason that $f_{\text{sas}}(p)$ is not a distance function to the SAS. On the other hand, having the signed distance function at hand would allow the construction of an implicit function for the SES due to the geometrical relationship between the SAS and the SES, i.e., they are separated by the fixed distance $r_p$.

In [93], we calculated the signed distance function to the SAS. Indeed, since the SAS is a closed surface, there exists a closest point on the SAS to any given point $p \in \mathbb{R}^3$, which is denoted by $x_{\text{sas}}^p$ depending on $p$. We emphasize that there might exist more than one closest point to $p$ and in this case, $x_{\text{sas}}^p$ is chosen to be one of these closest points. The signed distance function $f_{\text{sas}}(p)$ can then be easily written as:

$$f_{\text{sas}}(p) = \begin{cases} -\|p - x_{\text{sas}}^p\| & \text{if } p \text{ lies inside the SAS,} \\ \|p - x_{\text{sas}}^p\| & \text{if } p \text{ lies outside the SAS.} \end{cases} \quad (2.2.3)$$

The difficulty is however to find efficiently one closest point $x_{\text{sas}}^p$. In the Chapter 6 & 7 of [34], one can find a very detailed and general discussion about the properties of the signed (or more generally speaking, oriented) distance function.

According to the fact that any point on the SES has signed distance $-r_p$ to the SAS, an implicit function of the SES is obtained directly as:

$$f_{\text{ses}}(p) = f_{\text{sas}}(p) + r_p, \quad (2.2.4)$$

which motivates the choice of using the signed distance function denoted by $f_{\text{sas}}(p)$ to represent the SAS. From the above formula, the SES can be represented by a level set $f_{\text{sas}}^{-1}(-r_p)$, associated with the signed distance function $f_{\text{sas}}$ to the SAS. Therefore, the key point becomes how to compute the signed distance $f_{\text{sas}}(p)$ from a point $p \in \mathbb{R}^3$ to the SAS. Generally speaking, given a general surface $S \subset \mathbb{R}^3$ and any arbitrary point $p \in \mathbb{R}^3$, it is difficult or expensive to compute the signed distance from $p$ to $S$. The fast marching method [106] and the fast sweeping method [117] are two famous methods for computing such signed distance. However, considering that the SAS is a special surface composed of spherical patches, this computation can be done analytically [93].

Further, the region enclosed by the VdW-surface is called the VdW-cavity, that is, any point $p$ in the VdW-cavity satisfies $f_{\text{vdw}}(p) \leq 0$. More generally, the region enclosed by a molecular surface is called by its corresponding molecular cavity. As a consequence, the region enclosed by the SAS is called the SAS-cavity, and the region enclosed by the SES is called the SES-cavity. Similarly, any point $p$ in the SAS-cavity satisfies $f_{\text{sas}}(p) \leq 0$, and any point $p$ in the SES-cavity satisfies $f_{\text{ses}}(p) \leq 0$. 

Complete characterization of the SES

In this subsection, we present the main results about the complete characterization of the SES in Chapter 1 or in [93]. For any point $x_{\text{sas}}$ on the SAS, we define a mapping $R$ by

$$R(x_{\text{sas}}) = \{ p \in \Omega | \text{dist}(p, \Gamma_{\text{sas}}) = |p - x_{\text{sas}}| \},$$

where $\Omega$ is the SAS-cavity, $\Gamma_{\text{sas}}$ is the SAS and dist$(p, \Gamma_{\text{sas}})$ denotes the distance from $p$ to $\Gamma_{\text{sas}}$. Consequently, we have the following theorem according to [93]:

**Theorem 2.2.1.** The following equivalence statements hold:

1. If $x_{\text{sas}}$ lies on a spherical patch of the SAS, part of the sphere $S_i$, then $R(x_{\text{sas}}) = [c_i, x_{\text{sas}}]$ is a line segment.
2. If $x_{\text{sas}}$ lies on a circular arc of the SAS, part of the intersection circle $S_i \cap S_j$, then $R(x_{\text{sas}}) = \triangle x_{\text{sas}}c_ic_j$ is a triangle.
3. If $x_{\text{sas}}$ is an intersection point of the SAS, then $R(x_{\text{sas}})$ is a polyhedron.

**Remark 2.2.1.** The first equivalence in the above theorem states that in the case where $x_{\text{sas}}$ lies on a spherical patch of the SAS associated with $S_i$, $x_{\text{sas}}$ is a closest point to an arbitrary point $p \in \Omega$ if and only if $p$ lies on the line segment with endpoints $c_i$ and $x_{\text{sas}}$. The second equivalence states that in the case where $x_{\text{sas}}$ lies on a circular arc of the SAS associated with $S_i$ and $S_j$, $x_{\text{sas}}$ is a closest point to an arbitrary point $p \in \Omega$ if and only if $p$ lies on the triangle with three vertices $c_i, c_j$ and $x_{\text{sas}}$.

We can generalize the mapping $R$ by

$$R(X) = \bigcup_{x \in X} R(x), \quad \forall X \subseteq \Gamma_{\text{sas}}.$$

As a consequence, for a spherical SAS patch denoted by $P_{i_1}$ with index $i_1$, its corresponding convex SES patch $P_+$ can be written as $P_+ = R(P_{i_1}) \cap \Gamma_{\text{ses}}$ where $\Gamma_{\text{ses}}$ is the SES; for a circular SAS arc denoted by $l_{i_2}$ with index $i_2$, its corresponding toroidal SES patch $P_t$ can be written as $P_t = R(l_{i_2}) \cap \Gamma_{\text{ses}}$; for an SAS intersection point denoted by $x_{i_3}$ with index $i_3$, its corresponding concave SES patch $P_-$ can be written as $P_- = R(x_{i_3}) \cap \Gamma_{\text{ses}}$. In Figure 1, for instance, $P_+$ is a red convex patch, $P_t$ is a yellow toroidal patch and $P_-$ is a blue concave patch. Further, according to Theorem 2.2.1, we know that $R(P_{i_1})$ is a spherical sector, $R(l_{i_2})$ is a double-cone region and $R(x_{i_3})$ is a polyhedron.

There is no difficulty to compute the convex SES patches or the toroidal SES patches given all spherical SAS patches and circular SAS arcs, see details in [30]. However, the self-intersection problem might occur among the concave SES patches.
Figure 2: On the left, the concave spherical triangle $P_\Delta$ with vertices $(a_1, a_2, a_3)$ corresponds to an intersection point $x_I$. In the case where $P_-$ coincide with $P_\Delta$, there will be no singularity on the concave spherical patch. On the right, the concave spherical patch $P_-$ does not coincide with $P_\Delta$ and there are singular circular arcs as parts of its boundary. The vertices of $P_-$ are $(a_1, a_2, a_3, a_4, b_1, b_2, a_5)$. The two flat grey regions $D_1$ and $D_2$ are formed by the intersection of $P_\Delta$ with two other nearby spherical probes. These two regions have the boundaries composed of line segments and circular arcs.

For an SAS intersection point $x_{i_3} \in S_{j_1} \cap S_{j_2} \cap S_{j_3}$, denote by $P_\Delta$ the concave spherical triangle corresponding to $x_{i_3}$ (see the left of Figure 2 for a schematic), that is,

$$P_\Delta = \partial B_{r_p}(x_{i_3}) \cap \left\{ p : p = x_{i_3} + \sum_{k=1}^{3} \lambda_k (c_{j_k} - x_{i_3}), \forall \lambda_k \geq 0 \right\},$$

where $c_{j_k}$ is the center of the sphere $S_{j_k}$ associated with $x_{i_3}$, $k = 1, 2, 3$, and $B_r(c)$ denotes the open ball with center $c$ and radius $r$. Notice that $P_\Delta$ is a triangle-shaped patch generated by the sphere $\partial B_{r_p}(x_{i_3})$ and three planes, see the left of Figure 2 for an illustration. Then, we have the following result from [13] for computing the concave SES patch $P_-$ corresponding to $x_{i_3}$.

**Theorem 2.2.2.** $P_- = P_\Delta \setminus \bigcup_{x \in K} B_{r_p}(x)$ where $K = \{ x \in I : \| x - x_{i_3} \| < 2r_p, \ x \neq x_{i_3} \}$ and $I$ denotes the set of all SAS intersection points.

If $P_-$ does not coincide with $P_\Delta$, singular arcs occur on the concave SES patch and constitute part of its boundary $\partial P_-$, see the right of Figure 2 for an illustration. In fact, Theorem 2.2.2 says that an arbitrary concave SES patch $P_-$ can be characterized as $P_\Delta$ excluding all spherical probes centered at the "nearby" intersection points in $K$.

**Remark 2.2.2.** Theoretically speaking, the geometry of a concave SES patch can be as complicated as possible. But Theorem 2.2.2 provides an analytical representation of each concave SES, which finally gives a complete characterization of the SES.
Chapter 2: Meshing molecular surfaces based on analytical implicit representation

2.3 Construction of molecular surfaces

This section will focus on the construction of molecular surfaces by calculating different components of them. We only consider the SAS and the SES since the construction of the VdW-surface is the same as the SAS. Furthermore, we assume that any SAS-ball is not included in another one (otherwise, the inner SAS-ball can be ignored).

2.3.1 Data structures of the SAS

We first need to compute an intersection matrix in which the $i$-th row records the neighboring SAS-spheres intersected with the $i$-th SAS-sphere. To retrieve all neighboring SAS-spheres, we can use a data structure called a binary spatial division tree proposed by Barnes and Hut with the average complexity $O(\log M)$ \[12, 102\] where $M$ is the number of atoms. In practice, the maximum number of intersected SAS-spheres for a given SAS-sphere is bounded by a constant $k_{\text{max}}$. As a consequence, the intersection matrix is defined of size $M \times k_{\text{max}}$ and each row reports the indices of the neighboring spheres. Based on this intersection matrix, we can establish data structures of the components of both the SAS and then the SES.

An intersection point on the SAS can in theory be formed by the intersection of more than three SAS-spheres. This can appear quite often due to symmetries. In this case, the intersection can however be divided into multiple triplets of SAS-spheres for simplicity. Therefore, we assume that any intersection point is formed by the intersection of three SAS-spheres. On each SAS-sphere, we calculate the intersection points formed by the intersection with any two neighboring SAS-spheres. After calculating the intersection points on each SAS-sphere, we obtain the set of all intersection points $I$. An intersection point has the following data structure:

<table>
<thead>
<tr>
<th>SAS Intersection Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(x, y, z)$: coordinate of the point</td>
</tr>
<tr>
<td>$(i, j, k)$: indices of three corresponding SAS-spheres</td>
</tr>
</tbody>
</table>

For each pair of neighboring SAS-spheres, we can calculate all circular arcs on the intersection circle of two intersecting SAS-spheres since all intersection points on this
circle are known. To represent each circular arc, we record the starting and ending point, its center, radius, radian and the pair of SAS-spheres with the following data structure:

<table>
<thead>
<tr>
<th>Circular SAS Arc</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(i_1, i_2)$: index of the starting and ending intersection point</td>
</tr>
<tr>
<td>$(x, y, z)$: coordinate of the center</td>
</tr>
<tr>
<td>$r$: radius</td>
</tr>
<tr>
<td>$\beta$: radian</td>
</tr>
<tr>
<td>$(i, j)$: indices of two corresponding SAS-spheres</td>
</tr>
</tbody>
</table>

On each SAS-sphere, there are loops composed of circular arcs, which also form the boundaries of spherical patches. As a consequence, we can represent each loop by a set of constituting circular arcs and then each spherical patch by a set of loops forming the boundary. The following data structures are used:

<table>
<thead>
<tr>
<th>SAS Loop</th>
</tr>
</thead>
<tbody>
<tr>
<td>$l_1, l_2, \ldots, l_n$: the consisting circular arcs</td>
</tr>
<tr>
<td>$i$: index of the SAS-sphere on which the loop lies</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Spherical SAS Patch</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n$: the loops forming the boundary of the patch</td>
</tr>
<tr>
<td>$\hat{i}$: index of the SAS-sphere on which the patch lies</td>
</tr>
</tbody>
</table>

### 2.3.2 Assembling spherical patches

The crucial problem in the above data structures is to associate a spherical patch with a set of loops forming its boundary. This is tightly connected with determining whether two loops lie on the boundary of a common spherical patch or not, given all loops on a sphere. This motivates us to propose a method based on a binary tree structure. To do so, we define the "interior" and "exterior" of a loop on a sphere. More precisely, a loop divides the sphere into the "interior", the part which is not hidden by any intersected sphere forming this loop, and the "exterior", the remaining part. Denote by $\{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n\}$ the list of loops on the sphere. For any two loops $\mathcal{L}_i$ and $\mathcal{L}_j$ belonging to the boundary of a common spherical patch, we then have $\mathcal{L}_i \subset \mathcal{L}^o_j$ and $\mathcal{L}_j \subset \mathcal{L}^o_i$, where $\mathcal{L}^o_i$ and $\mathcal{L}^o_j$ represent respectively the "interior" of $\mathcal{L}^o_i$ and $\mathcal{L}^o_j$. As
a consequence, each spherical patch on the sphere has a boundary composed of loops which are "inside" each other. Here, one loop "inside" another means that the loop is in the "interior" of the other and further, a loop is said to be "inside" itself.

The problem is to classify all loops into different classes such that each loop in one class is "inside" another one in the same class. We first divide the set of loops on a sphere into two subsets by checking whether an element of the set is "inside" a given loop or not. Then, we can look at each of the subsets (loops) and for each subset, we check again if each loop of this subset is "inside" the first loop of this subset. If yes, we continue to check for the next loop of the subset until we find one loop that is "outside" another and then we build two new subsets for this subset. Otherwise, if each loop of the subset is "inside" all the others, we leave this subset as a leaf of the binary tree representing the boundary of a spherical patch. Given an initial loop, we can therefore derive a binary tree whose leaves identify the boundaries of different spherical patches. See Figure 15 in Chapter 1 for a schematic of this process.

This method is also suitable for assembling a concave spherical SES patch denoted by $P_-$ corresponding to an SAS intersection point $x_{i_3}$ using the formula in Theorem 2.2.2. We first calculate all loops forming the boundary of $P_-$ each having a data structure as an SAS loop. Then, we classify the set of loops into different subsets each identifying the boundary of a sub-patch using the above method which is based on a binary tree.

2.3.3 Assembling surfaces

2.3.3.1 SAS

With the above data structures of different molecular components, we are ready to construct complete and exterior molecular surfaces, i.e., assemble these data structures. The cSAS is composed of all spherical patches on each SAS-sphere, which have been calculated in Section 2.3.2. We map a faraway point from the molecule onto a spherical SAS patch and then use this patch as the first element of the eSAS. Two spherical SAS patches are neighbors if they have a common circular arc or circle on their boundaries. By adding neighboring spherical patches one by one, we finally obtain all spherical patches on the eSAS. Since the data structure of each patch contains all necessary information about its neighbors, this is straightforward.

2.3.3.2 SES

As already mentioned, a spherical SAS patch (cSAS or eSAS) corresponds to a similar convex spherical SES patch (respectively cSES or eSES), a circular SAS arc corresponds to a (rectangle-shaped or double-triangle-shaped, see Figure 11 in Chapter 1) toroidal SES patch and an SAS intersection point corresponds to a concave
spherical SES patch obtained from Theorem 2.2.2. A graphical illustration of the caffeine molecule is presented in Figure 9 in Chapter 1.

With this geometrical relationship, the construction of the cSES and the eSES can be done directly based on the construction (i.e., the assembling of the data structures) of the cSAS and the eSAS, where the data structures of different SES patches are established as follows:

<table>
<thead>
<tr>
<th>Rectangle-shaped Toroidal Patch</th>
<th>Double-triangle-shaped Toroidal Patch</th>
</tr>
</thead>
<tbody>
<tr>
<td>• $i_l$: index of the corresponding circular SAS arc</td>
<td>• $i_l$: index of the corresponding circular SAS arc</td>
</tr>
<tr>
<td>• $\mathcal{L}_1^l$ or $(\mathcal{L}_1^l, \mathcal{L}_2^l)$: one loop composed of four circular arcs or two circles forming the boundary</td>
<td>• $(\mathcal{L}_1^l, \mathcal{L}_2^l)$: two loops each composed of up to three circular arcs forming the boundary</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Convex Spherical SES Patch</th>
<th>Concave Spherical SES Patch (or Sub-patch)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• $i_P$: index of the corresponding spherical SAS patch</td>
<td>• $i_l$: index of the corresponding SAS intersection point</td>
</tr>
<tr>
<td>• $\mathcal{L}_1^+, \mathcal{L}<em>2^+, \ldots, \mathcal{L}</em>{m_1}^+$: loops forming the boundary of the patch</td>
<td>• $\mathcal{L}_1^-, \mathcal{L}<em>2^-, \ldots, \mathcal{L}</em>{m_2}^-$: loops forming the boundary of the patch</td>
</tr>
</tbody>
</table>

In the above data structures of the SES, a loop $(\mathcal{L}_1^l)$ on a toroidal patch contains only the corresponding circular arcs (or circles), each having the same structure as a circular SAS arc introduced in the previous subsection. A loop $(\mathcal{L}_1^+ \text{ or } \mathcal{L}_1^-)$ on a spherical SES patch has the same structure as an SAS loop containing both the corresponding circular arcs (or circles) and the index of the sphere on which the loop lies.

### 2.4 Molecular inner holes

As already mentioned, there might exist inner holes in the cSAS-cavity (or the VdW-cavity). In implicit solvation models, one might be interested in filling these inner holes since it is unphysical that the solvent is present in these holes. One possibility is to construct virtual atoms to fill these inner holes, which simultaneously doesn’t influence the construction of the eSAS. In particular, these virtual atoms can
be treated as completely artificial with the purpose to fill these inner holes. One has however to consider that these virtual balls should not intersect the exterior region of the SAS-cavity.

Denote the exterior SAS by $\Gamma^e$ and the separate inner subsurfaces of the cSAS by $\Gamma^i_j$, $1 \leq j \leq n$. Further, denote the set of intersection points on $\Gamma^e$ by $I^e$ and the set of intersection points on $\Gamma^i_j$ by $I^i_j$. With the above notations, we state the following lemma:

**Lemma 2.4.1.** The distance between two subsurfaces of the cSAS can be characterized by

$$\text{dist}(\Gamma^e, \Gamma^i_j) = \min_{x \in I^e, y \in I^i_j} \|x - y\|, \quad \forall 1 \leq j \leq n,$$

and

$$\text{dist}(\Gamma^i_j, \Gamma^i_k) = \min_{x \in I^i_j, y \in I^i_k} \|x - y\|, \quad \forall 1 \leq j, k \leq n,$$

where $\text{dist}(X, Y)$ denotes the minimum distance between two sets $X$ and $Y$ defined by $\text{dist}(X, Y) = \min_{x \in X, y \in Y} \|x - y\|$ and $\| \cdot \|$ denotes the Euclidean norm.

**Proof.** Consider any point $p \in \Gamma^i_j$. If it has a closest point $x^p_{\text{sas}} \in \Gamma^e$ lying on a spherical patch or a circular SAS arc, then $p$ must lie on the corresponding line segment or the corresponding triangle according to Theorem 2.2.1 (and Remark 2.2.1). Notice that such a line segment or triangle is covered by the union of SAS-spheres. However, $p \in \Gamma^i_j$ is not included by the union of all SAS-spheres. This is a contradiction. As a consequence, $p$ can only have a closest point to $\Gamma^e$ in the set of intersection points $I^e$. On the other hand, we have the same conclusion that for any point on $\Gamma^e$, it can only have a closest point to $\Gamma^i_j$ in $I^i_j$. Therefore, the distance between $\Gamma^e$ and $\Gamma^i_j$ is equal to the distance between the two sets of intersection points $I^e$ and $I^i_j$. That is to say,

$$\text{dist}(\Gamma^e, \Gamma^i_j) = \min_{x \in \Gamma^e, y \in \Gamma^i_j} \|x - y\| = \min_{x \in \Gamma^e, y \in \Gamma^i_j} \|x - y\| = \min_{x \in I^e, y \in I^i_j} \|x - y\|.$$

Further, we have the same result for $\Gamma^i_j$ and $\Gamma^i_k$, i.e., the distance between $\Gamma^i_j$ and $\Gamma^i_k$ is equal to the distance between the two sets $I^i_j$ and $I^i_k$. \qed

The above lemma allows a fast computation of the distance from an inner cavity to the exterior SAS or to another inner cavity since the right-hand sides only compute the distances among the finite number of intersection points. With the distances between each inner subsurface and the exterior subsurface $\Gamma^e$ of the cSAS (i.e. the eSAS), we can add virtual spheres with small enough radii so that each virtual sphere doesn’t intersect $\Gamma^e$. This prevents the added virtual spheres affecting the geometry of $\Gamma^e$.

Given an inner subsurface $\Gamma^i_j$, we want to fill the inner hole enclosed by it with virtual spheres. We propose the following three steps to do this:
2.5. Meshing

The SAS is composed of spherical patches like the VdW-surface but with increased radii $r_i + r_p$. As a consequence, meshing the SAS or the VdW-surface can be reduced to developing a meshing algorithm for an arbitrary spherical patch given its SAS-center, its SAS-radius and its boundary information obtained from the above data structures. This algorithm can also be applied to mesh a convex or concave SES patch, since its center, its radius and its boundary information are known. In this

Figure 3: The left figure shows the SAS (blue) of 1ETN with $r_p = 0.7\text{Å}$ and the added virtual spheres (yellow) filling the inner holes. On the right, the inner holes and the added virtual spheres are magnified for a better view.

- First, we fix the radii of virtual atoms as $\delta_j = \frac{1}{2} \text{dist} (\Gamma^e, \Gamma^i_j)$.
- Second, we take a small rectangular box containing the inner hole and take a set of virtual spheres with radius $\delta_j$ which cover this box completely, implying that the inner hole is also covered.
- Third, we remove all virtual spheres whose centers lie outside the inner hole and at the same time have distance to $\Gamma^i_j$ greater than $\delta_j$. In other words, we remove those virtual spheres which don’t intersect the inner hole and therefore don’t contribute to fill the inner hole.

As a consequence, the remaining virtual spheres (atoms) covers the inner hole enclosed by $\Gamma^i_j$ while do not intersect the exterior region of the SAS-cavity. By repeating the above steps for each inner cavity with the boundary $\Gamma^i_j$, $\forall 1 \leq j \leq n$, we can finally fill all inner holes with virtual spheres. Figure 3 illustrates the added virtual spheres for filling four inner holes of molecule 1ETN, one sphere for each hole.
section, we propose a meshing algorithm of molecular surface consisting of two sub-algorithms respectively for meshing a spherical patch and meshing a toroidal patch.

### 2.5.1 Boundary division

We first give a strategy for dividing the boundary of each (toroidal or spherical) patch on the SAS or the SES, which ensures that the meshes of two neighboring patches match on their interface, i.e., that the final global mesh will be conforming.

To divide the boundary of a toroidal or a spherical patch, we set initially the triangle size (the approximate length of a triangle edge) to $d$, which should be relatively small compared to the radius of the spherical patch. Since the boundary of a patch consists of loops which are composed of circular arcs, we make a uniform division of each circular arc on the boundary. The radius and the radian of a circular arc $l_m$ are denoted by $r_{lm}$ and $\theta_{lm}$. At the same time, we set a maximum allowed angle variation between two neighboring division points to $\alpha_0$ (in our codes, we use $\alpha_0 = 60^\circ$) in the case where the radius of the circular arc $r_{lm}$ is small compared to $d$. Then, the number of elements of the discretization of this circular arc denoted by $N_{lm}$ is set as follows:

$$N_{lm} = \max \left\{ \left\lfloor \frac{r_{lm} \theta_{lm}}{d} \right\rfloor + 1, \left\lfloor \frac{\theta_{lm}}{\alpha_0} \right\rfloor + 1 \right\}, \quad (2.5.1)$$

where $\lfloor \cdot \rfloor$ is the floor function which maps a real number to its largest smaller integer. As a consequence, this ensures that the distance and the angle variation between two neighboring division points are respectively smaller than $d$ and $\alpha_0$.

### 2.5.2 Spherical patches

In this subsection, we use the basic advancing-front method, see \[45,85,75,105,41\] for an overview of this technique, for meshing a (convex or concave, SAS or SES) spherical patch uniformly, with its center, its radius and its boundary information known. We only present the brief scheme of this method working on a sphere. However, we emphasize that any other suitable meshing algorithm can be applied to the spherical patch as long as they conserve the given boundary partition. For instance, the marching cubes algorithm \[77,42\] or the Delaunay refinement algorithm \[20,43\], can be taken into account. Therefore, the following algorithm can be replaced with another algorithm of choice.

The process of any advancing-front method can be summarized as follows:

1. Initialization of the front.
2. Creation of an internal element
   - determination of the departure zone;
Figure 4: Meshing two spherical patches.

- analysis of the entity and creation of (an) internal point(s) and (an) internal element(s);
- update the front.

(3) Repeat the creation of elements as long as the front is not empty.

For a given spherical patch, the initial front is chosen naturally to be its boundary which has been divided in Section 2.5.1. Then, a departure edge in the front is analyzed from which one or several new internal triangles are created. The front is updated and the process repeated until the front is empty, that is, when the front has merged and the spherical patch is entirely meshed.

To keep the completeness, we present in the Appendix B our implementation of the advancing-front algorithm. Also, we illustrate the mesh of a spherical patch that is obtained using this advancing-front algorithm in Figure 4.

Remark 2.5.1. Considering that the goal is to visualize molecular surfaces, the quality of mesh will not be focused on. If one aims to high-quality meshes, a remeshing process might be helpful or one can also use a different advancing-front algorithm.

2.5.2.1 Data structure

Denote by $P \in \mathbb{R}^{N_p \times 3}$ the array of coordinates of the $N_p$ points in the mesh, initialized to be the coordinates of all points of the boundary division. We define the orientation of any loop on the spherical patch satisfying that the interior of this patch is always on the right-hand side if one goes along the loop. As a consequence, each edge on a loop is endowed with an orientation, implying that we can classify its two endpoints to the right endpoint (or the starting endpoint) and the left endpoint (or the ending endpoint) viewing from the outside of the sphere where the patch lies, see the left of Figure 5 for a schematic.

Since the front might consist of several loops, we choose one of them as the active loop. Any edge on this active loop is called an active edge and any point on the active
Figure 5: On the left, the left endpoint and the right endpoint of edge $e$ are determined by the orientation of the loop. On the right, two dashed edges $e_1$ (in blue) and $e_2$ (in red) "intersect" in the sense that their two projected chords in form of circular arcs intersect on the sphere.

loop is called an active point. All active edges are sorted by the orientation of the active loop. We always choose the first active edge as the departure edge mentioned in section 2.5.2 and create a new triangle having the active edge as one side. Then, we update the set of active edges and go to the next active edge.

At each step, the set of active edges is represented by a matrix $A_e$ of size $N_{ae} \times 2$ where $N_{ae}$ is the number of active edges. Any active edge in $A_e$ is represented by the indices of its two endpoints in $P$. The set of triangles of the mesh at each step are represented by a matrix $T$ of size $N_t \times 3$ recording the indices of the three vertices of a triangle where $N_t$ is the number of triangles in the mesh.

2.5.2.2 Check front points

For a given active edge denoted by $e$ with the right endpoint $P_1$ and the left endpoint $P_2$, we want to construct a point $P_0$ for creating a new triangle with the edge $e$ and the opposite point $P_0$. The unit normal vectors at $P_1$ and $P_2$ on the spherical patch are denoted respectively by $\vec{n}_1$ and $\vec{n}_2$. Then, we define a unit vector $\vec{n}_e$ at the middle point of $e$ by

$$\vec{n}_e = \frac{\vec{P}_1 \vec{P}_2}{|\vec{P}_1 \vec{P}_2|} \times \vec{n}_1 + \vec{n}_2,$$

which is perpendicular to the edge $e$ and is tangent to the spherical patch at the same time. Geometrically speaking, $\vec{n}_e$ is a tangential unit vector to the sphere pointing towards the unmeshed region of the patch.

In the following, the notion of point-edge distance between a point and an edge is introduced and defined as the sum of the Euclidean distances between the point and two endpoints of the edge. Notice that each edge in the mesh is a chord of the corresponding sphere. Note that the projection of the chord onto the sphere is a
circular arc which lies on the plane generated by the chord and the spherical center. We say that two edges "intersect" if their projected chords in form of circular arcs intersect on the sphere, see the right of Figure 5 for a schematic of two "intersecting" edges.

For the edge $e$, the neighboring active edge with $P_1$ as a common endpoint is called the right neighboring active edge denoted by $e_1$. The one with $P_2$ as a common endpoint is called the left neighboring active edge denoted by $e_2$. The endpoint of $e_1$, other than $P_1$, is denoted by $P_1'$ and the endpoint of $e_2$, other than $P_2$, is similarly denoted by $P_2'$. By projecting the vectors $P_1P_1'$ and $P_1P_2'$ to the tangent plane to the sphere at point $P_1$, we obtain two vectors $\vec{1}\perp$ and $\vec{1}2$. The angle between $e$ and $e_1$, denoted by $\alpha_1$, is then defined to be the angle between vectors $\vec{1}\perp$ and $\vec{1}2$ if $\det(\vec{1}\perp, \vec{1}2, \vec{n}_1) \geq 0$ and to be $2\pi$ minus the angle between vectors $\vec{1}\perp$ and $\vec{1}2$ if $\det(\vec{1}\perp, \vec{1}2, \vec{n}_1) < 0$ where $\det(\cdot)$ denotes the determinant of a matrix. The angle between $e$ and $e_2$, denoted $\alpha_2$, is defined similarly to $\alpha_1$.

We first check if there exists possible points among the front points for the creation of a new triangle and collect all candidates in a set $S_f$. This means that we first try to create a new triangle with the existing front points without creating a new point. We propose two criterions for adding front points to $S_f$ as follows:

- **Angle Criterion**
  Set a minimal angle between two neighboring active edges to $\varepsilon_\alpha$. If $\alpha_1 < \varepsilon_\alpha$, add the left neighboring active point to $S_f$, see the left of Figure 6 for a schematic. If $\alpha_2 < \varepsilon_\alpha$, add the right neighboring active point to $S_f$.

- **Point-edge Criterion**
  Set a point-edge distance tolerance between a front point and a front edge to $\varepsilon_e$. For a front point $P_f$ from the set of points of all loops, other than $P_1$ or $P_2$, we check if it has a distance to $e$ smaller than $|e| + \varepsilon_e$ where $|e|$ is the length of $e$, see the right of Figure 6. Further, we check if the scalar product between $\vec{n}_e$ and the vector from the middle point of $e$ to $P_f$ is positive, and simultaneously if both edges $P_fP_1$ and $P_fP_2$ do not "intersect" any other front edge. If all conditions are satisfied, add $P_f$ to $S_f$.

The angle criterion is used to check if the angle between $e$ and one of its neighboring edges is small and the point-edge criterion is used to check if there exists any front point close to the edge $e$. As a consequence, $S_f$ is a list of front points $P_f$ that are possibly suited for creating a new triangle with the edge $e$ and the opposite point $P_f$.

If $S_f$ is not empty, we sort it with respect to the point-edge distance to the edge $e$ and choose the one denoted by $P_0$ that has the minimal distance to $e$, i.e.,

$$P_0 = \arg\min_{p \in S_f} \text{dist}(p, e),$$

(2.5.2)
where \( \text{dist}(p, e) \) is the point-edge distance between a point \( p \) and an edge \( e \). This ensures that any other point in \( S_f \) does not lie in the triangle \( \triangle P_0P_1P_2 \).

### 2.5.2.3 Create a new point

In the above section, we first scan the set of front points that can be used to create a new triangle for a given edge \( e \). Nevertheless, if \( S_f \) is empty, we should consider to create a new testing point \( P_{\text{test}} \) on the spherical patch. This testing point is constructed such that the edges \( P_{\text{test}}P_1 \) and \( P_{\text{test}}P_2 \) have the same length and that \( \triangle P_1P_2P_{\text{test}} \) has a fixed height \( h \) (we take \( h = \frac{\sqrt{3}}{2}d \)). In the case where \( |e| = d \), the triangle \( \triangle P_{\text{test}}P_1P_2 \) is equilateral. Then, we check if \( P_{\text{test}} \) is suited as a new vertex of the mesh with the following two criterions:

- **Point-edge Criterion**

  Check first if there exists a front edge \( e_f \) having a point-edge distance to \( P_{\text{test}} \) smaller than \( |e_f| + \varepsilon_e \), see the left of Figure 7. If yes, we further check for each endpoint of \( e_f \) (still denoted by \( P_f \)) if the scalar product between \( \overrightarrow{n_e} \) and the vector from the middle point of \( e_f \) to \( P_f \) is positive and simultaneously the edges \( P_fP_1 \) and \( P_fP_2 \) do not "intersect" any other front edge. If all conditions are satisfied for the edge \( e_f \) and the endpoint \( P_f \), we add \( P_f \) to \( S_f \).

- **Distance Criterion**

  Set a distance tolerance between a testing point and a front point to \( \varepsilon_d \). If there exists a front point \( P_f \) with distance to \( P_{\text{test}} \) smaller than \( \varepsilon_d \) (see the right of Figure 7) and both edges \( P_fP_1 \) and \( P_fP_2 \) do not "intersect" any other front edge, then we add \( P_f \) to \( S_f \).

If \( S_f \) is not empty now, we still select the point \( P_0 \) using formula (2.5.2) with this \( S_f \). If \( S_f \) is empty, we determine \( P_0 \) as \( P_{\text{test}} \).
2.5.2.4 Update front

The previous section was devoted to determine the point $P_0$ given an edge $e$. Then, a new triangle can be created by connecting the endpoints of $e$ and the point $P_0$.

After the creation of the triangle, we update the active loop including $A_e$ and $N_{ae}$, the set of vertices of the mesh including $P$ and $N_p$, as well the set of triangles $T$ and $N_t$. However, we should pay attention to two special cases of updating the active loop. If the point $P_0$ is a front point on the active loop but not a neighboring active point of $e$, the active loop is divided into two parts and we chose one of them to be the new active loop. If the point $P_0$ is a front point but not on the active loop, we add the loop on which $P_0$ lies to the active loop to form an active larger loop. After updating, we go to the next active edge on the active loop and repeat the process until the front has merged.

To obtain a mesh that is as uniform as possible, it is also necessary to control the length of each newly created edge. From the boundary division of circular arcs in Section 2.5.1, the length of any edge on the initial front is (in most cases, slightly) smaller than $d$. After the initialization, we bisect the newly created edge whenever its length is larger than a given tolerance $d_0$ and then map its middle point to the closest point on the sphere in order to obtain two new shorter edges. This technique ensures that each edge of the mesh will not become too large. Like this, we control the maximal diameter of each triangle.

In the advancing-front algorithm for meshing a spherical patch, we generate a surface mesh on the sphere which is essentially as difficult as generating a mesh for a planar domain in 2D. In this process, a new triangle can always be created by determining an optimal point $P_0$ and therefore the front will finally merge. This implies that a dead lock [105] of the front will not appear. From another point of view, we can first transform the spherical patch to a planar region, mesh it using a 2D mesh generator such as the NetGen [105] and transform the mesh back to the original spherical patch. As a consequence, the robustness of the proposed advancing-front algorithm can be guaranteed.
To mesh a toroidal SES patch, we should distinguish the cases of a rectangle-shaped patch and a double-triangle-shaped patch. We can parameterize a rectangle-shaped patch by defining a mapping from the toroidal patch to a rectangle having the same side length and the same boundary division as this patch. As a consequence, given the boundary division of the rectangle, we first mesh it in a trivial way (see the left of Figure 8 for a schematic) and then map the vertices of this mesh back to the patch to obtain the mesh of the rectangle-shaped patch. Similarly, we can parameterize a double-triangle-shaped patch by defining a mapping from this patch to two isosceles triangles respectively having the same side lengths and the same boundary divisions. We first mesh the two isosceles triangles (see the right of Figure 8 for a schematic) and then map the vertices of this mesh back to the patch to obtain the mesh of the double-triangle-shaped patch. Since the toroidal SES patch is part of a torus generated by rotating the spherical probe around an axis, these two mappings can be explicitly given using the parametric representation of a torus [115].

We illustrate the mesh of a rectangle-shaped toroidal patch on the left of Figure 9 and the mesh of a double-triangle-shaped toroidal patch on the right.
2.5.4 Mesh refinement

Once the mesh of a molecular patch is established, it is easy to refine it uniformly. Indeed, we can bisect each edge of the mesh and map its middle point to the closest point on the patch which can be computed given the data structure of the patch. Then, each triangle is replaced with four smaller triangles formed by the three vertices of this triangle and three closest points to the middle points of the three edges. As a consequence, the refined mesh consists of these smaller triangles. This process of refinement is quite efficient with the complexity proportional to the number of triangles in the mesh.

2.5.5 Remeshing

In many cases, high-quality meshes of molecular surfaces are required. One way to further improve the meshes generated by the above advancing-front algorithm is to use surface remeshing tools. With a remeshing tool, one can usually set some quality requirements and obtain an output mesh approximating well the input mesh. Figure 10 illustrates the mesh of caffeine using the remeshing tool MMGS developed by Dapogny et al. as an example. One should notice that in this case, the remeshing can not ensure that all vertices lie exactly on the molecular surfaces anymore but it can keep the initial vertices of the input mesh.

2.5.6 Illustrations

We visualize in Figure 11 the meshes of the eSES of some artificial molecules and in Figure 12 and Figure 13 the meshes of eSES of some non-artificial molecules. In both
cases, they are generated by the above-proposed meshing algorithm. In these figures, the green curves are the boundaries of all singular concave SES patches which are computed according to Theorem 2.2.2. On these singular patches, some singular arcs of the surface appear which might cause the self-intersection problem encountered in previous implementations. Generally speaking, the existence of singularities is quite often and the larger the molecule is, the more singularities exist.

In addition, we should mention that the techniques of molecular visualization (for example, [59, 58, 61, 64]) can also be combined with the molecular data structures constructed in Section 2.3, taking into account the complete characterization of the SES.
2.5.7 Computational cost

To get a first idea about the efficiency of the proposed algorithm, we present the total run time with respect to the number of atoms of different molecules. This program was run on a laptop with 2.5GHz quad-core Intel Core i7 processor in Matlab. Figure 14 demonstrates the relationship between the total run time and the size of molecule, where we observe almost a linear relationship. In order to lower the pre-constant of the linear scaling, one has to refer to a proper and more professional implementation in a better performing language. We therefore expect a better performance of the proposed method in fortran or C++.

2.6 Conclusion

In this chapter, we presented the construction of data structures for different molecular surfaces containing all information of their components. At the heart of our method is the recently developed singularity analysis of the SES which avoids the problem of self-intersection. This allows us to develop a meshing algorithm by meshing separately each patch, which includes two sub-algorithms respectively for meshing a (convex or concave, SAS or SES) spherical patch with an advancing-front method and for meshing a toroidal (SES) patch. It is also worth mentioning that each vertex of the created mesh lies exactly on the molecular surface. In addition, we propose an algorithm for filling molecular inner holes with virtual spheres since in some cases, the presence of these inner holes is unphysical, in particular in the context of solvation model. We provide therefore a way to treat these inner holes for

Figure 13: The eSES of molecule 4S19 with 3333 atoms (left) and the eSES of molecule 1IA0 with 9606 atoms (right) where $r_p = 1.5\text{Å}$. The green curves are the boundaries of singular concave patches.
more accurate chemical computation. In the future, we will focus on the computation of solvation energy in the polarizable continuum solvation model (PCM) using the SES-cavity, based on the complete characterization of the SES, together with the proposed algorithm for filling inner holes.
Part II

Domain Decomposition Method for Implicit Solvation Models
Chapter 3

Domain Decomposition Method for the Polarizable Continuum Model based on the Solvent Excluded Surface

Contents

3.1 Introduction ................................................. 98
  3.1.1 Problem statement ........................................ 99
  3.1.2 Contribution ............................................. 101
  3.1.3 Outline .................................................. 103
3.2 Solute-solvent boundary ...................................... 103
3.3 Dielectric permittivity function ................................ 105
3.4 Problem formulation and global strategy ..................... 107
  3.4.1 Problem formulation ....................................... 107
  3.4.2 Equivalent transformation .................................. 108
  3.4.3 Global strategy .......................................... 110
3.5 Domain decomposition strategy .............................. 111
3.6 Single-domain solvers ...................................... 112
  3.6.1 Laplace equation in a ball .............................. 112
  3.6.2 Generalized Poisson equation in a ball .................. 114
3.7 Numerical results .......................................... 119
  3.7.1 GP-solver test ........................................... 119
  3.7.2 Convergence of the global strategy ....................... 120
  3.7.3 Graphical illustration of the reaction potential ........ 121
This chapter has been submitted to a journal and has published online as a preprint [95]. In this chapter, an efficient solver for the polarizable continuum model in quantum chemistry is developed which takes the solvent excluded surface (the smooth molecular surface) as the solute-solvent boundary. This model requires to solve a generalized Poisson (GP) equation defined in $\mathbb{R}^3$ with a space-dependent dielectric permittivity function. First, the original GP-equation is transformed into a system of two coupled equations defined in a bounded domain. Then, this domain is decomposed into overlapping balls and the Schwarz domain decomposition method is used. This method involves a direct Laplace solver and an efficient GP-solver to solve the local sub-equations in balls. For each solver, the spherical harmonics are used as basis functions in the angular direction of the spherical coordinate system. A series of numerical experiments are presented to test the performance of this method.

### 3.1 Introduction

Many physical and chemical phenomena of interest in chemistry and biology take place in the liquid phase, and the solvent effects consequently play a crucial role in these processes. The typical situation is that a solute molecule (biological molecule for example) is embedded in the solvent. In this case, there are commonly two approaches to describing the solvent effects on the solute. The first one is to use an explicit solvation model, in which the simulated chemical system is composed of the solute molecule and a large number of explicit solvent molecules. The second one is to use an implicit solvation model, in which the solute molecule is embedded in a cavity surrounded by a continuous medium representing the solvent, i.e., the average response of the solvent molecules in the sense of statistical mechanics. Note that the cavity here is not physical but is artificially defined in an ad hoc way.

The implicit solvation models commonly take into account the polarization effects of the solvent and involve elliptic operators from the mathematical point of view. Despite the fact that these models are widely used in the chemistry community, little interaction with applied mathematics can be observed. There are still many open problems, in particular on the suitable definition of the cavity, the nature of the continuum media, the sense in which it is a proper approximation, and also that, the solutions to partial differential equations (PDEs) need to be approximated.

In this chapter, we focus on developing an efficient solver for a popular implicit
solvent model, the polarizable continuum model (PCM), in which the solvent is represented as a polarizable continuum (see [21, 109, 82] for an overview). Within this chapter, the ESU-CGS (electrostatic units, centimetre-gram-second) system of units [13, Appendix C] is used for all equations.

3.1.1 Problem statement

Once the cavity surrounding the solute molecule is defined, the electrostatic potential $\psi$ in an implicit solvation model is described by the following partial differential equation (PDE)

$$-\nabla \cdot \varepsilon(x) \nabla \psi(x) = 4\pi \rho(x), \quad \text{in} \ \mathbb{R}^3$$

(3.1.1)

where it is assumed that $\psi(x) \sim O(\frac{1}{|x|})$ as $|x| \to \infty$, $\varepsilon(x)$ represents the space-dependent dielectric permittivity parameter and $\rho(x)$ represents the charge distribution of the solvation system.

Based on Eq. (3.1.1), one can derive two different models, respectively for the non-ionic solvent and the ionic solvent. First, one can obtain a generalized Poisson (GP) equation if $\rho$ is independent on $\psi$ as for the non-ionic solvent and second, a Poisson-Boltzmann equation [33] is obtained if the solvent contains ions whose movement is accounted for by Boltzmann statistics. Since the end of 1980s, many codes have been proposed for solving the Poisson-Boltzmann (or Poisson) equation for example using the boundary element method [15, 78], the finite element method [3, 53] or the finite difference method including UHBD [32], DelPhi [68], APBS [9, 36, 58] and the other work [39]. In particular, the APBS software is popular and widely used which can calculate the biomolecular electrostatics for large molecules. In the case of ionic solvent, the electrostatic potential decays exponentially with respect to the distance from the solute-solvent interface, which allows researchers to develop methods on bounded domains. However, in the case of non-ionic solvent, the electrostatic potential only decays as $\frac{1}{|x|}$ due to the absence of ionic screening, which might require to take into account an integral representation of the potential on the entire unbounded domain for better accuracy. In this chapter, we focus on the case of non-ionic solvent and aim to solve the GP equation (3.1.1) for the PCM without ions.

In a classical PCM with solute cavity $\Omega$, the dielectric permittivity $\varepsilon(x)$ is defined as follows (see [26])

$$\varepsilon(x) = \begin{cases} 
1 & x \in \Omega, \\
\varepsilon_s & x \in \Omega^c := \mathbb{R}^3 \setminus \bar{\Omega},
\end{cases}$$

(3.1.2)

where $\varepsilon_s$ is the solvent dielectric constant and $\varepsilon(x)$ has a jump on the solute-solvent boundary $\Gamma := \partial \Omega$. Since the solvent is modeled as a polarizable continuum without ions, $\rho(x)$ is a function supported in the solute cavity. Physically speaking, this corresponds to the situation where the solute molecule is embedded in a homogeneous
solvent of dielectric constant $\varepsilon_s$.

A reduced version of PCM is the conductor-like screening model (COSMO) \cite{60} where the solvent is idealized as a conductor-like continuum, that is to say, $\varepsilon_s = \infty$. In this case, the original equation (3.1.1) is simply a Laplace equation with homogeneous Dirichlet boundary condition

$$\begin{align*}
-\Delta \psi(x) &= 4\pi \rho(x) \quad \text{in } \Omega, \\
\psi(x) &= 0 \quad \text{on } \Gamma.
\end{align*}$$

This reduced model is usually employed to approximate the PCM when the solvent dielectric constant $\varepsilon_s$ is relatively large, for example, the (relative) dielectric constant of water is $\varepsilon_s = 78.4$ at room temperature ($25^\circ C$). To solve the COSMO, the finite element method or the finite difference method can be used. However, the computational cost is too large to obtain a high degree of accuracy for a large realistic molecule. In particular, meshing the solute cavity of a complicated molecule is already costly. In the past several years, a Schwarz domain decomposition method for solving the COSMO \cite{96, 107} (called the ddCSOMO) has been developed, for the solute cavity constituted by a union of balls \cite{25, 72, 69}. This method has attracted much attention due to its impressive efficiency, that is, it performs about two orders of magnitude faster than the other equivalent methods \cite{69}. Further, the convergence of the ddCOSMO has been analyzed recently in \cite{29}, where the contraction property is obtained in a simplified setting.

Later on, a similar Schwarz domain decomposition method to solve Eqs (3.1.1)–(3.1.2) was proposed \cite{108} (called the ddPCM), which is based on the integral equation formalism of the PCM \cite{26}. The basic idea of the ddPCM (resp. the ddCOSMO) is to decompose the solute cavity into a union of balls and to solve each sub-equation in a ball, which avoids meshing the cavity. These two methods are limited to a domain constituted by the union of balls, such as the van der Waals (VdW) cavity and the solvent accessible surface (SAS) cavity \cite{30}, see the upcoming Section 2 where we will introduce those notions.

Due to the fast resolution of the COSMO and the PCM (see also website \cite{70}), one starts to raise the question of a better definition of the solute cavity $\Omega$ in order to improve the methodology. The solvent excluded surface (SES, see Figure 2) cavity \cite{30, 74, 93} seems to be a more suitable choice, because it has a stronger physical meaning that this cavity represents the region where the solvent molecules (represented by idealized spheres) can not touch. It has been confirmed in some chemical calculation that using the SES-cavity $\Omega_{\text{ses}}$ can yield more accurate results, such as in \cite{98, 91}. In fact, an SES-based solver for the implicit solvation model has been developed in \cite{50, 17}, using the integral equation of the PCM and an efficient mesh generator on the SES. In this chapter, we take $\Omega_{\text{ses}}$ as the solute cavity, i.e., $\Omega = \Omega_{\text{ses}}$. One should note that the SES has a more complicated geometry than the VdW surface and the SAS.
Further, there have been some attempts to introduce a continuous dielectric permittivity near the solute-solvent boundary \[56, 47, 80, 13\]. Mathematically speaking, this means that \(\varepsilon(x)\) is also space-dependent in the solvent region:

\[
\varepsilon(x) = \begin{cases} 
1 & x \in \Omega, \\
\varepsilon_s(x) & x \in \Omega^c.
\end{cases}
\] (3.1.4)

Due to the presence of the solute, the solvent density is not constant (especially for large biomolecules) near the solute-solvent boundary and therefore, the solvent dielectric permittivity is not constant. But at positions far from the solute molecule, the solvent dielectric permittivity \(\varepsilon_s(x)\) should be approximately the (bulk) solvent dielectric constant \(\varepsilon_s\), since the solvent molecules are influenced little by the solute molecule. Denoting by \(\Omega_\infty\) the region far from the solute molecule, \(\varepsilon_s(x)\) then satisfies

\[
\varepsilon_s(x) = \varepsilon_s, \quad \text{in } \Omega_\infty,
\] (3.1.5)

which yields that \(\psi\) of Eq. (3.1.1) is harmonic in \(\Omega_\infty\), i.e. \(-\Delta \psi = 0\) in \(\Omega_\infty\).

In summary, we want to solve Eq. (3.1.1) with a space-dependent parameter \(\varepsilon(x)\) in the form of (3.1.4) – (3.1.5) where \(\Omega\) is taken as \(\Omega_{\text{ses}}\).

### 3.1.2 Contribution

The contribution of this chapter is two-fold. On the one hand, we construct a continuous dielectric permittivity function for the SES-based PCM. On the other hand, we introduce a Schwarz domain decomposition method to solve the established model. This method consists of transforming the original equation into two coupled (by a non-local condition) equations defined on a bounded domain, decomposing the bounded domain into balls and solving only local equations in balls by a spectral method.

The idea of this domain decomposition method is illustrated briefly in Figure 1. We consider the following problem in a general form

\[
\begin{cases} 
-\nabla \cdot \varepsilon \nabla u = f & \text{in } \Omega_0 := \Omega^c, \\
-\Delta u = 0 & \text{in } \Omega_\infty, \\
[u] = 0 & \text{on } \Gamma_0 := \partial \Omega_0, \\
[\partial_n u] = 0 & \text{on } \Gamma_0,
\end{cases}
\]

where \([u]\) and \([\partial_n u]\) denote the jump of \(u\) and its normal derivative on \(\Gamma_0\), \(\varepsilon\) is a continuous function (in particular across \(\Gamma_0\)). The above equation can be transformed
into two coupled equations defined on $\Omega_0$ as follows

$$
\begin{cases}
-\nabla \cdot \varepsilon \nabla u = f & \text{in } \Omega_0, \\
u = g & \text{on } \Gamma_0,
\end{cases}
$$

and

$$
\begin{cases}
-\Delta u_\infty = 0 & \text{in } \Omega_0, \\
u_\infty = g & \text{on } \Gamma_0,
\end{cases}
$$

where a coupling condition arises through the auxiliary function $g$ defined as

$$
g = \frac{1}{4\pi} S_{\Gamma_0}(\partial_n u_\infty - \partial_n u), \quad \text{on } \Gamma_0.
$$

The operator $S_{\Gamma_0} : H^{-\frac{1}{2}}(\Gamma_0) \to H^{\frac{1}{2}}(\Gamma_0)$ is the single-layer operator on $\Gamma_0$, which maps the function $\partial_n u_\infty - \partial_n u$ to another function in $H^{\frac{1}{2}}(\Gamma_0)$, see Section 3.4.2 for the definition of $S_{\Gamma_0}$. Here, $H^{-\frac{1}{2}}(\Gamma_0)$ and $H^{\frac{1}{2}}(\Gamma_0)$ denote the usual Sobolev spaces of order $\pm \frac{1}{2}$ on the surface $\Gamma_0$, see [2]. The bounded domain $\Omega_0$ is taken as a union of balls, derived from the geometric data structures of the solute molecule. Thus, we can use the Schwarz domain decomposition method and only solve local problems in balls.

Ultimately, a Laplace solver and a GP solver are developed respectively to solve the Laplace equation and the GP equation in a ball. For each solver, we propose an efficient spectral method within each ball, using the spherical harmonics as basis functions in the angular direction. It is important to note that this algorithm does

Figure 1: Schematic diagram of the domain decomposition method.
not require any mesh, but only involves problems in balls that are coupled to each other.

3.1.3 Outline

In Section 3.2, we first introduce different solute-solvent boundaries including the VdW surface, the SAS and the SES, which are fundamental and classical concepts of the implicit solvation models, which however are mostly unknown to the applied mathematics community. In Section 3.3, we construct a continuous dielectric permittivity function \( \varepsilon(x) \) of PCM, ensuring that the SES-cavity always has the dielectric constant of vacuum. Then, in Section 3.4, we present the electrostatic problem of the PCM, its equivalent transformation, and a global iterative strategy for solving it. In Section 3.5, we introduce the scheme of the domain decomposition method for solving the associated partial differential equations in the global strategy. This requires to develop a Laplace solver and a GP-solver in the ball, which are presented in Section 3.6. After that, in Section 3.7, we give a series of numerical experiments on the performance of the proposed method. In the last section, we draw some conclusions.

3.2 Solute-solvent boundary

An important ingredient of an implicit solvation model is the solute-solvent boundary, that is, the interface between the solute and the solvent.

Indeed, the choice of a proper solute cavity is important as pointed out in [110, Section II. C.]: \textit{The shape and size of the cavity are critical factors in the elaboration of a method. An ideal cavity should reproduce the shape of the solute, with the inclusion of the whole charge distribution and with the exclusion of empty spaces which can be filled by the solvent continuous distribution.} If the cavity is too large, the solvation effects are damped; if it is too small, serious errors may arise in the evaluation of the interaction energy for the portions (atoms or bonds) near the solute-solvent boundary. One simple choice of the solute-solvent boundary is the VdW surface which is the topological boundary of the union of solute’s VdW-atoms with radii experimentally fitted. Another choice is the SAS denoted by \( \Gamma_{\text{sas}} \), which is defined by tracing the center of an idealized (spherical) solvent probe (representing a solvent molecule) when rolling over the solute molecule. The region enclosed by the SAS is called the SAS-cavity, denoted by \( \Omega_{\text{sas}} \). See Figure 2 for a 2D schematic diagram of different molecular surfaces.

The problem is that the VdW-cavity and the SAS-cavity are topologically not the correct answers to the cavity problem as they describe poorly the region where the solvent can touch. However, they are topologically simple and therefore attractive for numerical computations. Another solute-solvent boundary is the SES denoted
3.2. Solute-solvent boundary

Figure 2: 2D schematic diagram of the VdW surface (green), the SAS (blue) and the SES (red).

Figure 3: The SAS (left, $\Gamma_{\text{sas}}$) and the SES (right, $\Gamma_{\text{ses}}$) of caffeine. The SES is composed of convex spherical patches (red), toroidal patches (yellow) and concave spherical patches (blue).

by $\Gamma_{\text{ses}}$, which represents the boundary of the region where the probe has no access due to the presence of the solute. The region enclosed by the SES is the SES-cavity, denoted by $\Omega_{\text{ses}}$. See Figure 3 for a graphical illustration of the SAS and the SES of caffeine derived from our meshing algorithm [93, 94]. In fact, any point on the SES has a constant distance to the SAS, equal to the solvent probe radius. From the geometrical point of view, the SES is smoother but also more complicated than the SAS.

To set the notations, we suppose that the solute molecule is composed of $M$ atoms and the $j$th atom has center $c_j$ and VdW radius $r_j$. The solvent probe radius is denoted by $r_p$. Furthermore, for each atom, we define an “enlarged” ball $\Omega_j$ with center $c_j$ and radius $R_j = r_j + r_p + r_0$, where $r_0$ is a nonnegative constant. It is assumed that the solvent dielectric permittivity is constant outside the union of these enlarged balls. This is reasonable because the solvent density at positions far from the solute molecule (bulk) is approximately the same and therefore, the dielectric permittivity determined by the solvent density is almost the same [17, 13]. The SES-cavity is
completely covered by the union $\Omega_0$ of enlarged balls, that is,

$$\Omega_{ses} \subset \Omega_0 := \bigcup_{j=1}^{M} \Omega_j, \quad \text{with} \quad \Omega_j := B_{R_j}(c_j). \quad (3.2.1)$$

Here, we have specified the notation $\Omega_0$ mentioned in the introduction.

Denote the signed distance function to the SAS by $f_{sas}$ (negative inside the SAS and positive outside the SAS). We then have a mathematical characterization of the two cavities:

$$\Omega_{ses} = \{ x \in \mathbb{R}^3 : f_{sas}(x) \leq -r_p \} \quad \text{and} \quad \Omega_0 = \{ x \in \mathbb{R}^3 : f_{sas}(x) \leq r_0 \}.$$ 

Also, we have a characterization of their boundary surfaces

$$\Gamma_{ses} = f_{sas}^{-1}(-r_p) \quad \text{and} \quad \Gamma_0 = f_{sas}^{-1}(r_0).$$

An efficient method was proposed in [93] to compute $f_{sas}$ analytically, which is based on finding a closest point on the SAS to an arbitrary point. This allows us to construct a continuous permittivity function based on the value of $f_{sas}$.

### 3.3 Dielectric permittivity function

In this section, we construct a SES-based dielectric permittivity function $\varepsilon(x)$ associated with $f_{sas}$ as mentioned above.

![Dielectric Permittivity](image)

**Figure 4:** Schematic diagram of the dielectric permittivity function $\varepsilon(x)$ with respect to $f_{sas}$. The intermediate layer $\mathcal{L}$ (switching region) is bounded by two dashed lines (red), i.e., the region where $-r_p \leq f_{sas} \leq r_0$. 


In the following papers about implicit solvation models [56, 47, 80, 13] (see [109, Section 4.5] for more references), the motivation for taking into account continuous dielectric permittivity functions comes from the fact that the solvent density varies quite importantly around the solute molecule, especially for large biomolecules. As a consequence of this variation of the solvent density itself between near the solute-solvent boundary and at a faraway distance, there is a modification of the value of dielectric permittivity. Taking the SES as the solute-solvent boundary implies that the dielectric permittivity in the SES-cavity is always 1, i.e., the dielectric constant of vacuum. As assumed in the previous section, the dielectric permittivity outside the enlarged cavity $\Omega_0$ is the solvent dielectric constant $\varepsilon_s$. The remaining work is to determine $\varepsilon(x)$ in the intermediate layer (the switching region) $\mathcal{L} := \Omega_0 \setminus \Omega_{\text{ses}}$. To construct a more physical permittivity function, we propose the following definition of $\varepsilon(x)$ depending on $f_{\text{sas}}$,

$$
\varepsilon(x) = \begin{cases} 
1 & x \in \Omega_{\text{ses}}, \\
\xi \left( \frac{f_{\text{sas}}(x) + r_p}{r_p + r_0} \right) & x \in \mathcal{L}, \\
\varepsilon_s & x \in \Omega_\infty := \Omega_{0}^c,
\end{cases}
$$

(3.3.1)

where $\xi$ is a continuous function defined on $[0, 1]$, satisfying $\xi(0) = 1$, $\xi(1) = \varepsilon_s$. $\varepsilon(x)$ can be seen as a distance-dependent function where the “distance” represents the signed distance to the SAS, see Figure 4 for a schematic diagram. To guarantee the smoothness of the dielectric permittivity on the interfaces $\Gamma_{\text{ses}}$ and $\Gamma_0$, one can impose more conditions to the function $\xi$, for example,

$$
\xi'(0) = 0, \quad \xi'(1) = 0, \quad \xi''(0) = 0, \quad \xi''(1) = 0.
$$

(3.3.2)

**Remark 3.3.1.** We emphasize that the function $\xi$ can be chosen in many different manners satisfying various conditions. This characterizes the way the dielectric permittivity varies in the intermediate switching region.

The above SES-based PCM can be used to approximate the classical PCM with the discontinuous dielectric permittivity function (3.1.2). If one chooses a sequence of functions $\xi(t)$ approaching the discontinuous function

$$
\chi_{\text{ses}}(t) = \begin{cases} 
1 & t < 0, \\
\varepsilon_s & t \geq 0,
\end{cases}
$$

(3.3.3)

then the governing equations of the SES-based PCM approach the equations of the classical PCM with $\Omega = \Omega_{\text{ses}}$. Similarly, if one chooses another sequence of functions $\xi(t)$ approaching the discontinuous function

$$
\chi_{\text{sas}}(t) = \begin{cases} 
1 & t < 1, \\
\varepsilon_s & t \geq 1,
\end{cases}
$$

(3.3.4)
then the governing equations of the SES-based PCM approach the equations of the classical PCM with \( \Omega = \Omega_{\text{sas}} \).

### 3.4 Problem formulation and global strategy

In this section, we first introduce the electrostatic problem of the PCM and then present the techniques of transforming the original problem into two coupled problems defined on a bounded domain. After the transformation, we propose a global iterative strategy to solve the electrostatic problem.

#### 3.4.1 Problem formulation

One crucial issue on the PCM is to compute the electrostatic contribution to the solvation energy, denoted by \( E^s \). This electrostatic contribution can be written as

\[
E^s = \frac{1}{2} \int_{\mathbb{R}^3} \rho(x) W(x) \, dx, \tag{3.4.1}
\]

where \( W \) denotes the (solvent) reaction potential generated by the solute’s charge density \( \rho \) in presence of the polarizable continuum. The governing equations of the potential \( W \) will be explained later in this section.

We assume that the electrostatic potential \( \Phi \) generated by the charge distribution \( \rho \) in vacuum is already known and \( \rho \) is supported in \( \Omega_{\text{ses}} \), i.e. \( \text{supp}(\rho) \subset \Omega_{\text{ses}} \), which is a standard assumption in the content of PCM. As a consequence, \( \Phi \) satisfies the following PDE

\[
-\Delta \Phi(x) = 4\pi \rho(x), \quad \text{in } \mathbb{R}^3, \tag{3.4.2}
\]

As in [25], one always takes a neutral solute molecule with a classical charge distribution for simplicity,

\[
\rho(x) = \sum_{j=1}^{M} q_j \delta_{c_j}(x), \quad \forall x \in \mathbb{R}^3, \tag{3.4.3}
\]

where \( q_j \) represents the charge of the \( j \)th atom and \( \delta_{c_j} \) is the Dirac function at the atomic center \( c_j \). Then, the electrostatic potential \( \Phi \) generated by \( \rho \) is derived

\[
\Phi(x) = \sum_{j=1}^{M} \frac{q_j}{|x - c_j|}, \quad \forall x \in \mathbb{R}^3. \tag{3.4.4}
\]

The reaction potential \( W := \psi - \Phi \) represents the difference between the electrostatic potentials with and without the presence of solvent. For a SES-based PCM
with \( \varepsilon(x) \) given in (3.3.1), \( W \) satisfies consequently the following PDEs

\[
\begin{aligned}
-\Delta W &= 0 \quad \text{in } \Omega_{\text{ses}}, \\
-\nabla \cdot \varepsilon \nabla (W + \Phi) &= 0 \quad \text{in } \mathcal{L}, \\
-\Delta W &= 0 \quad \text{in } \Omega_{\infty}.
\end{aligned}
\] (3.4.5)

Further, since \( \varepsilon(x) \) is continuous across the interfaces \( \Gamma_{\text{ses}} \) and \( \Gamma_0 \), the following classical jump-conditions are required

\[
\begin{aligned}
[W] &= 0 \quad \text{on } \Gamma_{\text{ses}} \text{ and } \Gamma_0, \\
[\partial_n W] &= 0 \quad \text{on } \Gamma_{\text{ses}} \text{ and } \Gamma_0,
\end{aligned}
\] (3.4.6)

where \([W]\) denotes the jump (inside minus outside) of the reaction potential \( W \) on the boundary surfaces \( \Gamma_{\text{ses}} \) and \( \Gamma_0 \), \([\partial_n W]\) denotes the jump of the normal derivative of the reaction potential, \( n \) denotes the unit normal vector on \( \Gamma_0 \) pointing outwards with respect to \( \Omega_0 \), \( \partial_n W \) denotes the normal derivative \( \nabla W \cdot n \).

### 3.4.2 Equivalent transformation

Since \( \text{supp}(\rho) \subset \Omega_{\text{ses}} \), we have \( -\Delta \Phi = 0 \) in \( \mathcal{L} \). Combining with \( \varepsilon(x) = 1 \) in \( \Omega_{\text{ses}} \), the above Eqs (3.4.5)–(3.4.6) can be recast as

\[
\begin{aligned}
-\nabla \cdot \varepsilon \nabla W &= \nabla \cdot (\varepsilon - 1) \nabla \Phi \quad \text{in } \Omega_0, \\
-\Delta W &= 0 \quad \text{in } \Omega_{\infty},
\end{aligned}
\] (3.4.7)

with

\[
\begin{aligned}
[W] &= 0 \quad \text{on } \Gamma_0, \\
[\partial_n W] &= 0 \quad \text{on } \Gamma_0.
\end{aligned}
\] (3.4.8)

See Figure 5 for the schematic diagram of the PDEs in different regions.

In the second equation of (3.4.7), a single layer potential \( \widetilde{S}_{\Gamma_0} : H^{-\frac{1}{2}}(\Gamma_0) \to H^1(\mathbb{R}^3) \) can be used to represent the electrostatic potential \( W \) restricted in \( \Omega_{\infty} \) as follows

\[
W|_{\Omega_{\infty}}(x) = (\widetilde{S}_{\Gamma_0} \sigma)(x) := \int_{\Gamma_0} \sigma(s') \frac{1}{|x - s'|} ds', \quad \forall x \in \Omega_{\infty},
\] (3.4.9)

where \( H^{-\frac{1}{2}}(\Gamma_0) \) and \( H^1(\Gamma_0) \) denote the Sobolev spaces, \( \sigma \) is some function in \( H^{-\frac{1}{2}}(\Gamma_0) \). From the continuity of the single-layer potential across the interface, see for example [10, 49], we take the limit to \( \Gamma_0 \), so as to obtain the integral equation

\[
W|_{\Gamma_0}(s) = (S_{\Gamma_0} \sigma)(s) := \int_{\Gamma_0} \sigma(s') \frac{1}{|s - s'|} ds', \quad \forall s \in \Gamma_0,
\] (3.4.10)
where $S_{\Gamma_0} : H^{-\frac{1}{2}}(\Gamma_0) \rightarrow H^{\frac{1}{2}}(\Gamma_0)$ is the single-layer operator which is invertible (see also [103, 49]) and thus defines $\sigma \in H^{-\frac{1}{2}}(\Gamma_0)$. Further, we can artificially extend the potential in (3.4.9) to $\Omega_0$ as follows

$$W_1(x) := \int_{\Gamma_0} \frac{\sigma(s')}{|x-s'|} \, ds', \quad \forall x \in \Omega_0, \quad \text{(3.4.11)}$$

satisfying

$$-\Delta W_\infty = 0, \quad \text{in } \Omega_0 \quad \text{(3.4.12)}$$

and

$$W_\infty = W|_{\Omega_\infty}, \quad \text{on } \Gamma_0. \quad \text{(3.4.13)}$$

Recalling the relationship between the charge density $\sigma$ and the jump of the normal derivative of the potential on $\Gamma_0$ generated by $\sigma$ [108], we actually have

$$\sigma = \frac{1}{4\pi} (\partial_n W_\infty - \partial_n W|_{\Omega_\infty}), \quad \text{on } \Gamma_0. \quad \text{(3.4.14)}$$

Combining with $[\partial_n W] = 0$ on $\Gamma_0$ in (3.4.8), i.e., $\partial_n W|_{\Omega_0} = \partial_n W|_{\Omega_\infty}$, we therefore obtain

$$\sigma = \frac{1}{4\pi} (\partial_n W_\infty - \partial_n W|_{\Omega_0}), \quad \text{on } \Gamma_0. \quad \text{(3.4.15)}$$

Also, since $[W] = 0$ in (3.4.8) and $W_\infty = W|_{\Omega_\infty}$ on $\Omega_0$, we have $W_\infty = W$ on $\Gamma_0$.

Finally, let’s summarize what we have in hand now. We have obtained two PDEs of $W$ and $W_\infty$ both defined on $\Omega_0$

$$\begin{cases} -\nabla \cdot \varepsilon \nabla W = \nabla \cdot (\varepsilon - 1) \nabla \Phi & \text{in } \Omega_0, \\ -\Delta W_\infty = 0 & \text{in } \Omega_0, \end{cases} \quad \text{(3.4.16)}$$
and two boundary conditions coupling them
\[
\begin{aligned}
W &= W_\infty \quad \text{on } \Gamma_0, \\
\sigma &= \frac{1}{4\pi} (\partial_n W_\infty - \partial_n W) \quad \text{on } \Gamma_0,
\end{aligned}
\]  
(3.4.17)
where \(\sigma\) is the density generating \(W_\infty\) by \(3.4.11\). To compute the electrostatic contribution to the solvation energy, we need to compute \(W\) which involves solving Eqs \((3.4.16)-(3.4.17)\). Although we solve for a problem in an unbounded domain, we need to solve equations only on the bounded domain \(\Omega_0\) which is a remarkable property.

### 3.4.3 Global strategy

From the above formulation, we propose the following iterative procedure for solving Eqs \((3.4.16)-(3.4.17)\): Let \(g^0\) be an initial guess for the Dirichlet condition \(W|_{\Gamma_0} = W_\infty|_{\Gamma_0}\) on \(\Gamma_0\) and set \(k = 1\).

[1] Solve the following Dirichlet boundary problem for \(W^k\):
\[
\begin{aligned}
-\nabla \cdot \varepsilon \nabla W^k &= \nabla \cdot (\varepsilon - 1) \nabla \Phi \quad \text{in } \Omega_0, \\
W^k &= g^{k-1} \quad \text{on } \Gamma_0,
\end{aligned}
\]  
(3.4.18)
and derive its Neumann boundary trace \(\partial_n W^k\) on \(\Gamma_0\).

[2] Solve the following Dirichlet boundary problem for \(W^\infty_k\):
\[
\begin{aligned}
-\Delta W^\infty_k &= 0 \quad \text{in } \Omega_0, \\
W^\infty_k &= g^{k-1} \quad \text{on } \Gamma_0,
\end{aligned}
\]  
(3.4.19)
and derive similarly its Neumann boundary trace \(\partial_n W^\infty_k\) on \(\Gamma_0\).

[3] Build the charge density \(\sigma^k = \frac{1}{4\pi} (\partial_n W^\infty_k - \partial_n W^k)\) and compute a new Dirichlet condition \(g^k = S_{\Gamma_0} \sigma^k\).

[4] Compute the electrostatic contribution \(E_s^k\) to the solvation energy following \((3.4.11)\) based on \(W^k\) at the \(k\)-th iteration, set \(k \leftarrow k + 1\), go back to Step [1] and repeat until the increment of electrostatic interaction \(|E_s^k - E_s^{k-1}|\) becomes smaller than a given tolerance \(Tol \ll 1\).

**Remark 3.4.1.** In order to provide a suitable initial guess of \(g^0\), we consider the (unrealistic) scenario where the whole space \(\mathbb{R}^3\) is covered by the solvent medium with the dielectric constant \(\varepsilon_s\). In consequence, the electrostatic potential \(\psi\) is given by \(\psi = \frac{1}{\varepsilon_s} \Phi\) and the reaction potential is provided by \(W = \psi - \Phi = \left(\frac{1}{\varepsilon_s} - 1\right) \Phi\).
We therefore propose the following initial Dirichlet boundary function \( g^0 \) for the first iteration:

\[
g^0 = \left( \frac{1}{\varepsilon_s} - 1 \right) \Phi. \tag{3.4.20}
\]

**Remark 3.4.2.** This iterative procedure has the remarkable property that we solve a problem on an unbounded domain by a sequence of problems on bounded domains only. It can be seen as a domain decomposition method on the two non-overlapping domains \( \Omega_0 \) and \( \Omega_\infty \) where only problems on the bounded domain \( \Omega_0 \) are solved. Similar property has been obtained for the integral equation formulation of implicit solvation models, see for example [22, 24, 62, 11].

In the next section, we propose to use the classical Schwarz domain decomposition method to solve the PDE (3.4.18) in Step [1] and the PDE (3.4.19) in Step [2] by introducing sub-iterations.

### 3.5 Domain decomposition strategy

The Schwarz domain decomposition method [96, 107] aims at solving partial differential equations defined on complex domains which can be decomposed as a union of overlapping and possibly simple subdomains. For each subdomain, the same equation is solved but with boundary conditions that depend on the global boundary condition on one hand and on the neighboring solutions on the other hand.

Recalling that we have a natural domain decomposition of \( \Omega_0 \) as follows

\[
\Omega_0 = \bigcup_{j=1}^{M} \Omega_j, \quad \Omega_j = B_{R_j(c_j)},
\]

the Schwarz domain decomposition method can be applied to solve the PDE (3.4.18). We replace the global equation (3.4.18) by the following coupled equations, each restricted to \( \Omega_j \):

\[
\begin{align*}
-\nabla \cdot \varepsilon \nabla W_j &= \nabla \cdot (\varepsilon - 1) \nabla \Phi \quad \text{in } \Omega_j, \\
W_j &= h_j \quad \text{on } \Gamma_j,
\end{align*}
\]

where \( W_j = W|_{\Omega_j}, \Gamma_j = \partial \Omega_j \) and

\[
h_j = \begin{cases} W^N_j & \text{on } \Gamma^i_j, \\
g & \text{on } \Gamma^e_j. \end{cases} \tag{3.5.2}
\]

Here, we omit the superscript due to the (outer) iteration index \( k \). \( \Gamma^e_j \) is the external part of \( \Gamma_j \) not contained in any other ball \( \Omega_i \) (\( i \neq j \)), i.e., \( \Gamma^e_j = \Gamma_0 \cap \Gamma_j \), \( \Gamma^i_j \) is the
internal part of $\Gamma_j$, i.e., $\Gamma^i_j = \Omega_0 \cap \Gamma_j$ (see Figure 1 for an illustration), and

$$W_j^N(s) = \frac{1}{|\mathcal{N}(j, s)|} \sum_{i \in \mathcal{N}(j, s)} W_i(s), \quad \forall s \in \Gamma^i_j,$$ (3.5.3)

where $\mathcal{N}(j, s)$ represents the index set of all balls that overlap $\Omega_j$ at $s$. In fact, for a fixed point $s \in \Gamma^i_j$, we enforce $W_j = W_j^N(s)$.

In the next section, we will develop a GP-solver for solving the local problems (3.5.1). For each local problem defined on $\Omega_j$, this solver provides an approximate weak solution. Based on this solver, an iterative procedure can be applied to solve the coupled equations (3.5.1)–(3.5.2), such as the parallel and alternating Schwarz algorithms as presented in [25]. The idea of the parallel Schwarz algorithm is to solve each local problem based on the boundary condition of the neighboring solutions derived from the previous iteration. During this iterative procedure, the computed value of $W|_{\Gamma_j}$ is updated step by step and converges to the exact value.

The parallel and alternating Schwarz algorithms might not be the most efficient way to solve this set of equations, but is well-suited to illustrate the idea of domain decomposition. In practice, a global problem (linear system after introducing a discretization) can be solved with GMRes for example.

Remark 3.5.1. Notice that the global strategy for computing the electrostatic contribution to the solvation energy in Section 3.4.3 is also an iterative process (it was indexed by $k$). To distinguish these two iterations, the global iteration in Section 3.4.3 is called the outer iteration and the iteration of solving the GP-equation (3.4.18) by iteratively solving the set of local problems (3.5.1) is called the inner iteration. Of course, for the sake of efficiency, these two iterations (inner and outer) should be intertwined.

As to solve the Laplace equation (3.4.19), we still use the domain decomposition method which is called the ddCOSMO method in this context and has been developed in [25]. In fact, Eq. (3.4.19) can been seen as a special case of Eq. (3.5.1) when $\varepsilon = 1$. As a consequence, the same domain decomposition method as presented in (3.5.1)–(3.5.2) can be used where each local problem (3.5.1) simplifies to a Laplace problem.

3.6 Single-domain solvers

3.6.1 Laplace equation in a ball

As seen above, it is required to solve a set of coupled Laplace equations, each of which is restricted to a ball as in the ddCOSMO. For the sake of completeness, we
now introduce briefly the ideas. Without loss of generality, we consider the following Laplace equation with the Dirichlet boundary condition $\phi_0$ and defined in the unit ball with center $0$:

$$\begin{cases}
-\Delta u_0 = 0 & \text{in } B_1(0), \\
u_0 = \phi_0 & \text{on } \partial B_1(0),
\end{cases}
$$

where $B_1(0)$ represents the unit ball centered at the origin $0$. As a consequence, the unique solution to (3.6.1) in $H^1(B_1(0))$ can be written as

$$u_0(r, \theta, \varphi) = \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} \left[ \phi_0 \right]_{m}^{\ell} r^\ell Y_\ell^m(\theta, \varphi), \quad 0 \leq r \leq 1, \ 0 \leq \theta \leq \pi, \ 0 \leq \varphi < 2\pi,$n

where $Y_\ell^m$ is the (orthonormal) spherical harmonic of degree $\ell$ and order $m$ defined on $S^2$ and

$$[\phi_0]_{m}^{\ell} = \int_{S^2} \phi_0(s) Y_\ell^m(s) \, ds$$

is the real coefficient of $u_0$ corresponding to the mode $Y_\ell^m$.

To compute (3.6.2) numerically, we first approximate $[\phi_0]_{m}^{\ell}$ using the Lebedev quadrature rule defined by the integration points $s_n \in S^2$ and their weights $w_{n}^{\text{leb}}$ as follows

$$[\tilde{\phi}_0]_{m}^{\ell} = \sum_{n=1}^{N_{\text{leb}}} w_{n}^{\text{leb}} \phi_0(s_n) Y_\ell^m(s_n),$$

where $N_{\text{leb}}$ represents the number of Lebedev points. Then, $u_0$ can be approximated by $\tilde{u}_0$ in the discretization space spanned by a truncated basis of spherical harmonics $\{Y_\ell^m\}_{0 \leq \ell \leq \ell_{\text{max}}, \ -\ell \leq m \leq \ell}$, defined as

$$\tilde{u}_0(r, \theta, \varphi) = \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} \tilde{\phi}_0 Y_\ell^m(\theta, \varphi), \quad 0 \leq r \leq 1, \ 0 \leq \theta \leq \pi, \ 0 \leq \varphi < 2\pi,$n

where $\ell_{\text{max}}$ denotes the maximum degree of spherical harmonics. This approximate solution $\tilde{u}_0$ converges to the exact solution $u_0$ to Eq. (3.6.1) when $\ell_{\text{max}} \to \infty$ and $N_{\text{leb}} \to \infty$. One should note however that there is no systematic manner to derive
Remark 3.6.1. To compute the surface integral over the unit sphere, any other quadrature in the angular variables $\theta$ and $\varphi$ can be used. However, the Lebedev quadrature is very efficient, in the sense that fewer grid points are required to obtain similar accuracy, which has been pointed out in [14]. In addition, Lebedev grids yield exact numerical integration of spherical harmonics up to a certain degree (depending on the number of integration points) and satisfy some invariance properties of certain rotational groups.

3.6.2 Generalized Poisson equation in a ball

To solve the local equations (3.5.1), we need to develop a solver for the GP-equation defined on $\Omega_j$ with the Dirichlet boundary condition (3.5.2). Without loss of generality, we discuss how to solve the following GP-equation in the unit ball with center $0$ in the general form of

$$\begin{aligned}
-\nabla \cdot \varepsilon_1(\mathbf{x}) \nabla u_1(\mathbf{x}) &= f_1(\mathbf{x}) \quad \text{in } B_1(0), \\
u_1(\mathbf{x}) &= \phi_1(\mathbf{x}) \quad \text{on } \partial B_1(0).
\end{aligned}$$

(3.6.4)

In fact, for any $j = 1, \ldots, M$, we take $\varepsilon_1(\mathbf{x}) = \varepsilon(\mathbf{c}_j + R_j \mathbf{x})$, $u_1(\mathbf{x}) = W_j(\mathbf{c}_j + R_j \mathbf{x})$, $f_1(\mathbf{x}) = \nabla \cdot (\varepsilon_1(\mathbf{x}) - 1) \nabla \Phi(\mathbf{c}_j + R_j \mathbf{x})$ and $\phi_1(\mathbf{x}) = h_j(\mathbf{c}_j + R_j \mathbf{x})$ where $\varepsilon$, $W_j$, $\Phi$, $h_j$ are as in Eqs (3.5.1)–(3.5.2).

3.6.2.1 Variational formulation

From the discussion in Section 3.6.1, we know that there exists a unique harmonic function $\hat{u}_1 \in H^1(B_1(0))$, s.t.

$$\begin{aligned}
-\Delta \hat{u}_1 &= 0 \quad \text{in } B_1(0), \\
\hat{u}_1 &= \phi_1 \quad \text{on } \partial B_1(0),
\end{aligned}$$

(3.6.5)

and $\hat{u}_1$ can be efficiently approximated by Eq. (3.6.3). Let $v = u_1 - \hat{u}_1 \in H^1_0(B_1(0))$ and in consequence, $v$ satisfies the following PDE

$$\begin{aligned}
-\nabla \cdot \varepsilon_1 \nabla v &= f \quad \text{in } B_1(0), \\
v &= 0 \quad \text{on } \partial B_1(0),
\end{aligned}$$

(3.6.6)

where $f(\mathbf{x}) = \nabla \cdot (\varepsilon_1(\mathbf{x}) - 1) \nabla \Phi(\mathbf{c}_j + R_j \mathbf{x}) + \nabla \cdot \varepsilon_1(\mathbf{x}) \nabla \hat{u}_1(\mathbf{x})$.

Since the VdW-ball $B_{r_j}(\mathbf{c}_j) \subset \Omega_j$ and $\varepsilon \equiv 1$ holds in $B_{r_j}(\mathbf{c}_j)$, we know that $W_j$ defined in (3.5.1) is harmonic when restricted to the smaller ball $B_{r_j}(\mathbf{c}_j)$. As a
consequence, \( u_1 \) of Eq. (3.6.4) and \( v \) of Eq. (3.6.6) are both harmonic in \( B_\delta(0) \), where \( \delta \) is defined by
\[
\delta = \frac{r_j}{r_j + r_0 + r_p} \in (0, 1).
\]
Let \( \mathcal{D} := B_1(0) \setminus \overline{B}_\delta(0) \) represent the region between \( \partial B_1(0) \) and \( \partial B_\delta(0) \) and define the subspace \( H^1_{0, \delta}(\mathcal{D}) \) of the Sobolev space \( H^1(\mathcal{D}) \) as follows
\[
H^1_{0, \delta}(\mathcal{D}) = \{ w \in H^1(\mathcal{D}) : w|_{\partial B_1(0)} = 0 \}.
\]
In order to find the weak solution restricted to \( H^1_{0, \delta}(\mathcal{D}) \), we can write a variational formulation as: find \( v \in H^1_{0, \delta}(\mathcal{D}) \), s.t. \( \forall w \in H^1_{0, \delta}(\mathcal{D}) \),
\[
\int_{\mathcal{D}} \varepsilon_1 \nabla v \cdot \nabla w + \int_{\partial B_1(0)} (T v) w = \int_{\mathcal{D}} f w, \tag{3.6.7}
\]
where we use the fact that \( \varepsilon_1|_{\partial B_\delta(0)} = 1 \). The operator \( T \) is the Dirichlet-to-Neumann operator of the harmonic extension in \( B_\delta(0) \), that in terms of spherical harmonics is given below.

Assume that we have an expansion of the Dirichlet boundary condition \( v|_{\partial B_\delta(0)} \) as follows
\[
v|_{\partial B_\delta(0)}(\delta, \theta, \varphi) = \sum_{k=0}^{\infty} \sum_{m=-k}^{k} \alpha_{km} Y^m_k(\theta, \varphi), \quad 0 \leq \theta \leq \pi, \ 0 \leq \varphi < 2\pi. \tag{3.6.8}
\]
Then, we can extend \( v|_{\partial B_\delta(0)} \) harmonically from \( \partial B_\delta(0) \) to the ball \( B_\delta(0) \), i.e.,
\[
v|_{B_\delta(0)}(r, \theta, \varphi) = \sum_{k=0}^{\infty} \sum_{m=-k}^{k} \alpha_{km} \left( \frac{r}{\delta} \right)^k Y^m_k(\theta, \varphi), \quad 0 \leq r \leq \delta, \ 0 \leq \theta \leq \pi, \ 0 \leq \varphi < 2\pi. \tag{3.6.9}
\]
Denote by \( n_\delta \) the unit normal vector on the sphere \( \partial B_\delta(0) \) pointing outwards with respect to the ball \( B_\delta(0) \). As a consequence, we can compute \( \partial n_\delta v = \nabla v \cdot n_\delta \) consisting of the normal derivative of \( v \) on \( \partial B_\delta(0) \):
\[
T v(\delta, \theta, \varphi) := \partial n_\delta v(\delta, \theta, \varphi) = \sum_{k=0}^{\infty} \sum_{m=-k}^{k} \alpha_{km} \left( \frac{r}{\delta} \right)^k Y^m_k(\theta, \varphi), \quad 0 \leq \theta \leq \pi, \ 0 \leq \varphi < 2\pi. \tag{3.6.10}
\]
It is also easy to see that the bilinear form on the left side of the variational formulation (3.6.7) is symmetric and coercive due to properties of the Dirichlet-to-Neumann operator \( T \).
3.6.2.2 Galerkin discretization

In order to find basis functions belonging to $H^1_{0,\delta}(\mathcal{D})$, we first introduce the radial functions
\[ \varphi_i(r) = (1 - r)L_i'(\frac{2(r - \delta)}{1 - \delta} - 1), \]
implying that $\varphi_i(1) = 0$. Here, $L_i$ denotes the Legendre polynomial of $i$-th degree. We then discretize both, the radial part and the spherical part of the unknown $v$, meaning that we represent $v$ by linear combination of the basis functions $\{\varphi_i(r)Y^m_\ell(\theta, \varphi)\}$ with $1 \leq i \leq N$, $0 \leq \ell \leq \ell_{\text{max}}$ and $-\ell \leq m \leq \ell$, where $N$ denotes the maximum degree of Legendre polynomials and $\ell_{\text{max}}$ denotes the maximum degree of spherical harmonics as in Section 3.6.1. The spanned space of these functions is denoted by $V_{N,\ell_{\text{max}}}(\mathcal{D})$ which is defined as follows
\[ V_{N,\ell_{\text{max}}}(\mathcal{D}) = \text{span} \{\varphi_i(r)Y^m_\ell(\theta, \varphi) \mid 1 \leq i \leq N, 0 \leq \ell \leq \ell_{\text{max}}, -\ell \leq m \leq \ell\} \subset H^1_{0,\delta}(\mathcal{D}). \]

Then, we consider a Galerkin discretization of the variational formulation (3.6.1) that reads: find $\tilde{v} \in V_{N,\ell_{\text{max}}}(\mathcal{D})$, such that
\begin{equation}
\forall \tilde{w} \in V_{N,\ell_{\text{max}}}(\mathcal{D}) : \quad \int\limits_{\mathcal{D}} \varepsilon_1 \nabla \tilde{v} \cdot \nabla \tilde{w} + \int\limits_{\partial B_{\delta}(0)} (\mathcal{T} \tilde{v}) \tilde{w} = \int\limits_{\mathcal{D}} f \tilde{w}. \tag{3.6.11}
\end{equation}

Since $\tilde{v} \in V_{N,\ell_{\text{max}}}(\mathcal{D})$, we can write $\tilde{v}$ in the form of
\begin{equation}
\tilde{v}(r, \theta, \varphi) = \sum_{i=0}^{N} \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} v_{i\ell m} \varphi_i(r)Y^m_\ell(\theta, \varphi), \quad \forall \delta \leq r \leq 1, \ 0 \leq \theta \leq \pi, \ 0 \leq \varphi \leq 2\pi, \tag{3.6.12}
\end{equation}
and we consequently have
\begin{equation}
\mathcal{T} \tilde{v}|_{B_{\delta}(0)}(\delta, \theta, \varphi) = \sum_{i=0}^{N} \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} v_{i\ell m} \left(\frac{\ell}{\delta}\right) \varphi_i(\delta)Y^m_\ell(\theta, \varphi). \tag{3.6.13}
\end{equation}
By substituting (3.6.12)–(3.6.13) into (3.6.11) and taking the test function $\tilde{w} = \varphi_{i'}(r)Y^m_{\ell'}(\theta, \varphi)$, we then obtain a system of linear equations: $\forall 1 \leq i' \leq N, 0 \leq \ell' \leq \ell_{\text{max}}, -\ell' \leq m' \leq \ell'$,
\begin{equation}
\sum_{i=0}^{N} \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} v_{i\ell m} \left(\int\limits_{\mathcal{D}} \varepsilon_1 \nabla (\varphi_i Y^m_\ell) \cdot \nabla (\varphi_{i'} Y^{m'}_{\ell'}) + \frac{\ell}{\delta} \int\limits_{\partial B_{\delta}(0)} \varphi_i Y^m_\ell \varphi_{i'} Y^{m'}_{\ell'}\right) = \int\limits_{\mathcal{D}} f \varphi_{i'} Y^{m'}_{\ell'} . \tag{3.6.14}
\end{equation}
In order to write the corresponding system of linear equations, we define an index
\[ k = N(\ell^2 + m + \ell) + i \in \{1, 2, \ldots, N(\ell_{\text{max}} + 1)^2\} \]
which corresponds to the triple $(i, \ell, m)$ through a one-to-one mapping between $k$
and \((i, \ell, m)\). Assume that \(k\) corresponds to \((i, \ell, m)\) and \(k'\) corresponds to \((i', \ell', m')\). Then, we can recast the set of equations (3.6.14) as a linear system of the form

\[
\mathbf{A} \mathbf{X} = \mathbf{F}. \tag{3.6.15}
\]

Here, \(\mathbf{A}\) is a symmetric matrix of dimension \(N(\ell_{\text{max}}+1)^2 \times N(\ell_{\text{max}}+1)^2\) with elements \((\mathbf{A})_{kk'}\), for all \(1 \leq k, k' \leq N(\ell_{\text{max}}+1)^2\), defined by

\[
(\mathbf{A})_{k'k} = \int_\mathcal{D} \varepsilon_1 \nabla (\varphi_i Y^m_\ell) \cdot \nabla (\varphi_{i'} Y'^{m'}_{\ell'}) + \frac{\ell}{\delta} \int_{\partial \mathcal{B}_3(0)} \varphi_i Y^m_\ell \varphi_{i'} Y'^{m'}_{\ell'}, \tag{3.6.16}
\]

\(X\) is the column vector of \(N(\ell_{\text{max}}+1)^2\) unknowns \(v_{i\ell m}\), i.e.,

\[
(X)_k = v_{i\ell m}, \quad \forall 1 \leq k \leq N(\ell_{\text{max}}+1)^2 \tag{3.6.17}
\]

and \(F\) is also a column vector with \(N(\ell_{\text{max}}+1)^2\) entities defined by

\[
(F)_{k'} = \int_\mathcal{D} f \varphi_{i'} Y'^{m'}_{\ell'}, \quad \forall 1 \leq k' \leq N(\ell_{\text{max}}+1)^2. \tag{3.6.18}
\]

In summary, to solve Eq. (3.6.7), we finally need to solve the linear system (3.6.15) to obtain all coefficients \(v_{i\ell m}\) and then obtain an approximate solution \(\bar{v}(r, \theta, \varphi) \in \mathcal{V}_{N,\ell_{\text{max}}}(\mathcal{D})\) according to Eq. (3.6.12). Considering that \(v\) is harmonic in \(\mathcal{B}_3(0)\), \(\bar{v}\) can then be extended harmonically in the ball \(\mathcal{B}_4(0)\) following (3.6.9). Therefore, we obtain an approximate solution defined on \(\mathcal{B}_1(0)\) to Eq. (3.6.6).

**Remark 3.6.2.** In the global iterative procedure of Section 3.4.3, we compute the matrix \(\mathbf{A}\) of Eq. (3.6.14) for each subdomain \(\Omega_j\) a priori, since each \(\mathbf{A}\) associated with \(\Omega_j\) can be reused within the iterative GP-solver for solving Eq. (3.4.18) and for the outer iterations [1]–[4]. This helps to reduce considerably the cost of the global procedure at the cost of more memory requirements.

### 3.6.2.3 Numerical integration

In order to implement the method, the integrals in Eq. (3.6.10) and Eq. (3.6.18) need to be further computed. We start by observing that the second term in Eq. (3.6.10) can be simplified

\[
\frac{\ell}{\delta} \int_{\partial \mathcal{B}_3(0)} \varphi_i Y^m_\ell \varphi_{i'} Y'^{m'}_{\ell'} = \ell \delta \varphi_i(\delta) \varphi_{i'}(\delta) \int_{S^2} Y^m_\ell Y'^{m'}_{\ell'}, \tag{3.6.19}
\]

where \(\delta_{\ell\ell'}\) and \(\delta_{mm'}\) are both the Kronecker deltas. As a consequence, the solution matrix \(\mathbf{A}\) is symmetric. Next, we present the numerical integration over \(\mathcal{D}\) that is used to approximate the integral in the first term in Eq. (3.6.10) and the integral in Eq. (3.6.18).
The integral over $D$ can be divided into two parts: the radial part and spherical part, that is to say, for any given function $h \in L^1(B_1(0))$, we can compute its integral over $D$ separately as below

$$
\int_D h(x) \, dx = \int_\delta^1 r^2 \int_{S^2} h(r, s) \, ds \, dr,
$$

(3.6.20)

where $s \in S^2$ and $x = rs$. To compute the spherical part of this integral, we use again the Lebedev quadrature rule for the unit sphere as in Section 3.6.1. To integrate the radial part numerically, we use the Legendre-Gauss-Lobatto (LGL) quadrature rule defined by the integration points $x_m \in [-1, 1]$ and their weights $w_{lgl}^m, 1 \leq m \leq N_{lgl}$, where $N_{lgl}$ represents the number of LGL points. Using the change of variable $r = \frac{1}{2}(x+1)+\delta, x \in [-1, 1]$, we approximate the integral by the following quadrature rule

$$
\int_D h(x) \, dx = \int_\delta^1 r^2 \int_{S^2} h(r, s) \, ds \, dr
= \int_{-1}^1 \frac{1-\delta}{2} \left( \frac{1-\delta}{2} (x+1) + \delta \right)^2 \int_{S^2} h \left( \frac{1-\delta}{2} (x+1) + \delta, s \right) \, ds \, dx
\approx \frac{1-\delta}{2} \sum_{m=1}^{N_{lgl}} \sum_{n=1}^{N_{leb}} w_{lgl}^m w_{leb}^n \left( \frac{1-\delta}{2} (x_m+1) + \delta \right)^2 h \left( \frac{1-\delta}{2} (x_m+1) + \delta, s_n \right).
$$

(3.6.21)

The first integral of Eq. (3.6.16) and the integral of Eq. (3.6.18) can then be numerically computed using this quadrature. See Figure 7 for a schematic diagram of the Lebedev points and the LGL points.
Chapter 3: Domain Decomposition Method for the Polarizable Continuum Model
based on the Solvent Excluded Surface

3.7 Numerical results

In this section, we present some numerical results of the proposed method for solving the PDEs (3.4.7)–(3.4.8). Before investigating a realistic solute molecule, we will start by testing the GP-solver for the unit sphere. We then consider small molecules and compute the electrostatic contribution to the solvation energy to study its dependency with respect to numerous parameters. In the following tests, we simply choose a polynomial function to be $\xi$ in Eq. (3.3.1) as follows

$$\xi(t) = 1 + (\varepsilon_s - 1)t^3(10 + 3t(-5 + 2t)), \quad 0 \leq t \leq 1,$$

which satisfies the previously-mentioned conditions (3.3.2) at $t = 0$ and $t = 1$. This function does not come from experimental fitting nor have a physical meaning, but taken for the numerical tests of the proposed method only.

3.7.1 GP-solver test

In Section 3.6.2, we present the GP-solver in the unit ball, which will be called repeatedly in the domain decomposition scheme. In this part, we first test the GP-solver for solving Eq. (3.6.4) to have a first look at the solver itself. For the sake of simplicity, we assume here that the dielectric permittivity function $\varepsilon$ only depends on the radial variable $r$, i.e., the dielectric permittivity is symmetric. Let $r_1 = 0.4$, $r_p = 0.3$, $r_0 = 0.3$ and define the dielectric permittivity function $\varepsilon(r)$ by

$$\varepsilon(r) = \begin{cases} 1 & \text{if } 0 \leq r \leq r_1, \\ \xi \left( \frac{r - r_1}{r_p + r_0} \right) & \text{if } r_1 \leq r \leq 1, \\ \varepsilon_s & \text{if } r \geq 1, \end{cases}$$

where $\varepsilon_s = 10$. Assuming that the sphere carries a point charge $1$ at its center, i.e., $q_1 = 1$, we take $\Phi = \frac{1}{r}$, $f_1 = -\varepsilon(r)$ and $\phi_1 = -1$ in Eq. (3.6.3). Since both the dielectric permittivity function and the solution only depend on the radial variable, the discretization of the spherical part can be neglected, i.e., $\ell_{\max} = 0$. To have a good approximation in the radial direction, we set a high maximum degree of Legendre polynomials $N = 20$ and a large number of LGL points $N_{lgl} = 200$. By running the GP-solver, we then obtain Figure 8 illustrating the dielectric permittivity $\varepsilon(r)$ and the numerical solution $u_1(r)$ to Eq. (3.6.4). It is observed that $u_1(r)$ is constant in $[0, r_1]$ because of the harmonicity.
3.7. Numerical results

Figure 8: The left figure illustrates the dielectric permittivity function \( \varepsilon(r) \); the right plots the numerical solution to Eq. (3.4.18). Only the blue part is subject to the computational domain, the red part of the curve depicts the harmonic extension.

3.7.2 Convergence of the global strategy

With the domain decomposition method using the GP-solver for solving Eq. (3.4.18) and the Laplace solver for solving (3.4.19), we can then solve numerically Eqs (3.4.16)–(3.4.17) following the iterative procedure presented in Section 3.4.3.

The Schwarz domain decomposition method (for solving the PDE (3.4.18) and (3.4.19)) is well-studied and its convergence can be guaranteed [107]. To study the convergence of the proposed method with respect to the number of outer iterations, we first take the example of formaldehyde. The following discretization parameters are used: the maximum degree of spherical harmonics \( \ell_{\text{max}} = 11 \), the number of Lebedev points \( N_{\text{leb}} = 1202 \), the maximum degree of Legendre polynomials \( N = 15 \), the number of LGL nodes \( N_{\text{lgl}} = 50 \). These parameters are given based on a further study in Section 3.7.6. Furthermore, we take the solvent permittivity \( \varepsilon_s = 78.4 \) (water, at room temperature 25°C), the solvent probe radius \( r_p = 1.5\text{Å} \) and \( r_0 = 1\text{Å} \). We use the convention that in the numerical tests, the stopping criterion Tol is set to \( 10^{-7} \) by default. In this chapter, the unit of energy is set to \( e^2/\text{Å} = 332.06364 \text{ kcal/mol} \) (see [54, Section 1.4] for this equality), where \( e \) denotes the elementary charge. In addition, we use another convention that at each outer iteration of the global strategy, the number of inner iterations for solving the GP-equation (3.4.18) is fixed to be 8. This number is determined empirically and it allows us to obtain an accurate enough numerical solution to Eq. (3.4.18) at each outer iteration.

The error of the electrostatic contribution is computed as follows

\[
\text{Error}(N_{\text{it}}) = |E_{N_{\text{it}}} - E_{\infty}|,
\]  

(3.7.3)

where \( E_{N_{\text{it}}} \) is the electrostatic contribution computed at the \( N_{\text{it}} \)-th outer iteration in
Figure 9: On the left, the curve illustrates the computed electrostatic contribution to the solvation energy of formaldehyde with respect to $N_{it}$; on the right, the curve illustrates the error of the electrostatic contribution with respect to $N_{it}$. The following parameters are used: $\ell_{\text{max}} = 11$, $N_{\text{leb}} = 1202$, $N = 15$, $N_{\text{lgl}} = 50$, $\varepsilon_s = 78.4$, $r_p = 1.5\text{Å}$, $r_0 = 1\text{Å}$.

Section 3.4.3 and the “exact” electrostatic contribution $E_{\infty}^s$ is obtained after 15 outer iterations (this number is large enough) with the same discretization parameters as mentioned above. In Figure 9, which illustrates the numerical electrostatic contribution to the solvation energy of formaldehyde with respect to the number of outer iterations $N_{it}$, it is observed that the error of electrostatic contribution converges geometrically with respect to $N_{it}$ and the procedure stops at $N_{it} = 5$ when the stopping criterion is reached, i.e., $|E_{N_{it}}^s - E_{N_{it}-1}^s| < \text{Tol}$.

Then, we present the example of a larger molecule, the caffeine consisting of 24 atoms, with the parameters $\ell_{\text{max}} = 9$, $N_{\text{leb}} = 350$, $N = 15$, $N_{\text{lgl}} = 30$, $\varepsilon_s = 78.4$, $r_p = 1.5\text{Å}$, $r_0 = 1\text{Å}$. The discretization parameters are taken smaller than those for formaldehyde because of the running time. This does not matter since we only want to study the convergence of the method with respect to the number of outer iterations. As above, the “exact” electrostatic contribution $E_{\infty}^s$ is obtained after 15 outer iterations and the error is computed from (3.7.3). As can be seen in Figure 10, the error of electrostatic contribution converges also geometrically with respect to $N_{it}$.

### 3.7.3 Graphical illustration of the reaction potential

We give here some graphical illustration of the reaction potential in the enlarged cavity $\Omega_0$ and on the SES. In Figure 11, we illustrate the reaction potential $W$ of hydrogen-fluoride and the magnitude of the corresponding reaction field (i.e., $\nabla W$). In Figure 12 and 13, we illustrate the reaction potential of hydrogen-fluoride, formaldehyde and caffeine in water. The rotational symmetry of hydrogen-fluoride...
3.7. Numerical results

Figure 10: On the left, the curve illustrates the numerical electrostatic contribution to the solvation energy of caffeine with respect to $N_{lt}$; on the right, the curve illustrates the error of the electrostatic contribution with respect to $N_{lt}$. The following parameters are used: $\ell_{\text{max}} = 9$, $N_{\text{leb}} = 350$, $N = 15$, $N_{\text{lg}} = 30$, $\varepsilon_s = 78.4$, $r_p = 1.5\text{Å}$, $r_0 = 1\text{Å}$.

and the mirror symmetry of formaldehyde are observed as expected.

3.7.4 Thickness of the intermediate layer

We then draw the attention to the influence of the thickness of the intermediate layer, which is given by $r_p + r_0$. For the sake of simplicity, we fix the parameter $r_0 = 0\text{Å}$, implying that $\Gamma_0$ and $\Gamma_{\text{ss}}$ coincide. In the ddPCM algorithm presented in [108], the dielectric permittivity is discontinuous across the solute-solvent boundary and jumps from 1 to $\varepsilon_s$. In the following test, we take $\Gamma_0$ as the solute-solvent boundary in the ddPCM algorithm and expect that the numerical electrostatic contribution to the solvation energy computed from our method tends to the one computed from the ddPCM algorithm as the thickness of the intermediate layer $L$ tends to zero, i.e., $r_p \to 0$. This implies that both, $\Gamma_{\text{ss}}$ and $\Gamma_0$, tend to the VdW-surface.

To verify this, we take again the example of hydrogen-fluoride. Figure 14 illustrates the numerical electrostatic contribution to the solvation energy with respect to $n$ where the thickness of the intermediate layer is parametrized by $r_p = 2^{-n}\text{Å}$. We observe that the electrostatic contributions from both algorithms almost tend to coincide when the layer vanishes. This means that the proposed method is consistent with the ddPCM in this case.
Figure 11: The reaction potential (left) and the magnitude of the corresponding reaction field (right) in the YZ-plane for the hydrogen-fluoride molecule with two atoms, with the parameters $\ell_{\text{max}} = 11$, $N_{\text{leb}} = 1202$, $N = 15$, $N_{\text{lgl}} = 50$, $\varepsilon_s = 2$, $r_p = 1.5\text{Å}$, $r_0 = 1\text{Å}$. The colorbars represent respectively the reaction potential value and the magnitude of the reaction field.

Figure 12: Reaction potential of hydrogen-fluoride and formaldehyde on the SES, both computed with the parameters $\ell_{\text{max}} = 11$, $N_{\text{leb}} = 1202$, $N = 15$, $N_{\text{lgl}} = 50$, $\varepsilon_s = 78.4$, $r_p = 1.5\text{Å}$, $r_0 = 1\text{Å}$.

Figure 13: Reaction potential of caffeine on the SES, computed with the parameters $\ell_{\text{max}} = 9$, $N_{\text{leb}} = 350$, $N = 15$, $N_{\text{lgl}} = 30$, $\varepsilon_s = 78.4$, $r_p = 1.5\text{Å}$, $r_0 = 1\text{Å}$. 
3.7. Numerical results

Figure 14: The blue curve plots the electrostatic contribution of hydrogen-fluoride with respect to \( n \) (the layer thickness \( 2^{-n} \AA \)), computed from the proposed algorithm with the parameters \( \ell_{\text{max}} = 11, N_{\text{leb}} = 1202, N = 15, N_{\text{lg}} = 50, \varepsilon_s = 2 \); the red curve plots the results computed from the ddPCM algorithm with the same parameters of spherical harmonics \( \ell_{\text{max}} = 11, N_{\text{leb}} = 1202 \) and the same solvent permittivity \( \varepsilon_s = 2 \).

3.7.5 Solvent dielectric constant

We illustrate in Figure 15 how the electrostatic contribution to the solvation energy and the total number of outer iterations varies with respect to different solvent permittivities \( \varepsilon_s = 2^k, k = 1, 2, \ldots, 15 \). We observe that the numerical electrostatic contribution to the solvation energy varies smoothly and converges to some quantity for increasing solvent permittivities \( \varepsilon_s \). Another interesting observation is that when \( \varepsilon_s \) becomes large, the total number of outer iterations tends to decrease. This can be explained by the fact that the solvent is conductor-like and consequently, the initial guess of potential \( g^0 \) is very accurate for high permittivities \( \varepsilon_s \).

3.7.6 Discretization parameters

We consider the dichloromethane solvent (\( \varepsilon_s = 8.93 \)) at room temperature 25°C. First of all, we compute numerically the “exact” electrostatic contribution to the solvation energy of hydrogen-fluoride, denoted by \( E_{\text{exact}}^s \), with large discretization parameters \( \ell_{\text{max}} = 20, N_{\text{leb}} = 4334, N = 25, N_{\text{lg}} = 50 \) and the other parameters \( r_p = 1.5\AA, r_0 = 1\AA \). We treat \( E_{\text{exact}}^s \) as the benchmark of the electrostatic contribution to the solvation energy.

We then illustrate how the electrostatic contribution to the solvation energy of hydrogen-fluoride varies respectively with respect to the maximum degree \( \ell_{\text{max}} \) of spherical harmonics and to the number \( N_{\text{leb}} \) of Lebedev points in the dichloromethane
Chapter 3: Domain Decomposition Method for the Polarizable Continuum Model based on the Solvent Excluded Surface

Figure 15: The left figure shows the electrostatic contribution to the solvation energy of hydrogen-fluoride when the solvent dielectric constant $\varepsilon_s$ increases; the right figure plots the total number of outer iterations to reach the stopping criterion $Tol$ with respect to $\varepsilon_s$. The following parameters are used: $\ell_{\text{max}} = 11$, $N_{\text{leb}} = 1202$, $N = 20$, $N_{\text{lgl}} = 50$, $r_p = 1.5\text{Å}$, $r_0 = 1\text{Å}$.

solvent. On the left of Figure 16, we report the electrostatic contribution with respect to $\ell_{\text{max}}$ which varies from 3 to 15. Further, on the right of Figure 16, we report the electrostatic contribution with respect to $N_{\text{leb}}$ which varies from 350 to 4334. From the left figure, we observe that the proposed algorithm provides systematically improvable approximations when the parameters of spherical harmonics increases.

Similarly as above, we now illustrate how the electrostatic contribution to the solvation energy of hydrogen-fluoride varies with respect to the maximum degree $N$ of Legendre polynomials and to the number $N_{\text{lgl}}$ of LGL points in the dichloromethane solvent. On the left of Figure 17, we report the electrostatic contribution with respect to $N$ varying from 6 to 20. Further, on the right of Figure 17, we report the electrostatic contribution with respect to $N_{\text{lgl}}$ varying from 20 to 80. Again, it is observed that the proposed algorithm provides asymptotically systematically improvable approximations when the discretization of Legendre polynomials increases. These results help us to get a know-how in order to select, for a given molecule, the proper choice of discretization parameters $\ell_{\text{max}}$, $N_{\text{leb}}$, $N$, $N_{\text{lgl}}$ for an acceptable accuracy.

3.7.7 Robustness with respect to geometrical parameters

We now study in Figure 18 the variation of the electrostatic contribution to the solvation energy of hydrogen-fluoride while rotating the fluoride atom around the hydrogen atom. We observe that the variation in the electrostatic contribution, which should be invariant with respect to the rotation angle, is systematically controlled. For the two sets of parameters, see the caption of Figure 18, the variation is around
3.7. Numerical results

Figure 16: The left figure plots the electrostatic contribution to the solvation energy of hydrogen-fluoride with respect to $\ell_{\text{max}}$ when $N_{\text{leb}}$ is set to 1454. The right figure plots the electrostatic contribution with respect to $N_{\text{leb}}$ when $\ell_{\text{max}}$ is set to 15. In both figures, the blue line represents the “exact” electrostatic contribution $E_s^{\text{exact}}$. In addition, the following parameters are used: $N = 20$, $N_{\text{lgl}} = 50$, $\varepsilon_s = 8.93$, $r_p = 1.5\text{Å}$, $r_0 = 1\text{Å}$.

Figure 17: The left figure plots the electrostatic contribution to the solvation energy of hydrogen-fluoride with respect to $N$ when $N_{\text{lgl}}$ is set to 50. The right figure plots the electrostatic contribution with respect to $N_{\text{lgl}}$ when $N$ is set to 20. In both figures, the blue line represents the “exact” electrostatic contribution $E_s^{\text{exact}}$. In addition, the following parameters are used: $\ell_{\text{max}} = 15$, $N_{\text{leb}} = 1454$, $\varepsilon_s = 8.93$, $r_p = 1.5\text{Å}$, $r_0 = 1\text{Å}$.
Figure 18: The variation of the electrostatic contribution to the solvation energy of hydrogen-fluoride with respect to the angle of rotating the fluoride atom around the hydrogen atom. The blue curve corresponds to the parameters $\ell_{\text{max}} = 7$, $N_{\text{leb}} = 86$, $N = 10$, $N_{\text{lgl}} = 20$, the red curve corresponds to the parameters $\ell_{\text{max}} = 11$, $N_{\text{leb}} = 1202$, $N = 20$, $N_{\text{lgl}} = 40$. Furthermore, the following parameters are used: $\varepsilon_s = 8.93$, $r_p = 1.5\text{Å}$, $r_0 = 1\text{Å}$.

Further, we study the electrostatic contribution to the solvation energy under a dissociation of hydrogen-fluoride. We vary the separation distance between the hydrogen atom and the fluoride atom from $8.96\text{Å}$ to $9.36\text{Å}$ where the topology of the SES changes in the sense that the SES becomes two disconnected subsurfaces, see Figure 19 for a geometrical illustration. The left plot of Figure 20 illustrates that the energy profile is completely smooth when the topology of the SES changes. The right figure of Figure 20 provides the electrostatic contribution to the solvation energy when the separation distance varies in a wider range from $1.77\text{Å}$ to $9.37\text{Å}$.

We next study the smoothness of the numerical electrostatic contribution to the solvation energy of formaldehyde with respect to a topological change in the sense that the nature of the patches of the SES changes. We choose the coordinate system so that all nuclei lie in the $yz$-plane and that carbon and oxygen atoms have $y = 0$ coordinate. We then move the $z$-coordinate of the two hydrogen atoms further away from the carbon and oxygen atoms, see Figure 21 for an illustration. Figure 22 plots the numerical electrostatic contribution to the solvation energy with respect to the downwards displacement of the two hydrogen atoms. On the left, we plot the numerical electrostatic contribution with respect to displacement of the two hydrogen atoms in a large range from $0\text{Å}$ to $1\text{Å}$. On the right, we plot the numerical electrostatic contribution in the neighborhood of the threshold when the concave SES patches, marked in blue in Figure 21, first appear and we observe that the energy-profile is approximately smooth.
Figure 19: Topological change of the SES of hydrogen-fluoride when the distance between the two atomic centers increases. The most left figure illustrates the SES of hydrogen-fluoride in equilibrium and the three figures from the left to right corresponds to the distances between the centers increased respectively to $8.96\text{Å}$, $9.16\text{Å}$ and $9.36\text{Å}$.

Figure 20: Electrostatic contribution of hydrogen-fluoride with respect to the separation distance between the atomic centers with parameters $\ell_{\text{max}} = 11$, $N_{\text{leb}} = 1202$, $N = 20$, $N_{\text{lgl}} = 40$, $\varepsilon_s = 8.93$, $r_p = 1.5\text{Å}$, $r_0 = 0\text{Å}$. On the left, the separation distance varies between $8.96\text{Å}$ and $9.36\text{Å}$ where the topology of the SES changes as showed in Figure 19. On the right, the separation distance varies in a larger range.
Figure 21: Different SESs of formaldehyde when the two hydrogen atoms displace downwards. The left figure illustrates the SES of formaldehyde in equilibrium; the middle figure illustrates the SES when concave patches are about to appear; the right figure illustrates the SES when the concave patches (in blue) have appeared.

Figure 22: Electrostatic contribution to the solvation energy of formaldehyde with respect to the displacement of the two hydrogen atoms with parameters $\ell_{\text{max}} = 11$, $N_{\text{leb}} = 1454$, $N = 15$, $N_{\text{igl}} = 50$, $\varepsilon_s = 8.93$, $r_p = 1.5\text{Å}$, $r_0 = 1\text{Å}$. On the left, the displacement varies near the threshold when the concave SES patches first appear as showed in the middle of Figure 21. On the right, the displacement varies from 0Å to 1Å.
3.8 Conclusion

We have proposed a construction to deal with a continuous permittivity function $\varepsilon(x)$ for the SES-based PCM, in which a switching region on the solute-solvent boundary is introduced, using the signed distance function to the SAS. The switching region potentially also allows one to describe local effects close to the solute-solvent interface. Further, to solve this SES-based PCM, we lay out a mathematical framework to compute the solution on an unbounded domain that only involves computing problems in a bounded domain (the enlarged cavity $\Omega_0$). For each problem in the enlarged cavity, we propose to use the Schwarz domain decomposition method where the global problem is divided into simple sub-problems each defined in a ball. We provided numerical tests to illustrate the resulting energy profile is smooth with respect to geometrical parameters and that the energy is systematically improvable.

We focused here on the modeling part and the resulting discretization method. This work is meant to study the feasibility to solve a SES-based solvation model using a domain decomposition method and analyzing its properties. An upcoming paper will focus on an efficient implementation in Fortran which is a work in progress.
Chapter 4

Domain Decomposition Method for the Poisson-Boltzmann Solvation Model

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>132</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Previous work</td>
<td>132</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Contribution</td>
<td>134</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Outline</td>
<td>136</td>
</tr>
<tr>
<td>4.2</td>
<td>PB solvation model</td>
<td>137</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Solute-solvent boundary</td>
<td>137</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Poisson-Boltzmann equation</td>
<td>137</td>
</tr>
<tr>
<td>4.3</td>
<td>Problem transformation</td>
<td>139</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Problem setting</td>
<td>139</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Necessary tools from the potential theory</td>
<td>140</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Transformation</td>
<td>141</td>
</tr>
<tr>
<td>4.4</td>
<td>Strategy</td>
<td>142</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Global strategy</td>
<td>143</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Domain decomposition scheme</td>
<td>143</td>
</tr>
<tr>
<td>4.5</td>
<td>Single-domain solvers</td>
<td>145</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Laplace solver</td>
<td>145</td>
</tr>
<tr>
<td>4.5.2</td>
<td>HSP solver</td>
<td>146</td>
</tr>
<tr>
<td>4.6</td>
<td>Global linear system</td>
<td>148</td>
</tr>
<tr>
<td>4.6.1</td>
<td>Reformulation of the coupling conditions</td>
<td>148</td>
</tr>
</tbody>
</table>
This chapter will be submitted to a journal soon. In this chapter, a domain decomposition method is introduced to solve the Poisson-Boltzmann (PB) solvation model in quantum chemistry. The essential work is to solve the linearized Poisson-Boltzmann (LPB) equation defined in \( \mathbb{R}^3 \). First of all, the original equation is transformed to a system of two coupled equations defined in the bounded solute cavity: the Poisson equation and the homogeneous screened Poisson (HSP) equation. Then, a particular Schwarz domain decomposition is proposed to solve these two equations by decomposing the solute cavity into overlapping balls and solving a group of coupled sub-equations defined in balls. A Laplace solver and a HSP solver are developed respectively to solve the Poisson equation (after homogenization) and the HSP equation. Finally, a global linear system is derived after discretization, which can be solved iteratively. A series of numerical experiments are presented to test the robustness and the efficiency of this method.

4.1 Introduction

4.1.1 Previous work

The properties of numerous charged bio-molecules and their complexes with other molecules are dependent on the dielectric permittivity and the ionic strength of the environment. There are various methods to model ionic solution effects on molecular systems, which can be commonly divided into two broad categories according to whether they employ an explicit or implicit solvation model. Explicit solvataion models adopt microscopic representations of both the solute and solvent molecules, which produce accurate results, but are very expensive. Implicit solvation models adopt a semi-microscopic treatment of the solute, but characterize the solvent in terms of its macroscopic physical properties (for example, the solvent dielectric permittivity and the ionic strength). This reduces greatly the computational cost compared to an explicit description of the solvent. For this reason, implicit solvation models based on the Poisson-Boltzmann (PB) equation [110, 86] are now widely-used, taking into account both the solvent (relative) dielectric permittivity and the ionic strength. In
this chapter, we call these models as PB solvation models and we mention that the
ESU-CGS (electrostatic units, centimetre-gram-second) system of units \[15\] is used
for all equations.

For the sake of simplicity, we consider in particular the linearized Poisson-Boltzmann
(LPB) equation, which describes the electrostatic potential \( \psi \) of the PB solvation
model in the following form (see \[86\])

\[
- \nabla \cdot [\varepsilon(x) \nabla \psi(x)] + \kappa(x)^2 \psi(x) = 4\pi \rho_M(x), \quad \text{in } \mathbb{R}^3,
\] (4.1.1)

where \( \varepsilon(x) \) denotes the space-dependent dielectric permittivity function, \( \kappa(x) \) the
modified Debye-Hückel parameter, \( \rho_M(x) \) denotes the known solute’s charge distribu-
tion. Usually, \( \varepsilon(x) \) has the following form

\[
\varepsilon(x) = \begin{cases}
\varepsilon_1 & \text{in } \Omega, \\
\varepsilon_2 & \text{in } \Omega^c := \mathbb{R}^3 \setminus \overline{\Omega},
\end{cases}
\] (4.1.2)

where \( \varepsilon_1 \) and \( \varepsilon_2 \) are respectively the solute dielectric permittivity and the solvent
dielectric permittivity, \( \Omega \) and \( \Omega^c \) represents respectively the solute cavity and the
solvent region. In addition, \( \kappa(x) \) has the form

\[
\kappa(x) = \begin{cases}
0 & \text{in } \Omega, \\
\sqrt{\varepsilon_2} \kappa & \text{in } \Omega^c,
\end{cases}
\] (4.1.3)

where \( \kappa \) is the Debye-Hückel screening constant. Details on the solute cavity \( \Omega \), the
nonlinear PB equation and its linearization will be presented in Section \[11\].

### 4.1.1.1 Domain decomposition methods for implicit solvation models

The polarizable continuum model (PCM) \[21, 109, 82\] is a popular implicit solva-
tion model for calculating the solvation energy. In the classical PCM, the solvent is
represented as a polarizable continuous medium which is non-ionic, i.e., \( \kappa(x) \equiv 0 \) in
\( \mathbb{R}^3 \). A reduced version of the PCM is the conductor-like screening model (COSMO)
\[60\], where the solvent is simply represented by a conductor-like continuum implying
that \( \varepsilon_2 = \infty \). Both the PCM and the COSMO can be seen as two particular cases of
the PB solvation model.

In the past several years, a particular Schwarz domain decomposition method for
the COSMO (called the ddCOSMO method) has been proposed \[25, 72, 69, 71\]. This
method has attracted much attention because of its impressive efficiency \[69\], that is,
it performs two orders of magnitude faster than the equivalent algorithm implemented
in Gaussian, a popular and widely-used computational chemistry software. Later, a
similar discretization scheme for the classical PCM was proposed within the domain
decomposition paradigm (called the ddPCM method) \[108\]. Both the ddCOSMO and
the ddPCM work for the solute cavity constituted by overlapping balls, such as the van der Waals (VdW) cavity and the solvent accessible surface (SAS) cavity. In the case of the PCM based on a “smooth” solute cavity, i.e., the solvent excluded surface (SES) cavity, another domain decomposition method has been proposed in [85], which is called the ddPCM-SES method by us.

Inspired by the previous work, especially the ddPCM-SES method, we would like to develop a domain decomposition method for the PB solvation model, which actually solves the LPB equation. Here, note that this is a problem posed on the entire domain \( \mathbb{R}^3 \).

### 4.1.1.2 Numerical methods

We recall three commonly-used methods for solving the PB equation [79]: the boundary element method (BEM), the finite difference method (FDM) and the finite element method (FEM). As the names indicate, the BEM is based on solving an integral formulation defined on the solute-solvent boundary, while the FDM and the FEM are based on solving the partial differential equation (PDE) defined on the 3-dimensional domain.

In the BEM [116, 15, 4, 8], one first generates a surface mesh of the solute-solvent boundary (the interface between the solute and the solvent). This mesh might need to be of high-quality and adaptive in order to achieve a high degree of accuracy. In the FDM, one takes a big box with grid covering the region of interest and then imposes an approximate condition on the boundary of the box [86, 9, 40, 92, 68, 39]. To obtain accurate solutions, the box might need to be sufficiently large and the grid should be sufficiently dense. As a consequence, the storage and the running time can increase considerably with respect to the number of grid points. In this case, to accelerate the method, one can participate the problem domain into subdomains and solve the subproblems in parallel, for example in the APBS package [9]. However, since the variable parameters of the PB equation (including the dielectric permittivity parameter and the Debye-Hückel parameter, see Section 4.2.2) are discontinuous (there exists a jump on the solute-solvent boundary), the FDM can fail to converge at an optimal rate. As to the FEM [87, 16, 53, 3], it can be expensive as it replies on 3-dimensional volume meshes which themselves could be difficult to generate. In this method, due to the discontinuity of parameters, either the numerical integration needs to be carried out very accurately across the solute-solvent interface or the volume mesh needs to resolve the interface.

### 4.1.2 Contribution

A domain decomposition method is proposed in this chapter to solve the LPB equation (4.1.1) defined in \( \mathbb{R}^3 \). The LPB equation consists of a Poisson equation
defined in the bounded solute cavity $\Omega$ and a homogeneous screened Poisson (HSP) equation defined in the unbounded solvent region $\Omega^c$, which are coupled by some boundary conditions on $\Gamma$.

First, we homogenize the Poisson equation to the following Laplace equation of the reaction potential $\psi_r := \psi - \psi_0$,

$$-\Delta \psi_r = 0, \quad \text{in } \Omega,$$

(4.1.4)

where $\psi_0$ satisfies $-\Delta \psi_0 = \frac{4\pi}{\varepsilon_1} \rho_M$ in $\mathbb{R}^3$. Then, we use the single-layer potential to represent the electrostatic potential $\psi|_{\Omega^c}$ (an exterior Dirichlet problem), which simultaneously gives an extended potential $\psi_e$ satisfying the following HSP equation defined now in $\Omega$ (an interior Dirichlet problem)

$$-\Delta \psi_e(\mathbf{x}) + \kappa^2 \psi_e(\mathbf{x}) = 0, \quad \text{in } \Omega.$$

(4.1.5)

Based on the classical jump-conditions of $\psi$ on the solute-solvent boundary (which will be presented in Section 4.3), a coupling condition arises through an auxiliary function $g$ between the Laplace equation and the extended HSP equation as follows

$$g = S_\kappa \left( \frac{\varepsilon_1}{\varepsilon_2} \partial_n \psi_0 - \frac{\varepsilon_1}{\varepsilon_2} \partial_n (\psi_0 + \psi_r) \right), \quad \text{on } \Gamma,$$

(4.1.6)

where $S_\kappa : H^{-\frac{1}{2}}(\Gamma) \to H^{\frac{1}{2}}(\Gamma)$ denotes the single-layer operator on $\Gamma$ (the definition is given in Section 4.3). Here, $H^{-\frac{1}{2}}(\Gamma)$ and $H^{\frac{1}{2}}(\Gamma)$ denote the usual Sobolev spaces of order $\pm \frac{1}{2}$ on $\Gamma$, see [2]. The initial problem defined in $\mathbb{R}^3$ is therefore transformed into two equations (4.1.4)–(4.1.5) coupled by Eq. (4.1.6).

Fully taking advantage of the fact that we model the solute cavity as a union of overlapping balls here, a particular Schwarz domain decomposition can be adopted to solve Eqs (4.1.4)–(4.1.5) by respectively solving a group of coupled sub-equations in balls. The main idea of this domain decomposition method is illustrated in Figure 1. Ultimately, only a Laplace solver and a HSP solver in the unit ball should be developed for solving the local Laplace sub-equations and HSP sub-equations. In each solver, the spectral method is used to solve numerically the related PDE and the spherical harmonics are used as basis functions in the angular direction of the spherical coordinate system.

This domain decomposition method for solving the LPB equation (called the ddLPB method for short) has the same complexity as the ddPCM method. One highlight of this method is that it does not rely on a mesh nor a box grid, but only on the Lebedev quadrature points [10] for each atom. All integration points lie consequently on the 2-dimensional atomic spheres, the number of which is therefore much fewer than the number of grid points in the FDM and the FEM. Furthermore, one does not need to impose an approximate Dirichlet boundary condition, which is similar to the BEM.
4.1. Introduction

Figure 1: Schematic diagram of the domain decomposition method for solving the linearized Poisson-Boltzmann equation.

4.1.3 Outline

In Section 4.2, we introduce different kinds of solute-solvent boundary for the implicit solvation model and the PB equation defined in $\mathbb{R}^3$ as well as its linearization. Then, in Section 4.3, we transform equivalently the original PB equation defined on the unbounded domain $\mathbb{R}^3$ to two coupled equations both defined in the bounded solute cavity as briefly outlined above. Based on this transformation, we give a global strategy for solving the problem, where each equation is solved using the domain decomposition method in Section 4.4. The domain decomposition method requires to develop two single-domain solvers respectively for the Laplace equation and the HSP equation defined in a ball, which are presented in Section 4.5. After that, in Section 4.6, we give a reformulation of coupling conditions that should be discretized, and consequently, deduce a global linear system to be solved. In Section 4.7, we present some numerical results of the electrostatic contribution to the solvation energy computed by the proposed ddLPB method. In the last section, we draw some conclusions.
4.2 PB solvation model

In this section, we first introduce the solute-solvent boundary, which is a fundamental concept of the implicit solvation model. Furthermore, we introduce the well-known PB equation and its linearization, which describe the electrostatic potential in the implicit solvation model with ionic solutions.

4.2.1 Solute-solvent boundary

In an implicit solvation model, the whole space is divided into two parts: the solute cavity and the solvent region. The solute-solvent boundary is the interface between the solute and the solvent, which can be seen as the molecular surface of the solute. There are three widely-used molecular surfaces: the van der Waals (VdW) surfaces, the Solvent Accessible Surface (SAS) and the Solvent Excluded Surface (SES) [30]. Both the VdW surface and the SAS are constituted by the boundary of the union of balls (respectively the VdW-balls and the SAS-balls), while the SES has a more complicated geometry that can be characterized implicitly by the signed distance function to the SAS [93, 94]. In fact, in real applications, the scaled VdW surface is often used, which means that each atomic VdW-radius is multiplied by a factor $1.1 \sim 1.2$. In the following content, the solute-solvent boundary is taken to be the (scaled) VdW surface or the SAS, as from the geometrical point of view, they are much simpler to tackle than the SES.

4.2.2 Poisson-Boltzmann equation

In the PB solvation model, the solvent is represented by a polarizable continuum containing ions. The freedom of the ions to move in the solution is accounted for by Boltzmann statistics. This means that the Boltzmann equation is used to calculate the local ion density $c_i$ of the $i$-th type of ion as follows

$$c_i = c_i^\infty e^{\frac{-W_i}{k_BT}}, \quad (4.2.1)$$

where $c_i^\infty$ is the bulk ion concentration at an infinite distance from the solute molecule, $W_i$ is the work required to move the $i$-th type of ion to a given position from an infinitely far distance, $k_B$ is the Boltzmann constant, $T$ is the temperature in Kelvins (K). The electrostatic potential $\psi$ of a general implicit solvation model is characterized originally by the following Poisson equation

$$-\nabla \cdot \varepsilon(\mathbf{x}) \nabla \psi(\mathbf{x}) = 4\pi \rho(\mathbf{x}), \quad \text{in } \mathbb{R}^3, \quad (4.2.2)$$
where $\psi(\mathbf{x}) \sim \frac{1}{|\mathbf{x}|}$ as $|\mathbf{x}| \to \infty$. Here, $\varepsilon(\mathbf{x})$ represents the space-dependent dielectric permittivity and $\rho(\mathbf{x})$ represents the total charge distribution of the solvated system. With the solute’s charge distribution $\rho_M$ and the ionic distribution $c_i$ in (4.2.1), we then derive the Poisson-Boltzmann equation as follows (see [10])

$$
-\nabla \cdot [\varepsilon(\mathbf{x})\nabla \psi(\mathbf{x})] = 4\pi \rho_M(\mathbf{x}) + \sum_i z_i e \varepsilon_i e^{-\frac{|\varepsilon_i \psi(\mathbf{x})|}{k_B T}} \chi_{\Omega^c}(\mathbf{x}),
$$

(4.2.3)

where $z_i e$ is the charge of the $i$-th type of ion, $e$ is the elementary charge and $\chi_{\Omega^c}$ is the characteristic function of the solvent region $\Omega^c$.

In the PB solvation model with a 1 : 1 electrolyte, there are two types of ions respectively with charge $+e$ and $-e$ (see Figure 2 for a schematic diagram). With the assumption that $\psi$ satisfies the low potential condition, i.e., $|\varepsilon_i \psi(\mathbf{x})| \ll 1$, the nonlinear Poisson-Boltzmann equation (4.2.3) can be linearized to (see [50] for this form)

$$
-\nabla \cdot [\varepsilon(\mathbf{x})\nabla \psi(\mathbf{x})] + \bar{\kappa}(\mathbf{x})^2 \psi(\mathbf{x}) = 4\pi \rho_M(\mathbf{x}),
$$

(4.2.4)

where $\psi$ is determined by the data $\varepsilon(\mathbf{x})$, $\bar{\kappa}(\mathbf{x})$ and $\rho_M(\mathbf{x})$ that are introduced in Section 4.1.1.

**Remark 4.2.1.** If the ionic solution has more than two types of ions, the associated Poisson-Boltzmann equation can still be linearized to the same form (4.2.4).

In the definition (1.1.2) of the dielectric permittivity function $\varepsilon(\mathbf{x})$, the solute (relative) dielectric permittivity $\varepsilon_1$ should theoretically be set to 1 as in the vacuum (for example, in [26]). Sometimes, values different from $\varepsilon_1 = 1$ are used. For example, in [80] and [63], Section 1.4], the authors claim to obtain better approximations with the empirical value $\varepsilon_1 = 2$. The solvent dielectric permittivity $\varepsilon_2$ is determined by the solvent as well as the temperature, for example, $\varepsilon_2 = 78.4$ for water at room
temperature $25^\circ C$. The modified Debye-Hückel parameter in the implicit solvation model with a $1:1$ electrolyte is taken as

$$
\kappa(x) = \begin{cases} 
0 & \text{in } \Omega, \\
\sqrt{\varepsilon_2} \kappa & \text{in } \Omega^c,
\end{cases}
$$

(4.2.5)

where $\kappa$ is the Debye-Hückel screening constant representing the attenuation of interactions due to the presence of ions in the solvent region, which is related to the ionic strength $I$ of the ionic solution according to (see [86] and [63], Section 1.4) for the following formula

$$
\kappa^2 = \frac{8\pi e^2 N_A I}{1000\varepsilon_2 k_B T},
$$

(4.2.6)

where $N_A$ is the Avogadro constant.

Furthermore, it is usually assumed that the solute’s charge distribution $\rho_M$ is supported in $\Omega$. Here, we assume that $\rho_M$ is given by the sum of $M$ point charges in the following form

$$
\rho_M(x) = \sum_{i=1}^{M} q_i \delta(x - x_i),
$$

(4.2.7)

where $M$ is the number of solute atoms, $q_i$ represents the (partial) charge carried on the $i$th atom with center $x_i$, $\delta$ is the Dirac delta function. As a consequence, $\rho_M$ is supported in $\Omega$ as assumed.

### 4.3 Problem transformation

In this section, we first introduce the integral representation of the LPB equation in the potential theory. Based on this, we then transform the original electrostatic problem to two coupled equations restricted to the (bounded) solute cavity.

#### 4.3.1 Problem setting

The LPB equation can be divided into two equations: first, the Poisson equation in the solute cavity and second, the HSP equation in the solvent region. That is to say, the problem is recast in the following form

$$
\begin{cases}
- \Delta \psi(x) = \frac{4\pi}{\varepsilon_1} \rho_M(x) & \text{in } \Omega, \\
- \Delta \psi(x) + \kappa^2 \psi(x) = 0 & \text{in } \Omega^c,
\end{cases}
$$

(4.3.1)
with two classical jump-conditions

\[
\begin{align*}
[\psi] &= 0 \quad \text{on } \Gamma, \\
[\partial_n (\varepsilon \psi)] &= 0 \quad \text{on } \Gamma,
\end{align*}
\]

(4.3.2)

where \( \Gamma := \partial \Omega \) is the solute-solvent boundary, \( \mathbf{n} \) is the unit normal vector on \( \Gamma \) pointing outwards with respect to \( \Omega \) and \( \partial_n = \mathbf{n} \cdot \nabla \) is the notation of normal derivative. \([\psi]\) represents the jump (inside minus outside) of the potential and \([\partial_n (\varepsilon \psi)]\) represents the jump of the normal derivative of the electrostatic potential multiplied by the dielectric permittivity.

### 4.3.2 Necessary tools from the potential theory

The free-space Green’s function of the operator \(-\Delta\) is given as

\[
G(x, y) = \frac{1}{4\pi |x - y|}, \quad \forall x, y \in \mathbb{R}^3,
\]

(4.3.3)

and similarly, the free-space Green’s function of the operator \(-\Delta + \kappa^2\) is given as

\[
G_\kappa(x, y) = \frac{e^{-\kappa|x - y|}}{4\pi |x - y|}, \quad \forall x, y \in \mathbb{R}^3,
\]

(4.3.4)

which yields

\[
-\Delta x G(x, y) = \delta(x - y), \quad \forall y \in \mathbb{R}^3,
\]

(4.3.5)

and

\[
-\Delta x G_\kappa(x, y) + \kappa^2 G_\kappa(x, y) = \delta(x - y), \quad \forall y \in \mathbb{R}^3.
\]

(4.3.6)

In the solute cavity \( \Omega \), we define the reaction potential \( \psi_r := \psi - \psi_0 \), where \( \psi_0 \) is the potential generated by \( \rho_M \) in vacuum written as

\[
\psi_0 = \sum_{i=1}^{M} \frac{q_i}{|x - x_i|},
\]

(4.3.7)

satisfying \(-\Delta \psi_0 = \frac{4\pi}{\varepsilon_1} \rho_M \) in \( \mathbb{R}^3 \). Then, \( \psi_r \) is harmonic in \( \Omega \), that is,

\[
-\Delta \psi_r = 0, \quad \text{in } \Omega,
\]

(4.3.8)

which yields the following integral equation formulation

\[
\psi_r(x) = \tilde{S} \sigma_r(x) := \int_{\Gamma} \frac{\sigma_r(y)}{4\pi |x - y|}, \quad \forall x \in \Omega,
\]

(4.3.9)

where \( \sigma_r \) is some function in \( H^{-\frac{1}{2}}(\Gamma) \) and \( \tilde{S} : H^{-\frac{1}{2}}(\Gamma) \rightarrow H^{1}(\mathbb{R}^3 \setminus \Gamma) \) is the single-layer
potential associated with $G$.

Furthermore, according to the HSP equation in (4.3.1), the electrostatic potential in the solvent region $\Omega^e$ can be represented by

$$\psi|_{\Omega^e}(x) = \tilde{S}_\kappa \sigma_e(x) := \int_\Gamma \frac{e^{-\kappa|x-y|} \sigma_e(y)}{4\pi|x-y|}, \quad \forall x \in \Omega^e, \quad (4.3.10)$$

where $\sigma_e$ is another function in $H^{-\frac{1}{2}}(\Gamma)$ and $\tilde{S}_\kappa : H^{-\frac{1}{2}}(\Gamma) \rightarrow H^1(\mathbb{R}^3 \setminus \Gamma)$ is the single-layer potential associated with $G_\kappa$. Here, we also introduce the single-layer operator $S_\kappa : H^{-\frac{1}{2}}(\Gamma) \rightarrow H^\frac{1}{2}(\Gamma)$ defined by

$$S_\kappa \sigma_e(x) := \int_\Gamma \frac{e^{-\kappa|x-y|} \sigma_e(y)}{4\pi|x-y|}, \quad \forall x \in \Gamma, \quad (4.3.11)$$

which is an invertible operator (this is true according to the proof of the invertibility of the single-layer operator for the Helmholtz equation, see [10, Corollary 7.26] and [104, Theorem 3.9.1]). The invertibility of $S_\kappa$ implies that $\sigma_e$ can be characterized as $\sigma_e = S_\kappa^{-1} \psi|_{\Gamma}$.

### 4.3.3 Transformation

We will now transform the original problem defined in $\mathbb{R}^3$ equivalently to two coupled equations both defined in the solute cavity.

According to the continuity of the single-layer potential $\tilde{S}_\kappa$ across the interface $\Omega^e$, we can artificially extend the electrostatic potential $\psi|_{\Omega^e}$ from $\Omega^e$ to $\Omega$ as follows

$$\psi_e(x) := \tilde{S}_\kappa \sigma_e(x) = \int_\Gamma \frac{e^{-\kappa|x-y|} \sigma_e(y)}{4\pi|x-y|}, \quad \forall x \in \Omega, \quad (4.3.12)$$

where $\psi_e$ is called the extended potential in this chapter. As a consequence, $\psi_e$ satisfies the same HSP equation as $\psi|_{\Omega^e}$, but defined on $\Omega$, as follows

$$-\Delta \psi_e(x) + \kappa^2 \psi_e(x) = 0, \quad \text{in } \Omega, \quad (4.3.13)$$

with the same Dirichlet boundary conditions on $\Gamma$. Furthermore, from [104, Theorem 3.3.1], we have a relation among $\sigma_e$ and the normal derivatives of $\psi_e$ and $\psi|_{\Omega^e}$ on $\Gamma$ as follows

$$\sigma_e = \partial_n \psi_e - \partial_n \psi|_{\Omega^e}, \quad \text{on } \Gamma. \quad (4.3.14)$$

As introduced above, Eq. (4.3.8) of $\psi$ and (4.3.13) of $\psi_e$ are two PDEs defined on $\Omega$, that are derived from the original LPB equation (4.3.1). As a consequence, it is sufficient to couple these two equations. According to $[\psi] = 0$ on $\Gamma$ and the
continuity of \( \hat{S}_\kappa \) across \( \Gamma \), we then deduce a first coupling condition

\[
\psi_0 + \psi_r = \psi_e, \quad \text{on } \Gamma. \tag{4.3.15}
\]

Further, combining Eq. (4.3.14) with the second equation of the jump conditions (4.3.2), i.e.,

\[
\varepsilon_1 \partial_n \psi|_\Omega - \varepsilon_2 \partial_n \psi|_{\Omega^c} = 0, \quad \text{on } \Gamma, \tag{4.3.16}
\]

we deduce another coupling condition

\[
\sigma_e = \partial_n \psi_e - \frac{\varepsilon_1}{\varepsilon_2} \partial_n (\psi_0 + \psi_r), \quad \text{on } \Gamma. \tag{4.3.17}
\]

In summary, the original problem (4.3.11) is transformed into the following two equations defined on \( \Omega \)

\[
\begin{cases}
- \Delta \psi_l(x) = 0 & \text{in } \Omega, \\
- \Delta \psi_e(x) + \kappa^2 \psi_e(x) = 0 & \text{in } \Omega,
\end{cases} \tag{4.3.18}
\]

with two coupling conditions on \( \Gamma \) given by

\[
\begin{cases}
\psi_0 + \psi_r = \psi_e & \text{on } \Gamma, \\
\sigma_e = \partial_n \psi_e - \frac{\varepsilon_1}{\varepsilon_2} \partial_n (\psi_0 + \psi_r) & \text{on } \Gamma,
\end{cases} \tag{4.3.19}
\]

where \( \sigma_e \) is the charge density generating \( \psi_e \), as presented in (4.3.12). The second equation of (4.3.19) is also equivalent to

\[
\psi_e = S_\kappa \left( \partial_n \psi_e - \frac{\varepsilon_1}{\varepsilon_2} \partial_n (\psi_0 + \psi_r) \right), \quad \text{on } \Gamma, \tag{4.3.20}
\]

which is derived from letting \( S_\kappa \) act on both sides of the equation.

\section{4.4 Strategy}

In this section, we introduce a global strategy for solving Eqs (4.3.18)–(4.3.19) that are derived from the LPB equation (4.2.4). Then, we present how the domain decomposition method can be applied to solve the two PDEs defined on \( \Omega \), taking advantage of its particular geometrical structure (i.e., a union of overlapping balls). The scheme of this section is inspired by [92, Section 4.2 and 5], our previous work for the case of non-ionic solvent.
4.4.1 Global strategy

We propose the following iterative procedure for solving Eqs (4.3.18)–(4.3.19): let \( g^0 \) defined on \( \Gamma \) be an initial guess for the Dirichlet condition \( \psi_{\Gamma} \) and set \( k = 1 \).

1. Solve the following Dirichlet boundary problem for \( \psi_r^k \):

\[
\begin{cases}
-\Delta \psi_r^k = 0 & \text{in } \Omega, \\
\psi_r^k = g^{k-1} - \psi_0 & \text{on } \Gamma,
\end{cases}
\]

and derive its Neumann boundary trace \( \partial_n \psi_r^k \) on \( \Gamma \).

2. Solve the following Dirichlet boundary problem for \( \psi_e^k \):

\[
\begin{cases}
-\Delta \psi_e^k + \kappa^2 \psi_e^k = 0 & \text{in } \Omega, \\
\psi_e^k = g^{k-1} & \text{on } \Gamma,
\end{cases}
\]

and derive similarly its Neumann boundary trace \( \partial_n \psi_e^k \) on \( \Gamma \).

3. Build the charge density \( \sigma_e^k = \partial_n \psi_e^k - \frac{e_1}{e_2} \partial_n \left( \psi_0 + \psi_r^k \right) \) and compute a new Dirichlet condition \( g^k = S_\kappa \sigma_e^k \).

4. Compute the contribution \( E_s^k \) to the solvation energy based on \( \psi_r^k \) at the \( k \)-th iteration, set \( k \leftarrow k + 1 \), go back to Step [1] and repeat until the increment of interaction \( |E_s^k - E_s^{k-1}| \) becomes smaller than a given tolerance \( \text{Tol} \ll 1 \).

**Remark 4.4.1.** In order to provide a suitable initial guess of \( g^0 \) (defined on \( \Gamma \)), we consider the (unrealistic) scenario where the whole space \( \mathbb{R}^3 \) is covered by the solvent medium. Then, the electrostatic potential \( \psi \) in this case is given explicitly by

\[
\psi(x) = \sum_{i=1}^{M} \frac{4\pi q_i e^{-\kappa|x-x_i|}}{e_2 |x-x_i|}, \quad \forall x \in \mathbb{R}^3,
\]

see details in [63, Section 1.3.2]. As a consequence, we choose \( g^0 \) as this potential restricted on \( \Gamma \).

**Remark 4.4.2.** The above global strategy is an iterative procedure, which is presented for an easier understanding. However, the final convergent solution satisfies, after discretization, a global linear system that can be solved by different linear algebra algorithms. We will address this issue in the later Section 4.6.2.

4.4.2 Domain decomposition scheme

The Schwarz’s domain decomposition method [96] is a good choice to solve the PDE defined on a complex domain which can be composed as a union of overlapping
and possibly simple subdomains. According to the definition of $\Omega$, we have a natural domain decomposition as follows

$$\Omega = \bigcup_{j=1}^{M} \Omega_j, \quad \Omega_j = B_{r_j}(x_j),$$

where each $\Omega_j$ denotes the $j$-th atomic VdW-ball (or SAS-ball) with center $x_j$ and radius $r_j$. As a consequence, the Schwarz’s domain decomposition method can be applied to solve the PDEs (4.4.1) and (4.4.2).

Similar to the ddCOSMO method \[25\], Eq. (4.4.1) is equivalent to the following coupled local equations, each restricted to $\Omega_j$:

$$\begin{cases}
-\Delta \psi_r|_{\Omega_j} = 0 & \text{in } \Omega_j, \\
\psi_r|_{\Gamma_j} = \phi_r|_{\Gamma_j} & \text{on } \Gamma_j,
\end{cases} \quad (4.4.4)$$

where $\Gamma_j = \partial \Omega_j$ and

$$\phi_r|_{\Gamma_j} = \begin{cases}
\overline{\psi}_r & \text{on } \Gamma_j^i, \\
g - \psi_0 & \text{on } \Gamma_j^e. 
\end{cases} \quad (4.4.5)$$

Here, we omit the superscript due to the (outer) iteration index $k$. $\Gamma_j^e$ is the external part of $\Gamma_j$ not contained in any other ball $\Omega_i$ ($i \neq j$), i.e., $\Gamma_j^e = \Gamma \cap \Gamma_j$; $\Gamma_j^i$ is the internal part of $\Gamma_j$, i.e., $\Gamma_j^i = \Omega \cap \Gamma_j$ (see Figure 3 for an illustration). In addition,

$$\overline{\psi}_r(x) = \frac{1}{|\mathcal{N}(j,x)|} \sum_{i \in \mathcal{N}(j,x)} \psi_r|_{\Omega_i}(x), \quad \forall x \in \Gamma_j^i, \quad (4.4.6)$$

where $\mathcal{N}(j,x)$ represents the index set of all balls such that $x \in \Omega_i$. In fact, for a fixed point $x \in \Gamma_j^i$, we enforce $\psi_r|_{\Gamma_j}(x) = \overline{\psi}_r(x)$, which is the average value of $\psi_r|_{\Omega_i}(x)$ each computed by solving the local Laplace equation in $\Omega_i$.

Similarly, Eq. (4.4.2) is equivalent to the following coupled local equations, each restricted to $\Omega_j$:

$$\begin{cases}
-\Delta \psi_e|_{\Omega_j} + \kappa^2 \psi_e|_{\Omega_j} = 0 & \text{in } \Omega_j, \\
\psi_e|_{\Omega_j} = \phi_e|_{\Gamma_j} & \text{on } \Gamma_j,
\end{cases} \quad (4.4.7)$$

where

$$\phi_e|_{\Gamma_j} = \begin{cases}
\overline{\psi}_e & \text{on } \Gamma_j^i, \\
g & \text{on } \Gamma_j^e. 
\end{cases} \quad (4.4.8)$$

Here, in the same spirit as the definition of $\overline{\psi}_r$, we define

$$\overline{\psi}_e(x) = \frac{1}{|\mathcal{N}(j,x)|} \sum_{i \in \mathcal{N}(j,x)} \psi_e|_{\Omega_i}(x), \quad \forall x \in \Gamma_j^i, \quad (4.4.9)$$

meaning that for a fixed point $x \in \Gamma_j^i$, $\overline{\psi}_e(x)$ is taken to be the average value of
Figure 3: 2D schematic diagram of $\Gamma^i_j$ (red) and $\Gamma^e_j$ (blue) associated with $\Omega_j$.

$\psi_{e|\Omega_i}(\mathbf{x})$ each computed by solving the local HSP equation in $\Omega_i$. The notations $\Gamma^e_j$, $\Gamma^i_j$ and $N$ are defined as previously.

Generally speaking, given the Dirichlet boundary condition $g$ on $\Gamma$, an iterative procedure can be applied to solve the coupled equations (4.4.4)–(4.4.5) (respectively (4.4.7)–(4.4.8)), such as the parallel and alternating Schwarz algorithms as presented in the ddCOSMO [25]. For example, the idea of the parallel algorithm is to solve each local problem based on the boundary condition of the neighboring solutions derived from the previous iteration. During this iterative procedure, the computed value of $\psi_i|\Gamma^i_j$ (respectively $\psi_e|\Gamma^e_j$) is updated step by step and converges to the exact value.

However, the parallel or alternating Schwarz algorithms might not be the most efficient way to solve such a set of equations, but is well-suited to illustrate the idea of the domain decomposition method. In fact, the global linear system derived from discretization can be solved by different linear algebra algorithms (for example, the GMRes method). We will present more details about this in Section 4.6.2. Before that, we shall develop two single-domain solvers: a Laplace solver and a HSP solver both in the unit ball.

### 4.5 Single-domain solvers

In this section, we develop two single-domain solvers in the unit ball respectively for solving Eq. (4.4.4) and Eq. (4.4.7) within the domain decomposition scheme.

#### 4.5.1 Laplace solver

Developing a Laplace solver in a ball is not difficult, which has been presented in our previous work including the ddCOSMO [25], the ddPCM [108] and the ddPCM-SES [95]. For the sake of completeness, we recall briefly the Laplace solver in the following content.

We consider Eq. (4.4.4) defined on $\Omega_j$ which can actually be transformed easily...
into a Laplace equation defined in the unit ball $B_1(0)$ in the form of
\[
\begin{cases}
-\Delta u_r = 0 & \text{in } B_1(0), \\
u_r = \phi_r & \text{on } S^2.
\end{cases}
\] (4.5.1)

The unique solution in $H^1(B_1(0))$ can be written as
\[
u_r(r, \theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} [\phi_{\ell}][m] r^\ell Y_{\ell}^m(\theta, \varphi), \quad 0 \leq r \leq 1, \ 0 \leq \theta \leq \pi, \ 0 \leq \varphi < 2\pi. \] (4.5.2)

Here, $Y_{\ell}^m$ denotes the (real orthonormal) spherical harmonic of degree $\ell$ and order $m$ defined on $S^2$ and
\[ [\phi_{\ell}][m] = \int_{S^2} \phi_r(s)Y_{\ell}^m(s)ds, \]
is the real coefficient of $u_r$ corresponding to the mode $Y_{\ell}^m$. Then, $u_r$ can be numerically approximated by $\bar{u}_r$ in the discretization space spanned by a truncated basis of spherical harmonics $\{Y_{\ell}^m\}_{0 \leq \ell \leq \ell_{\text{max}}, -\ell \leq m \leq \ell}$, defined as
\[
\bar{u}_r(r, \theta, \varphi) = \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} [\bar{\phi}_{\ell}][m] r^\ell Y_{\ell}^m(\theta, \varphi), \quad 0 \leq r \leq 1, \ 0 \leq \theta \leq \pi, \ 0 \leq \varphi < 2\pi, \] (4.5.3)
where $\ell_{\text{max}}$ denotes the maximum degree of spherical harmonics and
\[
[\bar{\phi}_{\ell}][m] = \sum_{n=1}^{N_{\text{leb}}} u_n^{\text{leb}} \phi_{\ell}(s_n)Y_{\ell}^m(s_n). \] (4.5.4)

Here, $s_n \in S^2$ represent Lebedev quadrature points, $u_n^{\text{leb}}$ are the corresponding weights and $N_{\text{leb}}$ is the number of Lebedev quadrature points.

**4.5.2 HSP solver**

By a linear transformation of variables, Eq. (4.4.7) defined on $\Omega_j$ can be transformed into a HSP equation defined in the unit ball $B_1(0)$ in the form of
\[
\begin{cases}
-\Delta u_e + \kappa^2 u_e^2 = 0 & \text{in } B_1(0), \\
u_e = \phi_e & \text{on } S^2.
\end{cases}
\] (4.5.5)

Here, we actually take $u_e(y) = \psi_e(r_jy + x_j), \forall y \in B_1(0), \kappa_e = \kappa r_j$ and furthermore, $\phi_e(y) = \phi_{e,j}(r_jy + x_j), \forall y \in S^2$. By solving the above HSP equation in spherical coordinates by separation of variables, the radial equation corresponding to the angular dependency $Y_{\ell}^m$ has the form
\[
\frac{1}{R} \frac{d}{dr} \left( r^2 \frac{dR}{dr} \right) = \kappa^2 r^2 + \ell(\ell + 1), \quad n \geq 0, \] (4.5.6)
that is,
\[ r^2 \frac{d^2 R}{dr^2} + 2r \frac{dR}{dr} - (\kappa_e^2 r^2 + \ell(\ell + 1))R = 0, \tag{4.5.7} \]
which is called the modified spherical Bessel equation \[3\]. This equation has two linearly independent solutions as follows
\[ i_\ell(\kappa_e r) = \sqrt{\frac{\pi}{2\kappa_e r}} I_{\ell + \frac{1}{2}}(\kappa_e r); \quad k_\ell(\kappa_e r) = \sqrt{\frac{2}{\pi \kappa_e r}} K_{\ell + \frac{1}{2}}(\kappa_e r), \tag{4.5.8} \]
where \( i_\ell \) and \( k_\ell \) are the modified spherical Bessel functions of the first and second kind, see \[3\], Chapter 14 for details and Figure 4 for an illustration. Here, \( I_\ell(x) \) and \( K_\ell(x) \) with subscript \( \alpha \) are the modified Bessel functions of the first and second kind \[1\].

**Remark 4.5.1.** \( I_\alpha(x) \) and \( K_\alpha(x) \) satisfy the modified Bessel equation
\[ x^2 \frac{d^2 f}{dx^2} + x \frac{df}{dx} - (x^2 + \alpha^2) f = 0. \tag{4.5.9} \]
In fact, \( I_\alpha \) and \( K_\alpha \) are exponentially growing and decaying functions, respectively.

Since \( k_\ell \to \infty \) as \( r \to 0 \), we are interested in the family \( i_\ell \) of the first kind. That is, we write the solution to (4.5.5) in the form of
\[ u_e(r, \theta, \varphi) = \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} c_{\ell}^m i_\ell(\kappa_e r) Y_{\ell m}(\theta, \varphi), \quad 0 \leq r \leq 1, \ 0 \leq \theta \leq \pi, \ \ 0 \leq \varphi < 2\pi, \tag{4.5.10} \]
where \( c_{\ell}^m \) is the coefficient of the mode \( Y_{\ell m} \). With the same discretization as in Section 4.5.1, we derive the following approximate solution similar to Eq. (4.5.3):
\[ \tilde{u}_e(r, \theta, \varphi) = \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} \tilde{\phi}_{\ell}^m i_\ell(\kappa_e r) Y_{\ell m}(\theta, \varphi), \quad 0 \leq r \leq 1, \ 0 \leq \theta \leq \pi, \ \ 0 \leq \varphi < 2\pi, \tag{4.5.11} \]
where \( \tilde{\phi}_{\ell}^m \) is given similar to (4.5.4) as follows
\[ \tilde{\phi}_{\ell}^m = \sum_{n=1}^{N_{\text{leb}}} w_n^\text{leb} \phi_e(s_n) Y_{\ell m}(s_n), \tag{4.5.12} \]
with the same notations \( w_n^\text{leb} \) and \( N_{\text{leb}} \) as above.

**Remark 4.5.2.** We have \( \frac{i_\ell(\kappa_e r)}{i_\ell(\kappa_e)} \to r^\ell \) as \( \kappa_e \to 0 \) and therefore if \( \phi_t = \phi_e \), \( \tilde{u}_e \to \tilde{u}_t \) as \( \kappa_e \to 0 \). This means that the solution to Eq. (4.5.5) tends to the solution to Eq. (4.5.1) when \( \phi_t = \phi_e \) and \( \kappa_e \to 0 \), which makes sense.
4.6. Global linear system

The global strategy in Section 4.4 in combination with the domain decomposition schemes for solving Eq. (4.4.1) and Eq. (4.4.2) is an iterative procedure. This implies that the proposed algorithm can be parallelized since only a sequence of local problems on each $\Omega_j$ are solved. However, as mentioned in Remark 4.4.2, we will solve the problem in a global way, meaning that we finally solve a global linear system derived from discretization. To present this, we first introduce a reformulation of the coupling conditions and then present the global linear system for its discretization.

4.6.1 Reformulation of the coupling conditions

Let $\chi_i$ be the characteristic function of $\Omega_i$, i.e.,

$$\chi_i(x) := \begin{cases} 1 & \text{if } x \in \Omega_i \\ 0 & \text{if } x \not\in \Omega_i \end{cases}$$

(4.6.1)
and then let
\[
 w_{ji}(x) := \frac{\chi_i(x)}{|N(j, x)|} = \frac{\chi_i(x)}{\sum_{i \neq j} \chi_i(x)}, \quad \forall x \in \Gamma_j. \tag{4.6.2}
\]
Here, we make the convention that in the case of $|N(j, x)| = 0$ (i.e., $x \in \Gamma_j^i$), we define $w_{ji}(x) = 0$, $\forall i$. Furthermore, $\forall x \in \Gamma_j$, we define
\[
 \chi_j^e(x) := \begin{cases} 1 & \text{if } x \in \Gamma_j^e, \\ 0 & \text{if } x \in \Gamma_j^i, \end{cases} \tag{4.6.3}
\]
which is equivalent to
\[
 \chi_j^e(x) = 1 - \sum_{i \neq j} w_{ji}(x), \quad \forall x \in \Gamma_j. \tag{4.6.4}
\]

There are two local coupling conditions, Eq. (4.4.5) and (4.4.8), respectively for coupling the local Laplace equation (4.4.4) and the local HSP equation (4.4.7). Based on the above-defined notations, Eq. (4.4.5) can be recast as
\[
 \psi_{i|\Gamma_j}(x) - \sum_{i \neq j} w_{ji}(x) \psi_{i|\Omega_i}(x) = \chi_j^e(x) (g(x) - \psi_0(x)), \quad \forall x \in \Gamma_j. \tag{4.6.5}
\]
Similarly, Eq. (4.4.8) can be recast as
\[
 \psi_{e|\Gamma_j}(x) - \sum_{i \neq j} w_{ji}(x) \psi_{e|\Omega_i}(x) = \chi_j^e(x) g(x), \quad \forall x \in \Gamma_j. \tag{4.6.6}
\]

In addition, there is a global coupling condition on $\Gamma$, Eq. (4.3.20), between the global Laplace equation and the global HSP equation in (4.3.18), involving the nonlocal operator $S_\kappa$:
\[
 g(x) = \psi_{e|\Gamma}(x) = S_\kappa \left( \partial_n \psi_e - \frac{\varepsilon_1}{\varepsilon_2} \partial_n (\psi_0 + \psi_i) \right)(x), \quad \forall x \in \Gamma. \tag{4.6.7}
\]
The single-layer operator $S_\kappa$ involves an integral over the whole solute-solvent boundary $\Gamma$ which seems difficult to compute at a first glance. We now introduce a technique to compute the integral of $S_\kappa$ efficiently. For each sphere $\Gamma_i$, we define a local single-layer potential $\tilde{S}_{\kappa, \Gamma_i}$ as follows
\[
 \tilde{S}_{\kappa, \Gamma_i}(x) := \int_{\Gamma_i} \frac{e^{-\kappa|x-y|} \sigma(y)}{4\pi |x-y|}, \quad \forall x \in \mathbb{R}^3, \tag{4.6.8}
\]
where $\sigma$ is an arbitrary function in $H^{-\frac{1}{2}}(\Gamma_i)$. As a consequence, $\forall \sigma \in H^{-\frac{1}{2}}(\Gamma)$, we have
\[
 S_\kappa \sigma = \sum_{i=1}^{M} \tilde{S}_{\kappa, \Gamma_i} (\chi_j^e \sigma), \tag{4.6.9}
\]
where \( \chi_i^e \sigma \) extends \( \sigma|_{\Gamma_i} \) by zero to the whole sphere \( \Gamma_i \). The above equation implies that the integral over \( \Gamma \) can be divided into a group of integrals respectively over each sphere \( \Gamma_i \). Therefore, Eq. (4.6.7) can be recast as

\[
g(x) = \sum_{i=1}^{M} \tilde{S}_{x, \Gamma_i} \left[ \chi_i^e \left( \partial_n \psi_e - \frac{\varepsilon_1}{\varepsilon_2} \partial_n (\psi_0 + \psi_i) \right) \right](x), \quad \forall x \in \Gamma. \tag{4.6.10}
\]

### 4.6.2 Derivation of the linear system

In this part, we first present the discretization of the above reformulation and then introduce the global linear system derived from this discretization.

#### 4.6.2.1 Local discretization in each ball

In Section 4.5, without the loss of generalization, we have presented the discretization of the solutions to the Laplace equation and the HSP equation defined in the unit ball.

Based on this, for each sphere \( \Gamma_j \), we first approximate \( r_j \) and \( e_j \) respectively by a linear combination of spherical harmonics \( \{Y^m_\ell\} \) with \( 0 \leq \ell \leq \ell_{\text{max}} \) and \( -\ell \leq m \leq \ell \) as follows

\[
\psi_t|_{\Gamma_j}(x_j + r_j s) = \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} [X_t]_{j\ell m} Y^m_\ell(s), \quad s \in S^2, \tag{4.6.11}
\]

and

\[
\psi_e|_{\Gamma_j}(x_j + r_j s) = \sum_{\ell=0}^{\ell_{\text{max}}} \sum_{m=-\ell}^{\ell} [X_e]_{j\ell m} Y^m_\ell(s), \quad s \in S^2, \tag{4.6.12}
\]

where \( [X_t]_{j\ell m} \) and \( [X_e]_{j\ell m} \) are unknown coefficients of the mode \( Y^m_\ell \) respectively associated with \( \psi_t|_{\Gamma_j} \) and \( \psi_e|_{\Gamma_j} \). Here, for any point \( x \in \Gamma_j \), we actually use its spherical coordinates \( (r_j, s) \) s.t. \( x = x_j + r_j s \). According to the Laplace solver and the HSP solver presented in Section 4.5, we deduce directly

\[
\psi_t|_{\Omega_i}(x_i + r s) = \sum_{\ell' = 0}^{\ell_{\text{max}}} \sum_{m' = -\ell'}^{\ell'} [X_t]_{i\ell' m'} \left( \frac{r}{r_i} \right)^{\ell'} Y^{m'}_{\ell'}(s), \quad 0 \leq r \leq r_i, \quad s \in S^2, \tag{4.6.13}
\]

and

\[
\psi_e|_{\Omega_i}(x_i + r s) = \sum_{\ell' = 0}^{\ell_{\text{max}}} \sum_{m' = -\ell'}^{\ell'} [X_e]_{i\ell' m'} \frac{i_{\ell'} (kr)}{i_{\ell'} (kr_i)} Y^{m'}_{\ell'}(s), \quad 0 \leq r \leq r_i, \quad s \in S^2, \tag{4.6.14}
\]

where for any point \( x \in \Omega_i \), we take its spherical coordinates \( (r, s) \) s.t. \( x = x_i + rs \).
To compute the discretization of the coupling conditions, we can compute the normal derivative of $\psi_i$ on $\Gamma_i^e$ as follows:

$$\partial_n \psi_i(x_i + r_i s) = \sum_{\ell'=0}^{\ell_{\text{max}}} \sum_{m'=-\ell'}^{\ell'} [X_i]_{\ell' m'} \left( \frac{\ell'}{r_i} \right) Y^m_{\ell'}(s), \quad x_i + r_i s \in \Gamma_i^e,$$

(4.6.15)

and the normal derivative of $\psi_e$ on $\Gamma_i^e$

$$\partial_n \psi_e(x_i + r_i s) = \sum_{\ell'=0}^{\ell_{\text{max}}} \sum_{m'=-\ell'}^{\ell'} [X_e]_{\ell' m'} \frac{k_i}{i_{\ell'}(kr_i)} \frac{\ell'}{i_{\ell'}(kr_i)} Y^m_{\ell'}(s), \quad x_i + r_i s \in \Gamma_i^e,$$

(4.6.16)

where $i_{\ell'}$ represents the derivative of $i_{\ell'}$.

**Remark 4.6.1.** To compute $i'_{\ell'}(kr_i)$, we use the iterative property of the derivative of $i_n$ (see [3, Page 707]) given by

$$(2n + 1)i'_{n}(x) = ni_{n-1}(x) + (n + 1)i_{n+1}(x),$$

(4.6.17)

and in analogy, the derivative of $k_n$ as follows

$$-(2n + 1)k'_{n}(x) = nk_{n-1}(x) + (n + 1)k_{n+1}(x),$$

(4.6.18)

which will be used to compute $k'_{\ell'}(kr_i)$ in the Appendix [3].

### 4.6.2.2 Discretization of the coupling conditions

So far, we have written the Ansatz for the unknowns $\psi_i|_{\Gamma_j}$, $\psi_e|_{\Gamma_j}$, respectively $\psi_i|_{\Omega_i}$, $\psi_e|_{\Omega_i}$ with normal derivatives $\partial_n \psi_i$ and $\partial_n \psi_e$, following Eqs (4.6.11) - (4.6.10), which depend on the unknowns $X_i$ and $X_e$. This allows us to discretize the coupling conditions (4.6.5), (4.6.3) and (4.6.10).

First, we replace the variable $x \in \Gamma_j$ of Eq. (4.6.3) by $x = x_j + r_j s$ with $s \in S^2$ and derive the equation for each sphere $\Gamma_j$ as follows

$$\psi_i|_{\Gamma_j}(x_j + r_j s) - \sum_{i \neq j} w_{ji}(x_j + r_j s) \psi_i|_{\Omega_i}(x_j + r_j s) = \chi_j^e(x_j + r_j s) \left( g(x_j + r_j s) - \psi_0(x_j + r_j s) \right),$$

(4.6.19)

which induces the following local equation by multiplying by $Y^m_{\ell}$ and integrating over $S^2$ on both sides, $\forall j, \ell, m$,

$$\left\langle \psi_i|_{\Gamma_j}(x_j + r_j \cdot) - \sum_{i \neq j} w_{ji}(x_j + r_j \cdot) \psi_i|_{\Omega_i}(x_j + r_j \cdot), Y^m_{\ell} (\cdot) \right\rangle_{S^2} = \left\langle \chi_j^e(x_j + r_j \cdot) \left( g(x_j + r_j \cdot) - \psi_0(x_j + r_j \cdot) \right), Y^m_{\ell} (\cdot) \right\rangle_{S^2}.$$

(4.6.20)

Here, $\langle \cdot, \cdot \rangle_{S^2}$ represents the integral over the unit sphere $S^2$, which is numerically approximated using the Lebedev quadrature rule with $N_{\text{leb}}$ points. We therefore
denote such a numerical integration over $\mathbb{S}^2$ by the notation $\langle \cdot, \cdot \rangle_{\mathbb{S}^2, N_{\text{leb}}}$. Eq. (4.6.20) can be rewritten in the form of a linear system

$$\begin{bmatrix} A & \mathbf{X}_r \end{bmatrix}_{j\ell m} = \begin{bmatrix} \mathbf{G}_X & \mathbf{G}_0 \end{bmatrix}_{j\ell m}, \quad \forall j, \ell, m. \quad (4.6.21)$$

Here, $A$ is a square matrix of dimension $M(\ell_{\text{max}} + 1)^2 \times M(\ell_{\text{max}} + 1)^2$ and the $j\ell m$-th row of $AX_r$ is given by substituting (4.6.11) and (4.6.13) into (4.6.20) as follows

$$\begin{bmatrix} A & \mathbf{X}_r \end{bmatrix}_{j\ell m} = \left( \sum_{i\neq j} \sum_{m'} w_{n}^{\text{leb}} w_{ji}(x_j + r_j s_n) \left( \frac{r_{ij}}{r_i} \right)^{\ell'} Y^{m'}_{\ell'}(s_{ijn}) Y^{m}(s_n) \right)_{j\ell m}, \quad (4.6.22)$$

where $(r_{ijn}, s_{ijn})$ is the spherical coordinate associated with $\Gamma_i$ of the point $x_j + r_j s_n$ s.t.

$$x_j + r_j s_n = x_i + r_{ijn} s_{ijn}, \quad \text{with} \quad s_{ijn} \in \mathbb{S}^2.$$ 

Furthermore, the $j\ell m$-th element of the column vector $G_X$ is given as

$$\begin{bmatrix} G_X & \mathbf{G}_0 \end{bmatrix}_{j\ell m} = \left( \sum_{n=1}^{N_{\text{leb}}} w_{n}^{\text{leb}} \chi_n^e(x_j + r_j s_n) g(x_j + r_j s_n) Y^{m}(s_n) \right)_{j\ell m}, \quad (4.6.23)$$

which depends on the unknowns $X_r$ and $X_e$ through $g$ given by Eq. (4.6.10). The notation $X$ denotes the column of all unknowns, i.e.,

$$X = \begin{bmatrix} X_r \\ X_e \end{bmatrix} \in \mathbb{R}^{2M(\ell_{\text{max}} + 1)^2}. \quad (4.6.24)$$

Similarly, the $j\ell m$-th element of the column vector $G_0$ is given as

$$\begin{bmatrix} G_0 & \mathbf{G}_0 \end{bmatrix}_{j\ell m} = \left( - \sum_{n=1}^{N_{\text{leb}}} w_{n}^{\text{leb}} \chi_n^e(x_j + r_j s_n) \psi_0(x_j + r_j s_n) Y^{m}(s_n) \right)_{j\ell m}, \quad (4.6.25)$$

which can be computed a priori, since it is independent of $X$.

Similar to the linear system (4.6.21) and according to Eq. (4.6.6) for each sphere $\Gamma_j$, we have another linear system in the form of matrices

$$\begin{bmatrix} \mathbf{B} & \mathbf{X}_e \end{bmatrix}_{j\ell m} = \begin{bmatrix} \mathbf{G}_X \end{bmatrix}_{j\ell m}, \quad \forall j, \ell, m, \quad (4.6.26)$$
where the square matrix $B$ satisfies

$$
[BX_e]_{j\ell m} = [X_e]_{j\ell m} - \sum_{i \neq j} \sum_{\ell', m'} \left( \sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} w_{ji} (x_j + r_{js}) \frac{i_{\ell'} \left( k_{\ell' i j n} \right)}{i_{\ell'} \left( k_{\ell' i j n} \right)} Y_{\ell'}^{m'} (s_{ijn}) Y_{\ell}^{m} (s_n) \right) [X_e]_{i\ell' m'},
$$

and $[G_X]_{j\ell m}$ is given by (4.6.23).

**Remark 4.6.2.** For a fixed atomic ball $\Omega_i$ of a realistic molecule, especially when the molecule is large, the number of its intersections with other atomic balls $\Omega_j$ is bounded from above. From the definition (4.6.2) of $w_{ji}$, we have $w_{ji} (x_j + r_{js}) = 0$ if $r_{ijn} \geq r_i$. Since $w_{ji} \equiv 0$ when $\Omega_i \cap \Omega_j = \emptyset$, we deduce that $A$ and $B$ are both sparse matrices.

So far, we have derived two linear systems of the form

$$
\begin{align*}
\{ A X_r &= G_X + G_0, \\
B X_e &= G_X,
\end{align*}
$$

where $X_r$ and $X_e$ are the column vectors of unknowns $[X_r]_{j\ell m}$ and $[X_e]_{j\ell m}$ (respectively associated with the potentials $\psi_r$ and $\psi_e$). However, the column vector $G_X$ depending on both $X_r$ and $X_e$ is not specified yet. To do this, the coupling condition (4.6.10) in terms of $g$ should be used (which has not been used yet). Combining Eq. (4.6.10) with (4.6.23), we deduce the following form of $G_X$,

$$
G_X = F_0 - C_1 X_r - C_2 X_e,
$$

where the column vector $F_0$ is associated with $\partial_n \psi_0$, two dense square matrices $C_1$ and $C_2$ are respectively associated with $\partial_n \psi_r$ and $\partial_n \psi_e$. Considering the complexity of the formulas of $F_0$, $C_1$, $C_2$, we present them in Appendix. D.

With the form (4.6.29) of $G_X$, we finally obtain a global linear system written as

$$
\left( \begin{array}{cc}
A + C_1 & C_2 \\
C_1 & B + C_2
\end{array} \right) \left( \begin{array}{c}
X_r \\
X_e
\end{array} \right) = \left( \begin{array}{c}
G_0 + F_0 \\
F_0
\end{array} \right),
$$

which can also be reduced to

$$
\left( \begin{array}{cc}
A + C_1 & C_2 \\
A & -B
\end{array} \right) \left( \begin{array}{c}
X_r \\
X_e
\end{array} \right) = \left( \begin{array}{c}
G_0 + F_0 \\
G_0
\end{array} \right),
$$

where the matrix on the left-hand side is half-sparse, since both $A$ and $B$ are sparse (but not $C_1$ nor $C_2$).

To solve the linear system (4.6.30) or (4.6.31), the LU factorization method and the GMRes method can be used [101, 106], where the first one gives an exact solution while the second one gives an approximate solution. Furthermore, the global strategy
presented in Section 4.4 gives another iterative method with the following scheme

\[
\begin{pmatrix}
A & 0 \\
0 & B
\end{pmatrix}
\begin{pmatrix}
X_r^k \\
X_e^k
\end{pmatrix}
= -
\begin{pmatrix}
C_1 & C_2 \\
C_1 & C_2
\end{pmatrix}
\begin{pmatrix}
X_r^{k-1} \\
X_e^{k-1}
\end{pmatrix}
+ 
\begin{pmatrix}
G_0 + F_0 \\
F_0
\end{pmatrix},
\]

where \( k \) is the iteration number as in Section 4.4, \( X_r^k \) and \( X_e^k \) are respectively the values of \( X_r \) and \( X_e \) computed at the \( k \)-th iteration. This scheme implies that at each iteration, only a sparse linear system needs to be inverted, since \( A \) and \( B \) are sparse. We therefore employ the GMRes method to solve each sparse problem. Details about the efficiency of these three methods will be discussed in Section 4.7.4.

4.7 Numerical results

In an implicit solvation model, one crucial issue is to compute the solvation free energy, to which the electrostatic contribution plays an import role. In fact, the electrostatic contribution \( E_s \) to the solvation energy is computed from the reaction potential \( \psi_r \) according to the following formula (see [10] for the derivation of this formula)

\[
E_s = \frac{1}{2} \int_{\mathbb{R}^3} \rho_M(\mathbf{r}) \psi_r(\mathbf{r}) \, d\mathbf{r} = \frac{1}{2} \sum_{i=1}^{M} q_i \psi_r(\mathbf{x}_i),
\]

where the solute’s charge density \( \rho_M \) is given in Eq. (4.2.7) and \( \psi_r \) is obtained by solving the LPB equation. In the following content, we study the electrostatic contribution to the solvation energy computed numerically by the ddLPB method that is implemented in Matlab.

First of all, we take the dielectric permittivity in the solute cavity as in vacuum, that is, \( \varepsilon_1 = 1 \), and take the solvent to be water with the dielectric permittivity \( \varepsilon_2 = 78.4 \) at room temperature 25°C. In addition, the LU factorization method is used by default in the following sections to obtain the exact solutions to the global linear systems, while in Section 4.7.4, we test different linear algebra algorithms and present their running times.

4.7.1 Discretization parameters

We first determine the Debye-Hückel screening constant \( \kappa \) for a representative temperature of \( T = 298K \) (\( \approx 25^\circ C \)) and an ionic strength of \( I = 0.1 \), as in [63, Section 1.4]. According to Eq. (12.3), we compute \( \kappa = 0.1040 \). Then, we choose the SAS-cavity as the solute cavity, where the radius of each SAS-ball is equal to the atomic VdW-radius increased by the solvent radius. The solvent radius is set to \( r_p = 1.5 \AA \). Using the ddLPB method, we compute a highly accurate electrostatic contribution (denoted by \( E_s^{ex} \), the subscript “ex” represents “exact”) to the solvation
energy of formaldehyde with large discretization parameters $\ell_{\text{max}} = 25$, $N_{\text{leb}} = 4334$. This means that for each SAS-sphere of the molecule, we take $26^2$ spherical harmonics and 4334 discretization points. We treat $E_{\text{ex}}^s$ as the benchmark of the electrostatic contribution to the solvation energy.

In Figure 5, we illustrate how the electrostatic contribution $E^s$ to the solvation energy of formaldehyde varies with respect to the maximum degree of spherical harmonics $\ell_{\text{max}}$. The number of Lebedev quadrature points is set to $N_{\text{leb}} = 4334$. On the right side of Figure 5, the relative errors are computed according to the following formula

$$\text{Error} = \frac{|E^s - E_{\text{ex}}^s|}{|E_{\text{ex}}^s|}. \tag{4.7.2}$$

It is observed that $E^s$ converges exponentially and the ddLPB method provides improvable approximations when $\ell_{\text{max}}$ increases. Similarly, in Figure 5, we illustrate how the electrostatic contribution $E^s$ varies with respect to the number of Lebedev quadrature points $N_{\text{leb}}$, where we fix the maximum degree of spherical harmonics $\ell_{\text{max}} = 25$. Still, it is observed that the ddLPB method provides improvable approximations when $N_{\text{leb}}$ increases. However, the relative error does not decay monotonically as it is the case for the maximum degree of spherical harmonics.

### 4.7.2 Varying the Debye-Hückel screening constant

We now study the relationship between the electrostatic contribution to the solvation energy of formaldehyde and the Debye-Hückel screening constant. The discretization parameters are set to $\ell_{\text{max}} = 15$ and $N_{\text{leb}} = 974$. As given in the previous
Figure 6: The left figure plots the electrostatic contributions to the solvation energy of formaldehyde with respect to $N_{\text{leb}}$ and the right figure plots the relative errors. The blue line in the left figure represents the highly accurate electrostatic contribution.

In this subsection, we use the SAS-cavity and take the solvent radius $r_p = 1.5\text{Å}$. We plot in Figure 4 the electrostatic contribution to the solvation energy with various Debye-Hückel screening constants, respectively in a broad range $[2^{-12}, 2^{12}]$ and a narrow range $[0.1, 1]$. On the left side of Figure 4, the solid blue line gives the result for the COSMO (with conductor-like solvent), computed by the ddCOSMO method [25] with the same discretization parameters $\ell_{\text{max}}$ and $N_{\text{leb}}$. It is observed that when $\kappa$ tends to $\infty$, the result of the ddLPB method tends to the result of the ddCOSMO method. This means that when the ionic strength tends to $\infty$, the solvent becomes a perfect conductor, which makes sense. Furthermore, the dashed blue line gives the result for the PCM (the case of $\kappa = 0$, without ions), computed by the ddPCM method [108] with the same $\ell_{\text{max}}$ and $N_{\text{leb}}$. We observe that at the convergence of $\kappa = 0$, there is a slight difference (less than 0.1%) between the result of the ddLPB method and the result of the ddPCM method. The reason might be that the ddPCM method uses the integral equation formula [25] of PCM while the ddLPB method does not. As a consequence, these two methods are obtained by discretizing different formulations and then slightly different results are obtained.

**Remark 4.7.1.** The ddCOSMO method can be seen as a particular ddLPB method in the case of $\kappa = \infty$.

### 4.7.3 Varying the solvent radius

In Figure 8, we illustrate the relationship between the electrostatic contribution to the solvation energy of formaldehyde and the solvent radius $r_p$. In the numerical implementation, we use the parameters $\ell_{\text{max}} = 15$, $N_{\text{leb}} = 974$ and $\kappa = 0.1040$. We
Figure 7: The above two figures plot the electrostatic contributions to the solvation energy of formaldehyde with respect to \( \kappa \) respectively in a broad range (left) and a normal range (right). The solid blue line in the left figure represents the result of the COSMO with conductor-like solvent, computed by the ddCOSMO method. The dashed blue line in the left figure represents the result of the PCM without ions, computed by the ddPCM method.

observe a smooth change of the electrostatic contribution \( E^s \) with respect to \( r_p \) and \( E^s \) tends to 0 when \( r_p \) increases, which is as expected.

4.7.4 Different linear algebra algorithms

As mentioned in Section 4.6.2.2, different linear algebra algorithms can be used to solve the global linear system (4.6.31). We discuss here the LU factorization method, the GMRes method and the proposed iterative method (4.6.32) in combination with the GMRes method for the sub-problems. To study the efficiency, these methods are implemented in Matlab on a laptop with 2.5GHz quad-core Intel Core i7 processor. While the LU factorization and the GMRes method are built-in functions of Matlab, the iterative scheme (4.6.32) is implemented by us. For the GMRes method, we set the tolerance to \( \text{Tol} = 10^{-6} \), the number of restart iterations to 10, the maximum number of iterations to 50 and as a pre-conditioner, we use the diagonal of the matrix of the global linear system (4.6.31). At each iteration of the proposed iterative method (4.6.32), the GMRes method with the same settings of parameters is used to solve the sparse linear system involving \( A \) and \( B \). Further, the proposed iterative method is set to stop at the \( k \)-th iteration when \( \text{Error}_k < \text{Tol} \). Here,

\[
\text{Error}_k = \frac{\|X^k - X^{k-1}\|_2}{\|X^k\|_2}
\]
158 4.8. Conclusion

Figure 8: The above figure plots the electrostatic contributions to the solvation energy of formaldehyde with respect to the solvent radius $r_p$ (unit: Å).

denotes the relative error of the solution between previous and next iteration. We illustrate the running times of these methods for different molecules in Table 4.1, where we use the parameters $\ell_{\text{max}} = 15$, $N_{\text{leb}} = 974$, $\kappa = 0.1040$, $r_p = 1.5\text{Å}$. Each running time is computed as the average value of 10 testing results. It is observed that the proposed iterative method requires less computational time than the other two methods, when the size of the global linear system is large. In addition, the GMRes method fails to converge to the desired tolerance $10^{-6}$ after 50 outer iterations, with the chosen parameters in the case of caffeine.

4.7.5 Graphical illustration

Here, we give the graphical illustration of the reaction potential $\psi_r$ on the SAS of formaldehyde, benzene and caffeine in Figure 9 and 10. We use the following parameters: $\ell_{\text{max}} = 15$, $N_{\text{leb}} = 974$, $\kappa = 0.1040$, $r_p = 1.5\text{Å}$. The reaction potential of formaldehyde has mirror symmetry and the reaction potential of benzene has rotational symmetry, which match the geometrical structures of the molecules.

4.8 Conclusion

In this chapter, we have proposed a domain decomposition method for solving the LPB equation (the ddLPB method) describing the electrostatic potential in Poisson-Boltzmann solvation model. First, the original problem defined in $\mathbb{R}^3$ is transformed into two coupled equations defined in the bounded solute cavity, using the single-layer potential theory. Then, the Schwarz domain decomposition method was used to solve
Table 4.1: At the top, running times (unit: second) of the linear algebra algorithms for solving the linear systems of different molecules (size of linear system: $1024 \times 1024$, $2048 \times 2048$, $6144 \times 6144$, $12288 \times 12288$). At the bottom, the relative errors of the solutions obtained from the GMRes method and the proposed iterative method, with respect to the exact solution computed from the LU factorization method.

<table>
<thead>
<tr>
<th></th>
<th>Hydrogen-fluoride</th>
<th>Formaldehyde</th>
<th>Bezene</th>
<th>Caffeine</th>
</tr>
</thead>
<tbody>
<tr>
<td>LU factorization</td>
<td>0.0440</td>
<td>0.2570</td>
<td>6.0170</td>
<td>50.5490</td>
</tr>
<tr>
<td>GMRes method</td>
<td>0.2570</td>
<td>0.4670</td>
<td>11.1620</td>
<td>140.5670</td>
</tr>
<tr>
<td>Iterative method</td>
<td>0.1850</td>
<td>0.3720</td>
<td>4.6480</td>
<td>10.5430</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Hydrogen-fluoride</th>
<th>Formaldehyde</th>
<th>Bezene</th>
<th>Caffeine</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMRes method</td>
<td>$1.61 \times 10^{-6}$</td>
<td>$2.54 \times 10^{-6}$</td>
<td>$2.96 \times 10^{-6}$</td>
<td>0.25</td>
</tr>
<tr>
<td>Iterative method</td>
<td>$1.32 \times 10^{-6}$</td>
<td>$2.19 \times 10^{-6}$</td>
<td>$1.87 \times 10^{-6}$</td>
<td>$2.01 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

Figure 9: Reaction potential $\psi_t$ on the SAS of formaldehyde (left) and benzene (right).

Figure 10: Reaction potential $\psi_t$ on the SAS of caffeine.
these two problems by decomposing the solute cavity into balls, based on its particular geometrical structure, i.e., a union of overlapping balls. After that, we developed two direct single-domain solvers respectively for solving the Laplace equation and the HSP equation defined in the unit ball, which becomes easy to tackle by using the spherical harmonics in the angular direction. These two solvers provide the explicit formulas of the solutions, which allows us to obtain a global linear system after the discretization of the coupling conditions. A series of numerical results have been presented to show the performance of the ddLPB method. In the future work, we will continue to implement the ddLPB method in Fortran to further investigate its efficiency and the fast multipole method could be used to reduce the computational cost of assembling the global linear system, especially for the big molecules. In addition, we will devote ourselves to the computation of forces, which involve the derivatives of the solvation energy with respect to the nuclear coordinates.
Appendix A

Proof of Theorem 1.5.1

First, we provide a lemma about the geometric relationship between $P_0$ and $T = R(I)$ as follows.

**Lemma A.1.** There holds that $P_0 \setminus \bigcup_{x \in K} B_{r_p}(x) \subset T$, where $T = R(I)$ is the closed region composed of polyhedrons in the SAS-cavity.

**Proof.** We first recall an equivalence relationship for three sets $A$, $B$ and $C$:

$$A \setminus B \subset C \iff A \subset B \cup C \iff A \setminus C \subset B.$$ 

In consequence, to prove the lemma is equivalent to prove that $P_0 \setminus T \subset \bigcup_{x \in K} B_{r_p}(x)$. For simplicity, we only give the proof for the case of the eSAS in $\mathbb{R}^2$. There is no essential additional difficulty for the proof in $\mathbb{R}^3$. We define another polygon called the reduced polygon $T_0$ contained in $T$ (see Figure 3), which is obtained by removing from $T$ those vertices which are SAS intersection points, so that only the centers of the SAS-balls are left as vertices. In consequence, we have a simple relationship $T = (\bigcup_{x_m \in I} T_{x_m}) \cup T_0$, in which $T_{x_m}$ is the triangle with the three vertices $x_m$, $c_i$ and $c_j$. Each edge of $T_0$ is associated with an SAS intersection point.

With the above notations, three cases will be discussed as follows:

a) $P_0 \subset T_{x_m}$: Since $T_{x_m} \subset T$ implies that $P_0 \setminus T = \emptyset$, this is a trivial case.

b) $P_0 \subset T_{x_m} \cup T_0$: Since $T_{x_m} \cup T_0 \subset T$ implies that $P_0 \setminus T = \emptyset$, this is also a trivial case.

c) $\exists x_{m'} \neq x_m$ s.t. $P_0 \cap T_{x_{m'}} \neq \emptyset$: Figure 4 gives the schematics of the region (in grey) where $P_0$ can possibly appear outside $T_0$. If this region is included in $T_{x_{m'}}$, the proof is again trivial. If parts of this region lies outside $T_{x_{m'}}$, then this region is include by the probe $B_{r_p}(x_{m'})$ by symmetry. In consequence, all
Figure 1: A 2D schematics of the two different polygons in the eSAS-cavity of an artificial molecule for the proof of Lemma A.1. $T_{x_m}$ is the triangle with three vertices $(x_m, c_i, c_j)$ and the polygon in green is the reduced polygon $T_0$.

Figure 2: 2D schematics of two cases where $P_0 \cap T_{x_m'} \neq \emptyset$ with $x_m'$ being an SAS intersection point different from $x_m$. The green segment $[c_i', c_j']$ is an edge of $T_0$ and $a$ and $b$ are the two intersection points between the circle $\partial B_{r_p}(x_m')$ and the segment $[c_i', c_j']$. The grey region is the region where $P_0$ can possibly appear outside $T_0$ that may be included in the triangle $\triangle x_m'c_i'c_j'$ (left) or not (right).

such regions can be covered by the corresponding spherical probes, implying that $P_0 \setminus T \subset \bigcup_{x \in K} B_{r_p}(x)$.

With the above lemma, we can now prove Theorem 1.5.1. According to the definition of $K$, it accounts for all SAS intersection points that might intersect $B_{r_p}(x_m)$ and thus $P_0 \setminus \bigcup_{x \in K} B_{r_p}(x) = P_0 \setminus \bigcup_{x \in I} B_{r_p}(x)$. We first observe that

$$P_0 \setminus B_{r_p}(x) = P_0 \cap (B_{r_p}(x))^c = P_0 \cap \{p : \|p - x_m\| \leq \|p - x\|\}, \quad \forall x \neq x_m, \quad (A.1)$$
and therefore we obtain that

\[
P_0 \setminus \bigcup_{x \in K} B_{r_p}(x) = \left( P_0 \setminus \bigcup_{x \in K} B_{r_p}(x) \right) \cap T \\
= \bigcap_{x \in I} (P_0 \setminus B_{r_p}(x)) \cap T \\
= P_0 \cap T \cap \{ p : \|p - x_m\| \leq \|p - x\|, \forall x \neq x_m, x \in I \} \\
= P_-. 
\]  

(A.2)

In the above, the first equality results from Lemma A.1, the third one from equation (A.1) and the last one from (1.5.3).
Appendix B

Advancing-front algorithm for a spherical patch
Algorithm 1 Advancing-front algorithm for meshing a spherical patch

1: procedure Meshing Spherical Patch
2: Initialization of $A_e$, $N_{ae}$, $P$, $N_p$, $T$, $N_t$
3: while the front is not merged do
4:   for each active edge $e$ do
5:     Define the set of possibly suited front points $S_f$
6:     if $\alpha_1 < \varepsilon_\alpha$ then  \Comment{Angle criterion}
7:       Add $P_1$ to $S_f$
8:     else if $\alpha_2 < \varepsilon_\alpha$ then
9:       Add $P_2$ to $S_f$
10:   end if
11:   for each front point $P_f$ do  \Comment{Point-edge criterion}
12:     if $\text{dist}(P_f, e) < |e| + \varepsilon_e$ then
13:       Add $P_f$ to $S_f$ if $\triangle P_f P_1 P_2$ doesn’t “intersect” the front
14:     end if
15:   end for
16:   if $S_f$ is nonempty then
17:     $P_0 = \arg\min_{p \in S_f} \text{dist}(p, e)$
18:   else
19:     Create a new point $P_{test}$
20:     for each front edge $e_f$ do  \Comment{Point-edge criterion for $P_{test}$}
21:       if $\text{dist}(P_{test}, e_f) < |e_f| + \varepsilon_e$ then
22:         for each endpoint $P_f$ of $e_f$ do
23:           Add $P_f$ to $S_f$ if $\triangle P_f P_1 P_2$ doesn’t “intersect” the front
24:         end for
25:       end if
26:     end for
27:     for each front point $P_f$ do  \Comment{Distance criterion for $P_{test}$}
28:       if $\text{dist}(P_{test}, P_f) < \varepsilon_d$ then
29:         Add $P_f$ to $S_f$ if $\triangle P_f P_1 P_2$ doesn’t “intersect” the front
30:       end if
31:     end for
32:     if $S_f$ is nonempty then
33:       $P_0 = \arg\min_{p \in S_f} \text{dist}(p, e)$
34:     else
35:       $P_0 = P_{test}$
36:     end if
37:   end if
38:   Create a new triangle $\triangle P_0 P_1 P_2$ and update $A_e$, $N_{ae}$, $P$, $N_p$, $T$, $N_t$
39: end while
40: return $P$, $N_p$, $T$, $N_t$
41: end procedure
Appendix C

Appendices in Chapter 3

C.1 Well-posedness of (3.6.7)

In this appendix, we prove the well-posedness of the variational formulation (3.6.7).

We first show that the solution of Eq. (3.6.6), which is unique due to Lax-Milgram theorem, is a solution to the formulation (3.6.7). Indeed, assume that \( v \in H^1_0(B_1(0)) \) is the unique weak solution to Eq. (3.6.6), which implies that \( \forall w \in H^1_0(B_1(0)) \),

\[
\int_{B_1(0)} \varepsilon_1 \nabla v \cdot \nabla w = \int_{B_1(0)} f w. \tag{C.1}
\]

From the fact that \( \varepsilon_1|_{B_\delta(0)} = 1 \) and \( f|_{B_\delta(0)} = 0 \), we know that \( v \) is harmonic in \( B_\delta(0) \) and in consequence, \( \forall w \in H^1(B_\delta(0)) \),

\[
\int_{B_\delta(0)} \nabla v \cdot \nabla w + \int_{\partial B_\delta(0)} (\partial_n v) w = 0, \tag{C.2}
\]

where \( n \) points inwards \( B_\delta(0) \) at \( \partial B_\delta(0) \), i.e., outswards \( D \). Subtracting (C.2) from (C.1), we obtain: \( \forall w \in H^1_0(D) \),

\[
\int_D \varepsilon_1 \nabla v \cdot \nabla w - \int_{\partial B_\delta(0)} (T v) w = \int_D f w, \tag{C.3}
\]

where \( T v = \partial_n v \) is the Dirichlet to Neumann operator. This is just the variational formulation (3.6.7). Therefore, \( v|_D \in H^1_{0,\delta}(D) \) is a solution to (3.6.7), implying the existence of the solution to the formulation (3.6.7).

We now shed our attention to prove the coercivity of the bilinear form of the formulation (3.6.7). Let us equip the space \( H^1_{0,\delta}(D) \) with the norm \( \| \nabla \cdot \|_{L^2(D)} \), which is indeed a norm due to the Poincaré inequality. Recall that the bilinear form is
defined by
\[ a(v, w) = \int_D \varepsilon_1 \nabla v \cdot \nabla w - \int_{\partial B_\delta(0)} (Tv) w, \quad \forall v, w \in H^1_{0,\delta}(D), \]
and that \( \varepsilon_1 \in L^\infty(D) \) with \( 1 \leq \varepsilon_1 \leq \varepsilon_s \). For the coercivity of the bilinear form, it is clear that
\[ \int_D \varepsilon_1 \nabla v \cdot \nabla v \geq \| \nabla v \|_{L^2(D)}^2. \quad (C.4) \]
Furthermore, the following lemma can be derived.

**Lemma C.1.** There holds that
\[ -\int_{\partial B_\delta(0)} (Tv)v \geq 0. \quad (C.5) \]

*Proof.* The harmonic extension \( E_v \) of \( v \) in \( B_\delta(0) \) satisfies \( \Delta E_v = 0 \) and thus
\[ 0 = \int_{B_\delta(0)} \Delta E_v E_v = -\int_{B_\delta(0)} |\nabla E_v|^2 - \int_{\partial B_\delta(0)} (\nabla E_v \cdot n) E_v. \]
Since \( \nabla E_v \cdot n = T v \), we deduce that
\[ -\int_{\partial B_\delta(0)} (Tv)v = \int_{B_\delta(0)} |\nabla E_v|^2 \geq 0. \]

Combining the two inequalities (C.4) and (C.5), we then have \( a(v, v) \geq \| \nabla v \|_{L^2(D)}^2 \), which implies the coercivity of the bilinear form \( a \).

With the coercivity of the bilinear form \( a \), we can then prove that there exist a unique solution to the formulation (3.6.7) and this solution depends continuously on \( f \). Assuming that there are two solutions \( v_1 \) and \( v_2 \) to the formulation (3.6.7) in \( H^1_{0,\delta}(D) \), we then have
\[ a(v_1 - v_2, v_1 - v_2) = a(v_1, v_1 - v_2) - a(v_2, v_1 - v_2) = \int_D f(v_1 - v_2) - \int_D f(v_1 - v_2) = 0. \]
Further, according to the fact that
\[ a(v_1 - v_2, v_1 - v_2) \geq \| \nabla(v_1 - v_2) \|_{L^2(D)}^2, \]
we derive \( \| \nabla(v_1 - v_2) \|_{L^2(D)} = 0 \) and therefore \( v_1 = v_2 \). Since the existence of one solution has been presented above, we claim that there exist a unique solution to the formulation (3.6.7). Still denoting this unique solution by \( v \), we have
\[ \| \nabla v \|_{L^2(D)}^2 \leq a(v, v) = \int_D f v \leq \| f \|_{L^2(D)} \| v \|_{L^2(D)} \leq C_D \| \nabla f \|_{L^2(D)} \| \nabla v \|_{L^2(D)} \]
Chapter C: Appendices in Chapter 3

where \( C_D \) is a constant depending only on \( D \) due to the Poincaré inequality. This yields that
\[
\| \nabla v \|_{L^2(D)} \leq C \| \nabla f \|_{L^2(D)},
\]
that is, \( v \) depends continuously on \( f \). In summary, the formulation (3.6.7) is well-posed.

C.2 Computation of \( \frac{\partial}{\partial \theta} Y^m_\ell \) and \( \frac{\partial}{\partial \varphi} Y^m_\ell \)

We start with providing the definition of the real-valued spherical harmonics \( Y^m_\ell \):

\[
Y^m_\ell (\theta, \varphi) = \begin{cases} 
\sqrt{2} N^m_\ell P^{|m|}_\ell (\cos \theta) \sin |m| \varphi & \text{if } m < 0, \\
N^m_\ell P^m_\ell (\cos \theta) & \text{if } m = 0, \\
\sqrt{2} N^m_\ell P^m_\ell (\cos \theta) \cos m \varphi & \text{if } m > 0,
\end{cases}
\]

with
\[
N^m_\ell = \sqrt{\frac{(2\ell + 1)(\ell - |m|)!}{4\pi (\ell + |m|)!}},
\]
and where \( P^m_\ell \) represents the associated Legendre polynomials defined on \([-1, 1]\).

They satisfy the following property
\[
\sqrt{1 - x^2} \frac{d}{dx} P^m_\ell (x) = \frac{1}{2} \left[ (\ell + m)(\ell - m + 1)P^{m-1}_\ell (x) - P^{m+1}_\ell (x) \right], \quad x \in [-1, 1],
\]
which implies that
\[
\frac{d}{d\theta} P^m_\ell (\cos \theta) = -\frac{1}{2} \left[ (\ell + m)(\ell - m + 1)P^{m-1}_\ell (\cos \theta) - P^{m+1}_\ell (\cos \theta) \right].
\]

In consequence, we can then compute the derivative of \( Y^m_\ell \) with respect to \( \theta \) as below

\[
\frac{\partial}{\partial \theta} Y^m_\ell = \begin{cases} 
\sqrt{2} N^m_\ell \left( \frac{d}{d\theta} P^{|m|}_\ell (\cos \theta) \right) \sin |m| \varphi & \text{if } m < 0, \\
N^m_\ell \left( \frac{d}{d\theta} P^m_\ell (\cos \theta) \right) & \text{if } m = 0, \\
\sqrt{2} N^m_\ell \left( \frac{d}{d\theta} P^m_\ell (\cos \theta) \right) \cos m \varphi & \text{if } m > 0.
\end{cases}
\]

In addition, we can compute the derivative of \( Y^m_\ell \) with respect to \( \varphi \) as below

\[
\frac{\partial}{\partial \varphi} Y^m_\ell = \begin{cases} 
\sqrt{2} |m| N^m_\ell P^{|m|}_\ell (\cos \theta) \cos |m| \varphi & \text{if } m < 0, \\
0 & \text{if } m = 0, \\
-\sqrt{2} m N^m_\ell P^m_\ell (\cos \theta) \sin m \varphi & \text{if } m > 0.
\end{cases}
\]
C.3 Computation of \( f(x) \) in (3.6.6)

In Eq. (3.6.6), the right hand side \( f \) is defined as follows

\[
f(x) = \nabla \cdot \left[ (\varepsilon_1(x) - 1) \nabla \Phi(c_j + R_j x) + \nabla \cdot \varepsilon_1(x) \nabla \hat{u}_1(x) \right], \quad \forall x \in D,
\]

for some \( j = 1, \ldots, M \). Since \( \text{supp}(\Delta \Phi) = \text{supp}(\rho) \subset \Omega_{\text{sas}} \) and \( \text{supp}(\varepsilon - 1) = \Omega_{\text{sas}} \) (in fact, \( \varepsilon = 1 \) in \( \Omega_{\text{sas}} \)), it follows that \( (\varepsilon(x) - 1) \Delta \Phi(x) = 0 \), which implies that \( (\varepsilon_1(x) - 1) \Delta \Phi = 0 \). Further, combining with \( \Delta \hat{u}_1 = 0 \) yields

\[
f(x) = \nabla \varepsilon_1(x) \cdot \nabla \Phi(c_j + R_j x) + \nabla \varepsilon_1(x) \cdot \nabla \hat{u}_1(x), \quad \forall x \in D,
\]

where \( \varepsilon_1(x) = \varepsilon(c_j + R_j x) \). Further, letting \( p = c_j + R_j x \in \Omega_j \subset \Omega_0 \), we have

\[
\nabla \Phi(c_j + R_j x) = R_j \nabla_p \Phi(p) = R_j \sum_{i=1}^{M} \nabla_p q_i \frac{q_i}{|p - c_i|} = R_j \sum_{i=1}^{M} q_i \left( -\frac{p - c_i}{|p - c_i|^3} \right)
\]

and

\[
\nabla \hat{u}_1(x) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} [\phi_1]_{\ell}^m \nabla_x \left( r^\ell Y_{\ell m}(\theta, \varphi) \right)
= \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} [\phi_1]_{\ell}^m \left( \ell r^{\ell-1} Y_{\ell m}, r^{\ell-1} \frac{d}{d\theta} Y_{\ell m}, r^{\ell-1} \frac{d}{d\varphi} Y_{\ell m} \right) J,
\]

where \( [\phi_1]_{\ell}^m \) is the coefficient of \( Y_{\ell}^m \) in the spherical harmonic expansion of \( \phi_1 \), \( J \) is the Jacobian matrix from the spherical coordinates to the Cartesian coordinates, defined by

\[
J = \begin{pmatrix}
\sin \theta \cos \varphi & \sin \theta \sin \varphi & \cos \theta \\
\cos \theta \cos \varphi & \cos \theta \sin \varphi & -\sin \theta \\
-\sin \varphi & \cos \varphi & 0
\end{pmatrix}
\]

and further,

\[
\nabla \varepsilon_1(x) = \nabla_x \varepsilon(c_j + R_j x) = R_j \nabla_p \varepsilon(p).
\]

The remaining problem is to compute \( \nabla_p \varepsilon(p) \) where \( p \in \Omega_0 \) and this will be discussed as following.

Since \( p \in \Omega_0 \), we know that \( f_{\text{sas}}(p) < r_0 \). In the case when \( f_{\text{sas}}(p) \leq -r_p \), implying that \( p \) lies in the SES-cavity \( \Omega_{\text{sas}} \), we have \( \varepsilon(p) = 1 \) and consequently \( \nabla_p \varepsilon(p) = 0 \). In the other case when \( -r_p < f_{\text{sas}}(p) < r_0 \), implying that \( p \) lies in the dielectric boundary layer \( \Omega_0 \setminus \Omega_{\text{sas}} \), we can compute

\[
\nabla_p \varepsilon(p) = \nabla_p H \left( \frac{f_{\text{sas}}(p) + r_p}{r_p + r_0} \right) = \frac{H'(\xi)}{r_p + r_0} \nabla_p f_{\text{sas}}(p),
\]

where \( H' \) is the derivative of \( H \) with respect to \( \xi \).
where $\xi = \frac{f_{\text{sas}}(p) + r_p}{r_p + r_0} \in (0, 1)$. In the paper [93], a method of computing analytically one closest point $x_p$ on $\Gamma_{\text{sas}}$ to $p$ has been proposed, which is based on the data structure of each component of the SAS (an intersection point, a circular arc or a spherical patch). In consequence, we derive

$$f_{\text{sas}}(p) = \begin{cases} -|x_p - p| & \text{if } x_p \text{ is inside } \Gamma_{\text{sas}}, \\ |x_p - p| & \text{if } x_p \text{ is outside } \Gamma_{\text{sas}}, \end{cases} \quad (C.8)$$

and

$$\nabla_p f_{\text{sas}}(p) = \begin{cases} \frac{x_p - p}{|x_p - p|} & \text{if } x_p \text{ is inside } \Gamma_{\text{sas}}, \\ -\frac{x_p - p}{|x_p - p|} & \text{if } x_p \text{ is outside } \Gamma_{\text{sas}}, \end{cases} \quad (C.9)$$

which enables the computation of $\nabla_p \varepsilon(p)$ in Eq. (C.7). Combining all the above Eqs (C.1)–(C.3), we can therefore compute the function value $f(x)$. 
C.3. Computation of $f(x)$ in (3.5.1)
Appendix D

Computation of $C_1$, $C_2$, $F_0$

For each $\Gamma_i^e$, we first define a square matrix $P_{\chi_i^e}$ of dimension $(\ell_{\text{max}}+1)^2 \times (\ell_{\text{max}}+1)^2$ for each $\chi_i^e$, the $(\ell_0 m_0, \ell' m')$-th element of which is defined as

$$[P_{\chi_i^e}]_{\ell_0 m_0}^{\ell' m'} := \sum_{n=1}^{N_{\text{leb}}} w_{\text{leb}}^n \chi_i^e(x_i + r_is_n) Y_{\ell_0}^{m_0}(s_n) Y_{\ell'}^{m'}(s_n), \quad (D.1)$$

where $0 \leq \ell_0 \leq \ell_{\text{max}}$, $-\ell_0 \leq m_0 \leq \ell_0$, $0 \leq \ell' \leq \ell_{\text{max}}$, $-\ell' \leq m' \leq \ell'$. Based on Eq. (4.6.15), we can approximate $\chi_i^e \partial_n \psi_i$ (defined on $\Gamma_i$) by a linear combination of spherical harmonics $\{Y_{\ell_0}^{m_0}\}$ with $0 \leq \ell_0 \leq \ell_{\text{max}}$, $-\ell_0 \leq m_0 \leq \ell_0$ as follows

$$\chi_i^e \partial_n \psi_i (r_i, s) = \sum_{\ell_0=0}^{\ell_{\text{max}}} \sum_{m_0=-\ell_0}^{\ell_0} c_{r_i \ell_0 m_0} Y_{\ell_0}^{m_0}(s), \quad s \in S^2, \quad (D.2)$$

where the coefficient $c_{r_i \ell_0 m_0}$ is computed by the Lebedev quadrature rule as follows

$$c_{r_i \ell_0 m_0} = \sum_{\ell' = 0}^{\ell_{\text{max}}} \sum_{m' = -\ell'}^{\ell'} [P_{\chi_i^e}]_{\ell_0 m_0}^{\ell' m'} \frac{\ell'}{r_i} [X_i]_{\ell'}^{m'}. \quad (D.3)$$

Remark D.0.1. By writing $\chi_i^e \partial_n \psi_i$ as a linear combination of spherical harmonics, the single-layer potential $\tilde{S}_{\kappa, \Gamma_i}$ can act on it conveniently.

For an arbitrary Lebedev point $x_j + r_js_n = x_i + r_{ijn}s_{ijn} \in \Gamma_j^e$, we can then compute
as follows

\[
(\tilde{S}_{\kappa,\Gamma} \chi_i^e \partial_n \psi_i) (x_j + r_j s_n) = \sum_{\ell_0 = 0}^{\ell_{\max}} \sum_{m_0 = -\ell_0}^{\ell_0} c_{r, \ell_0 m_0} \left( \tilde{S}_{\kappa, \Gamma} Y_{\ell_0}^{m_0} \right) (x_j + r_j s_n) = \sum_{\ell_0 = 0}^{\ell_{\max}} \sum_{m_0 = -\ell_0}^{\ell_0} c_{r, \ell_0 m_0} \left( \kappa_{\ell_0} (\kappa \tau_i) - \kappa_{\ell_0} (\kappa \tau_i) \right) \frac{1}{k_{\ell_0} (\kappa \tau_i)} Y_{\ell_0}^{m_0} (s_{ijn}) \\
= \sum_{\ell' = 0}^{\ell_{\max}} \sum_{m' = -\ell'}^{\ell'} [Q]_{i\ell' m'} [r]_{i\ell' m'},
\]

where \( Q \) is a matrix of dimension \( M \ell_{\max}^2 \times MN_{\text{leb}} \), with the \((i \ell' m', jn)\)-th element defined by

\[
[Q]_{i\ell' m'} := \sum_{\ell' = 0}^{\ell_{\max}} \sum_{m' = -\ell'}^{\ell'} \left( \frac{\kappa_{\ell_0} (\kappa \tau_i)}{i_{\ell_0} (\kappa \tau_i)} - \frac{\kappa_{\ell_0} (\kappa \tau_i)}{k_{\ell_0} (\kappa \tau_i)} \right) \frac{1}{k_{\ell_0} (\kappa \tau_i)} Y_{\ell_0}^{m_0} (s_{ijn}).
\]

Therefore, we have the \((j \ell m, i \ell' m')\)-th element of \( C_1 \) as follows

\[
[C_1]_{j\ell m} = \frac{\varepsilon_1}{\varepsilon_2} \sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} \chi_j^e (x_j + r_j s_n) Y_{\ell}^m (s_n) [Q]_{i\ell' m'} \frac{1}{r_i}.
\]

Similarly, based on Eq. (1.6.10), we can approximate \( \chi_i^e \partial_n \psi_i \) (defined on \( \Gamma_i \)) by another linear combination of spherical harmonics \( \{ Y_{\ell_0}^{m_0} \} \) with \( 0 \leq \ell_0 \leq \ell_{\max}, -\ell_0 \leq m_0 \leq \ell_0 \) as follows

\[
\chi_i^e \partial_n \psi_i (r_i, s) = \sum_{\ell_0 = 0}^{\ell_{\max}} \sum_{m_0 = -\ell_0}^{\ell_0} c_{e, \ell_0 m_0} Y_{\ell_0}^{m_0} (s), \quad s \in S_2,
\]

where

\[
c_{e, \ell_0 m_0} = \sum_{\ell' = 0}^{\ell_{\max}} \sum_{m'' = -\ell'}^{\ell'} [P]_{\chi_i^e} Y_{\ell_0}^{m_0} \left( \frac{\kappa_{\ell_0} (\kappa \tau_i)}{i_{\ell_0} (\kappa \tau_i)} \right) [X_e]_{i\ell' m'}. \tag{D.8}
\]

For an arbitrary Lebedev point \( x_j + r_j s_n = x_i + r_{ijn} s_{ijn} \in \Gamma_j^e \), we can then compute

\[
(\tilde{S}_{\kappa, \Gamma} \chi_i^e \partial_n \psi_i) (x_j + r_j s_n) = \sum_{\ell_0 = 0}^{\ell_{\max}} \sum_{m_0 = -\ell_0}^{\ell_0} c_{e, \ell_0 m_0} \left( \tilde{S}_{\kappa, \Gamma} Y_{\ell_0}^{m_0} \right) (x_j + r_j s_n) = \sum_{\ell_0 = 0}^{\ell_{\max}} \sum_{m_0 = -\ell_0}^{\ell_0} c_{e, \ell_0 m_0} \left( \kappa_{\ell_0} (\kappa \tau_i) - \kappa_{\ell_0} (\kappa \tau_i) \right) \frac{1}{k_{\ell_0} (\kappa \tau_i)} Y_{\ell_0}^{m_0} (s_{ijn}) \]

\[
= \sum_{\ell' = 0}^{\ell_{\max}} \sum_{m' = -\ell'}^{\ell'} [Q]_{i\ell' m'} \left( \frac{\kappa_{\ell_0} (\kappa \tau_i)}{i_{\ell_0} (\kappa \tau_i)} - \frac{\kappa_{\ell_0} (\kappa \tau_i)}{k_{\ell_0} (\kappa \tau_i)} \right) \frac{1}{k_{\ell_0} (\kappa \tau_i)} Y_{\ell_0}^{m_0} (s_{ijn}). \tag{D.9}
\]
This yields that
\[
[C_2]_{j\ell m} = -\left( \sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} \chi_j^e(x_j + r_j s_n) Y_{\ell}^m(s_n) |Q|_{\ell\ell\ell m}^m - \frac{\kappa}{i'_{\ell_0}} (\kappa r_i) \right).
\] (D.10)

In addition, since \( \partial_{n} \psi_0 \) is known, for an arbitrary Lebedev point \( x_j + r_j s_n = x_i + r_{ij} n s_{ij} \in \Gamma_{i}^{e} \), we can compute the following column vector \( S \)
\[
[S]_{ij} = (\tilde{S}_{\ell_0, \Gamma}^{e} \chi^e_{\ell_0} (x_j + r_j s_n)) \]  
\[
= \sum_{\ell_0=0}^{\ell_{\text{max}}} \sum_{m=-\ell_0}^{\ell_0} c_{0, \ell m} \left( \tilde{S}_{\ell_0, \Gamma}^{e} Y_{\ell_0}^{m0} \right) (x_j + r_j s_n) 
\]  
\[
= \sum_{\ell_0=0}^{\ell_{\text{max}}} \sum_{m=-\ell_0}^{\ell_0} c_{0, \ell m} \left( \frac{\kappa}{i'_{\ell_0}} (\kappa r_i) - \frac{\kappa}{i'_{\ell_0}} (\kappa r_i) \right)^{-1} \frac{k_{\ell_0}}{k_{\ell_0}} (\kappa r_{ij} n) Y_{\ell_0}^{m0} (s_{ij}),
\] (D.11)

where
\[
c_{0, \ell m} = \sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} \chi_i^e (x_i + r_i s_n) \partial_{n} \psi_0 (x_i + r_i s_n) Y_{\ell_0}^{m0} (s_n).
\] (D.12)

This yields that
\[
[F_0]_{j\ell m} = -\frac{\varepsilon}{\varepsilon_2} \left( \sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} \chi_j^e (x_j + r_j s_n) Y_{\ell}^m (s_n) \sum_{i=1}^{M} [S]_{ij} \right).
\] (D.13)

\textbf{Remark D.0.2.} We first compute \( P_{\chi_i^e} \) a priori following (D.11) for each sphere \( \Gamma_{i}^{e} \), the complexity of which is \( O(MN_{\text{leb}}(\ell_{\text{max}} + 1)^4) \). Then, based on \( P_{\chi_i^e} \), we compute the matrix \( Q \) following (D.5) with a complexity \( O(M^2 N_{\text{leb}}(\ell_{\text{max}} + 1)^4) \). After that, we can compute \( C_1 \) and \( C_2 \) according to (D.6) and (D.11) with a complexity of \( O(M^2 N_{\text{leb}}(\ell_{\text{max}} + 1)^4) \). To compute the column vector \( F_0 \), we first compute \( S \) a priori with a complexity of \( O(M^2 N_{\text{leb}}(\ell_{\text{max}} + 1)^2) \). Then, based on \( S \), the complexity of computing \( F_0 \) according to (D.13) is reduced to \( O(M^2 N_{\text{leb}}(\ell_{\text{max}} + 1)^2) \). In summary, the total complexity of computing the global linear system (D.6.31) is \( O(M^2 N_{\text{leb}}(\ell_{\text{max}} + 1)^4) \).
Bibliography


[22] E. Cancès, B. Mennucci, and J. Tomasi. A new integral equation formalism for the polarizable continuum model: Theoretical background and applications to


Summary & Outlook

In this thesis, we have analyzed two topics of implicit solvation models: the molecular surface (the solute-solvent interface that defines the solute cavity) and the resolution of the underlying electrostatic problem. On the one hand, we gave for the first time a complete characterization of the solvent excluded surface (SES), with all singularities computed a priori. Based on this characterization, we then developed an efficient piecewise meshing algorithm for molecular surfaces, using the advancing-front method. On the other hand, we proposed two Schwarz domain decomposition methods for solving the electrostatic problems of the SES-based polarizable continuum model (PCM) and the Poisson-Boltzmann (PB) solvation model. In both methods, we first transformed the problem in \( \mathbb{R}^3 \) to two coupled problems defined in a bounded domain. Then, by decomposing the bounded domain into a union of balls, the global problem was recast as a group of coupled sub-problems, each restricted to a ball. To solve these sub-problems, single-domain solvers in the unit ball were hence developed, where the spherical harmonics were used as basis functions in the angular direction of the spherical coordinate system. A series of numerical tests have been presented to show the performance of these two methods.

Despite of the above achievements on implicit solvation models, there are many remaining questions and challenges that should be investigated in the future, for example,

- The complete characterization of SES might be useful in the application of molecular docking, where the complementarity between the molecular surfaces (for example, of the protein and the ligand) helps to find the optimal binding site.
• In the PB solvation model, the linearized PB equation is solved within the scheme of the Schwarz domain decomposition method, but the resolution of the nonlinear PB equation can potentially be intertwined with the iterations of the domain decomposition method.

• The implicit solvation model with homogenous but non-isotropic solvent can also be considered, for example, when the solvent is a liquid crystal. In this case, a similar Schwarz domain decomposition method can be developed for solving its associated electrostatic problem.

• The ddPCM-SES method and the ddLPB method for implicit solvation models have practical applications in quantum chemistry. They can either be used within the classical molecular dynamics simulation, thus coupling with a force-field, or within a fully quantum mechanical approach, thus coupling with a Hamiltonian. Therefore, the performance of these methods in the applications should be further studied.

We will continue to work in order to answer some of the above questions.
Abstract:
This thesis is devoted to study and improve the mathematical models and methods used in implicit solvation models in quantum chemistry. The manuscript is composed of two parts. In the first part where we analyze the solute-solvent interface, we give, for the first time, a complete characterization of the so-called “smooth” molecular surface, i.e., the solvent excluded surface (SES). Based on this characterization, we develop a piecewise meshing algorithm for different molecular surfaces, especially the SES, using the advancing-front triangulation. Further, it has been pointed out in the literature that the SES-cavity (the region enclosed by the SES) is a more accurate description of the solute cavity. In the second part, we therefore construct an SES-based polarizable continuum model (PCM), in which the dielectric permittivity parameter is continuous. The electrostatic problem of this model involves solving a Poisson equation defined in $\mathbb{R}^3$. We then develop a particular Schwarz domain decomposition method where only local equations restricted to balls need to be solved. Finally, the Poisson-Boltzmann solvation model, another implicit solvation model, is also investigated, which takes into account both the dielectric permittivity and the ionic strength of the solvent. A similar Schwarz domain decomposition method is proposed to solve the associated Poisson-Boltzmann equation by solving local equations restricted to balls as it is for the SES-based PCM.

Keywords: Solvent excluded surface, molecular visualization, implicit solvation model, polarizable continuum model, Poisson-Boltzmann equation, domain decomposition method

Méthodes mathématiques pour les modèles de solvabilité implicite en chimie quantique

Résumé:
Cette thèse est consacrée à étudier et à améliorer les modèles mathématiques et les méthodes utilisées pour les modèles de solvation implicite en chimie quantique. Ce manuscrit est composée de deux parties. Dans la première partie où nous analysons l’interface soluté-solvant, nous donnons, pour la première fois, une caractérisation complète de la surface moléculaire “lisse”, i.e., la surface exclue du solvant (SES). À partir de cette caractérisation, nous développons un algorithme de maillage par morceaux pour les surfaces moléculaires différentes, en particulier pour la SES, en utilisant la triangulation à front avançant. De plus, la cavité de la SES (la région entourée par la SES) est une description plus précise de la cavité de soluté. Dans la deuxième partie, nous construisions donc un modèles de continuum polarisable basé (PCM) sur la SES, dans lequel le paramètre de permittivité diélectrique est continu. Le problème électrostatique de ce modèle consiste à résoudre une équation de Poisson définie sur $\mathbb{R}^3$. Nous développons ensuite une méthode de Schwarz particulière, où seules les équations locales restreintes à des boules doivent être résolues. Enfin, nous étudions le modèle de solvation de Poisson-Boltzmann, un autre modèle de solvation implicite, qui tient compte à la fois de la permittivité diélectrique et de la force ionique du solvant. Une méthode de Schwarz similaire est proposée pour résoudre l’équation de Poisson-Boltzmann associée en résolvant des équations locales restreintes aux boules comme pour le PCM basé sur la SES.

Mot clés: Surface exclue du solvant, visualisation moléculaire, modèle de solvation implicite, modèle de continuum polarisable, équation de Poisson-Boltzmann, méthode de décomposition du domaine