



Mathematical methods for implicit solvation models in quantum chemistry

Chaoyu Quan

► To cite this version:

Chaoyu Quan. Mathematical methods for implicit solvation models in quantum chemistry. Mathematical Physics [math-ph]. Université Pierre et Marie Curie - Paris VI, 2017. English. NNT: 2017PA066587 . tel-01919793

HAL Id: tel-01919793

<https://tel.archives-ouvertes.fr/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 STAMM	Directeur de thèse
M.	Aihui ZHOU	Rapporteur



Chaoyu QUAN :

Sorbonne Universités, UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France.

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-

gardére, 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	9
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 representation	69
2.1 Introduction	70

2.2	Molecular surfaces	72
2.3	Construction of molecular surfaces	77
2.4	Molecular inner holes	80
2.5	Meshing	82
2.6	Conclusion	92
II	Domain Decomposition Method for Implicit Solvation Models	95
3	Domain Decomposition Method for the Polarizable Continuum Model based on the Solvent Excluded Surface	97
3.1	Introduction	98
3.2	Solute-solvent boundary	103
3.3	Dielectric permittivity function	105
3.4	Problem formulation and global strategy	107
3.5	Domain decomposition strategy	111
3.6	Single-domain solvers	112
3.7	Numerical results	119
3.8	Conclusion	130
4	Domain Decomposition Method for the Poisson-Boltzmann Solvation Model	131
4.1	Introduction	132
4.2	PB solvation model	137
4.3	Problem transformation	139
4.4	Strategy	142
4.5	Single-domain solvers	145

4.6 Global linear system	148
4.7 Numerical results	154
4.8 Conclusion	158
Appendices	160
A Proof of Theorem 1.5.1	161
B Advancing-front algorithm for a spherical patch	165
C Appendices in Chapter 3	167
C.1 Well-posedness of (3.6.7)	167
C.2 Computation of $\frac{\partial}{\partial\theta}Y_\ell^m$ and $\frac{\partial}{\partial\varphi}Y_\ell^m$	169
C.3 Computation of $f(\mathbf{x})$ in (3.6.6)	170
D Computation of C_1, C_2, F_0	173
Bibliography	186
Summary & Perspectives	187

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 Ψ , is described by the Schrödinger equation in its time-dependent

$$i \frac{\partial}{\partial t} \Psi(t) = \mathcal{H} \Psi(t), \quad (1.1)$$

or time-independent form

$$\mathcal{H} \Psi = \mathcal{E} \Psi, \quad (1.2)$$

where \mathcal{H} denotes the Hamiltonian of the molecular system under consideration and the constant \mathcal{E} is the energy of the stationary state Ψ . 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]:

$$\mathcal{H} = \mathcal{H}^M + \mathcal{H}^{MS}, \quad (1.3)$$

where \mathcal{H}^M is the Hamiltonian of the solute molecule M and \mathcal{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 [84, 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

$$G = G_{\text{cav}} + G_{\text{el}} + G_{\text{dis}} + G_{\text{rep}} + G_{\text{tm}}, \quad (1.4)$$

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 ψ of an implicit solvation model is characterized as follows

$$-\nabla \cdot \varepsilon(\mathbf{x}) \nabla \psi(\mathbf{x}) = 4\pi\rho(\mathbf{x}), \quad \text{in } \mathbb{R}^3, \quad (1.5)$$

where $\psi(\mathbf{x}) \sim \frac{1}{|\mathbf{x}|}$ as $|\mathbf{x}| \rightarrow \infty$. Here, $\varepsilon(\mathbf{x})$ represents the space-dependent dielectric constant and $\rho(\mathbf{x})$ 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

$$G_{\text{el}} = E^s = \frac{1}{2} \int_{\mathbb{R}^3} \rho(\mathbf{x}) (\psi(\mathbf{x}) - \psi_0(\mathbf{x})) d\mathbf{x}, \quad (1.6)$$

where

$$\psi_0(\mathbf{x}) = \int_{\mathbb{R}^3} \frac{\rho(\mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|} d\mathbf{x}' \quad (1.7)$$

is the electrostatic potential generated by ρ in vacuo.

For the sake of simplicity, it is usually assumed that the solute's charge distribution function ρ_M (part of the whole charge distribution ρ) is supported in the solute cavity Ω and is presented by the sum of M point charges in the form of

$$\rho_M(\mathbf{x}) = \sum_{i=1}^M q_i \delta(\mathbf{x} - \mathbf{x}_i), \quad (1.8)$$

where M is the number of solute atoms, q_i represents the charge carried on the i th atom with center \mathbf{x}_i , δ is the Dirac delta function. As a consequence, when the solvent does not contain any ion, we have $\rho = \rho_M$ and therefore, ψ_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 Ω , 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-

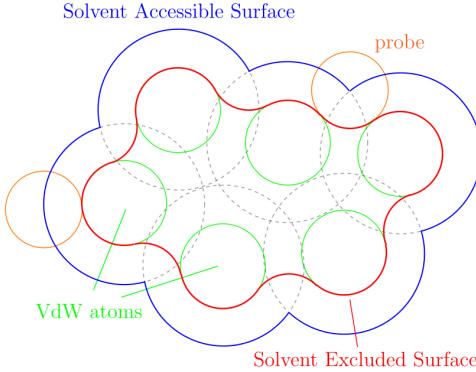


Figure 1: Schematic diagram of the VdW-surface (green), the SAS (blue) and the SES (red).

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

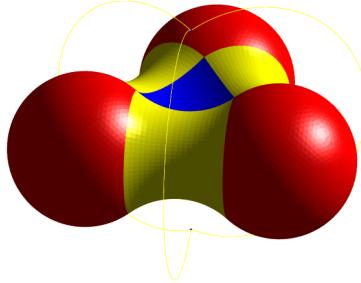


Figure 2: 3D schematic of the SES illustrating the convex spherical patches (red), the toroidal patches (yellow) and the concave spherical patches (blue).

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 1 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

number of contributions in literature [30, 102, 73, 52].

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 \mathbf{c}_j and a VdW radius r_j . The solvent probe radius is denoted by r_p . Thus, we can denote by f_{sas} the signed distance function to the SAS (negative inside the SAS and positive outside the SAS) as follows

$$f_{\text{sas}}(\mathbf{p}) = \begin{cases} -\|\mathbf{p} - \mathbf{x}_{\text{sas}}^{\mathbf{p}}\| & \text{if } \mathbf{p} \text{ lies inside the SAS,} \\ \|\mathbf{p} - \mathbf{x}_{\text{sas}}^{\mathbf{p}}\| & \text{if } \mathbf{p} \text{ lies outside the SAS,} \end{cases} \quad (2.1)$$

where $\mathbf{x}_{\text{sas}}^{\mathbf{p}}$ denotes a closest point on the SAS to \mathbf{p} . Note that there might exist more than one closest point on the SAS and in this case $\mathbf{x}_{\text{sas}}^{\mathbf{p}}$ is chosen as one of them. The implicit functions of the SAS (denoted by Γ_{sas}) and the SES (denoted by Γ_{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 $\mathbf{x}_{\text{sas}}^{\mathbf{p}}$ is not easy to compute because the shape of the molecule can be as complicated as possible.

In Chapter 1 of Part I, an efficient method for computing analytically the function value of f_{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-

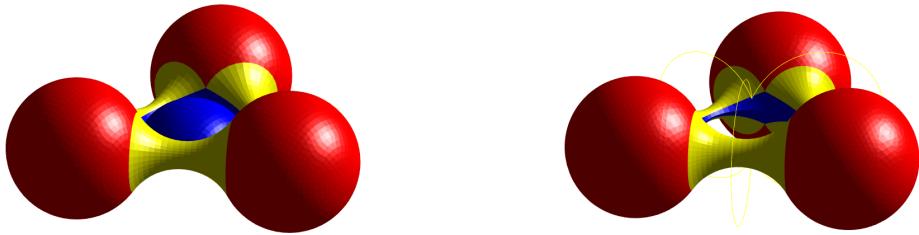


Figure 3: The SES of three balls having the self-intersection problem (left, triangulation provided by the MSMS-algorithm) and the realistic SES (right, triangulation provided by our algorithm).

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.

Benefiting 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 ρ_M supported in the solute cavity. In the classical PCM with a solute cavity Ω , the space-dependent dielectric permittivity $\varepsilon(\mathbf{x})$ is defined as (see the right of Figure 4)

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

where ε_s is the (bulk) solvent dielectric constant and $\varepsilon(\mathbf{x})$ 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 ψ can be rewritten as

$$\begin{cases} -\Delta\psi = 4\pi\rho_M, & \text{in } \Omega, \\ -\Delta\psi = 0, & \text{in } \Omega^c, \\ [\psi] = 0, & \text{on } \Gamma, \\ [\partial_{\mathbf{n}}(\varepsilon\psi)] = 0, & \text{on } \Gamma, \end{cases} \quad (2.3)$$

where \mathbf{n} denotes the unit normal vector pointing outwards with respect to Ω , $[\psi]$ and $[\partial_{\mathbf{n}}(\varepsilon\psi)]$ respectively denote the jump (inside Ω minus outside) of ψ and the jump of the normal derivative of $\varepsilon\psi$ on Γ . An integral equation formalism (IEF) [22, 81, 26] of this equation was proposed by E. Cancès, B. Mennucci and J. Tomasi, which has been the default PCM formulation in Gaussian [44]. 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.

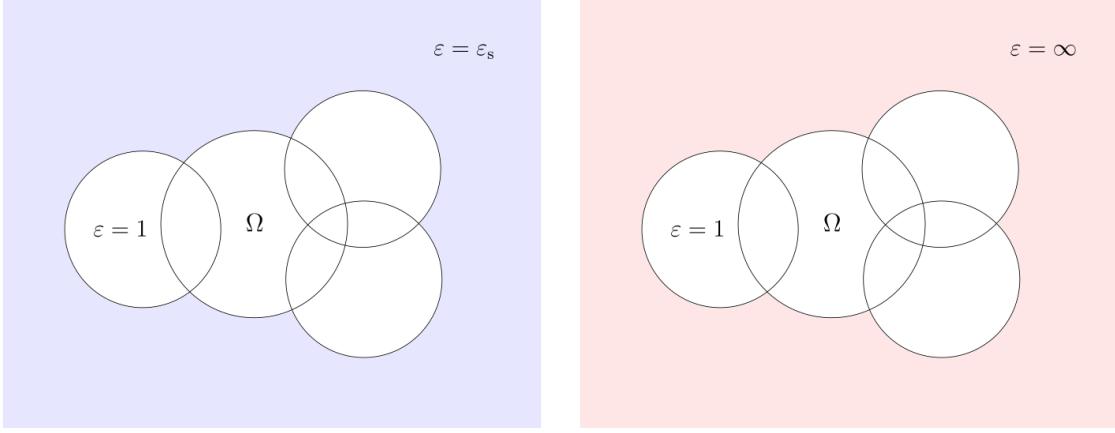


Figure 4: 2D schematic diagrams of the classical PCM (left) and the COSMO (right).

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. (2.2). That means that the dielectric permittivity function $\varepsilon(\mathbf{x})$ is taken as

$$\varepsilon(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega, \\ \infty & \mathbf{x} \in \Omega^c. \end{cases} \quad (2.4)$$

This reduced model is usually employed to approximate the PCM when the solvent dielectric permittivity ε_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 (1.5) of the electrostatic potential is simply a Poisson equation defined on a bounded domain with a homogeneous Dirichlet boundary condition

$$\begin{cases} -\Delta\psi(\mathbf{x}) = 4\pi\rho_M(\mathbf{x}), & \text{in } \Omega, \\ \psi(\mathbf{x}) = 0, & \text{on } \Gamma. \end{cases} \quad (2.5)$$

Here, the electrostatic potential ψ vanishes on the solute-solvent boundary Γ 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 Ω , which is consequently easier. As assumed, the charge distribution ρ_M of the solute molecule is already known. For calculating the solvent effects with a finite dielectric constant ε_s , the electrostatic contribution to the solvation energy E^s is usually approximated by

$$E^s = f(\varepsilon_s)E_\infty^s,$$

where E_∞^s is the electrostatic contribution to the solvation energy computed from the COSMO by solving the PDE (2.5) 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

$$c_i = c_i^\infty e^{\frac{-W_i}{k_B T}}, \quad (2.6)$$

where c_i^∞ 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])

$$-\nabla \cdot [\varepsilon(\mathbf{x}) \nabla \psi(\mathbf{x})] = 4\pi \rho_M(\mathbf{x}) + \sum_i z_i e c_i^\infty e^{\frac{-z_i e \psi(\mathbf{x})}{k_B T}} \chi_{\Omega^c}(\mathbf{x}), \quad (2.7)$$

where $z_i e$ is the charge of the i -th type of ion, e is the elementary charge and χ_{Ω^c} is the characteristic function of the solvent region Ω^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 ψ satisfies the low potential condition, i.e., $|\frac{e\psi}{k_B T}| \ll 1$, the above Poisson-Boltzmann equation can be linearized to (see [86] for this form)

$$-\nabla \cdot [\varepsilon(\mathbf{x}) \nabla \psi(\mathbf{x})] + \bar{\kappa}(\mathbf{x})^2 \psi(\mathbf{x}) = 4\pi \rho_M(\mathbf{x}), \quad (2.8)$$

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

Remark 2.1. If the solvent contains more than two types of ions, the associated

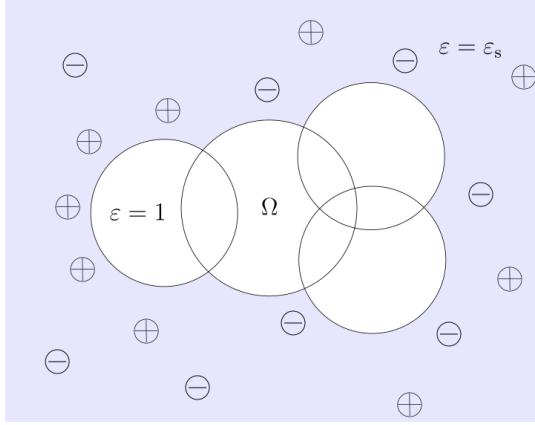


Figure 5: 2D schematic diagrams of the Poisson-Boltzmann solvation model.

Poisson-Boltzmann equation can still be linearized to the same form (2.8).

The dielectric permittivity function is defined in the classical way

$$\varepsilon(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega, \\ \varepsilon_s & \mathbf{x} \in \Omega^c, \end{cases} \quad (2.9)$$

where ε_s is the solvent dielectric constant as previous. The modified Debye-Hückel parameter is taken as

$$\bar{\kappa}(\mathbf{x}) = \begin{cases} 0 & \text{in } \Omega, \\ \sqrt{\varepsilon_s} \kappa & \text{in } \Omega^c, \end{cases} \quad (2.10)$$

where κ 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

$$\kappa^2 = \frac{8\pi e^2 N_A I}{1000 \varepsilon_s k_B T}, \quad (2.11)$$

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 Ω 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 Ω be the VdW-cavity or the

SAS-cavity, meaning that Ω is a union of overlapping balls in the following form

$$\Omega = \bigcup_{j=1}^M \Omega_j, \quad \Omega_j = B_{r_j}(\mathbf{x}_j), \quad (3.1)$$

where each Ω_j denotes the j -th atomic VdW-ball (or SAS-ball) with center \mathbf{x}_j and radius r_j .

We then homogenize the COSMO equation (2.5) by defining

$$\psi_0 = \sum_{i=1}^M \frac{q_i}{|\mathbf{x} - \mathbf{x}_i|}, \quad (3.2)$$

which satisfies $-\Delta\psi_0 = 4\pi\rho_M(\mathbf{x})$ in \mathbb{R}^3 . Here, ρ_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

$$\begin{cases} -\Delta\psi_r = 0, & \text{in } \Omega, \\ \psi_r = -\psi_0, & \text{on } \Gamma. \end{cases} \quad (3.3)$$

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 Ω , this Laplace equation can be recast as the following group of coupled sub-equations, each restricted to Ω_j :

$$\begin{cases} -\Delta\psi_r|_{\Omega_j} = 0 & \text{in } \Omega_j, \\ \psi_r|_{\Gamma_j} = \phi_{r,j} & \text{on } \Gamma_j, \end{cases} \quad (3.4)$$

where $\Gamma_j = \partial\Omega_j$ and

$$\phi_{r,j} = \begin{cases} \bar{\psi}_r & \text{on } \Gamma_j^i, \\ -\psi_0 & \text{on } \Gamma_j^e. \end{cases} \quad (3.5)$$

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

$$\bar{\psi}_r(\mathbf{x}) = \frac{1}{|\mathcal{N}(j, \mathbf{x})|} \sum_{i \in \mathcal{N}(j, \mathbf{x})} \psi_r|_{\Omega_i}(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma_j^i, \quad (3.6)$$

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

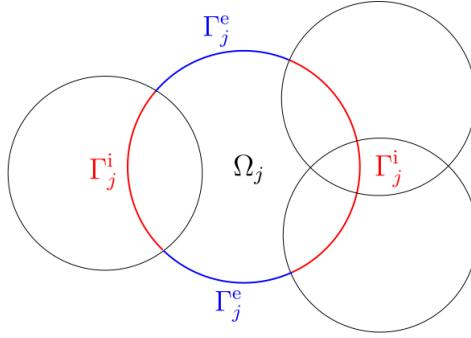


Figure 6: 2D schematic diagram of Γ_j^i (red) and Γ_j^e (blue) associated with Ω_j .

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 $\psi_r|_{\Gamma_j^i}$ 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 (1.5) of electrostatic potential and the definition (2.2) of $\varepsilon(\mathbf{x})$ was proposed in [108] (called ddPCM, see also website [70]), which is based on the integral equation formalism (IEF) of PCM [26].

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

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

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

$$\mathcal{R}_\varepsilon \mathcal{S}\sigma = -\mathcal{R}_\infty\psi_0, \quad \text{on } \Gamma, \quad (3.7)$$

where \mathcal{S} is the single-layer operator on Γ defined by

$$\mathcal{S}\sigma(\mathbf{x}) := \int_\Gamma \frac{\sigma(\mathbf{y})}{4\pi|\mathbf{x} - \mathbf{y}|}, \quad \forall \mathbf{x} \in \Gamma, \quad \sigma \in H^{-\frac{1}{2}}(\Gamma), \quad (3.8)$$

the two operators \mathcal{R}_ε and \mathcal{R}_∞ are defined using the double-layer operator \mathcal{D} as follows

$$\mathcal{R}_\varepsilon := \frac{\varepsilon + 1}{2(\varepsilon - 1)} - \mathcal{D}, \quad \mathcal{R}_\infty := \frac{1}{2} - \mathcal{D}, \quad (3.9)$$

and

$$\mathcal{D}\sigma(\mathbf{x}) := \int_{\Gamma} \left(\nabla_{\mathbf{y}} \frac{1}{4\pi|\mathbf{x} - \mathbf{y}|} \cdot \mathbf{n}_{\mathbf{y}} \right) \sigma(\mathbf{y}), \quad \forall \mathbf{x} \in \Gamma, \quad \sigma \in H^{-\frac{1}{2}}(\Gamma). \quad (3.10)$$

Here, $\mathbf{n}_{\mathbf{y}}$ denotes the unit normal vector at \mathbf{y} on Γ . Then, to solve the IEF-PCM equation (3.7), two steps are taken, including solving the following equation of an auxiliary function Φ

$$\mathcal{R}_\varepsilon \Phi = -\mathcal{R}_\infty \psi_0, \quad (3.11)$$

and then the following equation of σ

$$\mathcal{S}\sigma = \Phi. \quad (3.12)$$

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 Ω 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 ρ_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 ρ_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 Ω_{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 $\varepsilon(\mathbf{x})$ 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 $\varepsilon_s(\mathbf{x})$ have been proposed [47, 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 Ω_∞ far away from the solute molecule is equal to the bulk solvent dielectric permittivity ε_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 Ω_∞ , 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 $\varepsilon(\mathbf{x})$ of the following form (see the left of Figure 7)

$$\varepsilon(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega_{\text{ses}}, \\ \varepsilon_s(\mathbf{x}) & \mathbf{x} \in \mathcal{L}, \\ \varepsilon_s & \mathbf{x} \in \Omega_\infty. \end{cases} \quad (3.13)$$

See Figure 8 for an example of $\varepsilon(\mathbf{x})$, where $\varepsilon(\mathbf{x})$ is a distance-dependent function. The “distance” here represents the signed distance to the SAS, denoted by f_{sas} , which also characterizes the SES-cavity as follows

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

where r_p represents the probe radius. In addition, the bulk solvent region Ω_∞ is characterized as

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

where r_0 is a positive constant. We then deduce that ψ of Eq. (1.5) is harmonic in Ω_∞ , i.e. $-\Delta\psi = 0$ in Ω_∞ .

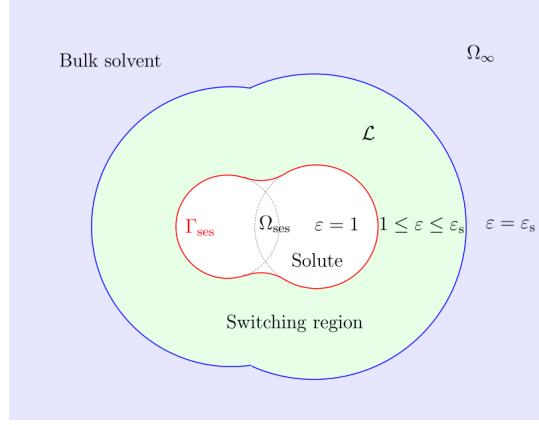


Figure 7: 2D schematic diagrams of the SES-based PCM.

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

3.3.2 Doman decomposition method for the SES-based PCM

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

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

where the bounded domain Ω_0 and the unbounded domain Ω_∞ are complementary in \mathbb{R}^3 , $[u]$ and $[\partial_{\mathbf{n}} u]$ 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 9. The unbounded problem defined in \mathbb{R}^3 is first transformed into two coupled problems both defined on Ω_0

$$\begin{cases} -\nabla \cdot \varepsilon \nabla u = f, & \text{in } \Omega_0, \\ u = g, & \text{on } \Gamma_0, \end{cases} \quad \text{and} \quad \begin{cases} -\Delta u_\infty = 0, & \text{in } \Omega_0, \\ u_\infty = g, & \text{on } \Gamma_0, \end{cases} \quad (3.14)$$

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

$$g = \frac{1}{4\pi} \mathcal{S}_{\Gamma_0} (\partial_{\mathbf{n}} u_\infty - \partial_{\mathbf{n}} u), \quad \text{on } \Gamma_0.$$

Here, \mathcal{S}_{Γ_0} denotes the single-layer potential of the interface Γ_0 , associated with the

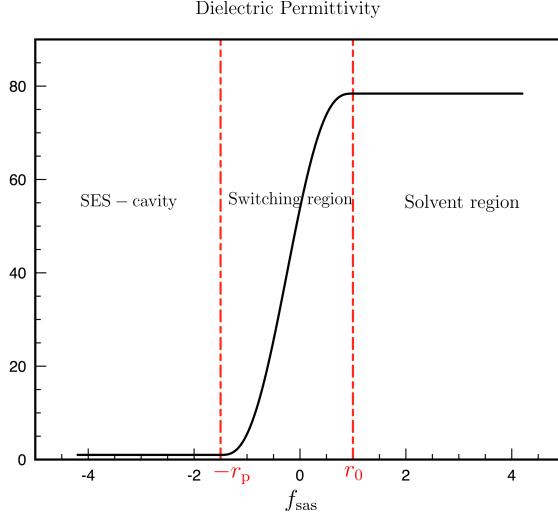


Figure 8: Schematic diagram of the dielectric permittivity function $\varepsilon(\mathbf{x})$ with respect to f_{sas} . The dielectric boundary layer \mathcal{L} (switching region) is bounded by two dashed lines (red), i.e., the region where $-r_p \leq f_{\text{sas}} \leq r_0$.

Laplace equation,

$$\mathcal{S}_{\Gamma_0} \sigma(\mathbf{x}) := \int_{\Gamma_0} \frac{\sigma(\mathbf{y})}{4\pi|\mathbf{x} - \mathbf{y}|}, \quad \forall \mathbf{x} \in \Gamma_0, \sigma \in H^{-\frac{1}{2}}(\Gamma_0). \quad (3.15)$$

The bounded domain Ω_0 is taken as a union of balls, inspired by the geometrical structure of the solute molecule. Since Ω_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 $\varepsilon(\mathbf{x})$ 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

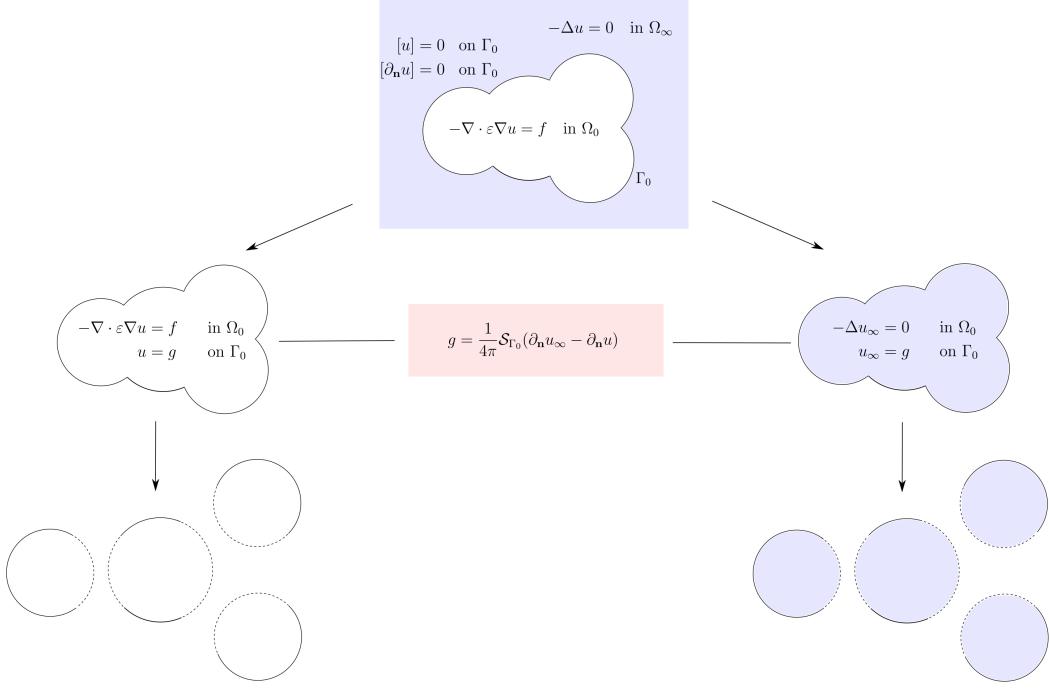


Figure 9: Schematic diagram of the domain decomposition method for the SES-based PCM (ddPCM-SES).

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 Ω and a homogeneous screened Poisson (HSP) equation defined in the unbounded solvent region Ω^c as follows

$$\begin{cases} -\Delta \psi(\mathbf{x}) = 4\pi \rho_M(\mathbf{x}) & \text{in } \Omega, \\ -\Delta \psi(\mathbf{x}) + \kappa^2 \psi(\mathbf{x}) = 0 & \text{in } \Omega^c, \end{cases} \quad (3.16)$$

combined with two classical jump-conditions

$$\begin{cases} [\psi] = 0 & \text{on } \Gamma, \\ [\partial_n(\varepsilon \psi)] = 0 & \text{on } \Gamma. \end{cases} \quad (3.17)$$

Here, κ is the Debye-Hückel screening constant. We remind that the solute cavity Ω 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 ψ_0 defined by (3.2). The reaction potential $\psi_r := \psi - \psi_0$ satisfies the following Laplace equation

$$\begin{cases} -\Delta\psi_r = 0, & \text{in } \Omega, \\ \psi_0 + \psi_r = g, & \text{on } \Gamma. \end{cases} \quad (3.18)$$

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

$$\begin{cases} -\Delta\psi_e(\mathbf{x}) + \kappa^2\psi_e(\mathbf{x}) = 0, & \text{in } \Omega, \\ \psi_e = g, & \text{on } \Gamma, \end{cases} \quad (3.19)$$

where ψ_e satisfies the same HSP equation as $\psi|_{\Omega^c}$ with the same Dirichlet boundary condition on Γ , but defined in Ω . The coupling condition between ψ_r and ψ_e arises through an auxiliary variable g defined as

$$g = \mathcal{S}_\kappa \left(\partial_{\mathbf{n}}\psi_e - \frac{1}{\varepsilon_s} \partial_{\mathbf{n}} (\psi_0 + \psi_r) \right), \quad \text{on } \Gamma, \quad (3.20)$$

where \mathcal{S}_κ is another single-layer potential defined as

$$\mathcal{S}_\kappa \sigma(\mathbf{x}) := \int_{\Gamma} \frac{e^{-\kappa|\mathbf{x}-\mathbf{y}|}\sigma(\mathbf{y})}{4\pi|\mathbf{x}-\mathbf{y}|}, \quad \forall \mathbf{x} \in \Gamma, \quad \sigma \in H^{-\frac{1}{2}}(\Gamma). \quad (3.21)$$

At this moment, the initial problem has been transformed equivalently into two coupled problems (3.18)–(3.19) both defined on the bounded domain Ω with a coupling condition (3.20). Taking advantage of the fact that Ω 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

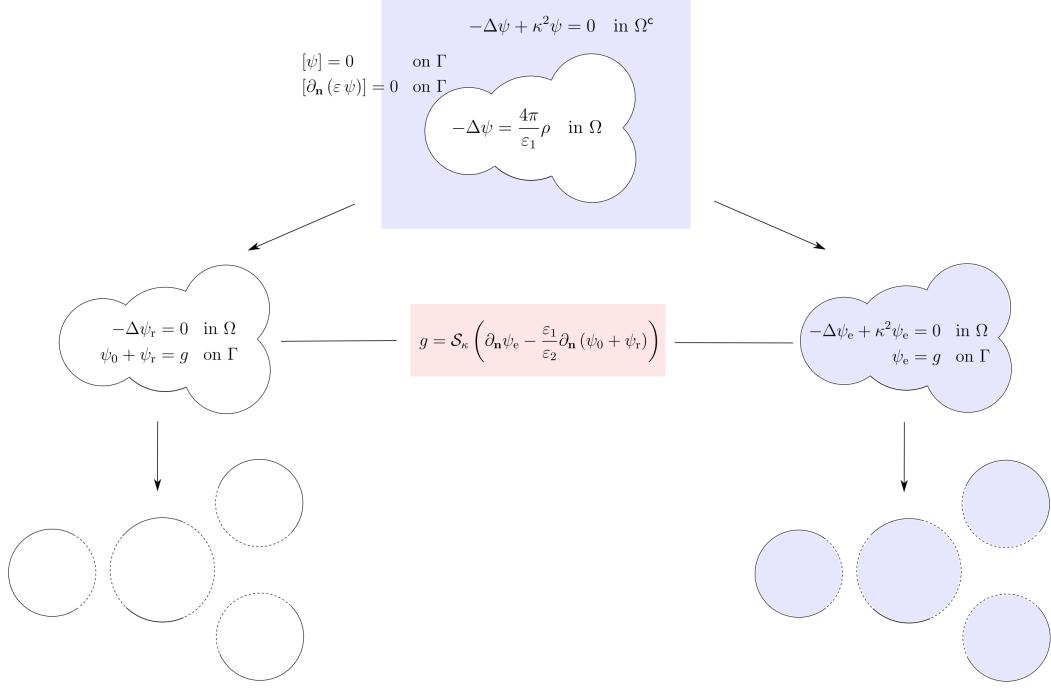


Figure 10: Schematic diagram of the domain decomposition method for the PB solvation model (ddLPB). Here, $\varepsilon_1 = 1$ and $\varepsilon_2 = \varepsilon_s$.

definitions of the solute-solvent boundary as in Chapter 3 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.

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
1.4	Solvent accessible surface	39
1.4.1	Mathematical definitions	39
1.4.2	Equivalence statements	41
1.4.3	New Voronoi-type diagram	47
1.4.4	SAS-area and SAS-volume	49
1.5	Solvent excluded surface	50
1.5.1	Mathematical definitions	51
1.5.2	SES-singularities	53
1.5.3	SES-area and SES-volume	56
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

1.7.1	A system of two atoms	63
1.7.2	Molecular areas and volumes	64
1.7.3	Comparison with the MSMS-algorithm	64
1.7.4	Computational cost	67
1.8	Conclusion	67

This chapter has been published as a journal paper [93]. As mentioned in the introduction of this thesis, we present in this chapter a complete characterization of the Solvent Excluded Surface (SES) for molecular systems, including a complete characterization of singularities of the surface. The theory is based on an implicit representation of the SES, which, in turn, is based on the signed distance function to the Solvent Accessible Surface (SAS). All proofs are constructive so that the theory allows for efficient algorithms in order to compute the area of the SES and the volume of the SES-cavity, or to visualize the surface. Further, we propose to refine the notion of SAS and SES in order to take inner holes in a solute molecule into account or not.

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 [109]. 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 [25, 72] 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 $\mathcal{O} \subset \mathbb{R}^n$ is called an *implicit object* if there exists a real-valued function $f : U \rightarrow \mathbb{R}^k$ with $\mathcal{O} \subset U \subset \mathbb{R}^n$, and a subset $V \subset \mathbb{R}^k$, such that $\mathcal{O} = f^{-1}(V)$. That is,

$$\mathcal{O} = \{p \in U : f(p) \in V\}.$$

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 $\mathcal{O} = f^{-1}(0)$, 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,

$$f_S(p) = \begin{cases} -\inf_{x \in S} \|p - x\| & \text{if } p \text{ lies inside } S, \\ \inf_{x \in S} \|p - x\| & \text{if } p \text{ lies outside } S. \end{cases} \quad (1.2.1)$$

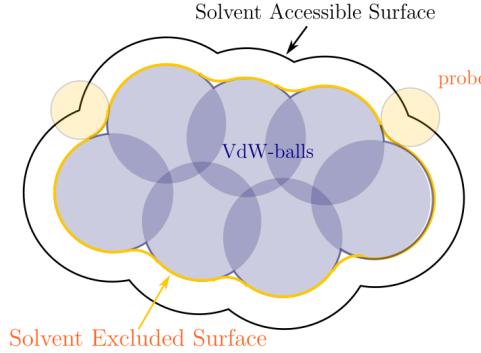


Figure 1: This is a 2-dimension (2D) schematics of the SAS and the SES, both defined by a spherical probe in orange rolling over the molecule atoms in dark blue.

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 [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 VdW-ball. The van der Waals surface can consequently be represented as an implicit surface $f_{vdw}^{-1}(0)$ with the following implicit function:

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

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. 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:

$$\tilde{f}_{\text{sas}}(p) = f_{\text{vdw}}(p) - r_p = \min_{i=1,\dots,M} \{\|p - c_i\|_2 - r_i - r_p\}, \quad \forall p \in \mathbb{R}^3. \quad (1.3.2)$$

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:

$$f_{\text{ses}}(p) = f_{\text{sas}}(p) + r_p, \quad (1.3.3)$$

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_{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:

$$\tilde{f}_{\text{ses}}(p) = \min_{1 \leq i < j < k \leq M} f_{ijk}(p), \quad \forall p \in \mathbb{R}^3, \quad (1.3.4)$$

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:

$$\tilde{f}_{\text{ses}}(p) = \min_{1 \leq i < j \leq 3} f_{i,j}(p), \quad \forall p \in \mathbb{R}^2, \quad (1.3.5)$$

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,

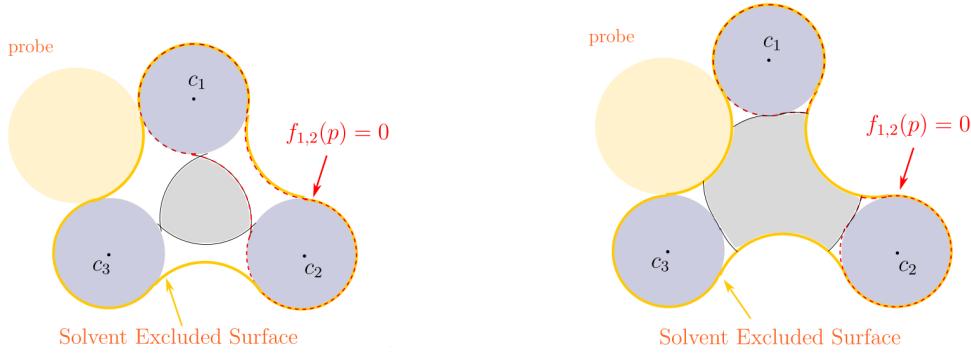


Figure 2: The above figures illustrate two SESs of two artificial molecules respectively containing three atoms. In each of them, $f_{1,2}$ denotes the signed distance function to the SES of the 1st and the 2nd atoms which is depicted with dashed red curves ($f_{2,3}$ and $f_{1,3}$ are similar).

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

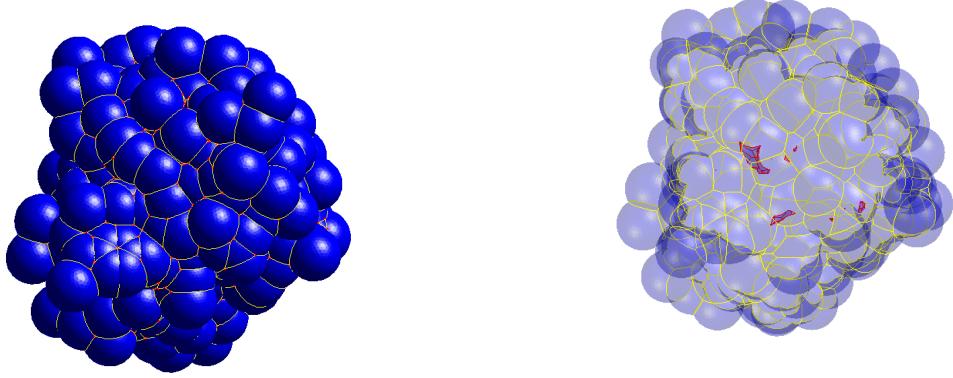


Figure 3: The left figure shows the components of the eSAS of the protein 1B17 (the probe radius $r_p = 1.2\text{\AA}$): open spherical patches in blue, open circular arcs in yellow and intersection points in red. The right figure shows the exterior surface (transparent) and the interior surfaces of 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_{i_0} 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_{sas}^p . Thus, the signed distance function f_{sas} can be written as:

$$f_{\text{sas}}(p) = \begin{cases} -\|p - x_{\text{sas}}^p\| & \text{if } p \text{ lies inside the SAS,} \\ \|p - x_{\text{sas}}^p\| & \text{if } p \text{ lies outside the SAS.} \end{cases} \quad (1.4.1)$$

In the above formula, x_{sas}^p depends on p . When p lies on the SAS, p coincides with x_{sas}^p and $f_{\text{sas}}(p) = 0$. It remains therefore to find a closest point x_{sas}^p 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_{sas}^p is chosen as one of them. In the context, this is not

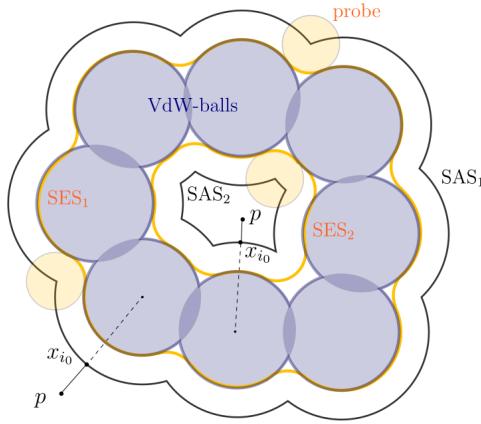


Figure 4: This is a 2D schematic of different molecular surfaces, including the VdW surface, the cSAS, the eSAS, the cSES and the eSES. The yellow discs denote the probes, representing solvent molecules, while the blue discs denote the VdW-balls. SAS_1 is the trace of the probe center when a probe rolls over the exterior of the VdW-balls, while SAS_2 is the trace of the probe center when a probe rolls over the inner holes of the VdW-balls. SES_1 and SES_2 respectively denote the corresponding solvent excluded surfaces to SAS_1 and SAS_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 $m = 1, \dots, m_1$. Denote by l_m the m -th SAS circular arc on the SAS where $m = 1, \dots, m_2$. Denote by x_m the m -th SAS intersection point on the SAS where $m = 1, \dots, m_3$. Furthermore,

denote by I the set of all SAS intersection points written as:

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

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 $i_0 \in \{1, \dots, M\}$ be such that*

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

Then, the point

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

is the closest point to p on the SAS and

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

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

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_{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_{sas} on the SAS and the corresponding region $R(x_{\text{sas}})$ for three cases where x_{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_{sas} is a point on the SAS. If x_{sas} is on an SAS spherical patch P_m on the i -th SAS-ball S_i , then x_{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_{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_{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_{sas} such that $V \subset P_m$, see Figure 5, and since x_{sas} is a closest point on the SAS to p , we have

$$\|p - x_{\text{sas}}\| \leq \|p - x\|, \quad \forall x \in V, \quad (1.4.5)$$

which yields that the vector from x_{sas} to p is perpendicular to any vector in the tangent plane of S_i at x_{sas} , thus the vector from p to x_{sas} is the normal vector at x_{sas} of S_i . As a consequence, p must lie on the line passing through x_{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_{sas} is obviously a closest point on the sphere S_i to p , see Figure 5. 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

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

That is, $\|p - x\| \geq \|p - x_{\text{sas}}\|, \forall x \in \text{SAS}$, which means that x_{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_{sas} is the unique closest point to p . \square

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_{sas} is a point on the SAS. If x_{sas} is on an SAS circular arc l_m associated with S_i and S_j , then x_{sas} is a closest point on the SAS to p if and only if p lies in the closed triangle $\Delta x_{\text{sas}}c_ic_j$ defined by three vertices x_{sas}, c_i and c_j . That is, $R(x_{\text{sas}}) = \Delta x_{\text{sas}}c_ic_j$.*

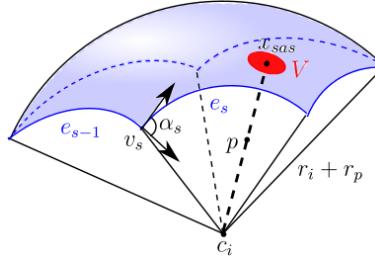


Figure 5: Illustration when the point x_{sas} lies on an SAS spherical patch P_m which is part of an SAS-sphere S_i , and p lies on the segment $[c_i, x_{\text{sas}}]$. V represents a neighborhood of x_{sas} on the spherical patch. α_s is the angle variation at the vertex v_s between two neighboring circular arcs e_{s-1} and e_s on the boundary of this spherical patch.

Further, the closest point x_{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_{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 γ_0 of x_{sas} with $\gamma_0 \subset l_m$, see Figure 6. Since x_{sas} is a closest point to p , we have

$$\|p - x_{\text{sas}}\| \leq \|p - x\|, \quad \forall x \in \gamma_0. \quad (1.4.7)$$

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

$$\|p' - x_{\text{sas}}\| \leq \|p' - x\|, \quad \forall x \in \gamma_0, \quad (1.4.8)$$

which yields that the vector from x_{sas} to p' is perpendicular to the tangent vector of γ_0 at x_{sas} . As a consequence, p' has to lie on the ray Ox_{sas} starting from O and passing through x_{sas} , where O is the center of the circular arc l_m . Thus, p must lie on the closed half plane Π defined by the three points x_{sas} , c_i and c_j and whose border is the line passing through c_i and c_j . The closed half plane Π contains the ray Ox_{sas} and is perpendicular to the plane where l_m lies. We then take another two neighboring curves γ_1 and γ_2 of x_{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, γ_1 has one closest endpoint x_{sas} and is open at the other end, which is the same for γ_2 . From the assumption that x_{sas} is a closest point on the SAS to p , we have the following inequality

$$\|p - x_{\text{sas}}\| \leq \|p - x\|, \quad \forall x \in \gamma_1 \cup \gamma_2. \quad (1.4.9)$$

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 Π with three vertices x_{sas} , c_i and c_j . Otherwise, we can find a point $x \in (\gamma_1 \cup \gamma_2) \setminus x_{\text{sas}}$ strictly

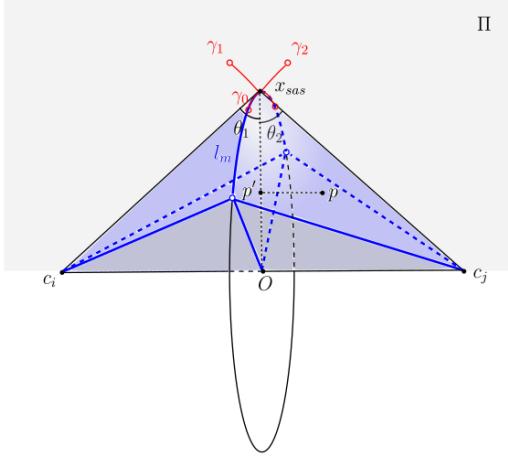


Figure 6: Illustration when l_m is an SAS circular arc associated with two SAS-spheres S_i and S_j . The point x_{sas} on l_m is a closest point on the SAS to p . γ_0 is a small neighborhood of x_{sas} on the circular arc l_m , whereas γ_1 and γ_2 represent two small neighboring curves of x_{sas} on the SAS, with x_{sas} as the endpoints, $\gamma_1 \subset \Pi \cap S_i \cap \text{SAS}$, and $\gamma_2 \subset \Pi \cap S_j \cap \text{SAS}$.

closer to p than x_{sas} , which contradicts the assumption.

On the other hand, suppose that $p \in \Delta x_{\text{sas}} c_i c_j$. As a consequence, it is not difficult to obtain that x_{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 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

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

That is, $\|p - x\| \geq \|p - x_{\text{sas}}\|, \forall x \in \text{SAS}$, which means that x_{sas} is a 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 $p \in \Delta x_{\text{sas}} c_i c_j \setminus [c_i, c_j]$, 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_{sas} is the unique closest point to p . \square

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.

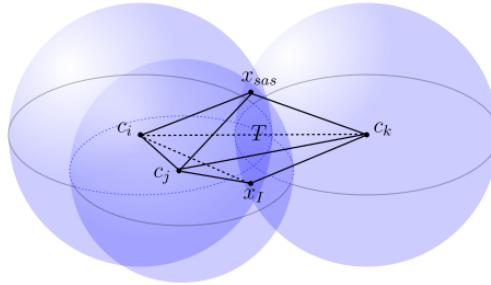


Figure 7: This figure illustrates the closed region T of three SAS-spheres with the centers (c_i, c_j, c_k) , where the tetrahedron T has five vertices $(x_{\text{sas}}, c_i, c_j, c_k, x_I)$. Here, x_{sas} is one SAS intersection point in I , and x_I is the other SAS intersection point.

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 7. 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:

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

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_{sas}^p on the SAS to p according to this decomposition. The signed distance function $f_{\text{sas}}(p)$ can therefore be calculated analytically by the formula (1.4.1), 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 [37] 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:

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

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 [6], 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 [7, 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 γ of the union of these discs can be classified into two types: open circular arcs $\{l_1, l_2, \dots, l_n\}$ and intersection points $\{x_1, x_2, \dots, x_n\}$, with the number of circular arcs equal to the number of intersection points. Take $A_1 = l_1, \dots, A_n = l_n, A_{n+1} = \{x_1, x_2, \dots, x_n\}$ in the Voronoi diagram. As a consequence, we obtain $n + 1$ corresponding new Voronoi-type regions $\{R_1, \dots, R_n, R_{n+1}\}$, 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 γ 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 closet point to x on γ .

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 $A_1 = P_1, \dots, A_{m_1} = P_{m_1}, A_{m_1+1} = l_1, \dots, A_{m_1+m_2} = l_{m_2}$ and $A_{m_1+m_2+1} = \{x_1, \dots, x_{m_3}\}$ with the above-mentioned notations. As a consequence, we obtain $m_1 + m_2 + 1$ corresponding new Voronoi-type regions $\{R_1, \dots, R_{m_1}, R_{m_1+1}, \dots, R_{m_1+m_2}, R_{m_1+m_2+1}\}$.

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 closet 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.

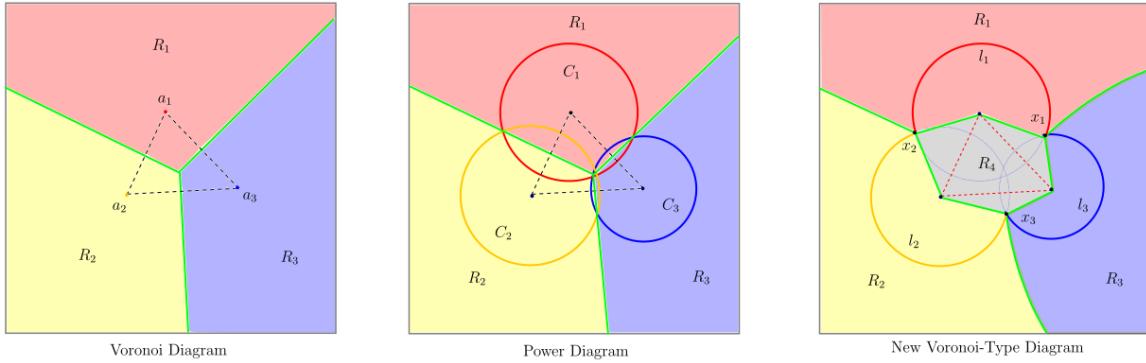


Figure 8: The left figure gives the Voronoi diagram of three points $\{a_1, a_2, a_3\}$ in \mathbb{R}^2 . R_1 is the Voronoi region associated with a_1 , while R_2 associated with a_2 and R_3 associated with a_3 . The middle figure gives the power diagram of three circles $\{C_1, C_2, C_3\}$ in \mathbb{R}^2 , with three corresponding power regions R_1 , R_2 and R_3 . The right figure gives the new Voronoi-type diagram of three circles in \mathbb{R}^2 , with four regions R_1 , R_2 , R_3 and R_4 respectively corresponding to l_1 , l_2 , l_3 and $\{x_1, x_2, x_3\}$.

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):

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

where α_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, χ 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_{sas} :

$$A_{\text{sas}} = \sum_{m=1}^{m_1} A_{P_m}. \quad (1.4.14)$$

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 6), 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_{sas} as below:

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

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. $\text{SES} = f_{\text{sas}}^{-1}(r_p)$, we propose an implicit function of the SES:

$$f_{\text{ses}}(p) = \begin{cases} -\|p - x_{\text{sas}}^p\| + r_p & \text{if } f_{\text{sas}}(p) \leq 0, \\ \|p - x_{\text{sas}}^p\| + r_p & \text{if } f_{\text{sas}}(p) \geq 0, \end{cases} \quad (1.5.1)$$

where x_{sas}^p 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:

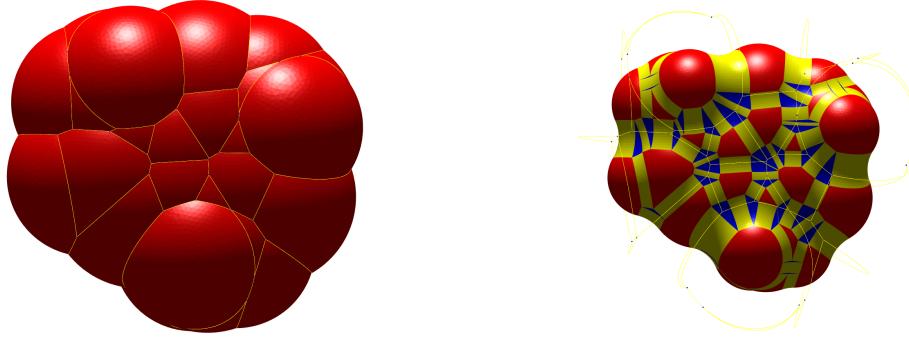


Figure 9: The SAS and the SES of caffeine with probe radius $r_p = 1.5\text{\AA}$. On the left, the SAS is composed of spherical patches in blue, circular arcs in yellow and intersection points in red. On the right, the patches in red (resp. in yellow or in blue) are convex spherical patches (resp. toroidal 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:

$$\begin{cases} P_+ = \{p : f_{\text{ses}}(p) = 0, p \in R(P_m)\}, & 1 \leq m \leq m_1 \\ P_t = \{p : f_{\text{ses}}(p) = 0, p \in R(l_m)\}, & 1 \leq m \leq m_2 \\ P_- = \{p : f_{\text{ses}}(p) = 0, p \in R(x_m)\}, & 1 \leq m \leq m_3 \end{cases} \quad (1.5.2)$$

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 $x_{\text{ses}} \in \text{SES}$ a *singularity* if the SES is not smooth at x_{ses} , which means that its implicit function $f_{\text{ses}}(p) = f_{\text{sas}}(p) + r_p$ is non-differentiable at x_{ses} in \mathbb{R}^3 . If $x_{\text{ses}} \in \text{SES}$ 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 $x_{\text{ses}} \in \text{SES}$ 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

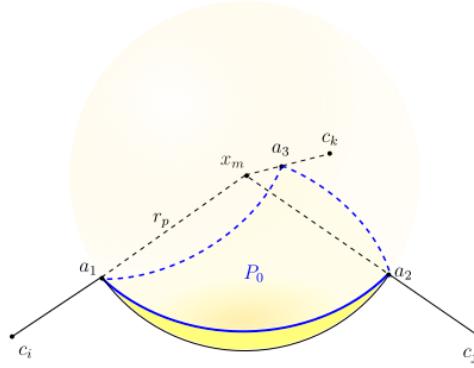


Figure 11: This is the schematic of the concave spherical triangle P_0 corresponding to an intersection point x_m , with the boundary composed of three circular arcs in blue ($\widehat{a_1a_2}$, $\widehat{a_2a_3}$ and $\widehat{a_3a_1}$) on the spherical probe. This spherical triangle touches at the same time three VdW-balls at vertices (a_1, a_2, a_3) .

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 (1.4.11), 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 $x_{\text{sas}} = x_m$ 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 $x_{\text{ses}} \in P_-$ but does not belong to $\cup_{x \in K} B_{r_p}(x)$, 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 $x_{\text{ses}} = c_i$ (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 $x_{\text{sas}} = x_m$ is the unique closest point among all SAS intersection points. \square

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 $P_0 \cap (\bigcup_{x \in K} B_{r_p}(x))$ 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_{ses} is not a singularity on the SES, and x_{ses} has an unique closest point $\{x_{\text{sas}}^1\}$ on the SAS.
- (2) $2 \leq N < \infty$: x_{ses} is a singularity on a concave spherical SES patch, and its closest points $\{x_{\text{sas}}^1, \dots, x_{\text{sas}}^N\}$ are among the SAS intersection points.
- (3) $N = \infty$: x_{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):

$$\sum_v \alpha_v + \sum_e k_e |e| + \frac{1}{r_i^2} A_{P_+} = 2\pi\chi, \quad (1.5.4)$$

where α_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 χ 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

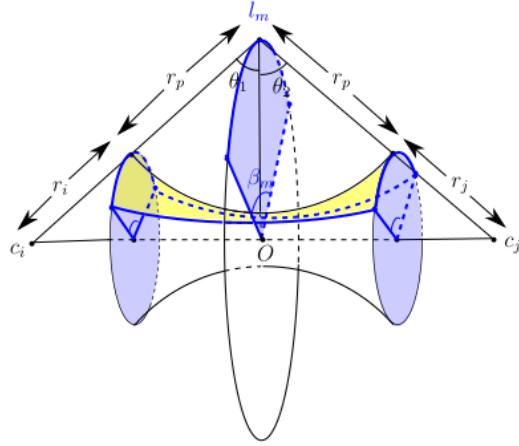


Figure 12: The yellow patch is a toroidal patch P_t on the SES corresponding to an SAS circular arc l_m with the radius $r_{l_m} > r_p$ and the radian β_m . θ_1 denotes the angle between the line connecting c_i with a point on l_m and the disc where l_m lies. Similarly, θ_2 denotes the angle between the line connecting c_j with the same point on l_m and the disc where l_m lies.

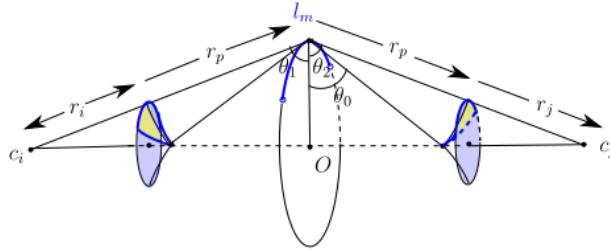


Figure 13: The two yellow parts form a toroidal patch P_t on the SES corresponding to an SAS circular arc l_m with the radius $r_{l_m} < r_p$ and the radian β_m . θ_0 denotes the angle between the line connecting a singularity on P_t with a point on l_m and the disc where l_m lies and θ_1, θ_2 are defined as in Fig. 12.

case where $r_{l_m} > r_p$, 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 :

$$A_{P_t} = r_p \beta_m [r_{l_m}(\theta_1 + \theta_2) - r_p(\sin \theta_1 + \sin \theta_2)]. \quad (1.5.5)$$

In the case where $r_{l_m} \leq r_p$, 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 :

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

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:

$$\sum_v \alpha_v + \sum_e k_e |e| + \frac{1}{r_p^2} A_{P_-} = 2\pi\chi, \quad (1.5.7)$$

where α_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 χ 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_{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

$$V_{R_s \cap R(P_m)} = \frac{1}{3} A_{P_m} (r_i + r_p) \left(1 - \frac{r_i^3}{(r_i + r_p)^3} \right), \quad (1.5.8)$$

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_{l_m} the radius of l_m and by β_m the radian of l_m . In the case where $r_{l_m} > r_p$, using the notations of Figure 12, the volume of the region $R_s \cap R(l_m)$ is given by

$$V_{R_s \cap R(l_m)} = \beta_m r_p^2 \left[\frac{r_{l_m}}{2} (\theta_1 + \theta_2) - \frac{r_p}{3} (\sin \theta_1 + \sin \theta_2) \right]. \quad (1.5.9)$$

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

$$V_{R_s \cap R(l_m)} = \beta_m r_p^2 \left[\frac{r_{l_m}}{2} (\theta_1 + \theta_2 - 2\theta_0) - \frac{r_p}{3} (\sin \theta_1 + \sin \theta_2 - 2 \sin \theta_0) \right] + \frac{1}{3} \beta_m r_{l_m}^2 \sqrt{r_p^2 - r_{l_m}^2}. \quad (1.5.10)$$

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

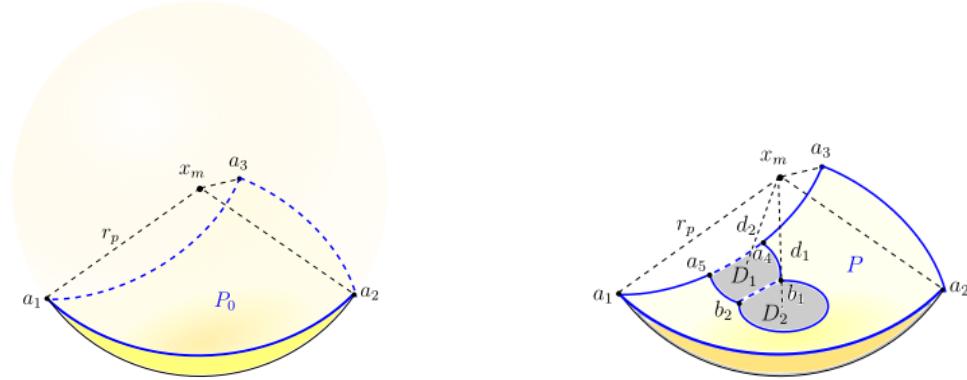


Figure 14: On the left, the concave spherical triangle P_0 with vertices (a_1, a_2, a_3) corresponds to an intersection point x_m . In the case where P_- coincide with P_0 , there will be no singularity on the concave spherical patch. On the right, the concave spherical patch P_- does not coincide with 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:

$$V_{R_s \cap R(x_m)} = \frac{1}{3} A_{P_-} r_p + \sum_i \frac{1}{3} A_{D_i} d_i, \quad (1.5.11)$$

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:

$$V_{R_s} = \sum_{\xi=P_m, l_m, x_m} V_{R_s \cap R(\xi)}. \quad (1.5.12)$$

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

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 $\{x_1, \dots, x_{m_1}\}$, 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 \mathcal{L}_k^i .
- 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_k^i 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_k^i and P_l^j 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_∞ 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

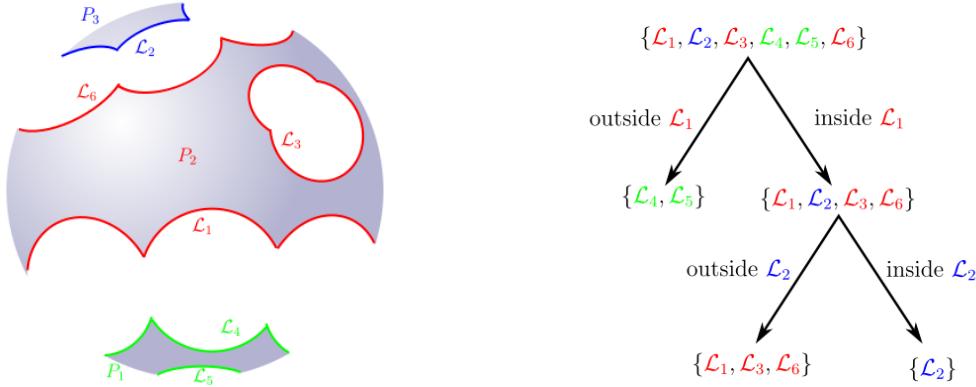


Figure 15: On the left is a brief schematic of an SAS-sphere and the loops on it. There are six loops on the SAS-sphere $\{\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4, \mathcal{L}_5, \mathcal{L}_6\}$, and three spherical patches with the boundaries formed by two loops in green $\{\mathcal{L}_4, \mathcal{L}_5\}$, three loops in red $\{\mathcal{L}_1, \mathcal{L}_3, \mathcal{L}_6\}$ and one loop (circle) in blue $\{\mathcal{L}_2\}$ 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 \mathcal{L} on a fixed SAS-sphere S_i , \mathcal{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 \mathcal{L} , denoted by \mathcal{L}° , while the other open part as the exterior of \mathcal{L} , denoted by \mathcal{L}^c . We say that another loop \mathcal{L}' on S_i is inside \mathcal{L} if $\mathcal{L}' \in \overline{\mathcal{L}^\circ}$, where $\overline{\mathcal{L}^\circ} = \mathcal{L} \cup \mathcal{L}^\circ$ is the closed hull of \mathcal{L}° on S_i . Notice that each loop \mathcal{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 \mathcal{L}' on S_i is inside the loop \mathcal{L} , we can classify all loops on S_i into two parts: the loops inside \mathcal{L} (including \mathcal{L} itself) and the remaining loops outside \mathcal{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 $\{\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4, \mathcal{L}_5, \mathcal{L}_6\}$ 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 $\{\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4, \mathcal{L}_5, \mathcal{L}_6\}$ and test with \mathcal{L}_1 , to divide into the set of loops inside \mathcal{L}_1 ($\{\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_6\}$) and the set of the remaining loops outside \mathcal{L}_1 ($\{\mathcal{L}_4, \mathcal{L}_5\}$). Then, we test with \mathcal{L}_2 , to find that \mathcal{L}_2 is inside itself, while $\{\mathcal{L}_1, \mathcal{L}_3, \mathcal{L}_6\}$ are outside, which implies that \mathcal{L}_2 itself forms the boundary of a spherical patch. Afterwards, we find that $\{\mathcal{L}_1, \mathcal{L}_6\}$ are both inside \mathcal{L}_3 by testing with \mathcal{L}_3 and that

$\{\mathcal{L}_1, \mathcal{L}_3\}$ are inside \mathcal{L}_6 by testing with \mathcal{L}_6 . This implies that $\{\mathcal{L}_1, \mathcal{L}_3, \mathcal{L}_6\}$ form the boundary of the second spherical patch, as these three loops are inside each other. Finally, we test respectively with \mathcal{L}_4 and \mathcal{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 \mathcal{L}' and \mathcal{L} be two loops on S_i . It is left to explain how to test whether \mathcal{L}' is inside \mathcal{L} or not. We assume that \mathcal{L} is composed by n circular arcs which are formed by the intersection of S_i and other SAS-spheres S'_1, \dots, S'_n . Denote all intersection circles by C_1, \dots, C_n , where $C_j = S_i \cap S'_j$. The corresponding SAS-balls to S'_1, \dots, S'_n are denoted by B'_1, \dots, B'_n . Since \mathcal{L}' and \mathcal{L} do not cross each other, we can determine if the loop \mathcal{L}' is inside \mathcal{L} or not by testing whether a particular point $x \in \mathcal{L}'$ is inside \mathcal{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 , $k = 1, \dots, n$, 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 \mathcal{L} or not, note that $\mathcal{L}' \cap \bigcup_{j=1}^n B'_j = \emptyset$.

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

Proof. We first prove the sufficiency. Assume that x is inside \mathcal{L} , i.e. $x \in \overline{\mathcal{L}^\circ} = \mathcal{L}^\circ \cup \mathcal{L}$. Assume by contradiction that $x_{k_0} \in C_{k_0}$ does not belong to \mathcal{L} . Then, since $C_{k_0} \subset \mathcal{L} \cup \mathcal{L}^c$ it follows that $x_{k_0} \in \mathcal{L}^c$. Then, the shortest path on S_i from $x \in \mathcal{L}^\circ$ to $x_{k_0} \in \mathcal{L}^c$ must cross \mathcal{L} . As a consequence, the intersection point between \mathcal{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 $x_{k_0} \in \mathcal{L}$. By contradiction again, we assume that x is not inside \mathcal{L} , i.e. $x \in \mathcal{L}^c$, which yields that $x \in \Omega := \mathcal{L}^c \setminus \bigcup_{j=1}^n B_j$ 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 Γ 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 \mathcal{L} . \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 15 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.

	A_{sas} (\AA^2)	V_{sas} (\AA^3)	A_{ses} (\AA^2)	V_{ses} (\AA^3)
Closed form	1.025416e+02	9.200259e+01	3.223514e+01	1.475567e+01
Proposed method	1.025416e+02	9.200259e+01	3.223514e+01	1.475567e+01
Difference	0	-2.842171e-14	7.105427e-15	-3.552714e-15

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.

Name	A_{cses}	A_{eses}	$\frac{\Delta A_{\text{ses}}}{A_{\text{eses}}}$	A_{csas}	A_{esas}	$\frac{\Delta A_{\text{sas}}}{A_{\text{esas}}}$
1ETN	956.636	956.636	0	1397.664	1397.664	0
1B17	3372.577	3372.577	0	4297.098	4297.098	0
101M	7335.438	7096.387	-0.034	8424.749	8413.275	-0.001
3WPE	28575.829	27390.001	-0.043	29747.737	29644.146	-0.003
1A0C	61006.132	54231.268	-0.125	53107.185	51933.798	-0.023
Name	V_{cses}	V_{eses}	$\frac{\Delta V_{\text{ses}}}{V_{\text{eses}}}$	V_{csas}	V_{esas}	$\frac{\Delta V_{\text{sas}}}{V_{\text{esas}}}$
1ETN	1567.728	1567.728	0	3333.077	3333.077	0
1B17	8630.389	8630.389	0	14435.224	14435.224	0
101M	25029.270	25195.072	0.007	36942.276	36943.700	0.000
3WPE	98651.803	99642.385	0.001	143181.258	143201.169	0.000
1A0C	238282.734	244765.065	0.026	326348.079	326601.929	0.001

Table 1.2: Different molecular areas (in \AA^2 , top) and volumes (in \AA^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>) where we use the notations $\Delta A_{\text{ses}} = A_{\text{eses}} - A_{\text{cses}}$ and $\Delta A_{\text{sas}} = A_{\text{esas}} - A_{\text{csas}}$ resp. $\Delta V_{\text{ses}} = V_{\text{eses}} - V_{\text{cses}}$ and $\Delta V_{\text{sas}} = V_{\text{esas}} - V_{\text{csas}}$. The probe radius r_p is fixed to be 1.5 \AA .

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),

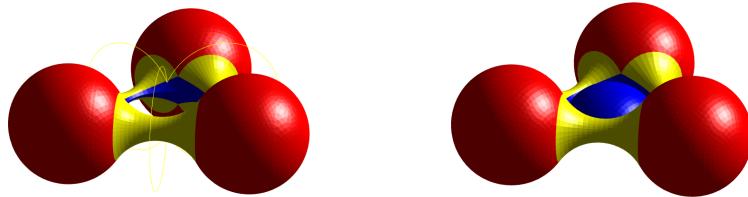


Figure 16: The SES of three balls with a singular circle on the concave spherical patches (left) and the SES with the intersection of two concave spherical patches (right, triangulation provided by the MSMS-algorithm). The probe radius $r_p = 1\text{\AA}$.

	$A_{\text{ses}} (r_p = 2\text{\AA})$	$A_{\text{sas}} (r_p = 2\text{\AA})$	$A_{\text{ses}} (r_p = 1\text{\AA})$	$A_{\text{sas}} (r_p = 1\text{\AA})$
MSMS	63.712	253.122	64.616	153.545
Proposed method	63.712	253.122	58.857	153.545

Table 1.3: Molecular areas (\AA^2) for three balls with different probe radii ($r_p = 2\text{\AA}, 1\text{\AA}$), 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 1.2\AA . In the case of a probe radius of $r_p = 2\text{\AA}$, 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 $r_p = 1\text{\AA}$ (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 $r_p = 1\text{\AA}$, 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.

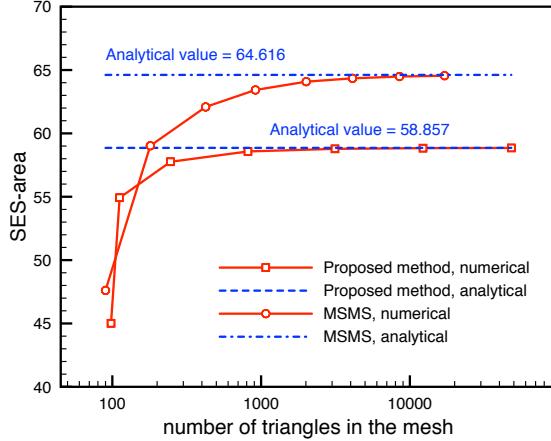


Figure 17: The numerical SES-areas of the three atoms reported in \AA^2 ($r_p = 1\text{\AA}$, see the left of Figure 16) with respect to the number of triangles in the mesh. For each method, the analytical and numerical area is computed.

Name	Number of atoms	MSMS		Proposed method		ΔA_{esas}	ΔA_{eeses}
		A_{esas}^*	A_{eeses}^*	A_{esas}	A_{eeses}		
1YJO	67	1325.395	822.629	1325.395	822.628	0	0
1ETN	160	1396.191	955.899	1397.664	956.636	-0.001	-0.001
1B17	483	4297.099	3372.577	4297.098	3372.577	0.000	0.000
101M	1413	8413.273	7096.392	8413.275	7096.387	0.000	0.000
2K4C	2443	13825.406	11404.567	13826.835	11409.523	0.000	0.000
3WPE	5783	29656.324	27387.289	29644.146	27390.001	0.000	0.000
1KJU	7671	34551.902	33172.387	34630.252	33148.676	-0.002	0.001
1A0T	9996	40821.195	38470.543	40757.735	38486.055	0.002	0.000
1A0C	13992	51231.840	53987.203	51933.798	54231.268	-0.014	-0.005
4XBG	19608	108054.734	101242.242	108158.716	101437.41	-0.001	-0.002

Table 1.4: Exterior molecular areas (\AA^2) of molecules at different sizes respectively obtained from the MSMS and the proposed method ($r_p = 1.5\text{\AA}$), where A_{esas}^* and A_{eeses}^* are results obtained from the MSMS, A_{esas} and A_{eeses} are results obtained from the proposed method, $\Delta A_{\text{esas}} = A_{\text{esas}}^* - A_{\text{esas}}$ and $\Delta A_{\text{eeses}} = A_{\text{eeses}}^* - A_{\text{eeses}}$.

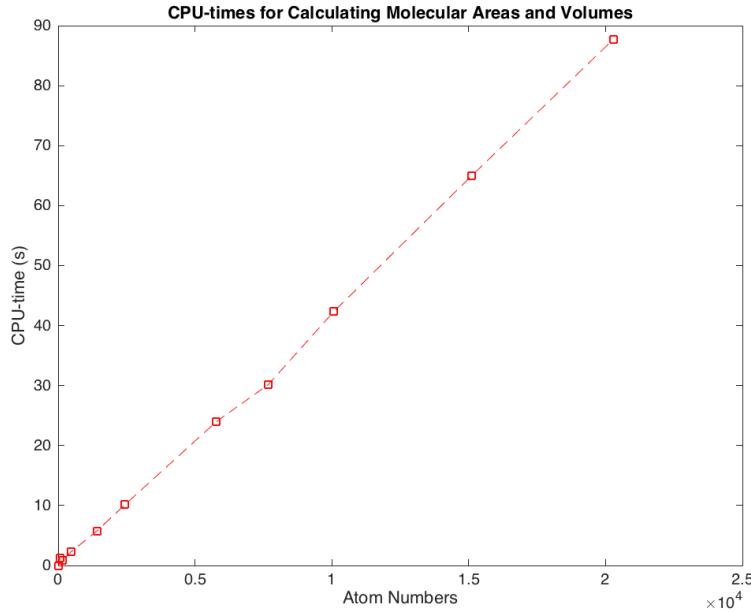


Figure 18: Total CPU-times of computing molecular areas and volumes for molecules at different sizes, from tens to tens of thousands ($r_p = 1\text{\AA}$).

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

2.6 Conclusion	92
---------------------------------	-----------

This chapter has been published as a journal paper [94]. In this chapter, we develop an algorithm for meshing molecular surfaces that is based on patch-wise meshing using an advancing-front method adapted to the particular case of molecular surfaces. We focus on the solvent accessible surface (SAS) and the solvent excluded surface (SES). The essential ingredient is a previously-developed analysis of molecular surfaces in Chapter 1 that allows us to describe all SES-singularities *a priori* and therefore gives a complete characterization of the SES. In addition, as mentioned in the introduction of this thesis, an algorithm for filling molecular inner holes is proposed based on the pre-computed data structures of molecular surfaces.

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 [109]. 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) [30]. 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 [98] or is more suitable for applications like docking [76], is more complicated.

2.1.1 Previous work

The definitions of the SAS and the SES were first introduced by Lee & Richards [67, 99] in the 1970s. The SES is also called the "smooth molecular surfaces" or "Connolly's surfaces" due to Connolly's fundamental calculation on it [30]. 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, 98, 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 16 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, 59, 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

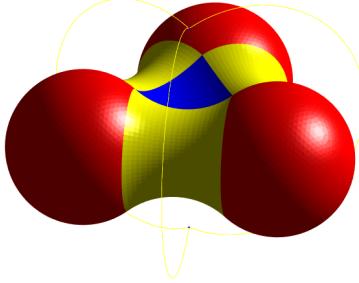


Figure 1: 3D schematic of the SES illustrating the convex spherical patches (red), the toroidal patches (yellow) and the concave spherical patches (blue).

and the 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). 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 $\tilde{f}_{\text{vdw}}^{-1}(0) = \{p \in \mathbb{R}^3 | \tilde{f}_{\text{vdw}}(p) = 0\}$ with the following implicit function:

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

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 $\tilde{f}_{\text{sas}}^{-1}(0)$ with the following implicit function:

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

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 $\tilde{f}_{\text{sas}}(p)$ 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_{sas}^p depending on p . We emphasize that there might exist more than one closest point to p and in this case, x_{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_{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_{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 $f_{\text{ses}}(p) \leq 0$.

2.2.3 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_{sas} on the SAS, we define a mapping by

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

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

Theorem 2.2.1. *The following equivalence statements hold:*

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

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

We can generalize the mapping \mathcal{R} by

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

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 $P_+ = \mathcal{R}(P_{i_1}) \cap \Gamma_{\text{ses}}$ where Γ_{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 $P_t = \mathcal{R}(l_{i_2}) \cap \Gamma_{\text{ses}}$; for an SAS intersection point denoted by x_{i_3} with index i_3 , its corresponding concave SES patch P_- can be written as $P_- = \mathcal{R}(x_{i_3}) \cap \Gamma_{\text{ses}}$. 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 $\mathcal{R}(P_{i_1})$ is a spherical sector, $\mathcal{R}(l_{i_2})$ is a double-cone region and $\mathcal{R}(x_{i_3})$ 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.

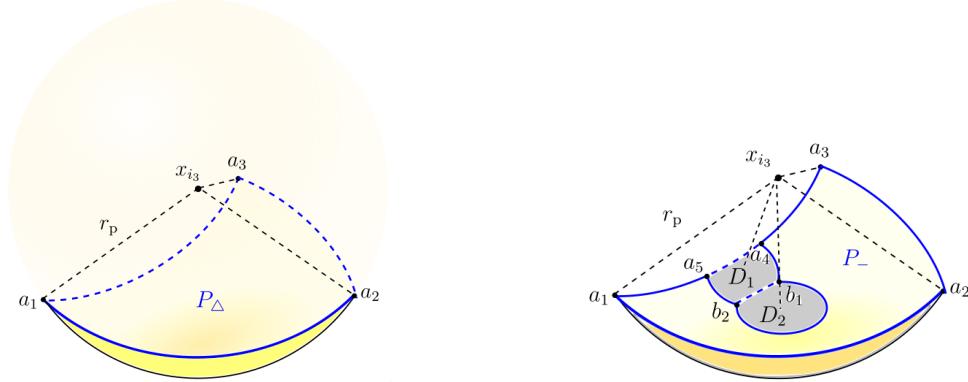


Figure 2: On the left, the concave spherical triangle P_Δ with vertices (a_1, a_2, a_3) corresponds to an intersection point x_{i3} . In the case where P_- coincide with P_Δ , there will be no singularity on the concave spherical patch. On the right, the concave spherical patch P_- does not coincide with P_Δ 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_Δ with two other nearby spherical probes. These two regions have the boundaries composed of line segments and circular arcs.

For an SAS intersection point $x_{i3} \in S_{j_1} \cap S_{j_2} \cap S_{j_3}$, denote by P_Δ the concave spherical triangle corresponding to x_{i3} (see the left of Figure 2 for a schematic), that is,

$$P_\Delta = \partial B_{r_p}(x_{i3}) \cap \left\{ p : p = x_{i3} + \sum_{k=1}^3 \lambda_k (c_{j_k} - x_{i3}), \forall \lambda_k \geq 0 \right\},$$

where c_{j_k} is the center of the sphere S_{j_k} associated with x_{i3} , $k = 1, 2, 3$, and $B_r(c)$ denotes the open ball with center c and radius r . Notice that P_Δ is a triangle-shaped patch generated by the sphere $\partial B_{r_p}(x_{i3})$ and three planes, see the left of Figure 2 for an illustration. Then, we have the following result from [93] for computing the concave SES patch P_- corresponding to x_{i3} .

Theorem 2.2.2. $P_- = P_\Delta \setminus \bigcup_{x \in K} B_{r_p}(x)$ where $K = \{x \in I : \|x - x_{i3}\| < 2r_p, x \neq x_{i3}\}$ and I denotes the set of all SAS intersection points.

If P_- does not coincide with P_Δ , singular arcs occur on the concave SES patch and constitute part of its boundary ∂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_Δ 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*

together with the analytical representation of each convex SES patch and each toroidal SES patch.

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_{\max} . As a consequence, the intersection matrix is defined of size $M \times k_{\max}$ 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:

SAS Intersection Point

- (x, y, z) : coordinate of the point
- (i, j, k) : indices of three corresponding SAS-spheres

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:

Circular SAS Arc

- (i_1, i_2) : index of the starting and ending intersection point
- (x, y, z) : coordinate of the center
- r : radius
- β : radian
- (i, j) : indices of two corresponding SAS-spheres

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:

SAS Loop

- l_1, l_2, \dots, l_{n_1} : the consisting circular arcs
- i : index of the SAS-sphere on which the loop lies

Spherical SAS Patch

- $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_{n_2}$: the loops forming the boundary of the patch
- i : index of the SAS-sphere on which the patch lies

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 $\{\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_n\}$ 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 $\mathcal{L}_i \subset \mathcal{L}_i^\circ$ and $\mathcal{L}_j \subset \mathcal{L}_i^\circ$, where \mathcal{L}_i° and \mathcal{L}_j° represent respectively the "interior" of \mathcal{L}_i and \mathcal{L}_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 10 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:

Rectangle-shaped Toroidal Patch

- i_l : index of the corresponding circular SAS arc
- \mathcal{L}_1^t or $(\mathcal{L}_1^t, \mathcal{L}_2^t)$: one loop composed of four circular arcs or two circles forming the boundary

Double-triangle-shaped Toroidal Patch

- i_l : index of the corresponding circular SAS arc
- $(\mathcal{L}_1^t, \mathcal{L}_2^t)$: two loops each composed of up to three circular arcs forming the boundary

Convex Spherical SES Patch

- i_P : index of the corresponding spherical SAS patch
- $\mathcal{L}_1^+, \mathcal{L}_2^+, \dots, \mathcal{L}_{m_1}^+$: loops forming the boundary of the patch

Concave Spherical SES Patch (or Sub-patch)

- i_I : index of the corresponding SAS intersection point
- $\mathcal{L}_1^-, \mathcal{L}_2^-, \dots, \mathcal{L}_{m_2}^-$: loops forming the boundary of the patch

In the above data structures of the SES, a loop (\mathcal{L}_i^t) 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 $(\mathcal{L}_i^+ \text{ or } \mathcal{L}_i^-)$ 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 Γ^e and the separate inner subsurfaces of the cSAS by Γ_j^i , $1 \leq j \leq n$. Further, denote the set of intersection points on Γ^e by I^e and the set of intersection points on Γ_j^i by I_j^i . 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*

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

and

$$\text{dist}(\Gamma_j^i, \Gamma_k^i) = \min_{x \in I_j^i, y \in I_k^i} \|x - y\|, \quad \forall 1 \leq j, k \leq n,$$

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 $p \in \Gamma_j^i$. If it has a closest point $x_{\text{sas}}^p \in \Gamma^e$ 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, $p \in \Gamma_j^i$ 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 Γ^e in the set of intersection points I^e . On the other hand, we have the same conclusion that for any point on Γ^e , it can only have a closest point to Γ_j^i in I_j^i . Therefore, the distance between Γ^e and Γ_j^i is equal to the distance between the two sets of intersection points I^e and I_j^i . That is to say,

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

Further, we have the same result for Γ_j^i and Γ_k^i , i.e., the distance between Γ_j^i and Γ_k^i is equal to the distance between the two sets I_j^i and I_k^i . \square

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 Γ^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 Γ^e . This prevents the added virtual spheres affecting the geometry of Γ^e .

Given an inner subsurface Γ_j^i , we want to fill the inner hole enclosed by it with virtual spheres. We propose the following three steps to do this:

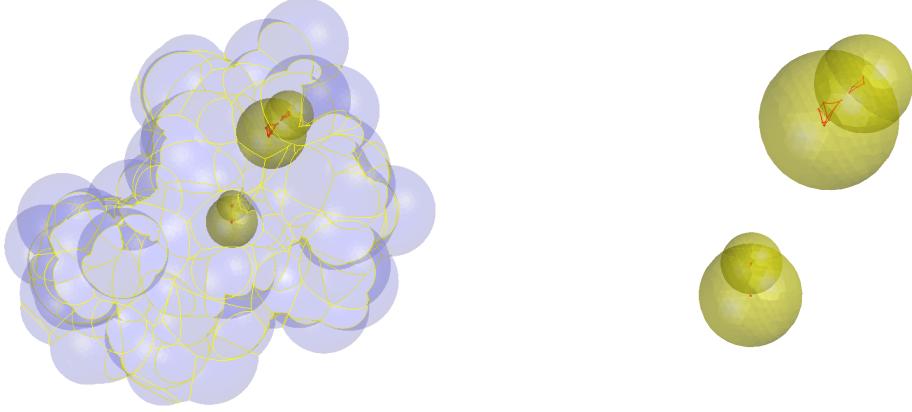


Figure 3: The left figure shows the SAS (blue) of 1ETN with $r_p = 0.7\text{\AA}$ and the added virtual spheres (yellow) filling the inner holes. On the right, the inner holes and the added virtual spheres are magnified for a better view.

- First, we fix the radii of virtual atoms as $\delta_j = \frac{1}{2}\text{dist}(\Gamma^e, \Gamma_j^i)$.
- Second, we take a small rectangular box containing the inner hole and take a set of virtual spheres with radius δ_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 Γ_j^i greater than δ_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 Γ_j^i while do not intersect the exterior region of the SAS-cavity. By repeating the above steps for each inner cavity with the boundary Γ_j^i , $\forall 1 \leq j \leq n$, 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.

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

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_{l_m} and θ_{l_m} . At the same time, we set a maximum allowed angle variation between two neighboring division points to α_0 (in our codes, we use $\alpha_0 = 60^\circ$) in the case where the radius of the circular arc r_{l_m} is small compared to d . Then, the number of elements of the discretization of this circular arc denoted by N_{l_m} is set as follows:

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

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 α_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;

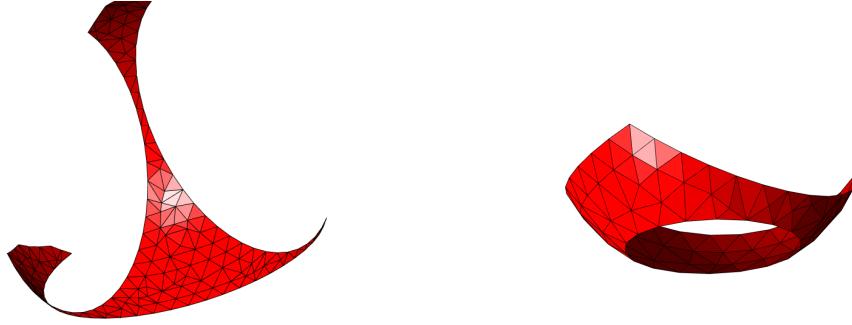


Figure 4: Meshing two spherical patches.

- 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

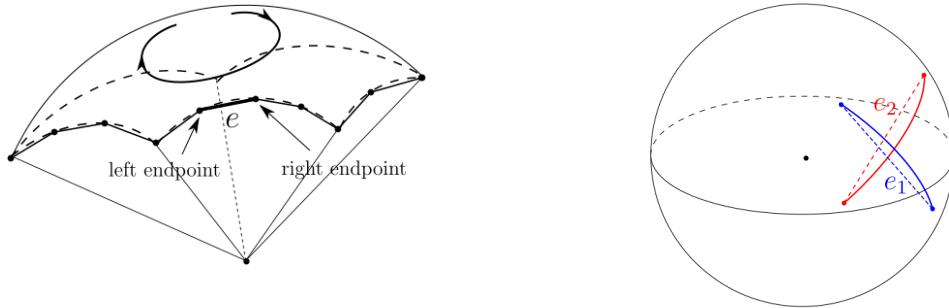


Figure 5: On the left, the left endpoint and the right endpoint of edge e are determined by the orientation of the loop. On the right, two dashed edges e_1 (in blue) and e_2 (in red) "intersect" in the sense that their two projected chords in form of circular arcs intersect on the sphere.

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

$$\vec{n}_e = \frac{\overrightarrow{P_1 P_2}}{|\overrightarrow{P_1 P_2}|} \times \frac{\vec{n}_1 + \vec{n}_2}{2},$$

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_r and the endpoint of e_2 , other than P_2 , is similarly denoted by P_l . By projecting the vectors $\overrightarrow{P_1 P_l}$ and $\overrightarrow{P_1 P_r}$ to the tangent plane to the sphere at point P_1 , we obtain two vectors $\vec{\tau}_{1l}$ and $\vec{\tau}_{1r}$. The angle between e and e_1 , denoted by α_1 , is then defined to be the angle between vectors $\vec{\tau}_{1l}$ and $\vec{\tau}_{1r}$ if $\det(\vec{\tau}_{1l}, \vec{\tau}_{1r}, \vec{n}_1) \geq 0$ and to be 2π minus the angle between vectors $\vec{\tau}_{1l}$ and $\vec{\tau}_{1r}$ if $\det(\vec{\tau}_{1l}, \vec{\tau}_{1r}, \vec{n}_1) < 0$ where $\det(\cdot)$ denotes the determinant of a matrix. The angle between e and e_2 , denoted α_2 , is defined similarly to α_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 ε_α . 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 ε_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 $P_f P_1$ and $P_f P_2$ 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.,

$$P_0 = \operatorname{argmin}_{p \in S_f} \operatorname{dist}(p, e), \quad (2.5.2)$$

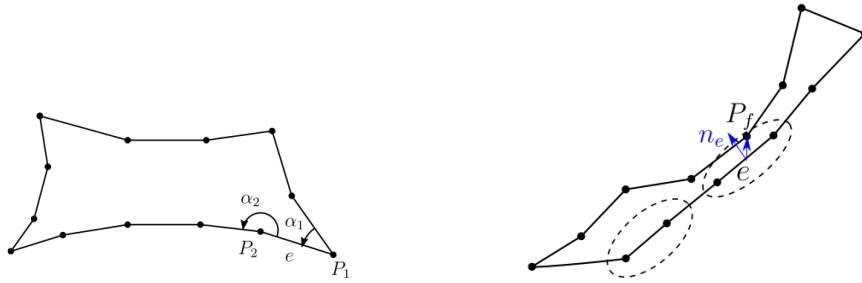


Figure 6: Planar schematics of two criterions for searching a possibly front point: angle criterion (left) and point-edge distance criterion (right).

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_{test} on the spherical patch. This testing point is constructed such that the edges $P_{test}P_1$ and $P_{test}P_2$ have the same length and that $\triangle P_1P_2P_{test}$ has a fixed height h (we take $h = \frac{\sqrt{3}}{2}d$). In the case where $|e| = d$, the triangle $\triangle P_{test}P_1P_2$ is equilateral. Then, we check if P_{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_{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 \vec{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 ε_d . If there exists a front point P_f with distance to P_{test} smaller than ε_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_{test} .



Figure 7: Planar schematics of two criterions for checking if P_{test} is suited as a new vertex of the mesh: point-edge criterion (left) and distance criterion (right).

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.



Figure 8: Schematics of meshing a rectangle (left) and a isosceles triangle (right) with the boundary division given.

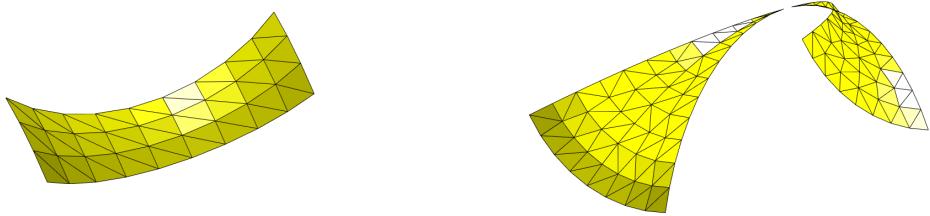


Figure 9: Meshing a rectangle-shaped patch (left) and a double-triangle-shaped patch (right).

2.5.3 Toroidal patches

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.

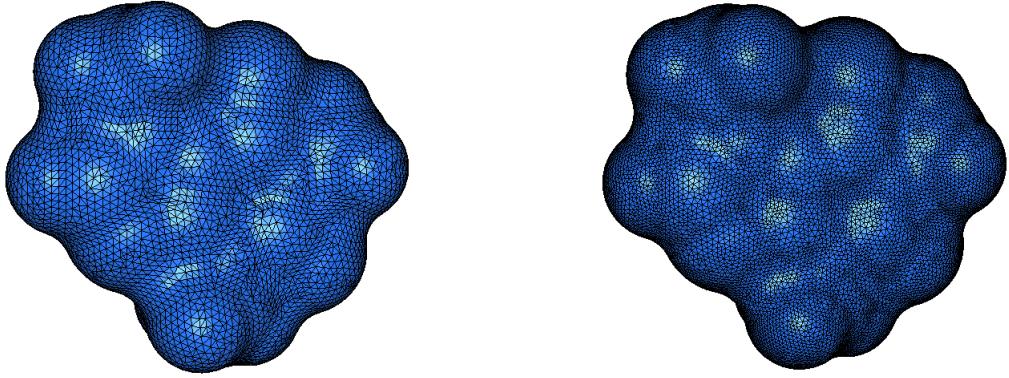


Figure 10: The mesh (right) of caffeine before remeshing with $r_p = 1\text{\AA}$ and the mesh (left) after remeshing with the MMGS tool.

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. [31] 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

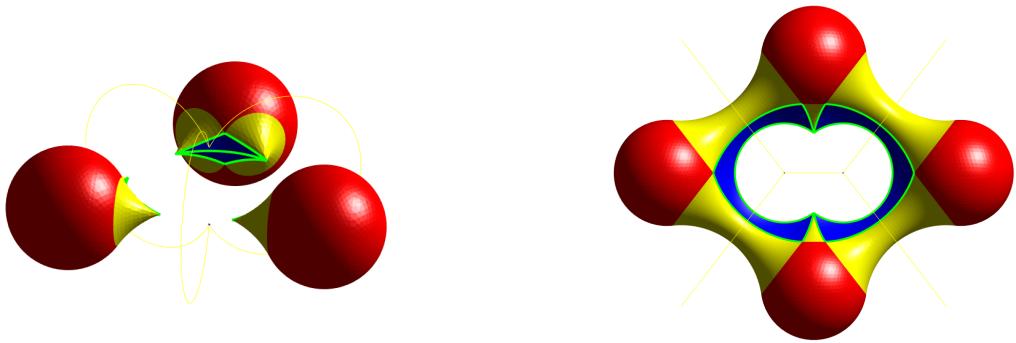


Figure 11: The eSES of three artificial spheres (left) and the eSES of four artificial spheres (right) where the green curves are the boundaries of singular concave patches.

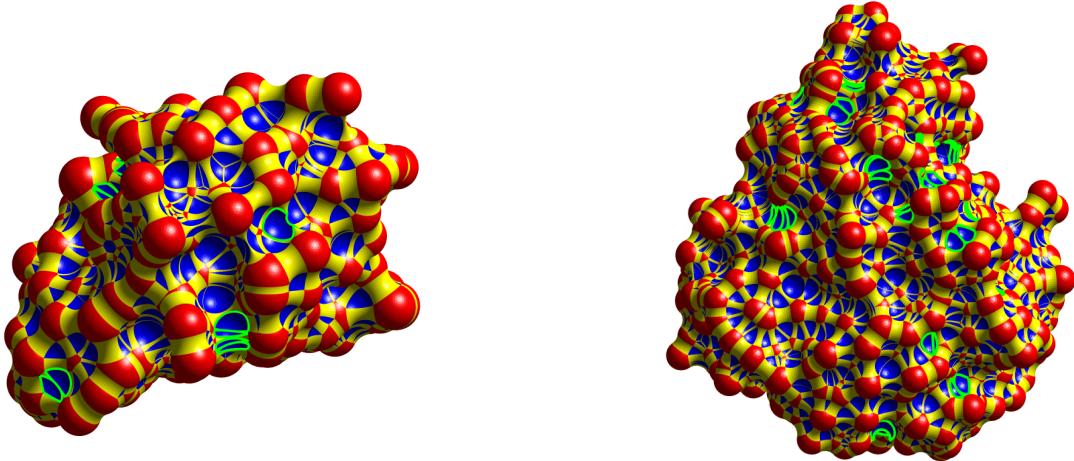


Figure 12: The eSES of molecule 1B17 with 485 atoms (left) and the eSES of molecule 101M with 1414 atoms (right) where $r_p = 1.5\text{\AA}$. The green curves are the boundaries of singular concave patches.

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, [55, 59, 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.

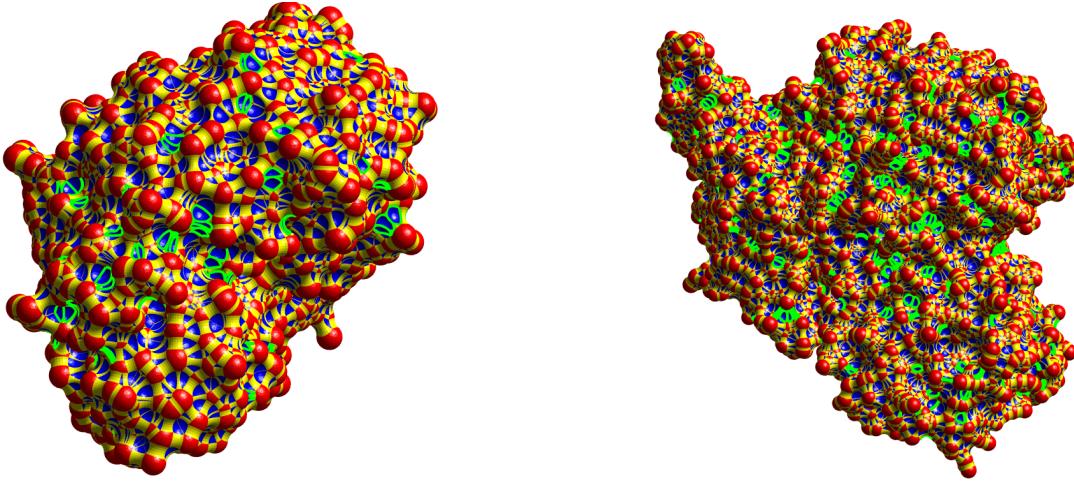


Figure 13: The eSES of molecule 4S19 with 3333 atoms (left) and the eSES of molecule IIA0 with 9606 atoms (right) where $r_p = 1.5\text{\AA}$. The green curves are the boundaries of singular concave patches.

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

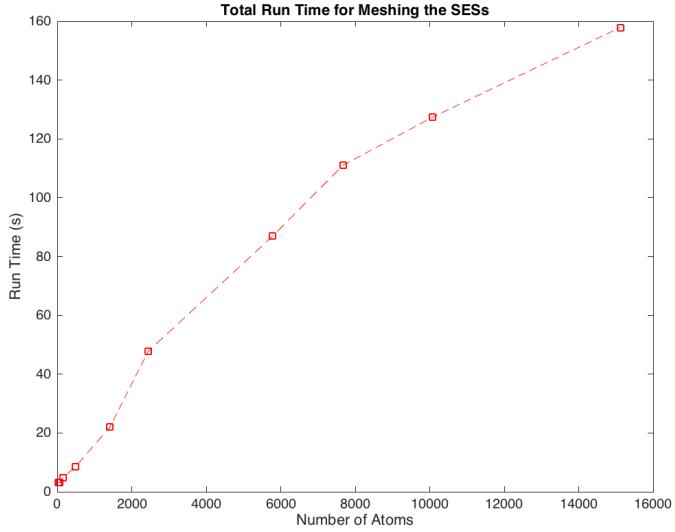


Figure 14: Total run time of the proposed algorithm for meshing the SESs of molecules with various sizes where the approximate triangle size d is set to be 0.5\AA and the probe radius $r_p = 1.5\text{\AA}$.

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

3.7.4	Thickness of the intermediate layer	122
3.7.5	Solvent dielectric constant	124
3.7.6	Discretization parameters	124
3.7.7	Robustness with respect to geometrical parameters	125
3.8	Conclusion	130

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

solvation 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 [48, Appendix C] is used for all equations.

3.1.1 Problem statement

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

$$-\nabla \cdot \varepsilon(\mathbf{x}) \nabla \psi(\mathbf{x}) = 4\pi\rho(\mathbf{x}), \quad \text{in } \mathbb{R}^3 \quad (3.1.1)$$

where it is assumed that $\psi(\mathbf{x}) \sim \mathcal{O}(\frac{1}{|\mathbf{x}|})$ as $|\mathbf{x}| \rightarrow \infty$, $\varepsilon(\mathbf{x})$ represents the space-dependent dielectric permittivity parameter and $\rho(\mathbf{x})$ 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 ρ is independent on ψ 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 $\frac{1}{|\mathbf{x}|}$ 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 Ω , the dielectric permittivity $\varepsilon(\mathbf{x})$ is defined as follows (see [26])

$$\varepsilon(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega, \\ \varepsilon_s & \mathbf{x} \in \Omega^c := \mathbb{R}^3 \setminus \overline{\Omega}, \end{cases} \quad (3.1.2)$$

where ε_s is the solvent dielectric constant and $\varepsilon(\mathbf{x})$ has a jump on the solute-solvent boundary $\Gamma := \partial\Omega$. Since the solvent is modeled as a polarizable continuum without ions, $\rho(\mathbf{x})$ 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 ε_s .

A reduced version of PCM is the conductor-like screening model (COSMO) [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

$$\begin{cases} -\Delta\psi(\mathbf{x}) = 4\pi\rho(\mathbf{x}) & \text{in } \Omega, \\ \psi(\mathbf{x}) = 0 & \text{on } \Gamma. \end{cases} \quad (3.1.3)$$

This reduced model is usually employed to approximate the PCM when the solvent dielectric constant ε_s is relatively large, for example, the (relative) dielectric constant of water is $\varepsilon_s = 78.4$ at room temperature (25°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 [96, 107] (called the ddCSOMO) has been developed, for the solute cavity constituted by a union of balls [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 [69]. Further, the convergence of the ddCOSMO has been analyzed recently in [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 [108] (called the ddPCM), which is based on the integral equation formalism of the PCM [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 [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 [70]), one starts to raise the question of a better definition of the solute cavity Ω in order to improve the methodology. The solvent excluded surface (SES, see Figure 2) cavity [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 Ω_{ses} can yield more accurate results, such as in [98, 91]. In fact, an SES-based solver for the implicit solvation model has been developed in [50, 17], using the integral equation of the PCM and an efficient mesh generator on the SES. In this chapter, we take Ω_{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 $\varepsilon(\mathbf{x})$ is also space-dependent in the solvent region:

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

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 $\varepsilon_s(\mathbf{x})$ should be approximately the (bulk) solvent dielectric constant ε_s , since the solvent molecules are influenced little by the solute molecule. Denoting by Ω_∞ the region far from the solute molecule, $\varepsilon_s(\mathbf{x})$ then satisfies

$$\varepsilon_s(\mathbf{x}) = \varepsilon_s, \quad \text{in } \Omega_\infty, \quad (3.1.5)$$

which yields that ψ of Eq. (3.1.1) is harmonic in Ω_∞ , i.e. $-\Delta\psi = 0$ in Ω_∞ .

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

$$\left\{ \begin{array}{ll} -\nabla \cdot \varepsilon \nabla u = f & \text{in } \Omega_0 := \Omega_\infty^c, \\ -\Delta u = 0 & \text{in } \Omega_\infty, \\ [u] = 0 & \text{on } \Gamma_0 := \partial\Omega_0, \\ [\partial_{\mathbf{n}} u] = 0 & \text{on } \Gamma_0, \end{array} \right.$$

where $[u]$ and $[\partial_{\mathbf{n}} u]$ denote the jump of u and its normal derivative on Γ_0 , ε is a continuous function (in particular across Γ_0). The above equation can be transformed

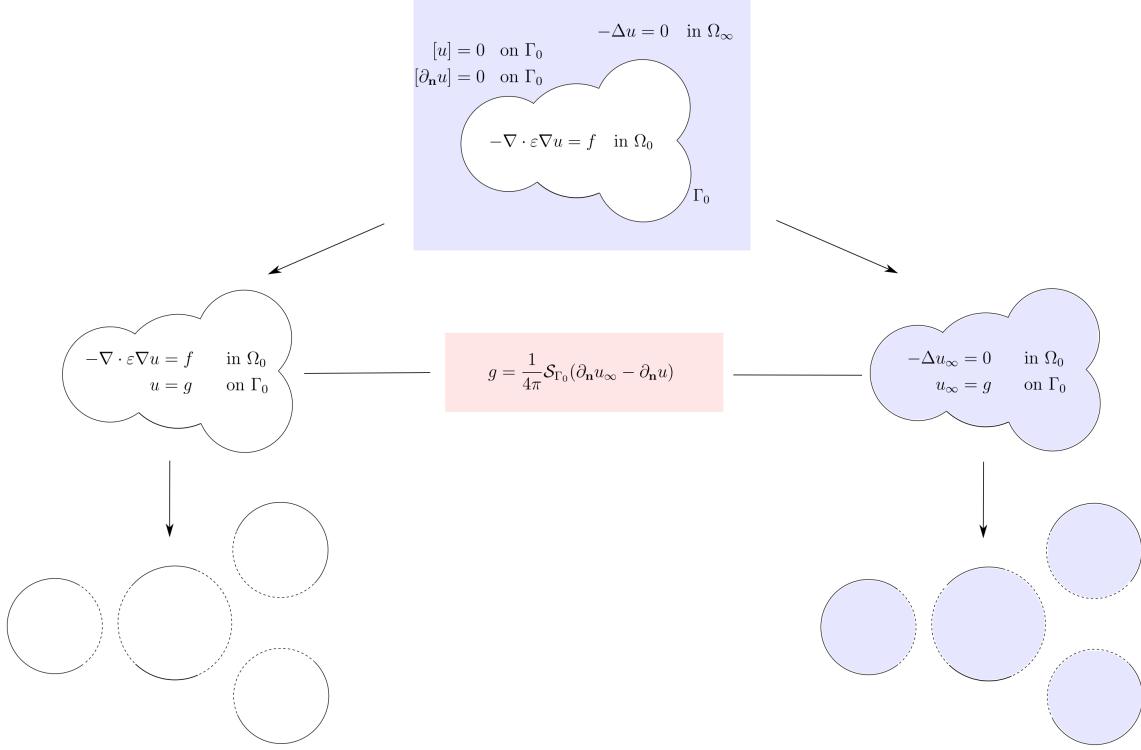


Figure 1: Schematic diagram of the domain decomposition method.

into two coupled equations defined on Ω_0 as follows

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

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

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

The operator $\mathcal{S}_{\Gamma_0} : H^{-\frac{1}{2}}(\Gamma_0) \rightarrow H^{\frac{1}{2}}(\Gamma_0)$ is the single-layer operator on Γ_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 \mathcal{S}_{Γ_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 Γ_0 , see [2]. The bounded domain Ω_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

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 $\varepsilon(\mathbf{x})$ 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.]: *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 Γ_{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 Ω_{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

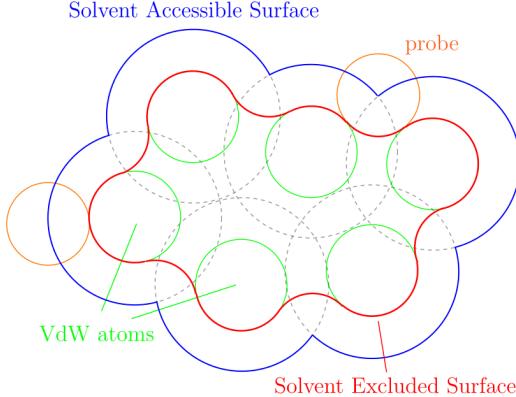


Figure 2: 2D schematic diagram of the VdW surface (green), the SAS (blue) and the SES (red).

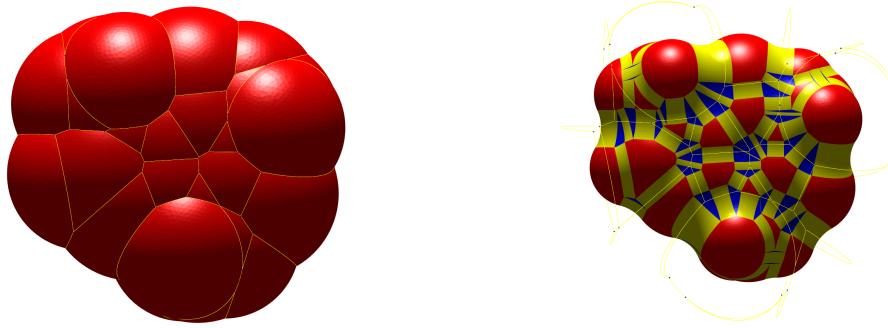


Figure 3: The SAS (left, Γ_{sas}) and the SES (right, Γ_{ses}) of caffeine. The SES is composed of convex spherical patches (red), toroidal patches (yellow) and concave spherical patches (blue).

by Γ_{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 Ω_{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 \mathbf{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 Ω_j with center \mathbf{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 [47, 13]. The SES-cavity is

completely covered by the union Ω_0 of enlarged balls, that is,

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

Here, we have specified the notation Ω_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:

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

Also, we have a characterization of their boundary surfaces

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

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 $\varepsilon(\mathbf{x})$ associated with f_{sas} as mentioned above.

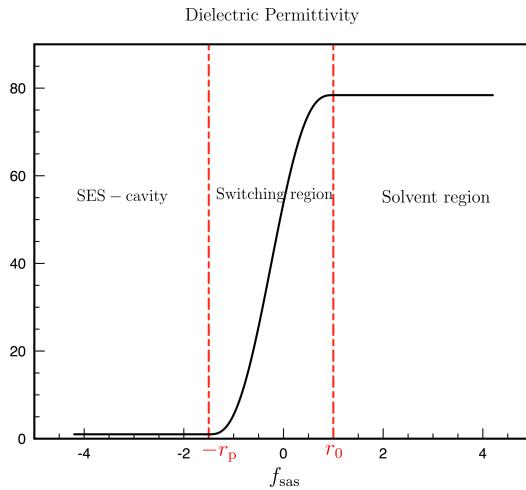


Figure 4: Schematic diagram of the dielectric permittivity function $\varepsilon(\mathbf{x})$ with respect to f_{sas} . The intermediate layer \mathcal{L} (switching region) is bounded by two dashed lines (red), i.e., the region where $-r_p \leq f_{\text{sas}} \leq r_0$.

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 Ω_0 is the solvent dielectric constant ε_s . The remaining work is to determine $\varepsilon(\mathbf{x})$ 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 $\varepsilon(\mathbf{x})$ depending on f_{sas} ,

$$\varepsilon(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in \Omega_{\text{ses}}, \\ \xi \left(\frac{f_{\text{sas}}(\mathbf{x}) + r_p}{r_p + r_0} \right) & \mathbf{x} \in \mathcal{L}, \\ \varepsilon_s & \mathbf{x} \in \Omega_\infty := \Omega_0^c, \end{cases} \quad (3.3.1)$$

where ξ is a continuous function defined on $[0, 1]$, satisfying $\xi(0) = 1$, $\xi(1) = \varepsilon_s$. $\varepsilon(\mathbf{x})$ 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 Γ_{ses} and Γ_0 , one can impose more conditions to the function ξ , for example,

$$\xi'(0) = 0, \quad \xi'(1) = 0, \quad \xi''(0) = 0, \quad \xi''(1) = 0. \quad (3.3.2)$$

Remark 3.3.1. *We emphasize that the function ξ 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

$$\chi_{\text{ses}}(t) = \begin{cases} 1 & t < 0, \\ \varepsilon_s & t \geq 0, \end{cases} \quad (3.3.3)$$

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

$$\chi_{\text{sas}}(t) = \begin{cases} 1 & t < 1, \\ \varepsilon_s & t \geq 1, \end{cases} \quad (3.3.4)$$

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

$$E^s = \frac{1}{2} \int_{\mathbb{R}^3} \rho(\mathbf{x}) W(\mathbf{x}) d\mathbf{x}, \quad (3.4.1)$$

where W denotes the (solvent) reaction potential generated by the solute's charge density ρ 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 Φ generated by the charge distribution ρ in vacuum is already known and ρ is supported in Ω_{ses} , i.e. $\text{supp}(\rho) \subset \Omega_{\text{ses}}$, which is a standard assumption in the context of PCM. As a consequence, Φ satisfies the following PDE

$$-\Delta \Phi(\mathbf{x}) = 4\pi\rho(\mathbf{x}), \quad \text{in } \mathbb{R}^3. \quad (3.4.2)$$

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

$$\rho(\mathbf{x}) = \sum_{j=1}^M q_j \delta_{\mathbf{c}_j}(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}^3, \quad (3.4.3)$$

where q_j represents the charge of the j th atom and $\delta_{\mathbf{c}_j}$ is the Dirac function at the atomic center \mathbf{c}_j . Then, the electrostatic potential Φ generated by ρ is derived

$$\Phi(\mathbf{x}) = \sum_{j=1}^M \frac{q_j}{|\mathbf{x} - \mathbf{c}_j|}, \quad \forall \mathbf{x} \in \mathbb{R}^3. \quad (3.4.4)$$

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 $\varepsilon(\mathbf{x})$ given in (3.3.1), W satisfies consequently the following PDEs

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

Further, since $\varepsilon(\mathbf{x})$ is continuous across the interfaces Γ_{ses} and Γ_0 , the following classical jump-conditions are required

$$\begin{cases} [W] = 0 & \text{on } \Gamma_{\text{ses}} \text{ and } \Gamma_0, \\ [\partial_{\mathbf{n}} W] = 0 & \text{on } \Gamma_{\text{ses}} \text{ and } \Gamma_0, \end{cases} \quad (3.4.6)$$

where $[W]$ denotes the jump (inside minus outside) of the reaction potential W on the boundary surfaces Γ_{ses} and Γ_0 , $[\partial_{\mathbf{n}} W]$ denotes the jump of the normal derivative of the reaction potential, \mathbf{n} denotes the unit normal vector on Γ_0 pointing outwards with respect to Ω_0 , $\partial_{\mathbf{n}} W$ denotes the normal derivative $\nabla W \cdot \mathbf{n}$.

3.4.2 Equivalent transformation

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

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

with

$$\begin{cases} [W] = 0 & \text{on } \Gamma_0, \\ [\partial_{\mathbf{n}} W] = 0 & \text{on } \Gamma_0. \end{cases} \quad (3.4.8)$$

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 $\tilde{\mathcal{S}}_{\Gamma_0} : H^{-\frac{1}{2}}(\Gamma_0) \rightarrow H^1(\mathbb{R}^3)$ can be used to represent the electrostatic potential W restricted in Ω_∞ as follows

$$W|_{\Omega_\infty}(\mathbf{x}) = (\tilde{\mathcal{S}}_{\Gamma_0} \sigma)(\mathbf{x}) := \int_{\Gamma_0} \frac{\sigma(\mathbf{s}')}{|\mathbf{x} - \mathbf{s}'|} d\mathbf{s}', \quad \forall \mathbf{x} \in \Omega_\infty, \quad (3.4.9)$$

where $H^{-\frac{1}{2}}(\Gamma_0)$ and $H^1(\Gamma_0)$ denote the Sobolev spaces, σ is some function in $H^{-\frac{1}{2}}(\Gamma_0)$. From the continuity of the single-layer potential across the interface, see for example [103, 49], we take the limit to Γ_0 , so as to obtain the integral equation

$$W|_{\Gamma_0}(\mathbf{s}) = (\mathcal{S}_{\Gamma_0} \sigma)(\mathbf{s}) := \int_{\Gamma_0} \frac{\sigma(\mathbf{s}')}{|\mathbf{s} - \mathbf{s}'|} d\mathbf{s}', \quad \forall \mathbf{s} \in \Gamma_0, \quad (3.4.10)$$

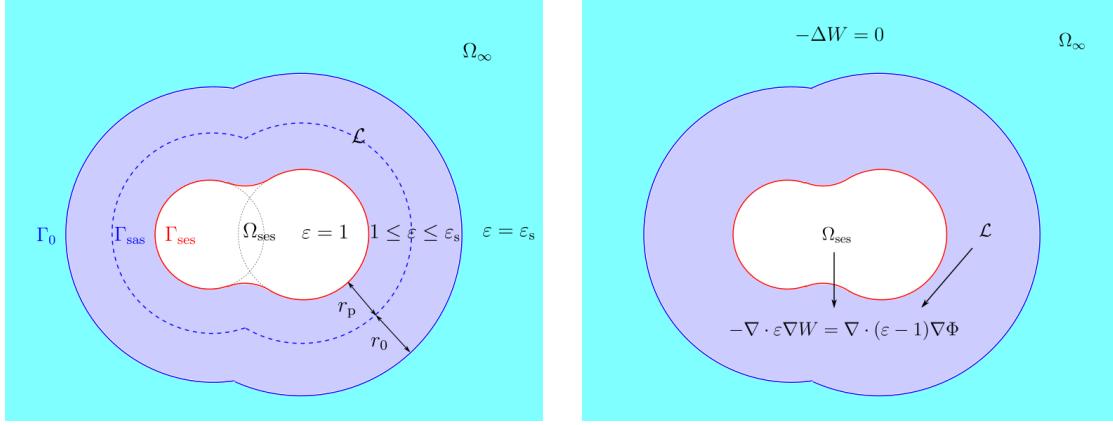


Figure 5: 2D schematic diagrams of some notations (left) and the PDEs of the PCM in different regions (right) for a system of two atoms.

where $\mathcal{S}_{\Gamma_0} : H^{-\frac{1}{2}}(\Gamma_0) \rightarrow H^{\frac{1}{2}}(\Gamma_0)$ 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 Ω_0 as follows

$$W_\infty(\mathbf{x}) := \int_{\Gamma_0} \frac{\sigma(\mathbf{s}')}{|\mathbf{x} - \mathbf{s}'|} d\mathbf{s}', \quad \forall \mathbf{x} \in \Omega_0, \quad (3.4.11)$$

satisfying

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

and

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

Recalling the relationship between the charge density σ and the jump of the normal derivative of the potential on Γ_0 generated by σ [108], we actually have

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

Combining with $[\partial_{\mathbf{n}} W] = 0$ on Γ_0 in (3.4.8), i.e., $\partial_{\mathbf{n}} W|_{\Omega_0} = \partial_{\mathbf{n}} W|_{\Omega_\infty}$, we therefore obtain

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

Also, since $[W] = 0$ in (3.4.8) and $W_\infty = W|_{\Omega_\infty}$ on Ω_0 , we have $W_\infty = W$ on Γ_0 .

Finally, let's summarize what we have in hand now. We have obtained two PDEs of W and W_∞ both defined on Ω_0

$$\begin{cases} -\nabla \cdot \varepsilon \nabla W = \nabla \cdot (\varepsilon - 1) \nabla \Phi & \text{in } \Omega_0, \\ -\Delta W_\infty = 0 & \text{in } \Omega_0, \end{cases} \quad (3.4.16)$$

and two boundary conditions coupling them

$$\begin{cases} W = W_\infty & \text{on } \Gamma_0, \\ \sigma = \frac{1}{4\pi} (\partial_{\mathbf{n}} W_\infty - \partial_{\mathbf{n}} W) & \text{on } \Gamma_0, \end{cases} \quad (3.4.17)$$

where σ is the density generating W_∞ 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 Ω_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 Γ_0 and set $k = 1$.

- [1] Solve the following Dirichlet boundary problem for W^k :

$$\begin{cases} -\nabla \cdot \varepsilon \nabla W^k = \nabla \cdot (\varepsilon - 1) \nabla \Phi & \text{in } \Omega_0, \\ W^k = g^{k-1} & \text{on } \Gamma_0, \end{cases} \quad (3.4.18)$$

and derive its Neumann boundary trace $\partial_{\mathbf{n}} W^k$ on Γ_0 .

- [2] Solve the following Dirichlet boundary problem for W_∞^k :

$$\begin{cases} -\Delta W_\infty^k = 0 & \text{in } \Omega_0, \\ W_\infty^k = g^{k-1} & \text{on } \Gamma_0, \end{cases} \quad (3.4.19)$$

and derive similarly its Neumann boundary trace $\partial_{\mathbf{n}} W_\infty^k$ on Γ_0 .

- [3] Build the charge density $\sigma^k = \frac{1}{4\pi} (\partial_{\mathbf{n}} W_\infty^k - \partial_{\mathbf{n}} W^k)$ and compute a new Dirichlet condition $g^k = \mathcal{S}_{\Gamma_0} \sigma^k$.
- [4] Compute the electrostatic contribution E_k^s to the solvation energy following (3.4.1) 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 $|E_k^s - E_{k-1}^s|$ becomes smaller than a given tolerance $\text{Tol} \ll 1$.

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 ε_s . In consequence, the electrostatic potential ψ 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:

$$g^0 = \left(\frac{1}{\varepsilon_s} - 1 \right) \Phi. \quad (3.4.20)$$

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 Ω_0 and Ω_∞ where only problems on the bounded domain Ω_0 are solved. Similar property has been obtained for the integral equation formulation of implicit solvation models, see for example [22, 26, 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 Ω_0 as follows

$$\Omega_0 = \bigcup_{j=1}^M \Omega_j, \quad \Omega_j = B_{R_j}(\mathbf{c}_j),$$

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 Ω_j :

$$\begin{cases} -\nabla \cdot \varepsilon \nabla W_j = \nabla \cdot (\varepsilon - 1) \nabla \Phi & \text{in } \Omega_j, \\ W_j = h_j & \text{on } \Gamma_j, \end{cases} \quad (3.5.1)$$

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

$$h_j = \begin{cases} W_j^N & \text{on } \Gamma_j^i, \\ g & \text{on } \Gamma_j^e. \end{cases} \quad (3.5.2)$$

Here, we omit the superscript due to the (outer) iteration index k . Γ_j^e is the external part of Γ_j not contained in any other ball Ω_i ($i \neq j$), i.e., $\Gamma_j^e = \Gamma_0 \cap \Gamma_j$, Γ_j^i is the

internal part of Γ_j , i.e., $\Gamma_j^i = \Omega_0 \cap \Gamma_j$ (see Figure 6 for an illustration), and

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

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

In the next section, we will develop a GP-solver for solving the local problems (3.5.1). For each local problem defined on Ω_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 $W|_{\Gamma_j^i}$ 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 be seen as a special case of Eq. (3.4.18) 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

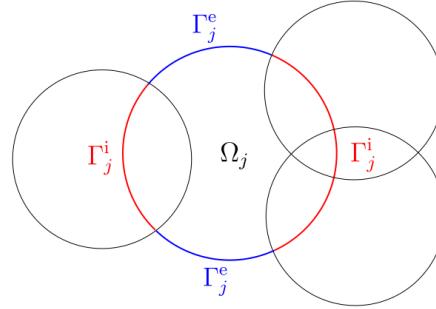


Figure 6: 2D schematic diagram of Γ_j^i (red) and Γ_j^e (blue) associated with Ω_j .

now introduce briefly the ideas. Without loss of generality, we consider the following Laplace equation with the Dirichlet boundary condition ϕ_0 and defined in the unit ball with center $\mathbf{0}$:

$$\begin{cases} -\Delta u_0 = 0 & \text{in } B_1(\mathbf{0}), \\ u_0 = \phi_0 & \text{on } \partial B_1(\mathbf{0}), \end{cases} \quad (3.6.1)$$

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

$$u_0(r, \theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} [\phi_0]_{\ell}^m r^{\ell} Y_{\ell}^m(\theta, \varphi), \quad 0 \leq r \leq 1, 0 \leq \theta \leq \pi, 0 \leq \varphi < 2\pi, \quad (3.6.2)$$

where Y_{ℓ}^m is the (orthonormal) spherical harmonic of degree ℓ and order m defined on \mathbb{S}^2 and

$$[\phi_0]_{\ell}^m = \int_{\mathbb{S}^2} \phi_0(\mathbf{s}) Y_{\ell}^m(\mathbf{s}) d\mathbf{s}$$

is the real coefficient of u_0 corresponding to the mode Y_{ℓ}^m .

To compute (3.6.2) numerically, we first approximate $[\phi_0]_{\ell}^m$ using the Lebedev quadrature rule [51] defined by the integration points $\mathbf{s}_n \in \mathbb{S}^2$ and their weights w_n^{leb} as follows

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

where N_{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 $\{Y_{\ell}^m\}_{0 \leq \ell \leq \ell_{\max}, -\ell \leq m \leq \ell}$, defined as

$$\tilde{u}_0(r, \theta, \varphi) = \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} [\tilde{\phi}_0]_{\ell}^m r^{\ell} Y_{\ell}^m(\theta, \varphi), \quad 0 \leq r \leq 1, 0 \leq \theta \leq \pi, 0 \leq \varphi < 2\pi, \quad (3.6.3)$$

where ℓ_{\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 $\ell_{\max} \rightarrow \infty$ and $N_{\text{leb}} \rightarrow \infty$. One should note however that there is no systematic manner to derive

Lebedev points of arbitrary accuracy.

Remark 3.6.1. *To compute the surface integral over the unit sphere, any other quadrature in the angular variables θ and φ 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 Ω_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 $\mathbf{0}$ in the general form of

$$\begin{cases} -\nabla \cdot \varepsilon_1(\mathbf{x}) \nabla u_1(\mathbf{x}) = f_1(\mathbf{x}) & \text{in } B_1(\mathbf{0}), \\ u_1(\mathbf{x}) = \phi_1(\mathbf{x}) & \text{on } \partial B_1(\mathbf{0}). \end{cases} \quad (3.6.4)$$

In fact, for any $j = 1, \dots, M$, 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 ε , W_j , Φ , 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 $\hat{u} \in H^1(B_1(\mathbf{0}))$, s.t.

$$\begin{cases} -\Delta \hat{u} = 0 & \text{in } B_1(\mathbf{0}), \\ \hat{u} = \phi_1 & \text{on } \partial B_1(\mathbf{0}), \end{cases} \quad (3.6.5)$$

and \hat{u}_1 can be efficiently approximated by Eq. (3.6.3). Let $v = u_1 - \hat{u}_1 \in H_0^1(B_1(\mathbf{0}))$ and in consequence, v satisfies the following PDE

$$\begin{cases} -\nabla \cdot \varepsilon_1 \nabla v = f & \text{in } B_1(\mathbf{0}), \\ v = 0 & \text{on } \partial B_1(\mathbf{0}), \end{cases} \quad (3.6.6)$$

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 $B_\delta(\mathbf{0})$, where δ is defined by

$$\delta = \frac{r_j}{r_j + r_0 + r_p} \in (0, 1).$$

Let $\mathcal{D} := B_1(\mathbf{0}) \setminus \overline{B}_\delta(\mathbf{0})$ represent the region between $\partial B_1(\mathbf{0})$ and $\partial B_\delta(\mathbf{0})$ and define the subspace $H_{0,\delta}^1(\mathcal{D})$ of the Sobolev space $H^1(\mathcal{D})$ as follows

$$H_{0,\delta}^1(\mathcal{D}) = \{w \in H^1(\mathcal{D}) : w|_{\partial B_1(\mathbf{0})} = 0\}.$$

In order to find the weak solution restricted to $H_{0,\delta}^1(\mathcal{D})$, we can write a variational formulation as: find $v \in H_{0,\delta}^1(\mathcal{D})$, s.t. $\forall w \in H_{0,\delta}^1(\mathcal{D})$,

$$\int_{\mathcal{D}} \varepsilon_1 \nabla v \cdot \nabla w + \int_{\partial B_\delta(\mathbf{0})} (\mathcal{T}v) w = \int_{\mathcal{D}} f w, \quad (3.6.7)$$

where we use the fact that $\varepsilon_1|_{\partial B_\delta(\mathbf{0})} = 1$. The operator \mathcal{T} is the Dirichlet-to-Neumann operator of the harmonic extension in $B_\delta(\mathbf{0})$, that in terms of spherical harmonics is given below.

Assume that we have an expansion of the Dirichlet boundary condition $v|_{\partial B_\delta(\mathbf{0})}$ as follows

$$v|_{\partial B_\delta(\mathbf{0})}(\delta, \theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \alpha_{\ell m} Y_\ell^m(\theta, \varphi), \quad 0 \leq \theta \leq \pi, 0 \leq \varphi < 2\pi. \quad (3.6.8)$$

Then, we can extend $v|_{\partial B_\delta(\mathbf{0})}$ harmonically from $\partial B_\delta(\mathbf{0})$ to the ball $B_\delta(\mathbf{0})$, i.e.,

$$v|_{B_\delta(\mathbf{0})}(r, \theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \alpha_{\ell m} \left(\frac{r}{\delta}\right)^\ell Y_\ell^m(\theta, \varphi), \quad 0 \leq r \leq \delta, 0 \leq \theta \leq \pi, 0 \leq \varphi < 2\pi. \quad (3.6.9)$$

Denote by \mathbf{n}_δ the unit normal vector on the sphere $\partial B_\delta(\mathbf{0})$ pointing outwards with respect to the ball $B_\delta(\mathbf{0})$. As a consequence, we can compute $\partial_{\mathbf{n}_\delta} v = \nabla v \cdot \mathbf{n}_\delta$ consisting of the normal derivative of v on $\partial B_\delta(\mathbf{0})$:

$$\mathcal{T}v(\delta, \theta, \varphi) := \partial_{\mathbf{n}_\delta} v(\delta, \theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \alpha_{\ell m} \left(\frac{\ell}{\delta}\right) Y_\ell^m(\theta, \varphi), \quad 0 \leq \theta \leq \pi, 0 \leq \varphi < 2\pi. \quad (3.6.10)$$

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 \mathcal{T} .

3.6.2.2 Galerkin discretization

In order to find basis functions belonging to $H_{0,\delta}^1(\mathcal{D})$, we first introduce the radial functions

$$\varphi_i(r) = (1-r)L'_i\left(\frac{2(r-\delta)}{1-\delta}-1\right),$$

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 $\{\varphi_i(r)Y_\ell^m(\theta, \varphi)\}$ with $1 \leq i \leq N$, $0 \leq \ell \leq \ell_{\max}$ and $-\ell \leq m \leq \ell$, where N denotes the maximum degree of Legendre polynomials and ℓ_{\max} denotes the maximum degree of spherical harmonics as in Section 3.6.1. The spanned space of these functions is denoted by $\mathcal{V}_{N,\ell_{\max}}(\mathcal{D})$ which is defined as follows

$$\mathcal{V}_{N,\ell_{\max}}(\mathcal{D}) = \text{span}\{\varphi_i(r)Y_\ell^m(\theta, \varphi) \mid 1 \leq i \leq N, 0 \leq \ell \leq \ell_{\max}, -\ell \leq m \leq \ell\} \subset H_{0,\delta}^1(\mathcal{D}).$$

Then, we consider a Galerkin discretization of the variational formulation (3.6.7) that reads: find $\tilde{v} \in \mathcal{V}_{N,\ell_{\max}}(\mathcal{D})$, such that

$$\forall \tilde{w} \in \mathcal{V}_{N,\ell_{\max}}(\mathcal{D}) : \quad \int_{\mathcal{D}} \varepsilon_1 \nabla \tilde{v} \cdot \nabla \tilde{w} + \int_{\partial B_\delta(\mathbf{0})} (\mathcal{T} \tilde{v}) \tilde{w} = \int_{\mathcal{D}} f \tilde{w}. \quad (3.6.11)$$

Since $\tilde{v} \in \mathcal{V}_{N,\ell_{\max}}(\mathcal{D})$, we can write \tilde{v} in the form of

$$\tilde{v}(r, \theta, \varphi) = \sum_{i=0}^N \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} v_{i\ell m} \varphi_i(r) Y_\ell^m(\theta, \varphi), \quad \forall \delta \leq r \leq 1, 0 \leq \theta \leq \pi, 0 \leq \varphi < 2\pi, \quad (3.6.12)$$

and we consequently have

$$\mathcal{T} \tilde{v}|_{B_\delta(\mathbf{0})}(\delta, \theta, \varphi) = \sum_{i=0}^N \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} v_{i\ell m} \left(\frac{\ell}{\delta}\right) \varphi_i(\delta) Y_\ell^m(\theta, \varphi). \quad (3.6.13)$$

By substituting (3.6.12)–(3.6.13) into (3.6.11) and taking the test function $\tilde{w} = \varphi_{i'}(r)Y_{\ell'}^{m'}(\theta, \varphi)$, we then obtain a system of linear equations: $\forall 1 \leq i' \leq N$, $0 \leq \ell' \leq \ell_{\max}$, $-\ell' \leq m' \leq \ell'$,

$$\sum_{i=0}^N \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} v_{i\ell m} \left(\int_{\mathcal{D}} \varepsilon_1 \nabla (\varphi_i Y_\ell^m) \cdot \nabla (\varphi_{i'} Y_{\ell'}^{m'}) + \frac{\ell}{\delta} \int_{\partial B_\delta(\mathbf{0})} \varphi_i Y_\ell^m \varphi_{i'} Y_{\ell'}^{m'} \right) = \int_{\mathcal{D}} f \varphi_{i'} Y_{\ell'}^{m'}. \quad (3.6.14)$$

In order to write the corresponding system of linear equations, we define an index

$$k = N(\ell^2 + m + \ell) + i \in \{1, 2, \dots, N(\ell_{\max} + 1)^2\}$$

which corresponds to the triple (i, ℓ, m) through a one-to-one mapping between k

and (i, ℓ, m) . Assume that k corresponds to (i, ℓ, m) and k' corresponds to (i', ℓ', m') . Then, we can recast the set of equations (3.6.14) as a linear system of the form

$$\mathbf{A}X = F. \quad (3.6.15)$$

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

$$(\mathbf{A})_{k'k} = \int_{\mathcal{D}} \varepsilon_1 \nabla (\varphi_i Y_\ell^m) \cdot \nabla (\varphi_{i'} Y_{\ell'}^{m'}) + \frac{\ell}{\delta} \int_{\partial B_\delta(\mathbf{0})} \varphi_i Y_\ell^m \varphi_{i'} Y_{\ell'}^{m'}, \quad (3.6.16)$$

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

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

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

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

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 $\tilde{v}(r, \theta, \varphi) \in \mathcal{V}_{N, \ell_{\max}}(\mathcal{D})$ according to Eq. (3.6.12). Considering that v is harmonic in $B_\delta(\mathbf{0})$, \tilde{v} can then be extended harmonically in the ball $B_\delta(\mathbf{0})$ following (3.6.9). Therefore, we obtain an approximate solution defined on $B_1(\mathbf{0})$ 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. (3.6.15) for each subdomain Ω_j a priori, since each \mathbf{A} associated with Ω_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. (3.6.16) and Eq. (3.6.18) need to be further computed. We start by observing that the second term in Eq. (3.6.16) can be simplified

$$\frac{\ell}{\delta} \int_{\partial B_\delta(\mathbf{0})} \varphi_i Y_\ell^m \varphi_{i'} Y_{\ell'}^{m'} = \ell \delta \varphi_i(\delta) \varphi_{i'}(\delta) \int_{\mathbb{S}^2} Y_\ell^m Y_{\ell'}^{m'} = \ell \delta \varphi_i(\delta) \varphi_{i'}(\delta) \delta_{\ell\ell'} \delta_{mm'}, \quad (3.6.19)$$

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. (3.6.16) and the integral in Eq. (3.6.18).

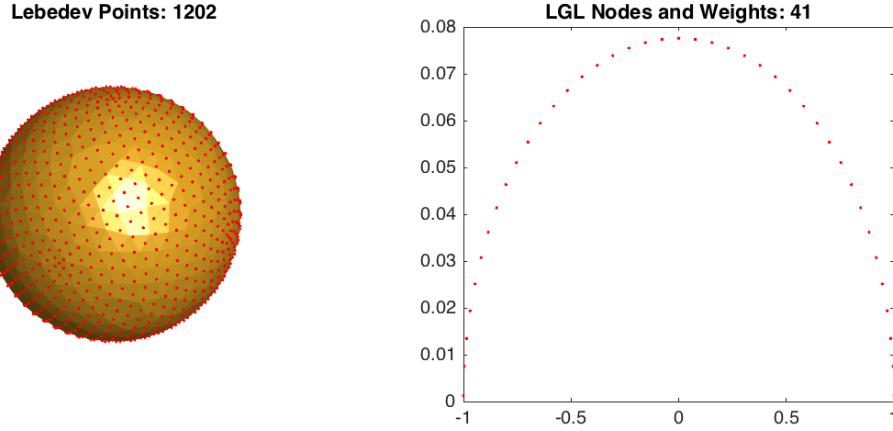


Figure 7: Schematic diagram of the Lebedev points (left) and the LGL points as well as their weights (right).

The integral over \mathcal{D} can be divided into two parts: the radial part and spherical part, that is to say, for any given function $h \in L^1(B_1(\mathbf{0}))$, we can compute its integral over \mathcal{D} separately as below

$$\int_{\mathcal{D}} h(\mathbf{x}) d\mathbf{x} = \int_{\delta}^1 r^2 \int_{\mathbb{S}^2} h(r, \mathbf{s}) d\mathbf{s} dr, \quad (3.6.20)$$

where $\mathbf{s} \in \mathbb{S}^2$ and $\mathbf{x} = r\mathbf{s}$. 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 [88] defined by the integration points $x_m \in [-1, 1]$ and their weights w_m^{lgl} , $1 \leq m \leq N_{\text{lgl}}$, where N_{lgl} represents the number of LGL points. Using the change of variable $r = \frac{1-\delta}{2}(x+1)+\delta$, $x \in [-1, 1]$, we approximate the integral by the following quadrature rule

$$\begin{aligned} \int_{\mathcal{D}} h(\mathbf{x}) d\mathbf{x} &= \int_{\delta}^1 r^2 \int_{\mathbb{S}^2} h(r, \mathbf{s}) d\mathbf{s} dr \\ &= \int_{-1}^1 \frac{1-\delta}{2} \left(\frac{1-\delta}{2}(x+1) + \delta \right)^2 \int_{\mathbb{S}^2} h \left(\frac{1-\delta}{2}(x+1) + \delta, \mathbf{s} \right) d\mathbf{s} dx \\ &\approx \frac{1-\delta}{2} \sum_{m=1}^{N_{\text{lgl}}} \sum_{n=1}^{N_{\text{leb}}} w_m^{\text{lgl}} w_n^{\text{leb}} \left(\frac{1-\delta}{2}(x_m+1) + \delta \right)^2 h \left(\frac{1-\delta}{2}(x_m+1) + \delta, \mathbf{s}_n \right). \end{aligned} \quad (3.6.21)$$

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.

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 ξ in Eq. (3.3.1) as follows

$$\xi(t) = 1 + (\varepsilon_s - 1)t^3(10 + 3t(-5 + 2t)), \quad 0 \leq t \leq 1, \quad (3.7.1)$$

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 ε 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

$$\varepsilon(r) = \begin{cases} 1 & \text{if } 0 \leq r \leq r_1, \\ \xi \left(\frac{r - r_1}{r_p + r_0} \right) & \text{if } r_1 \leq r \leq 1, \\ \varepsilon_s & \text{if } r \geq 1, \end{cases} \quad (3.7.2)$$

where $\varepsilon_s = 10$. Assuming that the sphere carries a point charge 1 at its center, i.e., $q_1 = 1$, we take $\Phi = \frac{1}{r}$, $f_1 = -\frac{\varepsilon'(r)}{r^2}$ and $\phi_1 = -1$ in Eq. (3.6.4). 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 $N_{\text{LGL}} = 200$. 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.

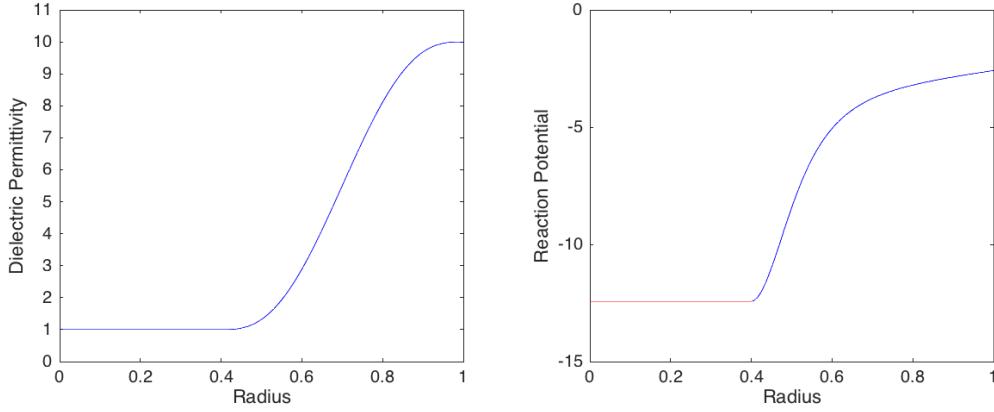


Figure 8: The left figure illustrates the dielectric permittivity function $\varepsilon(r)$; the right plots the numerical solution to Eq. (3.6.4). Only the blue part is subject to the computational domain, the red part of the curve depicts the harmonic extension.

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 $\ell_{\max} = 11$, 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 $r_p = 1.5\text{\AA}$ and $r_0 = 1\text{\AA}$. 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 $e^2/\text{\AA} = 332.06364 \text{ kcal/mol}$ (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

$$\text{Error}(N_{\text{it}}) = |E_{N_{\text{it}}}^s - E_\infty^s|, \quad (3.7.3)$$

where $E_{N_{\text{it}}}^s$ is the electrostatic contribution computed at the N_{it} -th outer iteration in

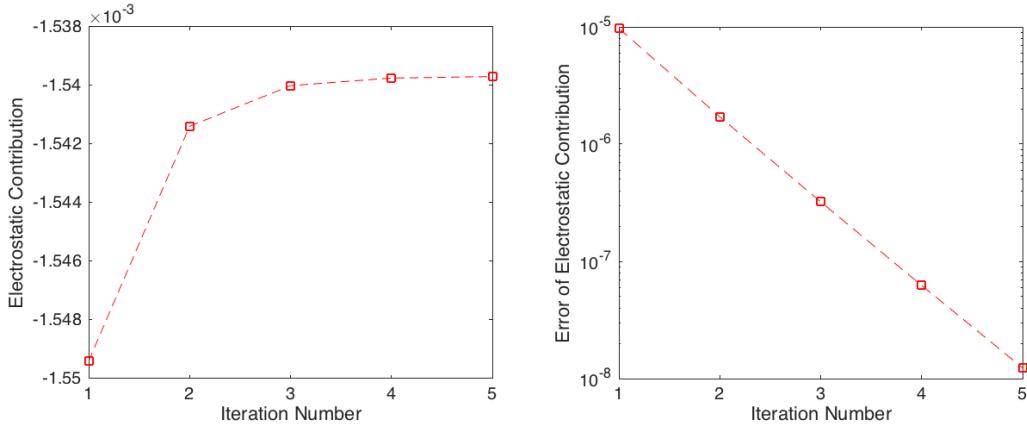


Figure 9: On the left, the curve illustrates the computed electrostatic contribution to the solvation energy of formaldehyde with respect to N_{it} ; on the right, the curve illustrates the error of the electrostatic contribution with respect to N_{it} . The following parameters are used: $\ell_{\max}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

Section 3.4.3 and the “exact” electrostatic contribution E_∞^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 $\ell_{\max} = 9$, $N_{leb} = 350$, $N = 15$, $N_{lgl} = 30$, $\varepsilon_s = 78.4$, $r_p = 1.5\text{\AA}$, $r_0 = 1\text{\AA}$. 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_∞^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 Ω_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., ∇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

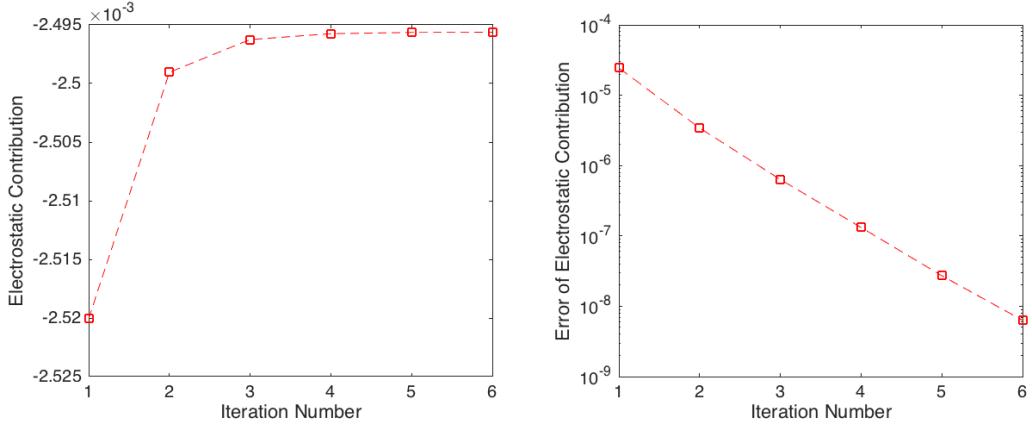


Figure 10: On the left, the curve illustrates the numerical electrostatic contribution to the solvation energy of caffeine with respect to N_{it} ; on the right, the curve illustrates the error of the electrostatic contribution with respect to N_{it} . The following parameters are used: $\ell_{\max}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

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 $r_0 = 0\text{\AA}$, implying that Γ_0 and Γ_{sas} coincide. In the ddPCM algorithm presented in [108], the dielectric permittivity is discontinuous across the solute-solvent boundary and jumps from 1 to ε_s . In the following test, we take Γ_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 \mathcal{L} tends to zero, i.e., $r_p \rightarrow 0$. This implies that both, Γ_{ses} and Γ_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 $r_p = 2^{-n}\text{\AA}$. 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.

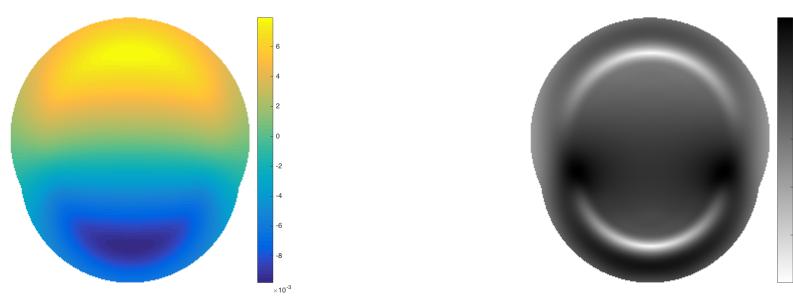


Figure 11: The reaction potential (left) and the magnitude of the corresponding reaction field (right) in the YZ-plane for the hydrogen-fluoride molecule with two atoms, with the parameters $\ell_{\max} = 11$, $N_{\text{leb}} = 1202$, $N = 15$, $N_{\text{lgl}} = 50$, $\varepsilon_s = 2$, $r_p = 1.5\text{\AA}$, $r_0 = 1\text{\AA}$. The colorbars represent respectively the reaction potential value and the magnitude of the reaction field.

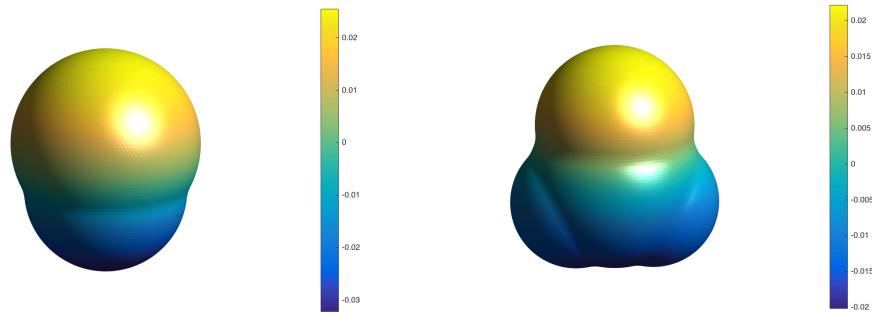


Figure 12: Reaction potential of hydrogen-fluoride and formaldehyde on the SES, both computed with the parameters $\ell_{\max} = 11$, $N_{\text{leb}} = 1202$, $N = 15$, $N_{\text{lgl}} = 50$, $\varepsilon_s = 78.4$, $r_p = 1.5\text{\AA}$, $r_0 = 1\text{\AA}$.

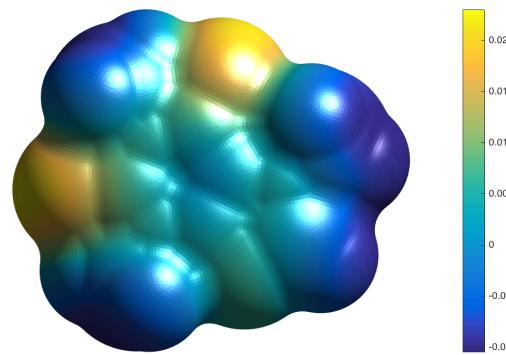


Figure 13: Reaction potential of caffeine on the SES, computed with the parameters $\ell_{\max} = 9$, $N_{\text{leb}} = 350$, $N = 15$, $N_{\text{lgl}} = 30$, $\varepsilon_s = 78.4$, $r_p = 1.5\text{\AA}$, $r_0 = 1\text{\AA}$.

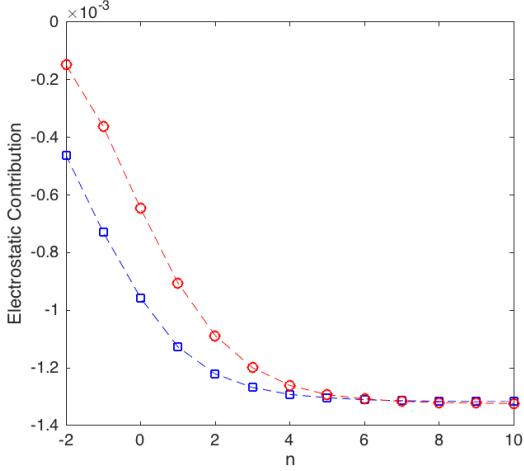


Figure 14: The blue curve plots the electrostatic contribution of hydrogen-fluoride with respect to n (the layer thickness 2^{-n}\AA), computed from the proposed algorithm with the parameters $\ell_{\max} = 11$, $N_{\text{leb}} = 1202$, $N = 15$, $N_{\text{lgl}} = 50$, $\varepsilon_s = 2$; the red curve plots the results computed from the ddPCM algorithm with the same parameters of spherical harmonics $\ell_{\max} = 11$, $N_{\text{leb}} = 1202$ 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 $\varepsilon_s = 2^k$, $k = 1, 2, \dots, 15$. We observe that the numerical electrostatic contribution to the solvation energy varies smoothly and converges to some quantity for increasing solvent permittivities ε_s . Another interesting observation is that when ε_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 ε_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_{exact}^s , with large discretization parameters $\ell_{\max} = 20$, $N_{\text{leb}} = 4334$, $N = 25$, $N_{\text{lgl}} = 50$ and the other parameters $r_p = 1.5\text{\AA}$, $r_0 = 1\text{\AA}$. We treat E_{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 ℓ_{\max} of spherical harmonics and to the number N_{leb} of Lebedev points in the dichloromethane

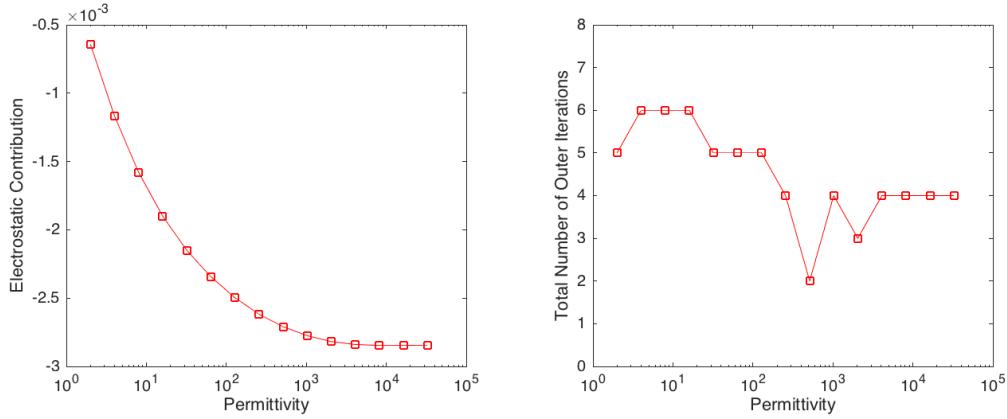


Figure 15: The left figure shows the electrostatic contribution to the solvation energy of hydrogen-fluoride when the solvent dielectric constant ϵ_s increases; the right figure plots the total number of outer iterations to reach the stopping criterion To1 with respect to ϵ_s . The following parameters are used: $\ell_{\max} = 11$, $N_{\text{leb}} = 1202$, $N = 20$, $N_{\text{lgl}} = 50$, $r_p = 1.5\text{\AA}$, $r_0 = 1\text{\AA}$.

solvent. On the left of Figure 16, we report the electrostatic contribution with respect to ℓ_{\max} which varies from 3 to 15. Further, on the right of Figure 16, we report the electrostatic contribution with respect to N_{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_{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_{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 ℓ_{\max} , N_{leb} , N , N_{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

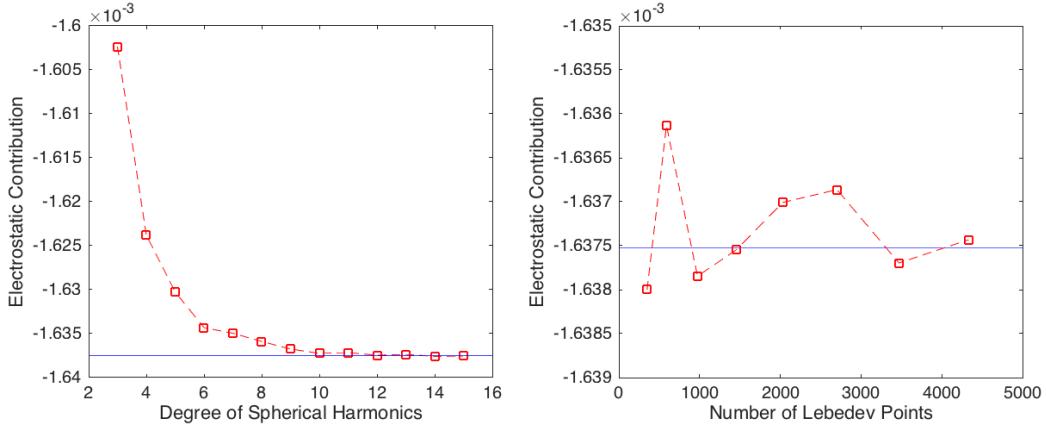


Figure 16: The left figure plots the electrostatic contribution to the solvation energy of hydrogen-fluoride with respect to ℓ_{\max} when N_{leb} is set to 1454. The right figure plots the electrostatic contribution with respect to N_{leb} when ℓ_{\max} is set to 15. In both figures, the blue line represents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

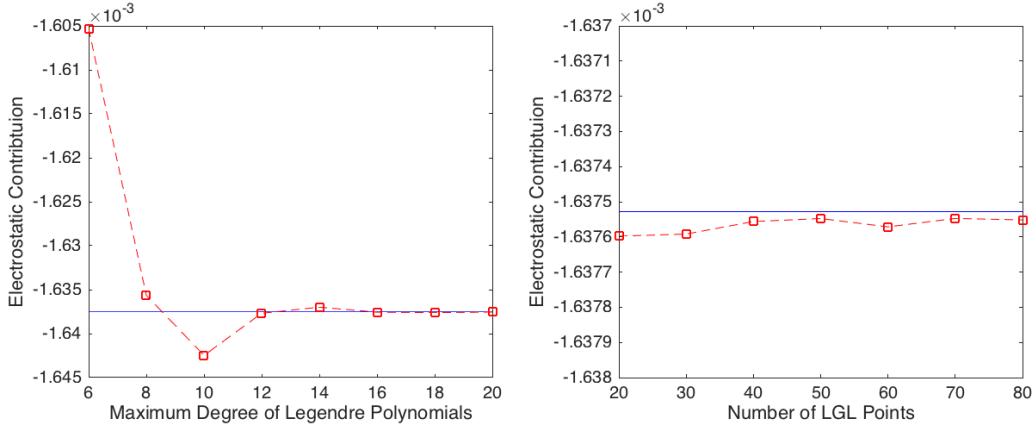


Figure 17: The left figure plots the electrostatic contribution to the solvation energy of hydrogen-fluoride with respect to N when N_{lgl} is set to 50. The right figure plots the electrostatic contribution with respect to N_{lgl} when N is set to 20. In both figures, the blue line represents the “exact” electrostatic contribution E_{exact}^s . In addition, the following parameters are used: $\ell_{\max} = 15$, $N_{\text{leb}} = 1454$, $\varepsilon_s = 8.93$, $r_p = 1.5\text{\AA}$, $r_0 = 1\text{\AA}$.

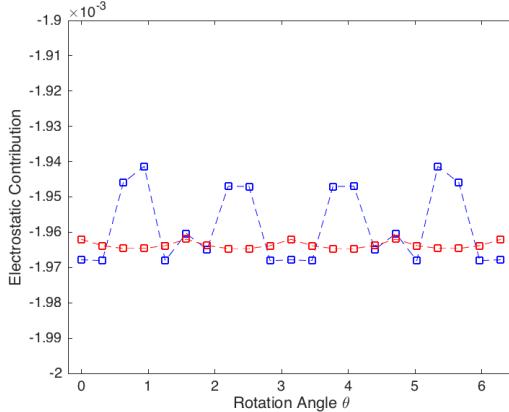


Figure 18: The variation of the electrostatic contribution to the solvation energy of hydrogen-fluoride with respect to the angle of rotating the fluoride atom around the hydrogen atom. The blue curve corresponds to the parameters $\ell_{\max} = 7$, $N_{\text{leb}} = 86$, $N = 10$, $N_{\text{lgl}} = 20$, the red curve corresponds to the parameters $\ell_{\max} = 11$, $N_{\text{leb}} = 1202$, $N = 20$, $N_{\text{lgl}} = 40$. Furthermore, the following parameters are used: $\varepsilon_s = 8.93$, $r_p = 1.5\text{\AA}$, $r_0 = 1\text{\AA}$.

1.5% and 0.25% respectively.

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\AA to 9.36\AA 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\AA to 9.37\AA .

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\AA to 1\AA . 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.

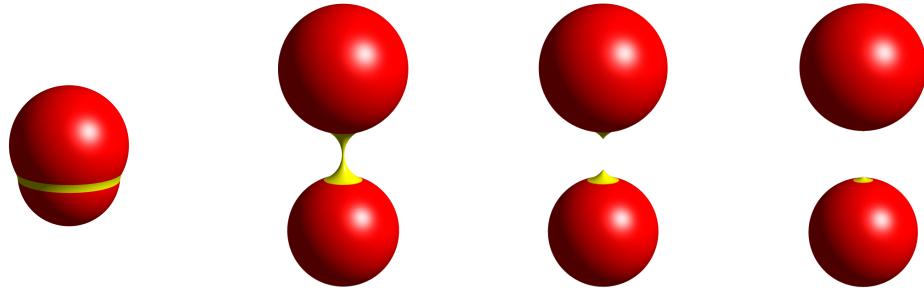


Figure 19: Topological change of the SES of hydrogen-fluoride when the distance between the two atomic centers increases. The most left figure illustrates the SES of hydrogen-fluoride in equilibrium and the three figures from the left to right corresponds to the distances between the centers increased respectively to 8.96Å, 9.16Å and 9.36Å.

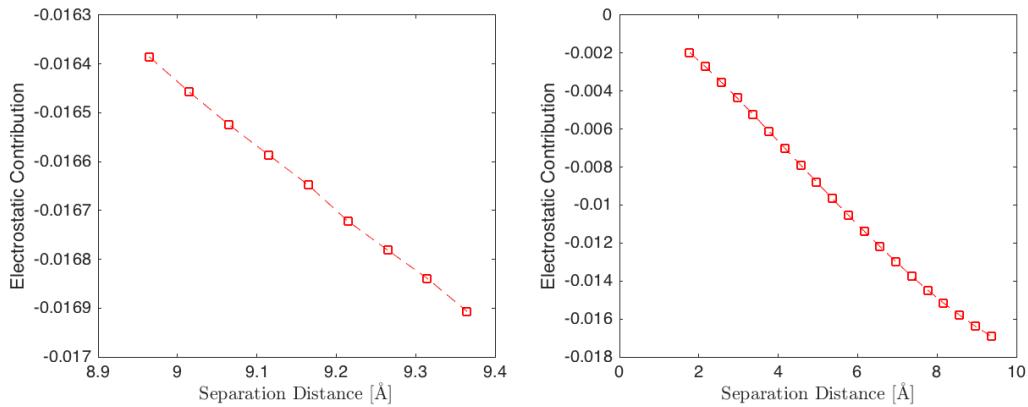


Figure 20: Electrostatic contribution of hydrogen-fluoride with respect to the separation distance between the atomic centers with parameters $\ell_{\max} = 11$, $N_{\text{leb}} = 1202$, $N = 20$, $N_{\text{lgI}} = 40$, $\varepsilon_s = 8.93$, $r_p = 1.5\text{\AA}$, $r_0 = 0\text{\AA}$. On the left, the separation distance varies between 8.96Å and 9.36Å where the topology of the SES changes as showed in Figure 19. On the right, the separation distance varies in a larger range.

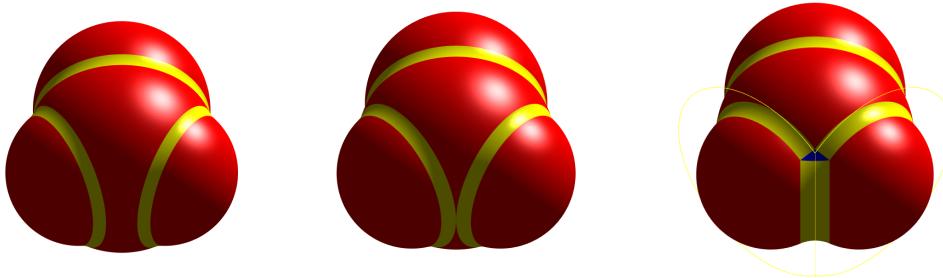


Figure 21: Different SESs of formaldehyde when the two hydrogen atoms displace downwards. The left figure illustrates the SES of formaldehyde in equilibrium; the middle figure illustrates the SES when concave patches are about to appear; the right figure illustrates the SES when the concave patches (in blue) have appeared.

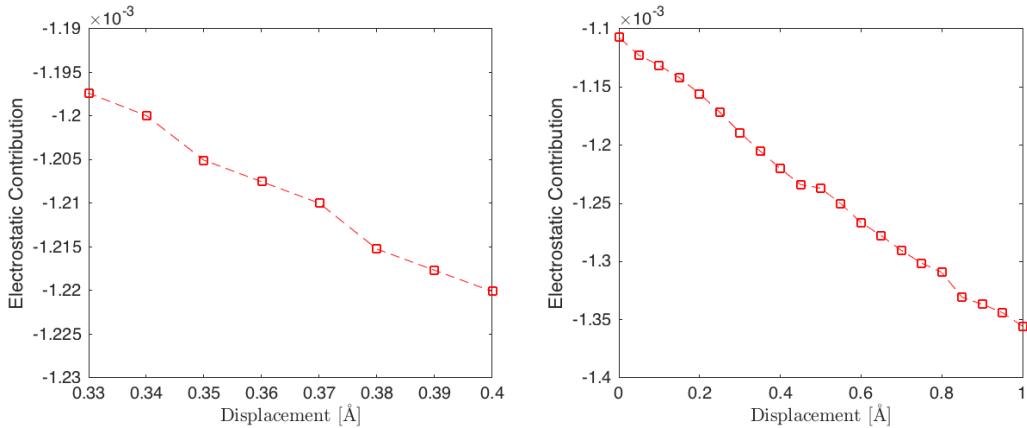


Figure 22: Electrostatic contribution to the solvation energy of formaldehyde with respect to the displacement of the two hydrogen atoms with parameters $\ell_{\max} = 11$, $N_{\text{leb}} = 1454$, $N = 15$, $N_{\text{lgl}} = 50$, $\varepsilon_s = 8.93$, $r_p = 1.5\text{\AA}$, $r_0 = 1\text{\AA}$. On 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 0\AA to 1\AA .

3.8 Conclusion

We have proposed a construction to deal with a continuous permittivity function $\varepsilon(\mathbf{x})$ 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 Ω_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

4.1	Introduction	132
4.1.1	Previous work	132
4.1.2	Contribution	134
4.1.3	Outline	136
4.2	PB solvation model	137
4.2.1	Solute-solvent boundary	137
4.2.2	Poisson-Boltzmann equation	137
4.3	Problem transformation	139
4.3.1	Problem setting	139
4.3.2	Necessary tools from the potential theory	140
4.3.3	Transformation	141
4.4	Strategy	142
4.4.1	Global strategy	143
4.4.2	Domain decomposition scheme	143
4.5	Single-domain solvers	145
4.5.1	Laplace solver	145
4.5.2	HSP solver	146
4.6	Global linear system	148
4.6.1	Reformulation of the coupling conditions	148

4.6.2	Derivation of the linear system	150
4.7	Numerical results	154
4.7.1	Discretization parameters	154
4.7.2	Varying the Debye-Hückel screening constant	155
4.7.3	Varying the solvent radius	156
4.7.4	Different linear algebra algorithms	157
4.7.5	Graphical illustration	158
4.8	Conclusion	158

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 solvation 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 [116, 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 [48] 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 ψ of the PB solvation model in the following form (see [86])

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

where $\varepsilon(\mathbf{x})$ denotes the space-dependent dielectric permittivity function, $\bar{\kappa}(\mathbf{x})$ the modified Debye-Hückel parameter, $\rho_M(\mathbf{x})$ denotes the known solute's charge distribution. Usually, $\varepsilon(\mathbf{x})$ has the following form

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

where ε_1 and ε_2 are respectively the solute dielectric permittivity and the solvent dielectric permittivity, Ω and Ω^c represents respectively the solute cavity and the solvent region. In addition, $\bar{\kappa}(\mathbf{x})$ has the form

$$\bar{\kappa}(\mathbf{x}) = \begin{cases} 0 & \text{in } \Omega, \\ \sqrt{\varepsilon_2} \kappa & \text{in } \Omega^c, \end{cases} \quad (4.1.3)$$

where κ is the Debye-Hückel screening constant. Details on the solute cavity Ω , the nonlinear PB equation and its linearization will be presented in Section 4.2.

4.1.1.1 Domain decomposition methods for implicit solvation models

The polarizable continuum model (PCM) [21, 109, 82] is a popular implicit solvation 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., $\bar{\kappa}(\mathbf{x}) \equiv 0$ 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 [67, 99]. 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 [95], 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 partition 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 relies 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 Ω and a homogeneous screened Poisson (HSP) equation defined in the unbounded solvent region Ω^c , which are coupled by some boundary conditions on Γ .

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

$$-\Delta\psi_r = 0, \quad \text{in } \Omega, \quad (4.1.4)$$

where ψ_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 ψ_e satisfying the following HSP equation defined now in Ω (an interior Dirichlet problem)

$$-\Delta\psi_e(\mathbf{x}) + \kappa^2\psi_e(\mathbf{x}) = 0, \quad \text{in } \Omega. \quad (4.1.5)$$

Based on the classical jump-conditions of ψ 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

$$g = \mathcal{S}_\kappa \left(\partial_{\mathbf{n}}\psi_e - \frac{\varepsilon_1}{\varepsilon_2} \partial_{\mathbf{n}}(\psi_0 + \psi_r) \right), \quad \text{on } \Gamma, \quad (4.1.6)$$

where $\mathcal{S}_\kappa : H^{-\frac{1}{2}}(\Gamma) \rightarrow H^{\frac{1}{2}}(\Gamma)$ denotes the single-layer operator on Γ (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 Γ , 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 [66] 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.

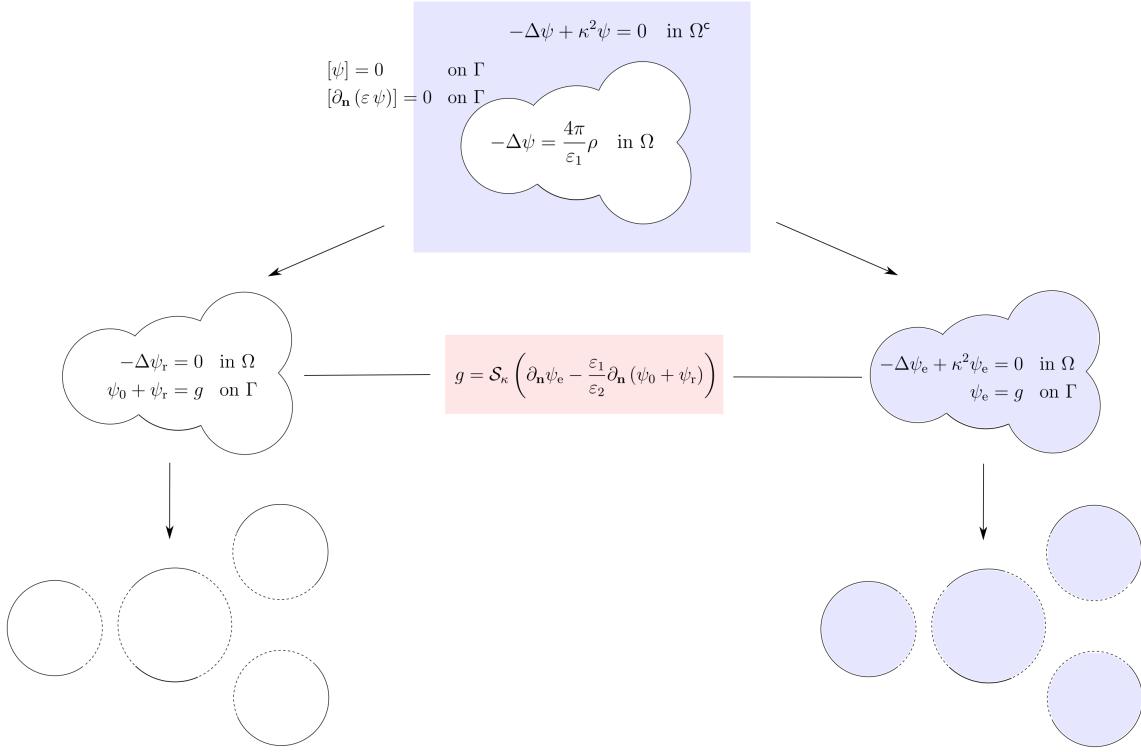


Figure 1: Schematic diagram of the domain decomposition method for solving the linearized Poisson-Boltzmann equation.

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

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

where c_i^∞ 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 ψ of a general implicit solvation model is characterized originally by the following Poisson equation

$$-\nabla \cdot \epsilon(\mathbf{x}) \nabla \psi(\mathbf{x}) = 4\pi \rho(\mathbf{x}), \quad \text{in } \mathbb{R}^3, \quad (4.2.2)$$

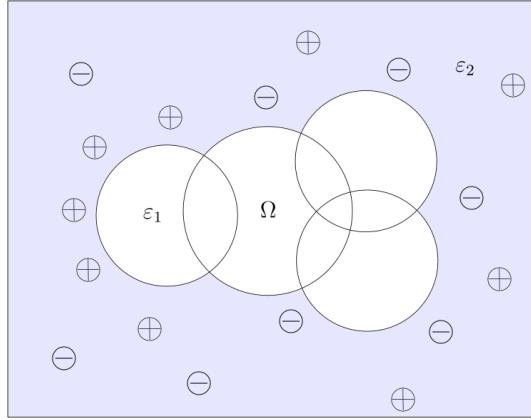


Figure 2: 2D schematic diagram of the implicit solvation model with ionic solutions, i.e., the PB solvation model.

where $\psi(\mathbf{x}) \sim \frac{1}{|\mathbf{x}|}$ as $|\mathbf{x}| \rightarrow \infty$. 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 ρ_M and the ionic distribution c_i in (4.2.1), we then derive the Poisson-Boltzmann equation as follows (see [40])

$$-\nabla \cdot [\varepsilon(\mathbf{x}) \nabla \psi(\mathbf{x})] = 4\pi \rho_M(\mathbf{x}) + \sum_i z_i e c_i^\infty e^{\frac{-z_i e \psi(\mathbf{x})}{k_B T}} \chi_{\Omega^c}(\mathbf{x}), \quad (4.2.3)$$

where $z_i e$ is the charge of the i -th type of ion, e is the elementary charge and χ_{Ω^c} is the characteristic function of the solvent region Ω^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 ψ satisfies the low potential condition, i.e., $\left| \frac{e\psi}{k_B T} \right| \ll 1$, the nonlinear Poisson-Boltzmann equation (4.2.3) can be linearized to (see [86] for this form)

$$-\nabla \cdot [\varepsilon(\mathbf{x}) \nabla \psi(\mathbf{x})] + \bar{\kappa}(\mathbf{x})^2 \psi(\mathbf{x}) = 4\pi \rho_M(\mathbf{x}), \quad (4.2.4)$$

where ψ 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 (4.1.2) of the dielectric permittivity function $\varepsilon(\mathbf{x})$, the solute (relative) dielectric permittivity ε_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 [86] and [63, Section 1.4], the authors claim to obtain better approximations with the empirical value $\varepsilon_1 = 2$. The solvent dielectric permittivity ε_2 is determined by the solvent as well as the temperature, for example, $\varepsilon_2 = 78.4$ for water at room

temperature 25°C. The modified Debye-Hückel parameter in the implicit solvation model with a 1 : 1 electrolyte is taken as

$$\bar{\kappa}(\mathbf{x}) = \begin{cases} 0 & \text{in } \Omega, \\ \sqrt{\varepsilon_2} \kappa & \text{in } \Omega^c, \end{cases} \quad (4.2.5)$$

where κ 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)

$$\kappa^2 = \frac{8\pi e^2 N_A I}{1000 \varepsilon_2 k_B T}, \quad (4.2.6)$$

where N_A is the Avogadro constant.

Furthermore, it is usually assumed that the solute's charge distribution ρ_M is supported in Ω . Here, we assume that ρ_M is given by the sum of M point charges in the following form

$$\rho_M(\mathbf{x}) = \sum_{i=1}^M q_i \delta(\mathbf{x} - \mathbf{x}_i), \quad (4.2.7)$$

where M is the number of solute atoms, q_i represents the (partial) charge carried on the i th atom with center \mathbf{x}_i , δ is the Dirac delta function. As a consequence, ρ_M is supported in Ω 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

$$\begin{cases} -\Delta\psi(\mathbf{x}) = \frac{4\pi}{\varepsilon_1} \rho_M(\mathbf{x}) & \text{in } \Omega, \\ -\Delta\psi(\mathbf{x}) + \kappa^2\psi(\mathbf{x}) = 0 & \text{in } \Omega^c, \end{cases} \quad (4.3.1)$$

with two classical jump-conditions

$$\begin{cases} [\psi] = 0 & \text{on } \Gamma, \\ [\partial_{\mathbf{n}}(\varepsilon\psi)] = 0 & \text{on } \Gamma, \end{cases} \quad (4.3.2)$$

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

$$G(\mathbf{x}, \mathbf{y}) = \frac{1}{4\pi|\mathbf{x} - \mathbf{y}|}, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^3, \quad (4.3.3)$$

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

$$G_{\kappa}(\mathbf{x}, \mathbf{y}) = \frac{e^{-\kappa|\mathbf{x} - \mathbf{y}|}}{4\pi|\mathbf{x} - \mathbf{y}|}, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^3, \quad (4.3.4)$$

which yields

$$-\Delta_{\mathbf{x}} G(\mathbf{x}, \mathbf{y}) = \delta(\mathbf{x} - \mathbf{y}), \quad \forall \mathbf{y} \in \mathbb{R}^3, \quad (4.3.5)$$

and

$$-\Delta_{\mathbf{x}} G_{\kappa}(\mathbf{x}, \mathbf{y}) + \kappa^2 G_{\kappa}(\mathbf{x}, \mathbf{y}) = \delta(\mathbf{x} - \mathbf{y}), \quad \forall \mathbf{y} \in \mathbb{R}^3. \quad (4.3.6)$$

In the solute cavity Ω , we define the reaction potential $\psi_r := \psi - \psi_0$, where ψ_0 is the potential generated by ρ_M in vacuum written as

$$\psi_0 = \sum_{i=1}^M \frac{q_i}{\varepsilon_1 |\mathbf{x} - \mathbf{x}_i|}, \quad (4.3.7)$$

satisfying $-\Delta\psi_0 = \frac{4\pi}{\varepsilon_1}\rho_M$ in \mathbb{R}^3 . Then, ψ_r is harmonic in Ω , that is,

$$-\Delta\psi_r = 0, \quad \text{in } \Omega, \quad (4.3.8)$$

which yields the following integral equation formulation

$$\psi_r(\mathbf{x}) = \tilde{\mathcal{S}}\sigma_r(\mathbf{x}) := \int_{\Gamma} \frac{\sigma_r(\mathbf{y})}{4\pi|\mathbf{x} - \mathbf{y}|}, \quad \forall \mathbf{x} \in \Omega, \quad (4.3.9)$$

where σ_r is some function in $H^{-\frac{1}{2}}(\Gamma)$ and $\tilde{\mathcal{S}} : H^{-\frac{1}{2}}(\Gamma) \rightarrow H^1(\mathbb{R}^3 \setminus \Gamma)$ 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 Ω^c can be represented by

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

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

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

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 \mathcal{S}_κ implies that σ_e can be characterized as $\sigma_e = \mathcal{S}_\kappa^{-1} \psi|_{\Gamma}$.

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 $\tilde{\mathcal{S}}_\kappa$ across the interface [104], we can artificially extend the electrostatic potential $\psi|_{\Omega^c}$ from Ω^c to Ω as follows

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

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

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

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

$$\sigma_e = \partial_{\mathbf{n}} \psi_e - \partial_{\mathbf{n}} \psi|_{\Omega^c}, \quad \text{on } \Gamma. \quad (4.3.14)$$

As introduced above, Eq. (4.3.8) of ψ_r and (4.3.13) of ψ_e are two PDEs defined on Ω , 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 Γ and the

continuity of $\tilde{\mathcal{S}}_\kappa$ across Γ [104], we then deduce a first coupling condition

$$\psi_0 + \psi_r = \psi_e, \quad \text{on } \Gamma. \quad (4.3.15)$$

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

$$\varepsilon_1 \partial_n \psi|_{\Omega} - \varepsilon_2 \partial_n \psi|_{\Omega^c} = 0, \quad \text{on } \Gamma, \quad (4.3.16)$$

we deduce another coupling condition

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

In summary, the original problem (4.3.1) is transformed into the following two equations defined on Ω

$$\begin{cases} -\Delta \psi_r(\mathbf{x}) = 0 & \text{in } \Omega, \\ -\Delta \psi_e(\mathbf{x}) + \kappa^2 \psi_e(\mathbf{x}) = 0 & \text{in } \Omega, \end{cases} \quad (4.3.18)$$

with two coupling conditions on Γ given by

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

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

$$\psi_e = \mathcal{S}_\kappa \left(\partial_n \psi_e - \frac{\varepsilon_1}{\varepsilon_2} \partial_n (\psi_0 + \psi_r) \right), \quad \text{on } \Gamma, \quad (4.3.20)$$

which is derived from letting \mathcal{S}_κ act on both sides of the equation.

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 Ω , taking advantage of its particular geometrical structure (i.e., a union of overlapping balls). The scheme of this section is inspired by [95, 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 Γ be an initial guess for the Dirichlet condition $\psi_e|_\Gamma$ and set $k = 1$.

- [1] Solve the following Dirichlet boundary problem for ψ_r^k :

$$\begin{cases} -\Delta \psi_r^k = 0 & \text{in } \Omega, \\ \psi_r^k = g^{k-1} - \psi_0 & \text{on } \Gamma, \end{cases} \quad (4.4.1)$$

and derive its Neumann boundary trace $\partial_n \psi_r^k$ on Γ .

- [2] Solve the following Dirichlet boundary problem for ψ_e^k :

$$\begin{cases} -\Delta \psi_e^k + \kappa^2 \psi_e^k = 0 & \text{in } \Omega, \\ \psi_e^k = g^{k-1} & \text{on } \Gamma, \end{cases} \quad (4.4.2)$$

and derive similarly its Neumann boundary trace $\partial_n \psi_e^k$ on Γ .

- [3] Build the charge density $\sigma_e^k = \partial_n \psi_e^k - \frac{\varepsilon_1}{\varepsilon_2} \partial_n (\psi_0 + \psi_r^k)$ and compute a new Dirichlet condition $g^k = \mathcal{S}_\kappa \sigma_e^k$.
- [4] Compute the contribution E_k^s to the solvation energy based on ψ_r^k at the k -th iteration, set $k \leftarrow k + 1$, go back to Step [1] and repeat until the increment of interaction $|E_k^s - E_{k-1}^s|$ 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 Γ), we consider the (unrealistic) scenario where the whole space \mathbb{R}^3 is covered by the solvent medium. Then, the electrostatic potential ψ in this case is given explicitly by*

$$\psi(\mathbf{x}) = \sum_{i=1}^M \frac{4\pi q_i e^{-\kappa|\mathbf{x}-\mathbf{x}_i|}}{\varepsilon_2 |\mathbf{x} - \mathbf{x}_i|}, \quad \forall \mathbf{x} \in \mathbb{R}^3, \quad (4.4.3)$$

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

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 Ω , we have a natural domain decomposition as follows

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

where each Ω_j denotes the j -th atomic VdW-ball (or SAS-ball) with center \mathbf{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 Ω_j :

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

where $\Gamma_j = \partial\Omega_j$ and

$$\phi_{r,j} = \begin{cases} \bar{\psi}_r & \text{on } \Gamma_j^i, \\ g - \psi_0 & \text{on } \Gamma_j^e. \end{cases} \quad (4.4.5)$$

Here, we omit the superscript due to the (outer) iteration index k . Γ_j^e is the external part of Γ_j not contained in any other ball Ω_i ($i \neq j$), i.e., $\Gamma_j^e = \Gamma \cap \Gamma_j$; Γ_j^i is the internal part of Γ_j , i.e., $\Gamma_j^i = \Omega \cap \Gamma_j$ (see Figure 3 for an illustration). In addition,

$$\bar{\psi}_r(\mathbf{x}) = \frac{1}{|\mathcal{N}(j, \mathbf{x})|} \sum_{i \in \mathcal{N}(j, \mathbf{x})} \psi_r|_{\Omega_i}(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma_j^i, \quad (4.4.6)$$

where $\mathcal{N}(j, \mathbf{x})$ represents the index set of all balls such that $\mathbf{x} \in \Omega_i$. In fact, for a fixed point $\mathbf{x} \in \Gamma_j^i$, we enforce $\psi_r|_{\Gamma_j}(\mathbf{x}) = \bar{\psi}_r(\mathbf{x})$, which is the average value of $\psi_r|_{\Omega_i}(\mathbf{x})$ each computed by solving the local Laplace equation in Ω_i .

Similarly, Eq. (4.4.2) is equivalent to the following coupled local equations, each restricted to Ω_j :

$$\begin{cases} -\Delta \psi_e|_{\Omega_j} + \kappa^2 \psi_e|_{\Omega_j} = 0 & \text{in } \Omega_j, \\ \psi_e|_{\Omega_j} = \phi_{e,j} & \text{on } \Gamma_j, \end{cases} \quad (4.4.7)$$

where

$$\phi_{e,j} = \begin{cases} \bar{\psi}_e & \text{on } \Gamma_j^i, \\ g & \text{on } \Gamma_j^e. \end{cases} \quad (4.4.8)$$

Here, in the same spirit as the definition of $\bar{\psi}_r$, we define

$$\bar{\psi}_e(\mathbf{x}) = \frac{1}{|\mathcal{N}(j, \mathbf{x})|} \sum_{i \in \mathcal{N}(j, \mathbf{x})} \psi_e|_{\Omega_i}(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma_j^i, \quad (4.4.9)$$

meaning that for a fixed point $\mathbf{x} \in \Gamma_j^i$, $\bar{\psi}_e(\mathbf{x})$ is taken to be the average value of

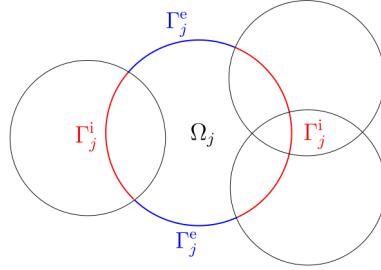


Figure 3: 2D schematic diagram of Γ_j^i (red) and Γ_j^e (blue) associated with Ω_j .

$\psi_e|_{\Omega_i}(\mathbf{x})$ each computed by solving the local HSP equation in Ω_i . The notations Γ_j^e , Γ_j^i and \mathcal{N} are defined as previously.

Generally speaking, given the Dirichlet boundary condition g on Γ , 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 $\psi_r|_{\Gamma_j^i}$ (respectively $\psi_e|_{\Gamma_j^i}$) 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 Ω_j which can actually be transformed easily

into a Laplace equation defined in the unit ball $B_1(\mathbf{0})$ in the form of

$$\begin{cases} -\Delta u_r = 0 & \text{in } B_1(\mathbf{0}), \\ u_r = \phi_r & \text{on } \mathbb{S}^2. \end{cases} \quad (4.5.1)$$

The unique solution in $H^1(B_1(\mathbf{0}))$ can be written as

$$u_r(r, \theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} [\phi_r]_{\ell}^m r^{\ell} Y_{\ell}^m(\theta, \varphi), \quad 0 \leq r \leq 1, \quad 0 \leq \theta \leq \pi, \quad 0 \leq \varphi < 2\pi. \quad (4.5.2)$$

Here, Y_{ℓ}^m denotes the (real orthonormal) spherical harmonic of degree ℓ and order m defined on \mathbb{S}^2 and

$$[\phi_r]_{\ell}^m = \int_{\mathbb{S}^2} \phi_r(\mathbf{s}) Y_{\ell}^m(\mathbf{s}) d\mathbf{s},$$

is the real coefficient of u_r corresponding to the mode Y_{ℓ}^m . Then, u_r can be numerically approximated by \tilde{u}_r in the discretization space spanned by a truncated basis of spherical harmonics $\{Y_{\ell}^m\}_{0 \leq \ell \leq \ell_{\max}, -\ell \leq m \leq \ell}$, defined as

$$\tilde{u}_r(r, \theta, \varphi) = \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} [\tilde{\phi}_r]_{\ell}^m r^{\ell} Y_{\ell}^m(\theta, \varphi), \quad 0 \leq r \leq 1, \quad 0 \leq \theta \leq \pi, \quad 0 \leq \varphi < 2\pi, \quad (4.5.3)$$

where ℓ_{\max} denotes the maximum degree of spherical harmonics and

$$[\tilde{\phi}_r]_{\ell}^m = \sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} \phi_r(\mathbf{s}_n) Y_{\ell}^m(\mathbf{s}_n). \quad (4.5.4)$$

Here, $\mathbf{s}_n \in \mathbb{S}^2$ represent Lebedev quadrature points [51], w_n^{leb} are the corresponding weights and N_{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 Ω_j can be transformed into a HSP equation defined in the unit ball $B_1(\mathbf{0})$ in the form of

$$\begin{cases} -\Delta u_e + \kappa_e^2 u_e^2 = 0 & \text{in } B_1(\mathbf{0}), \\ u_e = \phi_e & \text{on } \mathbb{S}^2. \end{cases} \quad (4.5.5)$$

Here, we actually take $u_e(\mathbf{y}) = \psi_e(r_j \mathbf{y} + \mathbf{x}_j)$, $\forall \mathbf{y} \in B_1(\mathbf{0})$, $\kappa_e = \kappa r_j$ and furthermore, $\phi_e(\mathbf{y}) = \phi_{e,j}(r_j \mathbf{y} + \mathbf{x}_j)$, $\forall \mathbf{y} \in \mathbb{S}^2$. By solving the above HSP equation in spherical coordinates by separation of variables, the radial equation corresponding to the angular dependency Y_{ℓ}^m has the form

$$\frac{1}{R} \frac{d}{dr} \left(r^2 \frac{dR}{dr} \right) = \kappa_e^2 r^2 + \ell(\ell+1), \quad n \geq 0, \quad (4.5.6)$$

that is,

$$r^2 \frac{d^2 R}{dr^2} + 2r \frac{dR}{dr} - (\kappa_e^2 r^2 + \ell(\ell + 1))R = 0, \quad (4.5.7)$$

which is called the modified spherical Bessel equation [5]. This equation has two linearly independent solutions as follows

$$i_\ell(\kappa_e r) = \sqrt{\frac{\pi}{2\kappa_e r}} I_{\ell+\frac{1}{2}}(\kappa_e r), \quad k_\ell(\kappa_e r) = \sqrt{\frac{2}{\pi\kappa_e r}} K_{\ell+\frac{1}{2}}(\kappa_e r), \quad (4.5.8)$$

where i_ℓ and k_ℓ are the modified spherical Bessel functions of the first and second kind, see [5, Chapter 14] for details and Figure 4 for an illustration. Here, $I_\alpha(x)$ and $K_\alpha(x)$ with subscript α 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

$$x^2 \frac{d^2 f}{dx^2} + x \frac{df}{dx} - (x^2 + \alpha^2)f = 0. \quad (4.5.9)$$

In fact, I_α and K_α are exponentially growing and decaying functions, respectively.

Since $k_\ell \rightarrow \infty$ as $r \rightarrow 0$, we are interested in the family i_ℓ of the first kind. That is, we write the solution to (4.5.5) in the form of

$$u_e(r, \theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} c_\ell^m i_\ell(\kappa_e r) Y_\ell^m(\theta, \varphi), \quad 0 \leq r \leq 1, \quad 0 \leq \theta \leq \pi, \quad 0 \leq \varphi < 2\pi, \quad (4.5.10)$$

where c_ℓ^m is the coefficient of the mode Y_ℓ^m . With the same discretization as in Section 4.5.1, we derive the following approximate solution similar to Eq. (4.5.3):

$$\tilde{u}_e(r, \theta, \varphi) = \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} [\tilde{\phi}_e]_\ell^m \frac{i_\ell(\kappa_e r)}{i_\ell(\kappa_e)} Y_\ell^m(\theta, \varphi), \quad 0 \leq r \leq 1, \quad 0 \leq \theta \leq \pi, \quad 0 \leq \varphi < 2\pi, \quad (4.5.11)$$

where $[\tilde{\phi}_e]_\ell^m$ is given similar to (4.5.4) as follows

$$[\tilde{\phi}_e]_\ell^m = \sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} \phi_e(\mathbf{s}_n) Y_\ell^m(\mathbf{s}_n), \quad (4.5.12)$$

with the same notations w_n^{leb} and N_{leb} as above.

Remark 4.5.2. We have $\frac{i_\ell(\kappa_e r)}{i_\ell(\kappa_e)} \rightarrow r^\ell$ as $\kappa_e \rightarrow 0$ and therefore if $\phi_r = \phi_e$, $\tilde{u}_e \rightarrow \tilde{u}_r$ as $\kappa_e \rightarrow 0$. This means that the solution to Eq. (4.5.5) tends to the solution to Eq. (4.5.1) when $\phi_r = \phi_e$ and $\kappa_e \rightarrow 0$, which makes sense.

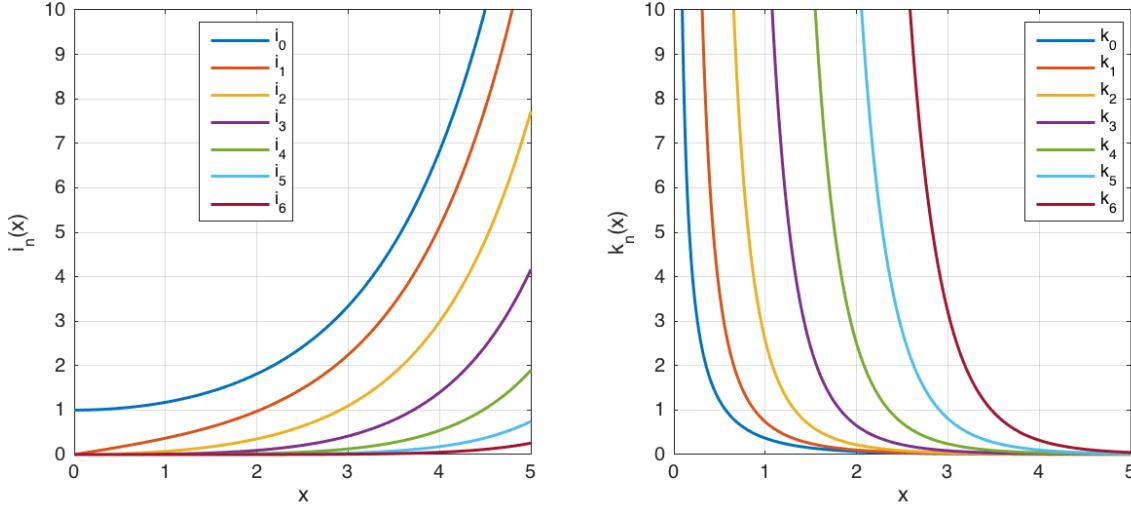


Figure 4: The modified spherical Bessel functions of the first kind (i_n , left) and the second kind (k_n , right).

Remark 4.5.3. Similar to the Laplace solver, the HSP solver actually reads: given an arbitrary Dirichlet condition on the unit sphere, we can derive directly the numerical solution to the HSP equation (4.5.5), according to the formula (4.5.11).

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 Ω_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 χ_i be the characteristic function of Ω_i , i.e.,

$$\chi_i(\mathbf{x}) := \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_i \\ 0 & \text{if } \mathbf{x} \notin \Omega_i \end{cases} \quad (4.6.1)$$

and then let

$$w_{ji}(\mathbf{x}) := \frac{\chi_i(\mathbf{x})}{|\mathcal{N}(j, \mathbf{x})|} = \frac{\chi_i(\mathbf{x})}{\sum_{i \neq j} \chi_i(\mathbf{x})}, \quad \forall \mathbf{x} \in \Gamma_j. \quad (4.6.2)$$

Here, we make the convention that in the case of $|\mathcal{N}(j, \mathbf{x})| = 0$ (i.e., $\mathbf{x} \in \Gamma_j^e$), we define $w_{ji}(\mathbf{x}) = 0$, $\forall i$. Furthermore, $\forall \mathbf{x} \in \Gamma_j$, we define

$$\chi_j^e(\mathbf{x}) := \begin{cases} 1 & \text{if } \mathbf{x} \in \Gamma_j^e, \\ 0 & \text{if } \mathbf{x} \in \Gamma_j^i, \end{cases} \quad (4.6.3)$$

which is equivalent to

$$\chi_j^e(\mathbf{x}) = 1 - \sum_{i \neq j} w_{ji}(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma_j. \quad (4.6.4)$$

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

$$\psi_r|_{\Gamma_j}(\mathbf{x}) - \sum_{i \neq j} w_{ji}(\mathbf{x}) \psi_r|_{\Omega_i}(\mathbf{x}) = \chi_j^e(\mathbf{x}) (g(\mathbf{x}) - \psi_0(\mathbf{x})), \quad \forall \mathbf{x} \in \Gamma_j. \quad (4.6.5)$$

Similarly, Eq. (4.4.8) can be recast as

$$\psi_e|_{\Gamma_j}(\mathbf{x}) - \sum_{i \neq j} w_{ji}(\mathbf{x}) \psi_e|_{\Omega_i}(\mathbf{x}) = \chi_j^e(\mathbf{x}) g(\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma_j. \quad (4.6.6)$$

In addition, there is a global coupling condition on Γ , Eq. (4.3.20), between the global Laplace equation and the global HSP equation in (4.3.18), involving the nonlocal operator \mathcal{S}_κ :

$$g(\mathbf{x}) = \psi_e|_\Gamma(\mathbf{x}) = \mathcal{S}_\kappa \left(\partial_{\mathbf{n}} \psi_e - \frac{\varepsilon_1}{\varepsilon_2} \partial_{\mathbf{n}} (\psi_0 + \psi_r) \right) (\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma. \quad (4.6.7)$$

The single-layer operator \mathcal{S}_κ involves an integral over the whole solute-solvent boundary Γ which seems difficult to compute at a first glance. We now introduce a technique to compute the integral of \mathcal{S}_κ efficiently. For each sphere Γ_i , we define a local single-layer potential $\tilde{\mathcal{S}}_{\kappa, \Gamma_i}$ as follows

$$\tilde{\mathcal{S}}_{\kappa, \Gamma_i} \sigma(\mathbf{x}) := \int_{\Gamma_i} \frac{e^{-\kappa|\mathbf{x}-\mathbf{y}|} \sigma(\mathbf{y})}{4\pi|\mathbf{x}-\mathbf{y}|}, \quad \forall \mathbf{x} \in \mathbb{R}^3, \quad (4.6.8)$$

where σ is an arbitrary function in $H^{-\frac{1}{2}}(\Gamma_i)$. As a consequence, $\forall \sigma \in H^{-\frac{1}{2}}(\Gamma)$, we have

$$\mathcal{S}_\kappa \sigma = \sum_{i=1}^M \tilde{\mathcal{S}}_{\kappa, \Gamma_i} (\chi_i^e \sigma), \quad (4.6.9)$$

where $\chi_i^e \sigma$ extends $\sigma|_{\Gamma_i^e}$ by zero to the whole sphere Γ_i . The above equation implies that the integral over Γ can be divided into a group of integrals respectively over each sphere Γ_i . Therefore, Eq. (4.6.7) can be recast as

$$g(\mathbf{x}) = \sum_{i=1}^M \tilde{\mathcal{S}}_{\kappa, \Gamma_i} \left[\chi_i^e \left(\partial_{\mathbf{n}} \psi_e - \frac{\varepsilon_1}{\varepsilon_2} \partial_{\mathbf{n}} (\psi_0 + \psi_r) \right) \right] (\mathbf{x}), \quad \forall \mathbf{x} \in \Gamma. \quad (4.6.10)$$

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 Γ_j , we first approximate $\psi_r|_{\Gamma_j}$ and $\psi_e|_{\Gamma_j}$ respectively by a linear combination of spherical harmonics $\{Y_\ell^m\}$ with $0 \leq \ell \leq \ell_{\max}$ and $-\ell \leq m \leq \ell$ as follows

$$\psi_r|_{\Gamma_j}(\mathbf{x}_j + r_j \mathbf{s}) = \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} [X_r]_{j\ell m} Y_\ell^m(\mathbf{s}), \quad \mathbf{s} \in \mathbb{S}^2, \quad (4.6.11)$$

and

$$\psi_e|_{\Gamma_j}(\mathbf{x}_j + r_j \mathbf{s}) = \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} [X_e]_{j\ell m} Y_\ell^m(\mathbf{s}), \quad \mathbf{s} \in \mathbb{S}^2, \quad (4.6.12)$$

where $[X_r]_{j\ell m}$ and $[X_e]_{j\ell m}$ are unknown coefficients of the mode Y_ℓ^m respectively associated with $\psi_r|_{\Gamma_j}$ and $\psi_e|_{\Gamma_j}$. Here, for any point $\mathbf{x} \in \Gamma_j$, we actually use its spherical coordinates (r_j, \mathbf{s}) s.t. $\mathbf{x} = \mathbf{x}_j + r_j \mathbf{s}$. According to the Laplace solver and the HSP solver presented in Section 4.5, we deduce directly

$$\psi_r|_{\Omega_i}(\mathbf{x}_i + r \mathbf{s}) = \sum_{\ell'=0}^{\ell_{\max}} \sum_{m'=-\ell'}^{\ell'} [X_r]_{i\ell' m'} \left(\frac{r}{r_i} \right)^{\ell'} Y_{\ell'}^{m'}(\mathbf{s}), \quad 0 \leq r \leq r_i, \quad \mathbf{s} \in \mathbb{S}^2, \quad (4.6.13)$$

and

$$\psi_e|_{\Omega_i}(\mathbf{x}_i + r \mathbf{s}) = \sum_{\ell'=0}^{\ell_{\max}} \sum_{m'=-\ell'}^{\ell'} [X_e]_{i\ell' m'} \frac{i_{\ell'}}{i_{\ell'}} \frac{(\kappa r)}{(\kappa r_i)} Y_{\ell'}^{m'}(\mathbf{s}), \quad 0 \leq r \leq r_i, \quad \mathbf{s} \in \mathbb{S}^2, \quad (4.6.14)$$

where for any point $\mathbf{x} \in \Omega_i$, we take its spherical coordinates (r, \mathbf{s}) s.t. $\mathbf{x} = \mathbf{x}_i + r \mathbf{s}$.

Also, for each sphere Γ_i , we can compute the normal derivative of ψ_r on Γ_i^e as follows

$$\partial_n \psi_r(\mathbf{x}_i + r_i \mathbf{s}) = \sum_{\ell'=0}^{\ell_{\max}} \sum_{m'=-\ell'}^{\ell'} [X_r]_{i\ell'm'} \left(\frac{\ell'}{r_i} \right) Y_{\ell'}^{m'}(\mathbf{s}), \quad \mathbf{x}_i + r_i \mathbf{s} \in \Gamma_i^e, \quad (4.6.15)$$

and the normal derivative of ψ_e on Γ_i^e

$$\partial_n \psi_e(\mathbf{x}_i + r_i \mathbf{s}) = \sum_{\ell'=0}^{\ell_{\max}} \sum_{m'=-\ell'}^{\ell'} [X_e]_{i\ell'm'} \frac{\kappa i'_{\ell'}(\kappa r_i)}{i_{\ell'}(\kappa r_i)} Y_{\ell'}^{m'}(\mathbf{s}), \quad \mathbf{x}_i + r_i \mathbf{s} \in \Gamma_i^e, \quad (4.6.16)$$

where $i'_{\ell'}$ represents the derivative of $i_{\ell'}$.

Remark 4.6.1. To compute $i'_{\ell'}(\kappa r_i)$, we use the iterative property of the derivative of i_n (see [5, Page 707]) given by

$$(2n+1)i'_n(x) = ni_{n-1}(x) + (n+1)i_{n+1}(x), \quad (4.6.17)$$

and in analogy, the derivative of k_n as follows

$$-(2n+1)k'_n(x) = nk_{n-1}(x) + (n+1)k_{n+1}(x), \quad (4.6.18)$$

which will be used to compute $k'_{\ell}(\kappa r_i)$ in the Appendix D.

4.6.2.2 Discretization of the coupling conditions

So far, we have written the Ansatz for the unknowns $\psi_r|_{\Gamma_j}$, $\psi_e|_{\Gamma_j}$, respectively $\psi_r|_{\Omega_i}$, $\psi_e|_{\Omega_i}$ with normal derivatives $\partial_n \psi_r$ and $\partial_n \psi_e$, following Eqs (4.6.11) – (4.6.16), which depend on the unknowns X_r and X_e . This allows us to discretize the coupling conditions (4.6.5), (4.6.6) and (4.6.10).

First, we replace the variable $\mathbf{x} \in \Gamma_j$ of Eq. (4.6.5) by $\mathbf{x} = \mathbf{x}_j + r_j \mathbf{s}$ with $\mathbf{s} \in \mathbb{S}^2$ and derive the equation for each sphere Γ_j as follows

$$\psi_r|_{\Gamma_j}(\mathbf{x}_j + r_j \mathbf{s}) - \sum_{i \neq j} w_{ji}(\mathbf{x}_j + r_j \mathbf{s}) \psi_r|_{\Omega_i}(\mathbf{x}_j + r_j \mathbf{s}) = \chi_j^e(\mathbf{x}_j + r_j \mathbf{s}) (g(\mathbf{x}_j + r_j \mathbf{s}) - \psi_0(\mathbf{x}_j + r_j \mathbf{s})), \quad (4.6.19)$$

which induces the following local equation by multiplying by Y_{ℓ}^m and integrating over \mathbb{S}^2 on both sides, $\forall j, \ell, m$,

$$\begin{aligned} & \left\langle \psi_r|_{\Gamma_j}(\mathbf{x}_j + r_j \cdot) - \sum_{i \neq j} w_{ji}(\mathbf{x}_j + r_j \cdot) \psi_r|_{\Omega_i}(\mathbf{x}_j + r_j \cdot), Y_{\ell}^m(\cdot) \right\rangle_{\mathbb{S}^2} \\ &= \langle \chi_j^e(\mathbf{x}_j + r_j \cdot) (g(\mathbf{x}_j + r_j \cdot) - \psi_0(\mathbf{x}_j + r_j \cdot)), Y_{\ell}^m(\cdot) \rangle_{\mathbb{S}^2}. \end{aligned} \quad (4.6.20)$$

Here, $\langle \cdot, \cdot \rangle_{\mathbb{S}^2}$ represents the integral over the unit sphere \mathbb{S}^2 , which is numerically approximated using the Lebedev quadrature rule with N_{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

$$[\mathbf{A}X_r]_{j\ell m} = [G_X]_{j\ell m} + [G_0]_{j\ell m}, \quad \forall j, \ell, m. \quad (4.6.21)$$

Here, \mathbf{A} is a square matrix of dimension $M(\ell_{\max}+1)^2 \times M(\ell_{\max}+1)^2$ and the $j\ell m$ -th row of $\mathbf{A}X_r$ is given by substituting (4.6.11) and (4.6.13) into (4.6.20) as follows

$$\begin{aligned} [\mathbf{A}X_r]_{j\ell m} &= \left\langle \psi_r|_{\Gamma_j}(\mathbf{x}_j + r_j \cdot) - \sum_{i \neq j} w_{ji}(\mathbf{x}_j + r_j \cdot) \psi_r|_{\Omega_i}(\mathbf{x}_j + r_j \cdot), Y_\ell^m(\cdot) \right\rangle_{\mathbb{S}^2, N_{\text{leb}}} \\ &= [X_r]_{j\ell m} - \sum_{i \neq j} \sum_{\ell', m'} \left(\sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} w_{ji}(\mathbf{x}_j + r_j \mathbf{s}_n) \left(\frac{r_{ijn}}{r_i} \right)^{\ell'} Y_{\ell'}^{m'}(\mathbf{s}_{ijn}) Y_\ell^m(\mathbf{s}_n) \right) [X_r]_{i\ell' m'}, \end{aligned} \quad (4.6.22)$$

where $(r_{ijn}, \mathbf{s}_{ijn})$ is the spherical coordinate associated with Γ_i of the point $\mathbf{x}_j + r_j \mathbf{s}_n$ s.t.

$$\mathbf{x}_j + r_j \mathbf{s}_n = \mathbf{x}_i + r_{ijn} \mathbf{s}_{ijn}, \quad \text{with } \mathbf{s}_{ijn} \in \mathbb{S}^2.$$

Furthermore, the $j\ell m$ -th element of the column vector G_X is given as

$$\begin{aligned} [G_X]_{j\ell m} &= \langle \chi_j^e(\mathbf{x}_j + r_j \cdot) g(\mathbf{x}_j + r_j \cdot), Y_\ell^m(\cdot) \rangle_{\mathbb{S}^2, N_{\text{leb}}} \\ &= \sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} \chi_j^e(\mathbf{x}_j + r_j \mathbf{s}_n) g(\mathbf{x}_j + r_j \mathbf{s}_n) Y_\ell^m(\mathbf{s}_n), \end{aligned} \quad (4.6.23)$$

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.,

$$X = \begin{pmatrix} X_r \\ X_e \end{pmatrix} \in \mathbb{R}^{2M(\ell_{\max}+1)^2}. \quad (4.6.24)$$

Similarly, the $j\ell m$ -th element of the column vector G_0 is given as

$$[G_0]_{j\ell m} = - \sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} \chi_j^e(\mathbf{x}_j + r_j \mathbf{s}_n) \psi_0(\mathbf{x}_j + r_j \mathbf{s}_n) Y_\ell^m(\mathbf{s}_n), \quad (4.6.25)$$

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 Γ_j , we have another linear system in the form of matrices

$$[\mathbf{B}X_e]_{j\ell m} = [G_X]_{j\ell m}, \quad \forall j, \ell, m, \quad (4.6.26)$$

where the square matrix \mathbf{B} satisfies

$$[\mathbf{B}X_e]_{j\ell m} = [X_e]_{j\ell m} - \sum_{i \neq j} \sum_{\ell', m'} \left(\sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} w_{ji}(\mathbf{x}_j + r_j \mathbf{s}_n) \frac{i_{\ell'}(\kappa r_{ijn})}{i_{\ell'}(\kappa r_i)} Y_{\ell'}^{m'}(\mathbf{s}_{ijn}) Y_{\ell'}^m(\mathbf{s}_n) \right) [X_e]_{i\ell' m'}, \quad (4.6.27)$$

and $[G_X]_{j\ell m}$ is given by (4.6.23).

Remark 4.6.2. For a fixed atomic ball Ω_i of a realistic molecule, especially when the molecule is large, the number of its intersections with other atomic balls Ω_j is bounded from above. From the definition (4.6.2) of w_{ji} , we have $w_{ji}(\mathbf{x}_j + r_j \mathbf{s}_n) = 0$ if $r_{ijn} \geq r_i$. Since $w_{ji} \equiv 0$ when $\Omega_i \cap \Omega_j = \emptyset$, we deduce that \mathbf{A} and \mathbf{B} are both sparse matrices.

So far, we have derived two linear systems of the form

$$\begin{cases} \mathbf{A} X_r = G_X + G_0, \\ \mathbf{B} X_e = G_X, \end{cases} \quad (4.6.28)$$

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 ψ_r and ψ_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 ,

$$G_X = F_0 - \mathbf{C}_1 X_r - \mathbf{C}_2 X_e, \quad (4.6.29)$$

where the column vector F_0 is associated with $\partial_{\mathbf{n}}\psi_0$, two dense square matrices \mathbf{C}_1 and \mathbf{C}_2 are respectively associated with $\partial_{\mathbf{n}}\psi_r$ and $\partial_{\mathbf{n}}\psi_e$. Considering the complexity of the formulas of F_0 , \mathbf{C}_1 , \mathbf{C}_2 , we present them in Append. D.

With the form (4.6.29) of G_X , we finally obtain a global linear system written as

$$\begin{pmatrix} \mathbf{A} + \mathbf{C}_1 & \mathbf{C}_2 \\ \mathbf{C}_1 & \mathbf{B} + \mathbf{C}_2 \end{pmatrix} \begin{pmatrix} X_r \\ X_e \end{pmatrix} = \begin{pmatrix} G_0 + F_0 \\ F_0 \end{pmatrix}, \quad (4.6.30)$$

which can also be reduced to

$$\begin{pmatrix} \mathbf{A} + \mathbf{C}_1 & \mathbf{C}_2 \\ \mathbf{A} & -\mathbf{B} \end{pmatrix} \begin{pmatrix} X_r \\ X_e \end{pmatrix} = \begin{pmatrix} G_0 + F_0 \\ G_0 \end{pmatrix}, \quad (4.6.31)$$

where the matrix on the left-hand side is half-sparse, since both \mathbf{A} and \mathbf{B} are sparse (but not \mathbf{C}_1 nor \mathbf{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, 46], 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

$$\begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix} \begin{pmatrix} X_r^k \\ X_e^k \end{pmatrix} = - \begin{pmatrix} \mathbf{C}_1 & \mathbf{C}_2 \\ \mathbf{C}_1 & \mathbf{C}_2 \end{pmatrix} \begin{pmatrix} X_r^{k-1} \\ X_e^{k-1} \end{pmatrix} + \begin{pmatrix} G_0 + F_0 \\ F_0 \end{pmatrix}, \quad (4.6.32)$$

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 \mathbf{A} and \mathbf{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 ψ_r according to the following formula (see [40] for the derivation of this formula)

$$E^s = \frac{1}{2} \int_{\mathbb{R}^3} \rho_M(\mathbf{r}) \psi_r(\mathbf{r}) d\mathbf{r} = \frac{1}{2} \sum_{i=1}^M q_i \psi_r(\mathbf{x}_i), \quad (4.7.1)$$

where the solute's charge density ρ_M is given in Eq. (4.2.7) and ψ_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 κ 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. (4.2.5), 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 $r_p = 1.5\text{\AA}$. Using the ddLPB method, we compute a highly accurate electrostatic contribution (denoted by E_{ex}^s , the subscript “ex” represents “exact”) to the solvation

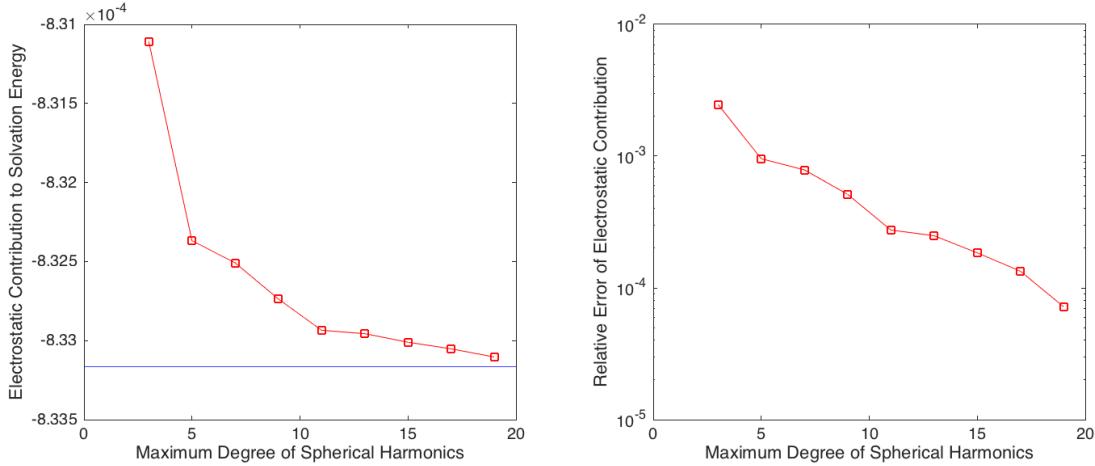


Figure 5: The left figure plots the electrostatic contributions to the solvation energy of formaldehyde with respect to ℓ_{\max} and the right figure plots the relative errors. The blue line in the left figure represents the highly accurate electrostatic contribution.

energy of formaldehyde with large discretization parameters $\ell_{\max} = 25$, $N_{\text{leb}} = 4334$. This means that for each SAS-sphere of the molecule, we take 26^2 spherical harmonics and 4334 discretization points. We treat E_{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 ℓ_{\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

$$\text{Error} = \frac{|E^{\text{s}} - E_{\text{ex}}^{\text{s}}|}{|E_{\text{ex}}^{\text{s}}|}. \quad (4.7.2)$$

It is observed that E^{s} converges exponentially and the ddLPB method provides improvable approximations when ℓ_{\max} increases. Similarly, in Figure 6, we illustrate how the electrostatic contribution E^{s} varies with respect to the number of Lebedev quadrature points N_{leb} , where we fix the maximum degree of spherical harmonics $\ell_{\max} = 25$. Still, it is observed that the ddLPB method provides improvable approximations when N_{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 $\ell_{\max} = 15$ and $N_{\text{leb}} = 974$. As given in the previous

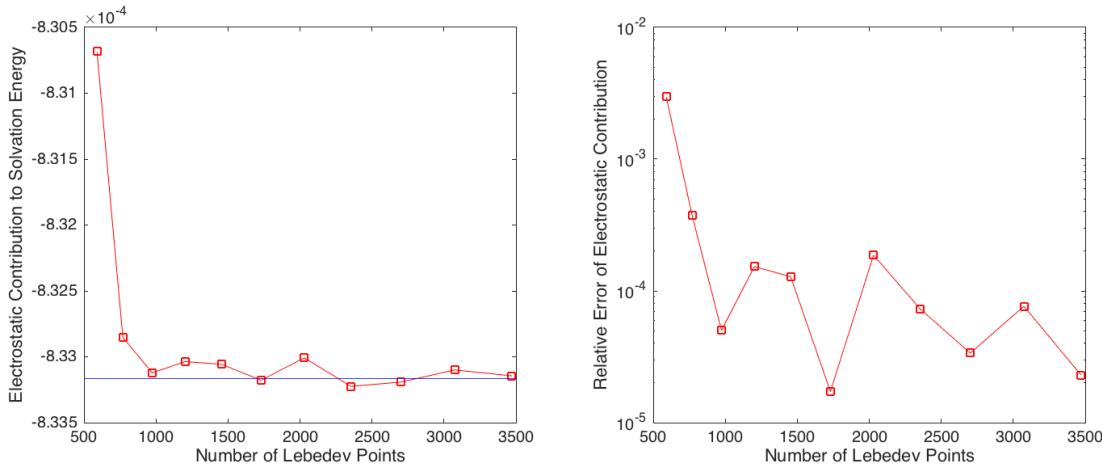


Figure 6: The left figure plots the electrostatic contributions to the solvation energy of formaldehyde with respect to N_{leb} and the right figure plots the relative errors. The blue line in the left figure represents the highly accurate electrostatic contribution.

subsection, we use the SAS-cavity and take the solvent radius $r_p = 1.5\text{\AA}$. We plot in Figure 7 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 7, 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 ℓ_{\max} and N_{leb} . It is observed that when κ tends to ∞ , the result of the ddLPB method tends to the result of the ddCOSMO method. This means that when the ionic strength tends to ∞ , the solvent becomes an 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 ℓ_{\max} and N_{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 [26] 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 been 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 $\ell_{\max} = 15$, $N_{\text{leb}} = 974$ and $\kappa = 0.1040$. We

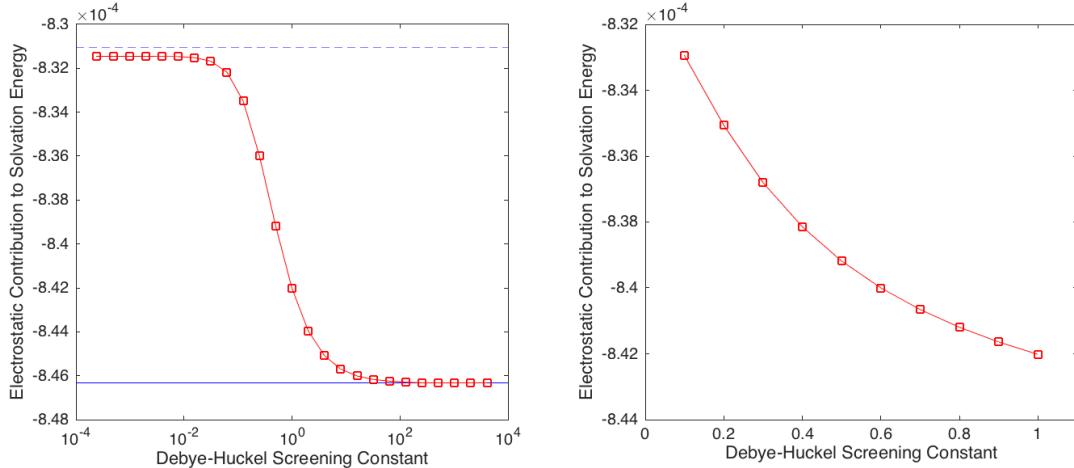


Figure 7: The above two figures plot the electrostatic contributions to the solvation energy of formaldehyde with respect to κ respectively in a broad range (left) and a normal range (right). The solid blue line in the left figure represents the result of the COSMO with conductor-like solvent, computed 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 \mathbf{A} and \mathbf{B} . Further, the proposed iterative method is set to stop at the k -th iteration when $\text{Error}_k < \text{Tol}$. Here,

$$\text{Error}_k = \frac{\|X^k - X^{k-1}\|_2}{\|X^k\|_2}$$

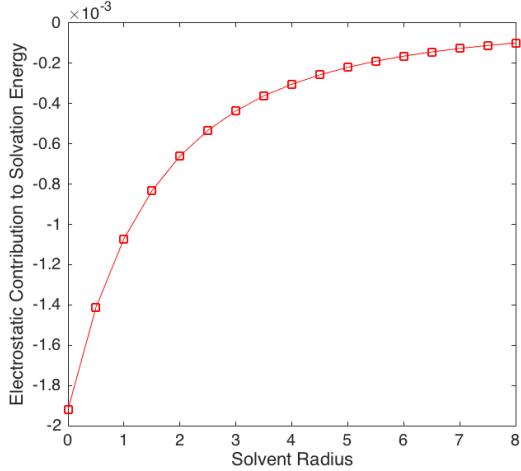


Figure 8: The above figure plots the electrostatic contributions to the solvation energy of formaldehyde with respect to the solvent radius r_p (unit: Å).

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 $\ell_{\max} = 15$, $N_{\text{leb}} = 974$, $\kappa = 0.1040$, $r_p = 1.5\text{\AA}$. 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 ψ_r on the SAS of formaldehyde, benzene and caffeine in Figure 9 and 10. We use the following parameters: $\ell_{\max} = 15$, $N_{\text{leb}} = 974$, $\kappa = 0.1040$, $r_p = 1.5\text{\AA}$. 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

Running time (s)	Hydrogen-fluoride	Formaldehyde	Bezene	Caffeine
LU factorization	0.0440	0.2570	6.0170	50.5490
GMRes method	0.2570	0.4670	11.1620	140.5670
Iterative method	0.1850	0.3720	4.6480	10.5430
Relative error	Hydrogen-fluoride	Formaldehyde	Bezene	Caffeine
GMRes method	1.61×10^{-6}	2.54×10^{-6}	2.96×10^{-6}	0.25
Iterative method	1.32×10^{-6}	2.19×10^{-6}	1.87×10^{-6}	2.01×10^{-6}

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: 1024×1024 , 2048×2048 , 6144×6144 , 12288×12288). 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.

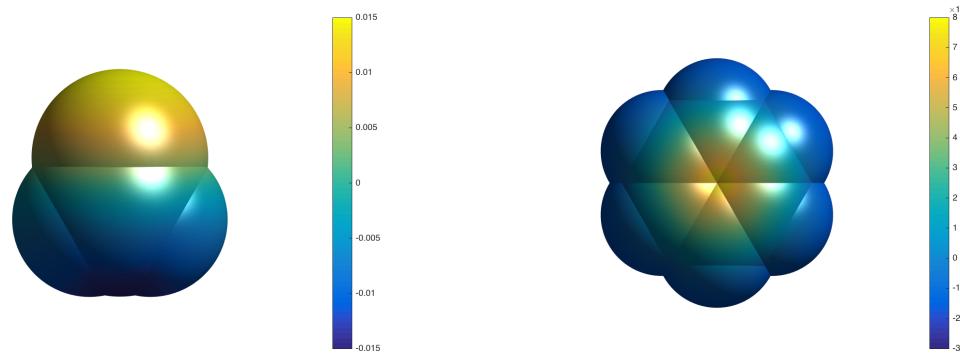


Figure 9: Reaction potential ψ_r on the SAS of formaldehyde (left) and benzene (right).

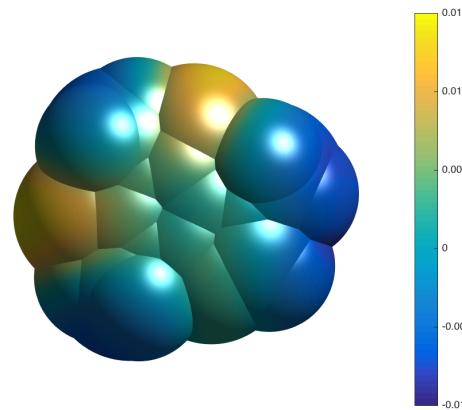


Figure 10: Reaction potential ψ_r on the SAS of caffeine.

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 1), 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 2 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

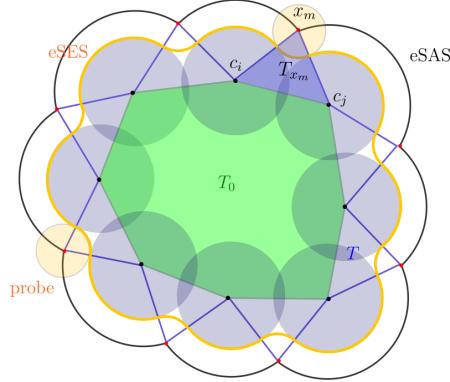


Figure 1: A 2D schematics of the two different polygons in the eSAS-cavity of an artificial molecule for the proof of Lemma A.1. T_{x_m} is the triangle with three vertices (x_m, c_i, c_j) and the polygon in green is the reduced polygon T_0 .

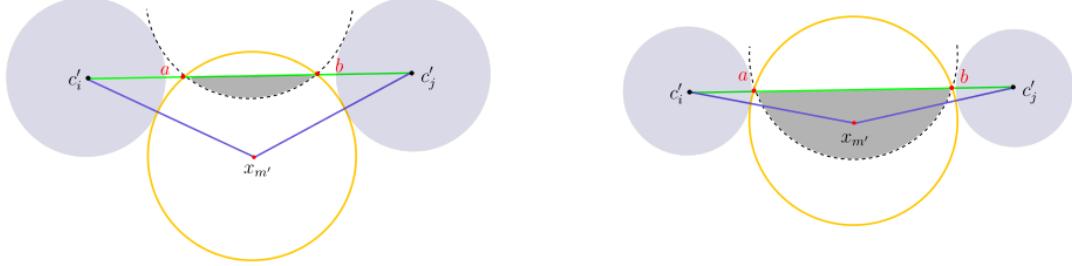


Figure 2: 2D schematics of two cases where $P_0 \cap T_{x_{m'}} \neq \emptyset$ with $x_{m'}$ being an SAS intersection point different from x_m . The green segment $[c'_i, c'_j]$ is an edge of T_0 and a and b are the two intersection points between the circle $\partial B_{r_p}(x_{m'})$ and the segment $[c'_i, c'_j]$. 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

$$P_0 \setminus B_{r_p}(x) = P_0 \cap (B_{r_p}(x))^c = P_0 \cap \{p : \|p - x_m\| \leq \|p - x\|\}, \quad \forall x \neq x_m, \quad (\text{A.1})$$

and therefore we obtain that

$$\begin{aligned}
 P_0 \setminus \bigcup_{x \in K} B_{r_p}(x) &= \left(P_0 \setminus \bigcup_{x \in K} B_{r_p}(x) \right) \cap T \\
 &= \bigcap_{x \in I} (P_0 \setminus B_{r_p}(x)) \cap T \\
 &= P_0 \cap T \cap \{p : \|p - x_m\| \leq \|p - x\|, \forall x \neq x_m, x \in I\} \\
 &= P_-.
 \end{aligned} \tag{A.2}$$

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                                 $\triangleright$  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                   $\triangleright$  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             $\triangleright$  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  $e$  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             $\triangleright$  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 for
40:  end while
41:  return  $P$ ,  $N_p$ ,  $T$ ,  $N_t$ 
42: 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 $v \in H_0^1(B_1(\mathbf{0}))$ is the unique weak solution to Eq. (3.6.6), which implies that $\forall w \in H_0^1(B_1(\mathbf{0}))$,

$$\int_{B_1(\mathbf{0})} \varepsilon_1 \nabla v \cdot \nabla w = \int_{B_1(\mathbf{0})} f w. \quad (\text{C.1})$$

From the fact that $\varepsilon_1|_{B_\delta(\mathbf{0})} = 1$ and $f|_{B_\delta(\mathbf{0})} = 0$, we know that v is harmonic in $B_\delta(\mathbf{0})$ and in consequence, $\forall w \in H^1(B_\delta(\mathbf{0}))$,

$$\int_{B_\delta(\mathbf{0})} \nabla v \cdot \nabla w + \int_{\partial B_\delta(\mathbf{0})} (\partial_{\mathbf{n}} v) w = 0, \quad (\text{C.2})$$

where \mathbf{n} points inwards $B_\delta(\mathbf{0})$ at $\partial B_\delta(\mathbf{0})$, i.e., outwards \mathcal{D} . Subtracting (C.2) from (C.1), we obtain: $\forall w \in H_{0,\delta}^1(\mathcal{D})$,

$$\int_{\mathcal{D}} \varepsilon_1 \nabla v \cdot \nabla w - \int_{\partial B_\delta(\mathbf{0})} (\mathcal{T} v) w = \int_{\mathcal{D}} f w, \quad (\text{C.3})$$

where $\mathcal{T} v = \partial_{\mathbf{n}} v$ is the Dirichlet to Neumann operator. This is just the variational formulation (3.6.7). Therefore, $v|_{\mathcal{D}} \in H_{0,\delta}^1(\mathcal{D})$ 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 $H_{0,\delta}^1(\mathcal{D})$ with the norm $\|\nabla \cdot\|_{L^2(\mathcal{D})}$, which is indeed a norm due to the Poincaré inequality. Recall that the bilinear form is

defined by

$$a(v, w) = \int_{\mathcal{D}} \varepsilon_1 \nabla v \cdot \nabla w - \int_{\partial B_\delta(\mathbf{0})} (\mathcal{T}v)w, \quad \forall v, w \in H_{0,\delta}^1(\mathcal{D}),$$

and that $\varepsilon_1 \in L^\infty(\mathcal{D})$ with $1 \leq \varepsilon_1 \leq \varepsilon_s$. For the coercivity of the bilinear form, it is clear that

$$\int_{\mathcal{D}} \varepsilon_1 \nabla v \cdot \nabla v \geq \|\nabla v\|_{L^2(\mathcal{D})}^2. \quad (\text{C.4})$$

Furthermore, the following lemma can be derived.

Lemma C.1. *There holds that*

$$-\int_{\partial B_\delta(0)} (\mathcal{T}v)v \geq 0. \quad (\text{C.5})$$

Proof. The harmonic extension E_v of v in $B_\delta(0)$ satisfies $\Delta E_v = 0$ and thus

$$0 = \int_{B_\delta(0)} \Delta E_v E_v = -\int_{B_\delta(0)} |\nabla E_v|^2 - \int_{\partial B_\delta(0)} (\nabla E_v \cdot \mathbf{n}) E_v.$$

Since $\nabla E_v \cdot \mathbf{n} = \mathcal{T}v$, we deduce that

$$-\int_{\partial B_\delta(0)} (\mathcal{T}v)v = \int_{B_\delta(0)} |\nabla E_v|^2 \geq 0.$$

□

Combining the two inequalities (C.4) and (C.5), we then have $a(v, v) \geq \|\nabla v\|_{L^2(\mathcal{D})}^2$, 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 $H_{0,\delta}^1(\mathcal{D})$, we then have

$$a(v_1 - v_2, v_1 - v_2) = a(v_1, v_1 - v_2) - a(v_2, v_1 - v_2) = \int_{\mathcal{D}} f(v_1 - v_2) - \int_{\mathcal{D}} f(v_1 - v_2) = 0.$$

Further, according to the fact that

$$a(v_1 - v_2, v_1 - v_2) \geq \|\nabla(v_1 - v_2)\|_{L^2(\mathcal{D})}^2,$$

we derive $\|\nabla(v_1 - v_2)\|_{L^2(\mathcal{D})} = 0$ 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

$$\|\nabla v\|_{L^2(\mathcal{D})}^2 \leq a(v, v) = \int_{\mathcal{D}} fv \leq \|f\|_{L^2(\mathcal{D})} \|v\|_{L^2(\mathcal{D})} \leq C_{\mathcal{D}} \|\nabla f\|_{L^2(\mathcal{D})} \|\nabla v\|_{L^2(\mathcal{D})}$$

where $C_{\mathcal{D}}$ is a constant depending only on \mathcal{D} due to the Poincaré inequality. This yields that

$$\|\nabla v\|_{L^2(\mathcal{D})} \leq C \|\nabla f\|_{L^2(\mathcal{D})},$$

that is, v depends continuously on f . In summary, the formulation (3.6.7) is well-posed.

C.2 Computation of $\frac{\partial}{\partial\theta}Y_\ell^m$ and $\frac{\partial}{\partial\varphi}Y_\ell^m$

We start with providing the definition of the real-valued spherical harmonics Y_ℓ^m :

$$Y_\ell^m(\theta, \varphi) = \begin{cases} \sqrt{2} N_\ell^m P_\ell^{|m|}(\cos \theta) \sin |m|\varphi & \text{if } m < 0, \\ N_\ell^m P_\ell^m(\cos \theta) & \text{if } m = 0, \\ \sqrt{2} N_\ell^m P_\ell^m(\cos \theta) \cos m\varphi & \text{if } m > 0, \end{cases} \quad (\text{C.1})$$

with

$$N_\ell^m = \sqrt{\frac{(2\ell+1)}{4\pi} \frac{(\ell-|m|)!}{(\ell+|m|)!}},$$

and where P_ℓ^m represents the associated Legendre polynomials defined on $[-1, 1]$. They satisfy the following property

$$\sqrt{1-x^2} \frac{d}{dx} P_\ell^m(x) = \frac{1}{2} [(\ell+m)(\ell-m+1)P_\ell^{m-1}(x) - P_\ell^{m+1}(x)], \quad x \in [-1, 1], \quad (\text{C.2})$$

which implies that

$$\frac{d}{d\theta} P_\ell^m(\cos \theta) = -\frac{1}{2} [(\ell+m)(\ell-m+1)P_\ell^{m-1}(\cos \theta) - P_\ell^{m+1}(\cos \theta)]. \quad (\text{C.3})$$

In consequence, we can then compute the derivative of Y_ℓ^m with respect to θ as below

$$\frac{\partial}{\partial\theta} Y_\ell^m = \begin{cases} \sqrt{2} N_\ell^m \left(\frac{d}{d\theta} P_\ell^{|m|}(\cos \theta) \right) \sin |m|\varphi & \text{if } m < 0, \\ N_\ell^m \left(\frac{d}{d\theta} P_\ell^m(\cos \theta) \right) & \text{if } m = 0, \\ \sqrt{2} N_\ell^m \left(\frac{d}{d\theta} P_\ell^m(\cos \theta) \right) \cos m\varphi & \text{if } m > 0. \end{cases} \quad (\text{C.4})$$

In addition, we can compute the derivative of Y_ℓ^m with respect to φ as below

$$\frac{\partial}{\partial\varphi} Y_\ell^m = \begin{cases} \sqrt{2} |m| N_\ell^m P_\ell^{|m|}(\cos \theta) \cos |m|\varphi & \text{if } m < 0, \\ 0 & \text{if } m = 0, \\ -\sqrt{2} m N_\ell^m P_\ell^m(\cos \theta) \sin m\varphi & \text{if } m > 0. \end{cases} \quad (\text{C.5})$$

C.3 Computation of $f(\mathbf{x})$ in (3.6.6)

In Eq. (3.6.6), the right hand side f is defined as follows

$$f(\mathbf{x}) = \nabla \cdot [(\varepsilon_1(\mathbf{x}) - 1) \nabla \Phi(\mathbf{c}_j + R_j \mathbf{x})] + \nabla \cdot \varepsilon_1(\mathbf{x}) \nabla \hat{u}_1(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{D}, \quad (\text{C.1})$$

for some $j = 1, \dots, M$. Since $\text{supp}(\Delta \Phi) = \text{supp}(\rho) \subset \Omega_{\text{ses}}$ and $\text{supp}(\varepsilon - 1) = \Omega_{\text{ses}}^c$ (in fact, $\varepsilon = 1$ in Ω_{ses}), it follows that $(\varepsilon(\mathbf{p}) - 1) \Delta_{\mathbf{p}} \Phi(\mathbf{p}) = 0$, which implies that $(\varepsilon_1(\mathbf{x}) - 1) \Delta \Phi = 0$. Further, combining with $\Delta \hat{u}_1 = 0$ yields

$$f(\mathbf{x}) = \nabla \varepsilon_1(\mathbf{x}) \cdot \nabla \Phi(\mathbf{c}_j + R_j \mathbf{x}) + \nabla \varepsilon_1(\mathbf{x}) \cdot \nabla \hat{u}_1(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{D}, \quad (\text{C.2})$$

where $\varepsilon_1(\mathbf{x}) = \varepsilon(\mathbf{c}_j + R_j \mathbf{x})$. Further, letting $\mathbf{p} = \mathbf{c}_j + R_j \mathbf{x} \in \Omega_j \subset \Omega_0$, we have

$$\nabla \Phi(\mathbf{c}_j + R_j \mathbf{x}) = R_j \nabla_{\mathbf{p}} \Phi(\mathbf{p}) = R_j \sum_{i=1}^M \nabla_{\mathbf{p}} \frac{q_i}{|\mathbf{p} - \mathbf{c}_i|} = R_j \sum_{i=1}^M q_i \left(-\frac{\mathbf{p} - \mathbf{c}_i}{|\mathbf{p} - \mathbf{c}_i|^3} \right) \quad (\text{C.3})$$

and

$$\begin{aligned} \nabla \hat{u}_1(\mathbf{x}) &= \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} [\phi_1]_{\ell}^m \nabla_{\mathbf{x}} (r^{\ell} Y_{\ell}^m(\theta, \varphi)) \\ &= \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} [\phi_1]_{\ell}^m \left(\ell r^{\ell-1} Y_{\ell}^m, r^{\ell-1} \frac{d}{d\theta} Y_{\ell}^m, \frac{r^{\ell-1}}{\sin \theta} \frac{d}{d\varphi} Y_{\ell}^m \right) \mathbf{J}, \end{aligned} \quad (\text{C.4})$$

where $[\phi_1]_{\ell}^m$ is the coefficient of Y_{ℓ}^m in the spherical harmonic expansion of ϕ_1 , \mathbf{J} is the Jacobian matrix from the spherical coordinates to the Cartesian coordinates, defined by

$$\mathbf{J} = \begin{pmatrix} \sin \theta \cos \varphi & \sin \theta \sin \varphi & \cos \theta \\ \cos \theta \cos \varphi & \cos \theta \sin \varphi & -\sin \theta \\ -\sin \varphi & \cos \varphi & 0 \end{pmatrix} \quad (\text{C.5})$$

and further,

$$\nabla \varepsilon_1(\mathbf{x}) = \nabla_{\mathbf{x}} \varepsilon(\mathbf{c}_j + R_j \mathbf{x}) = R_j \nabla_{\mathbf{p}} \varepsilon(\mathbf{p}). \quad (\text{C.6})$$

The remaining problem is to compute $\nabla_{\mathbf{p}} \varepsilon(\mathbf{p})$ where $\mathbf{p} \in \Omega_0$ and this will be discussed as following.

Since $\mathbf{p} \in \Omega_0$, we know that $f_{\text{sas}}(\mathbf{p}) < r_0$. In the case when $f_{\text{sas}}(\mathbf{p}) \leq -r_p$, implying that \mathbf{p} lies in the SES-cavity Ω_{ses} , we have $\varepsilon(\mathbf{p}) = 1$ and consequently $\nabla_{\mathbf{p}} \varepsilon(\mathbf{p}) = 0$. In the other case when $-r_p < f_{\text{sas}}(\mathbf{p}) < r_0$, implying that \mathbf{p} lies in the dielectric boundary layer $\Omega_0 \setminus \Omega_{\text{ses}}$, we can compute

$$\nabla_{\mathbf{p}} \varepsilon(\mathbf{p}) = \nabla_{\mathbf{p}} H \left(\frac{f_{\text{sas}}(\mathbf{p}) + r_p}{r_p + r_0} \right) = \frac{H'(\xi)}{r_p + r_0} \nabla_{\mathbf{p}} f_{\text{sas}}(\mathbf{p}), \quad (\text{C.7})$$

where $\xi = \frac{f_{\text{sas}}(\mathbf{p}) + r_p}{r_p + r_0} \in (0, 1)$. In the paper [93], a method of computing analytically one closest point \mathbf{x}_p on Γ_{sas} to \mathbf{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

$$f_{\text{sas}}(\mathbf{p}) = \begin{cases} -|\mathbf{x}_p - \mathbf{p}| & \text{if } \mathbf{x}_p \text{ is inside } \Gamma_{\text{sas}}, \\ |\mathbf{x}_p - \mathbf{p}| & \text{if } \mathbf{x}_p \text{ is outside } \Gamma_{\text{sas}}, \end{cases} \quad (\text{C.8})$$

and

$$\nabla_{\mathbf{p}} f_{\text{sas}}(\mathbf{p}) = \begin{cases} \frac{\mathbf{x}_p - \mathbf{p}}{|\mathbf{x}_p - \mathbf{p}|} & \text{if } \mathbf{x}_p \text{ is inside } \Gamma_{\text{sas}}, \\ -\frac{\mathbf{x}_p - \mathbf{p}}{|\mathbf{x}_p - \mathbf{p}|} & \text{if } \mathbf{x}_p \text{ is outside } \Gamma_{\text{sas}}, \end{cases} \quad (\text{C.9})$$

which enables the computation of $\nabla_{\mathbf{p}} \varepsilon(\mathbf{p})$ in Eq. (C.7). Combining all the above Eqs (C.1)–(C.9), we can therefore compute the function value $f(\mathbf{x})$.

Appendix D

Computation of \mathbf{C}_1 , \mathbf{C}_2 , F_0

For each Γ_i^e , we first define a square matrix $\mathbf{P}_{\chi_i^e}$ of dimension $(\ell_{\max}+1)^2 \times (\ell_{\max}+1)^2$ for each χ_i^e , the $(\ell_0 m_0, \ell' m')$ -th element of which is defined as

$$[\mathbf{P}_{\chi_i^e}]_{\ell_0 m_0}^{\ell' m'} := \sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} \chi_i^e(\mathbf{x}_i + r_i \mathbf{s}_n) Y_{\ell_0}^{m_0}(\mathbf{s}_n) Y_{\ell'}^{m'}(\mathbf{s}_n), \quad (\text{D.1})$$

where $0 \leq \ell_0 \leq \ell_{\max}, -\ell_0 \leq m_0 \leq \ell_0, 0 \leq \ell' \leq \ell_{\max}, -\ell' \leq m' \leq \ell'$. Based on Eq. (4.6.15), we can approximate $\chi_i^e \partial_{\mathbf{n}} \psi_r$ (defined on Γ_i) by a 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

$$\chi_i^e \partial_{\mathbf{n}} \psi_r(r_i, \mathbf{s}) = \sum_{\ell_0=0}^{\ell_{\max}} \sum_{m_0=-\ell_0}^{\ell_0} c_{r, \ell_0 m_0} Y_{\ell_0}^{m_0}(\mathbf{s}), \quad \mathbf{s} \in \mathbb{S}^2, \quad (\text{D.2})$$

where the coefficient $c_{r, \ell_0 m_0}$ is computed by the Lebedev quadrature rule as follows

$$c_{r, \ell_0 m_0} = \sum_{\ell'=0}^{\ell_{\max}} \sum_{m'=-\ell'}^{\ell'} [\mathbf{P}_{\chi_i^e}]_{\ell_0 m_0}^{\ell' m'} \frac{\ell'}{r_i} [X_r]_{i \ell' m'}. \quad (\text{D.3})$$

Remark D.0.1. By writing $\chi_i^e \partial_{\mathbf{n}} \psi_r$ as a linear combination of spherical harmonics, the single-layer potential $\tilde{\mathcal{S}}_{\kappa, \Gamma_i}$ can act on it conveniently.

For an arbitrary Lebedev point $\mathbf{x}_j + r_j \mathbf{s}_n = \mathbf{x}_i + r_{ijn} \mathbf{s}_{ijn} \in \Gamma_j^e$, we can then compute

as follows

$$\begin{aligned}
(\widetilde{\mathcal{S}}_{\kappa, \Gamma_i} \chi_i^e \partial_{\mathbf{n}} \psi_r) (\mathbf{x}_j + r_j \mathbf{s}_n) &= \sum_{\ell_0=0}^{\ell_{\max}} \sum_{m_0=-\ell_0}^{\ell_0} c_{r, \ell_0 m_0} (\widetilde{\mathcal{S}}_{\kappa, \Gamma_i} Y_{\ell_0}^{m_0}) (\mathbf{x}_j + r_j \mathbf{s}_n) \\
&= \sum_{\ell_0=0}^{\ell_{\max}} \sum_{m_0=-\ell_0}^{\ell_0} c_{r, \ell_0 m_0} \left(\frac{\kappa i'_{\ell_0}(\kappa r_i)}{i_{\ell_0}(\kappa r_i)} - \frac{\kappa k'_{\ell_0}(\kappa r_i)}{k_{\ell_0}(\kappa r_i)} \right)^{-1} \frac{k_{\ell_0}(\kappa r_{ijn})}{k_{\ell_0}(\kappa r_i)} Y_{\ell_0}^{m_0}(\mathbf{s}_{ijn}) \\
&= \sum_{\ell'=0}^{\ell_{\max}} \sum_{m'=-\ell'}^{\ell'} [\mathbf{Q}]_{i\ell'm'}^{jn} \frac{\ell'}{r_i} [X_r]_{i\ell'm'},
\end{aligned} \tag{D.4}$$

where \mathbf{Q} is a matrix of dimension $M(\ell_{\max} + 1)^2 \times MN_{\text{leb}}$, with the $(i\ell'm', jn)$ -th element defined by

$$[\mathbf{Q}]_{i\ell'm'}^{jn} := \sum_{\ell_0=0}^{\ell_{\max}} \sum_{m_0=-\ell_0}^{\ell_0} [\mathbf{P}_{\chi_i^e}]_{\ell_0 m_0}^{\ell'm'} \left(\frac{\kappa i'_{\ell_0}(\kappa r_i)}{i_{\ell_0}(\kappa r_i)} - \frac{\kappa k'_{\ell_0}(\kappa r_i)}{k_{\ell_0}(\kappa r_i)} \right)^{-1} \frac{k_{\ell_0}(\kappa r_{ijn})}{k_{\ell_0}(\kappa r_i)} Y_{\ell_0}^{m_0}(\mathbf{s}_{ijn}). \tag{D.5}$$

Therefore, we have the $(j\ell m, i\ell'm')$ -th element of \mathbf{C}_1 as follows

$$[\mathbf{C}_1]_{j\ell m}^{i\ell'm'} = \frac{\varepsilon_1}{\varepsilon_2} \left(\sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} \chi_j^e(\mathbf{x}_j + r_j \mathbf{s}_n) Y_{\ell}^m(\mathbf{s}_n) [\mathbf{Q}]_{i\ell'm'}^{jn} \frac{\ell'}{r_i} \right). \tag{D.6}$$

Similarly, based on Eq. (4.6.16), we can approximate $\chi_i^e \partial_{\mathbf{n}} \psi_e$ (defined on Γ_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

$$\chi_i^e \partial_{\mathbf{n}} \psi_e(r_i, \mathbf{s}) = \sum_{\ell_0=0}^{\ell_{\max}} \sum_{m_0=-\ell_0}^{\ell_0} c_{e, \ell_0 m_0} Y_{\ell_0}^{m_0}(\mathbf{s}), \quad \mathbf{s} \in \mathbb{S}^2, \tag{D.7}$$

where

$$c_{e, \ell_0 m_0} = \sum_{\ell'=0}^{\ell_{\max}} \sum_{m'=-\ell'}^{\ell'} [\mathbf{P}_{\chi_i^e}]_{\ell_0 m_0}^{\ell'm'} \frac{\kappa i'_{\ell'}(\kappa r_i)}{i_{\ell'}(\kappa r_i)} [X_e]_{i\ell'm'}. \tag{D.8}$$

For an arbitrary Lebedev point $\mathbf{x}_j + r_j \mathbf{s}_n = \mathbf{x}_i + r_{ijn} \mathbf{s}_{ijn} \in \Gamma_j^e$, we can then compute

$$\begin{aligned}
(\widetilde{\mathcal{S}}_{\kappa, \Gamma_i} \chi_i^e \partial_{\mathbf{n}} \psi_e) (\mathbf{x}_j + r_j \mathbf{s}_n) &= \sum_{\ell_0=0}^{\ell_{\max}} \sum_{m_0=-\ell_0}^{\ell_0} c_{e, \ell_0 m_0} (\widetilde{\mathcal{S}}_{\kappa, \Gamma_i} Y_{\ell_0}^{m_0}) (\mathbf{x}_j + r_j \mathbf{s}_n) \\
&= \sum_{\ell_0=0}^{\ell_{\max}} \sum_{m_0=-\ell_0}^{\ell_0} c_{e, \ell_0 m_0} \left(\frac{\kappa i'_{\ell_0}(\kappa r_i)}{i_{\ell_0}(\kappa r_i)} - \frac{\kappa k'_{\ell_0}(\kappa r_i)}{k_{\ell_0}(\kappa r_i)} \right)^{-1} \frac{k_{\ell_0}(\kappa r_{ijn})}{k_{\ell_0}(\kappa r_i)} Y_{\ell_0}^{m_0}(\mathbf{s}_{ijn}) \\
&= \sum_{\ell'=0}^{\ell_{\max}} \sum_{m'=-\ell'}^{\ell'} [\mathbf{Q}]_{i\ell'm'}^{jn} \frac{\kappa i'_{\ell'}(\kappa r_i)}{i_{\ell'}(\kappa r_i)} [X_e]_{i\ell'm'}.
\end{aligned} \tag{D.9}$$

This yields that

$$[\mathbf{C}_2]_{j\ell m}^{i\ell' m'} = - \left(\sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} \chi_j^e(\mathbf{x}_j + r_j \mathbf{s}_n) Y_\ell^m(\mathbf{s}_n) [\mathbf{Q}]_{i\ell' m'}^{jn} \frac{\kappa i'_{\ell'}(\kappa r_i)}{i_{\ell'}(\kappa r_i)} \right). \quad (\text{D.10})$$

In addition, since $\partial_{\mathbf{n}} \psi_0$ is known, for an arbitrary Lebedev point $\mathbf{x}_j + r_j \mathbf{s}_n = \mathbf{x}_i + r_{ijn} \mathbf{s}_{ijn} \in \Gamma_j^e$, we can compute the following column vector S

$$\begin{aligned} [S]_{ijn} &= (\tilde{\mathcal{S}}_{\kappa, \Gamma_i} \chi_i^e \partial_{\mathbf{n}} \psi_0)(\mathbf{x}_j + r_j \mathbf{s}_n) \\ &= \sum_{\ell_0=0}^{\ell_{\max}} \sum_{m=-\ell_0}^{\ell_0} c_{0,\ell_0 m} (\tilde{\mathcal{S}}_{\kappa, \Gamma_i} Y_{\ell_0}^{m_0})(\mathbf{x}_j + r_j \mathbf{s}_n) \\ &= \sum_{\ell_0=0}^{\ell_{\max}} \sum_{m=-\ell_0}^{\ell_0} c_{0,\ell_0 m} \left(\frac{\kappa i'_{\ell_0}(\kappa r_i)}{i_{\ell_0}(\kappa r_i)} - \frac{\kappa k'_{\ell_0}(\kappa r_i)}{k_{\ell_0}(\kappa r_i)} \right)^{-1} \frac{k_{\ell_0}(\kappa r_{ijn})}{k_{\ell_0}(\kappa r_i)} Y_{\ell_0}^{m_0}(\mathbf{s}_{ijn}), \end{aligned} \quad (\text{D.11})$$

where

$$c_{0,\ell_0 m_0} = \sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} \chi_i^e(\mathbf{x}_i + r_i \mathbf{s}_n) \partial_{\mathbf{n}} \psi_0(\mathbf{x}_i + r_i \mathbf{s}_n) Y_{\ell_0}^{m_0}(\mathbf{s}_n). \quad (\text{D.12})$$

This yields that

$$[F_0]_{j\ell m} = -\frac{\varepsilon_1}{\varepsilon_2} \left(\sum_{n=1}^{N_{\text{leb}}} w_n^{\text{leb}} \chi_j^e(\mathbf{x}_j + r_j \mathbf{s}_n) Y_\ell^m(\mathbf{s}_n) \sum_{i=1}^M [S]_{ijn} \right). \quad (\text{D.13})$$

Remark D.0.2. We first compute $\mathbf{P}_{\chi_i^e}$ a priori following (D.1) for each sphere Γ_i^e , the complexity of which is $O(M N_{\text{leb}} (\ell_{\max} + 1)^4)$. Then, based on $\mathbf{P}_{\chi_i^e}$, we compute the matrix \mathbf{Q} following (D.5) with a complexity $O(M^2 N_{\text{leb}} (\ell_{\max} + 1)^4)$. After that, we can compute \mathbf{C}_1 and \mathbf{C}_2 according to (D.6) and (D.10) with a complexity of $O(M^2 N_{\text{leb}} (\ell_{\max} + 1)^4)$. To compute the column vector F_0 , we first compute S a priori with a complexity of $O(M^2 N_{\text{leb}} (\ell_{\max} + 1)^2)$. Then, based on S , the complexity of computing F_0 according to (D.13) is reduced to $O(M^2 N_{\text{leb}} (\ell_{\max} + 1)^2)$. In summary, the total complexity of computing the global linear system (4.6.31) is $O(M^2 N_{\text{leb}} (\ell_{\max} + 1)^4)$.

Bibliography

- [1] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation, 1964.
- [2] Robert A Adams and John JF Fournier. *Sobolev spaces*, volume 140. Academic Press, 2003.
- [3] Burak Aksoylu, Stephen D. Bond, Eric C. Cyr, and Michael Holst. Goal-Oriented Adaptivity and Multilevel Preconditioning for the Poisson-Boltzmann Equation. *Journal of Scientific Computing*, 52(1):202–225, 2012.
- [4] Michael D Altman, Jaydeep P Bardhan, Jacob K White, and Bruce Tidor. Accurate solution of multi-region continuum biomolecule electrostatic problems using the linearized Poisson-Boltzmann equation with curved boundary elements. *Journal of Computational Chemistry*, 30(1):132–153, 2009.
- [5] G Arfken, H Weber, and FE Harris. Mathematical Methods for Physicists: A Comprehensive Guide 7th edn (London: Academic). 2012.
- [6] Franz Aurenhammer. Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96, 1987.
- [7] David Avis, Binay K Bhattacharya, and Hiroshi Imai. Computing the volume of the union of spheres. *Visual Computer*, 3(6):323–328, 1988.
- [8] Chandrajit Bajaj, Shun-Chuan Chen, and Alexander Rand. An efficient higher-order fast multipole boundary element solution for Poisson-Boltzmann-based molecular electrostatics. *SIAM Journal on Scientific Computing*, 33(2):826–848, 2011.
- [9] Nathan A Baker, David Sept, Simpson Joseph, Michael J Holst, and J Andrew McCammon. Electrostatics of nanosystems: application to microtubules and the ribosome. *Proceedings of the National Academy of Sciences*, 98(18):10037–10041, 2001.
- [10] L. Banjai. Boundary element methods, October 2007.

- [11] Jaydeep P Bardhan. Numerical solution of boundary-integral equations for molecular electrostatics. *Journal of Chemical Physics*, 130(9):094102, 2009.
- [12] J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324:446–449, December 1986.
- [13] Mikhail V Basilevsky, Fedor V Grigoriev, Ekaterina A Nikitina, and Jerzy Leszczynski. Implicit electrostatic solvent model with continuous dielectric permittivity function. *Journal of Physical Chemistry B*, 114(7):2457–2466, 2010.
- [14] CASPER HL Beentjes. Quadrature on a spherical surface. *Working note available on the website <http://people.maths.ox.ac.uk/beentjes/Essays>*, 2015.
- [15] Alexander H Boschitsch, Marcia O Fenley, and Huan-Xiang Zhou. Fast boundary element method for the linear Poisson-Boltzmann equation. *Journal of Physical Chemistry B*, 106(10):2741–2754, 2002.
- [16] Tahar Zamène Boulmezaoud. Inverted finite elements: a new method for solving elliptic problems in unbounded domains. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39(1):109–145, 2005.
- [17] Monica Bugeanu, Roberto Di Remigio, Krzysztof Mozgawa, Simen Sommerfelt Reine, Helmut Harbrecht, and Luca Frediani. Wavelet formulation of the polarizable continuum model. II. Use of piecewise bilinear boundary elements. *Phys. Chem. Chem. Phys.*, 17:31566–31581, 2015.
- [18] Felipe A Bulat, Alejandro Toro-Labbé, Tore Brinck, Jane S Murray, and Peter Politzer. Quantitative analysis of molecular surfaces: areas, volumes, electrostatic potentials and average local ionization energies. *Journal of molecular modeling*, 16(11):1679–1691, 2010.
- [19] Ján Buša, Jozef Džurina, Edik Hayryan, Shura Hayryan, Chin-Kun Hu, Ján Plavka, Imrich Pokorný, Jaroslav Skřivánek, and Ming-Chya Wu. ARVO: a Fortran package for computing the solvent accessible surface area and the excluded volume of overlapping spheres via analytic equations. *Computer physics communications*, 165(1):59–96, 2005.
- [20] James C Caendish, David A Field, and William H Frey. An apporach to automatic three-dimensional finite element mesh generation. *International journal for numerical methods in engineering*, 21(2):329–347, 1985.
- [21] Roberto Cammi and Benedetta Mennucci. *Continuum Solvation Models in Chemical Physics: From Theory to Applications*. John Wiley, 2007.
- [22] E. Cancès, B. Mennucci, and J. Tomasi. A new integral equation formalism for the polarizable continuum model: Theoretical background and applications to

- isotropic and anisotropic dielectrics. *Journal of Chemical Physics*, 107(8):3032–3041, 1997.
- [23] Eric Cancès, Mireille Defranceschi, Werner Kutzelnigg, Claude Le Bris, and Yvon Maday. Computational quantum chemistry: a primer. In *Handb. Numer. Anal., X*, pages 3–270. North-Holland, Amsterdam, 2003.
 - [24] Eric Cancès and Claude Le Bris. Mathematical modeling of point defects in materials science. *Mathematical Models & Methods in Applied Sciences*, 23(10):1795–1859, 2013.
 - [25] Eric Cancès, Yvon Maday, and Benjamin Stamm. Domain decomposition for implicit solvation models. *Journal of Chemical Physics*, 139(5):054111, 2013.
 - [26] Eric Cancès and Benedetta Mennucci. New applications of integral equations methods for solvation continuum models: ionic solutions and liquid crystals. *Journal of Mathematical Chemistry*, 23(3-4):309–326, 1998.
 - [27] Frederic Cazals, Harshad Kanhere, and Sébastien Loriot. Computing the volume of a union of balls: a certified algorithm. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):3, 2011.
 - [28] Calvin Chen and George Makhatadze. ProteinVolume: calculating molecular van der Waals and void volumes in proteins. *BMC Bioinformatics*, 16(1):101, 2015.
 - [29] G. Ciaramella and M. J. Gander. Analysis of the parallel Schwarz method for growing chains of fixed-size subdomains: part I. *Preprint. To appear in SIAM J. Numer. Anal.*, 2016.
 - [30] M. L. Connolly. Analytical molecular surface calculation. *Journal of Applied Crystallography*, 16(5):548–558, Oct 1983.
 - [31] Charles Dapogny, Cécile Dobrzynski, and Pascal Frey. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *Journal of Computational Physics*, 262:358–378, 2014.
 - [32] Malcolm E Davis, Jeffry D Madura, Brock A Luty, and J Andrew McCammon. Electrostatics and diffusion of molecules in solution: simulations with the University of Houston Brownian dynamics program. *Computer Physics Communications*, 62(2-3):187–197, 1991.
 - [33] Malcolm E Davis and J Andrew McCammon. Electrostatics in biomolecular structure and dynamics. *Chemical Reviews*, 90(3):509–521, 1990.

- [34] Michel C Delfour and J-P Zolésio. *Shapes and geometries: metrics, analysis, differential calculus, and optimization*, volume 22. Siam, 2011.
- [35] Manfredo Perdigao Do Carmo and Manfredo Perdigao Do Carmo. *Differential geometry of curves and surfaces*, volume 2. Prentice-hall Englewood Cliffs, 1976.
- [36] Todd J Dolinsky, Paul Czodrowski, Hui Li, Jens E Nielsen, Jan H Jensen, Gerhard Klebe, and Nathan A Baker. PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations. *Nucleic Acids Research*, 35(Web Server issue):W522–W525, 2007.
- [37] Robert Lewis (Scot) Drysdale, III. *Generalized Voronoi Diagrams and Geometric Searching*. PhD thesis, Stanford University, Stanford, CA, USA, 1979. AAI7917225.
- [38] Herbert Federer. The Gauss-Green theorem. *Transactions of the American Mathematical Society*, pages 44–76, 1945.
- [39] G. Fisicaro, L. Genovese, O. Andreussi, N. Marzari, and S. Goedecker. A generalized Poisson and Poisson-Boltzmann solver for electrostatic environments. *Journal of Chemical Physics*, 144(1):014103, 2016.
- [40] Federico Fogolari, Alessandro Brigo, and Henriette Molinari. The Poisson-Boltzmann equation for biomolecular electrostatics: a tool for structural biology. *Journal of Molecular Recognition*, 15(6):377–392, 2002.
- [41] Pascal J Frey, Houman Borouchaki, and Paul-Louis George. 3D Delaunay mesh generation coupled with an advancing-front approach. *Computer Methods in Applied Mechanics and Engineering*, 157(1):115–131, 1998.
- [42] Pascal Jean Frey and Paul-Louis George. *Mesh Generation: Application to Finite Elements*. ISTE, 2007.
- [43] William H Frey. Selective refinement: a new strategy for automatic node placement in graded triangular meshes. *International Journal for Numerical Methods in Engineering*, 24(11):2183–2200, 1987.
- [44] Aeleen Frisch. *Gaussian 03. U. User's reference: manual version 7.1 (corresponding to Gaussian 03 revision D. 1)*. Gaussian, 2005.
- [45] Paul Louis George and Éric Seveno. The advancing-front mesh generation method revisited. *International Journal for Numerical Methods in Engineering*, 37(21):3605–3619, 1994.
- [46] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

- [47] J Andrew Grant, Barry T Pickup, and Anthony Nicholls. A smooth permittivity function for Poisson-Boltzmann solvation methods. *Journal of Computational Chemistry*, 22(6):608–640, 2001.
- [48] David Griffiths. *Introduction to elementary particles*. John Wiley & Sons, 2008.
- [49] Wolfgang Hackbusch. *Integral equations: theory and numerical treatment*, volume 120. Birkhäuser, 2012.
- [50] Helmut Harbrecht and Maharavo Randrianarivony. Wavelet BEM on molecular surfaces: solvent excluded surfaces. *Computing*, 92(4):335–364, 2011.
- [51] Daniel J Haxton. Lebedev discrete variable representation. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 40(23):4443, 2007.
- [52] Pedro Hermosilla, Michael Krone, Victor Guallar, Pere-Pau Vázquez, Àlvar Vinacua, and Timo Ropinski. Interactive GPU-based generation of solvent-excluded surfaces. *Visual Computer*, pages 1–13, 2017.
- [53] Michael Holst, James Andrew Mccammon, Zeyun Yu, YC Zhou, and Yunrong Zhu. Adaptive finite element modeling techniques for the Poisson-Boltzmann equation. *Communications in computational physics*, 11(1):179–214, 2012.
- [54] Michael J Holst et al. The Poisson-Boltzmann equation: Analysis and multilevel numerical solution. 1994.
- [55] William Humphrey, Andrew Dalke, and Klaus Schulten. VMD: visual molecular dynamics. *Journal of Molecular Graphics*, 14(1):33–38, 1996.
- [56] Wonpil Im, Dmitrii Beglov, and Benoit Roux. Continuum solvation model: computation of electrostatic forces from numerical solutions to the Poisson–Boltzmann equation. *Computer Physics Communications*, 111(1):59–75, 1998.
- [57] Richard M Jackson and Michael JE Sternberg. A continuum model for protein–protein interactions: application to the docking problem. *Journal of Molecular Biology*, 250(2):258–275, 1995.
- [58] Elizabeth Jurrus, Dave Engel, Keith Star, Kyle Monson, Juan Brandi, Lisa E Felberg, David H Brookes, Leighton Wilson, Jiahui Chen, Karina Liles, et al. Improvements to the APBS biomolecular solvation software suite. *Protein Science*, 2017.
- [59] Daniel Kauker, Michael Krone, Alexandros Panagiotidis, Guido Reina, and Thomas Ertl. Rendering molecular surfaces using order-independent transparency. In *Proceedings of the 13th Eurographics Symposium on Parallel Graphics and Visualization*, pages 33–40. Eurographics Association, 2013.

- [60] Andreas Klamt and GJGJ Schüürmann. COSMO: a new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient. *Journal of the Chemical Society, Perkin Transactions 2*, (5):799–805, 1993.
- [61] Michael Krone, Katrin Bidmon, and Thomas Ertl. Interactive visualization of molecular surface dynamics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1391–1398, 2009.
- [62] Shihhsien S Kuo, Michael D Altman, Jaydeep P Bardhan, Bruce Tidor, and Jacob K White. Fast methods for simulation of biomolecule electrostatics. In *Proceedings of the 2002 IEEE/ACM International Conference on Computer-aided Design*, pages 466–473. ACM, 2002.
- [63] Gene Lamm. The Poisson-Boltzmann equation. *Reviews in Computational Chemistry*, 19:147–333, 2003.
- [64] Roman A Laskowski. SURFNET: a program for visualizing molecular surfaces, cavities, and intermolecular interactions. *Journal of Molecular Graphics*, 13(5):323–330, 1995.
- [65] Patrick Laug and Houman Borouchaki. Molecular surface modeling and meshing. *Engineering with Computers*, 18(3):199–210, 2002.
- [66] VI Lebedev and DN Laikov. A quadrature formula for the sphere of the 131st algebraic order of accuracy. In *Doklady. Mathematics*, volume 59, pages 477–481. MAIK Nauka/Interperiodica, 1999.
- [67] Byungkook Lee and Frederic M Richards. The interpretation of protein structures: estimation of static accessibility. *Journal of Molecular Biology*, 55(3):379–IN4, 1971.
- [68] Lin Li, Chuan Li, Subhra Sarkar, Jie Zhang, Shawn Witham, Zhe Zhang, Lin Wang, Nicholas Smith, Marharyta Petukh, and Emil Alexov. DelPhi: a comprehensive suite for DelPhi software and associated resources. *BMC Biophysics*, 5(1):9, 2012.
- [69] Filippo Lipparini, Louis Lagardère, Giovanni Scalmani, Benjamin Stamm, Eric Cancès, Yvon Maday, Jean-Philip Piquemal, Michael J Frisch, and Benedetta Mennucci. Quantum calculations in solution for large to very large molecules: A new linear scaling QM/continuum approach. *Journal of Physical Chemistry Letters*, 5(6):953–958, 2014.
- [70] Filippo Lipparini, Louis Lagardère, Benjamin Stamm, Eric Cancès, Yvon Maday, Jean-Philip Piquemal, and Benedetta Mennucci. ddCOSMO & ddPCM. <https://www.ddpcm.org>, 2015.

- [71] Filippo Lipparini, Giovanni Scalmani, Louis Lagardère, Benjamin Stamm, Eric Cancès, Yvon Maday, Jean-Philip Piquemal, Michael J Frisch, and Benedetta Mennucci. Quantum, classical, and hybrid QM/MM calculations in solution: General implementation of the ddCOSMO linear scaling strategy. *Journal of Chemical Physics*, 141(18):184108, 2014.
- [72] Filippo Lipparini, Benjamin Stamm, Eric Cancès, Yvon Maday, and Benedetta Mennucci. Fast domain decomposition algorithm for continuum solvation models: Energy and first derivatives. *Journal of Chemical Theory and Computation*, 9(8):3637–3648, 2013.
- [73] Beibei Liu, Bao Wang, Rundong Zhao, Yiyi Tong, and Guo-Wei Wei. ESES: Software for Eulerian solvent excluded surface. *Journal of Computational Chemistry*, 2017.
- [74] Beibei Liu, Bao Wang, Rundong Zhao, Yiyi Tong, and Guo-Wei Wei. ESES: Software for Eulerian solvent excluded surface. *Journal of Computational Chemistry*, 38(7):446–466, 2017.
- [75] Rainald Löhner. Progress in grid generation via the advancing front technique. *Engineering with Computers*, 12(3-4):186–210, 1996.
- [76] Anne Lopes, Sophie Sacquin-Mora, Viktoriya Dimitrova, Elodie Laine, Yann Ponty, and Alessandra Carbone. Protein-protein interactions in a crowded environment: an analysis via cross-docking simulations and evolutionary information. *PLoS Comput Biol*, 9(12):e1003369, 2013.
- [77] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *ACM Siggraph Computer Graphics*, volume 21, pages 163–169. ACM, 1987.
- [78] Benzhuo Lu, Xiaolin Cheng, Jingfang Huang, and J Andrew McCammon. An Adaptive Fast Multipole Boundary Element Method for Poisson-Boltzmann Electrostatics. *Journal of Chemical Theory and Computation*, 5(6):1692–1699, 2009.
- [79] BZ Lu, YC Zhou, MJ Holst, and JA McCammon. Recent progress in numerical methods for the Poisson-Boltzmann equation in biophysical applications. *Communications in Computational Physics*, 3(5):973–1009, 2008.
- [80] Buddhadeb Mallik, Artem Masunov, and Themis Lazaridis. Distance and exposure dependent effective dielectric function. *Journal of Computational Chemistry*, 23(11):1090–1099, 2002.

- [81] B. Mennucci, E. Cancès, and J. Tomasi. Evaluation of solvent effects in isotropic and anisotropic dielectrics and in ionic solutions with a unified integral equation method: theoretical bases, computational implementation, and numerical applications. *Journal of Physical Chemistry B*, 101(49):10506–10517, 1997.
- [82] Benedetta Mennucci. Continuum solvation models: What else can we learn from them? *Journal of Physical Chemistry Letters*, 1(10):1666–1674, 2010.
- [83] Benedetta Mennucci and Roberto Cammi. *Continuum solvation models in chemical physics: from theory to applications*. John Wiley & Sons, 2008.
- [84] S Miertuš, E Scrocco, and J Tomasi. Electrostatic interaction of a solute with a continuum. A direct utilization of AB initio molecular potentials for the prevision of solvent effects. *Chemical Physics*, 55(1):117–129, 1981.
- [85] Peter Möller and Peter Hansbo. On advancing front mesh generation in three dimensions. *International Journal for Numerical Methods in Engineering*, 38(21):3551–3569, 1995.
- [86] Anthony Nicholls and Barry Honig. A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson-Boltzmann equation. *Journal of Computational Chemistry*, 12(4):435–445, 1991.
- [87] William H Orttung. Direct solution of the Poisson equation for biomolecules of arbitrary shape, polarizability density, and charge distribution. *Annals of the New York Academy of Sciences*, 303(1):22–37, 1977.
- [88] Seymour V Parter. On the Legendre-Gauss-Lobatto points and weights. *Journal of Scientific Computing*, 14(4):347–355, 1999.
- [89] Julius Parulek and Ivan Viola. Implicit representation of molecular surfaces. In *Pacific Visualization Symposium (PacificVis), 2012 IEEE*, pages 217–224. IEEE, 2012.
- [90] Christian S Pomelli and Jacopo Tomasi. DefPol: New procedure to build molecular surfaces and its use in continuum solvation methods. *Journal of Computational Chemistry*, 19(15):1758–1776, 1998.
- [91] Makenzie R Provorose, Thomas Peev, Chou Xiong, and Christine M Isborn. Convergence of excitation energies in mixed quantum and classical solvent: Comparison of continuum and point charge models. *Journal of Physical Chemistry B*, 120(47):12148–12159, 2016.
- [92] Zhong-hua Qiao, Zhi-lin Li, and Tao Tang. A finite difference scheme for solving the nonlinear Poisson-Boltzmann equation modeling charged spheres. *Journal of Computational Mathematics*, pages 252–264, 2006.

- [93] Chaoyu Quan and Benjamin Stamm. Mathematical analysis and calculation of molecular surfaces. *Journal of Computational Physics*, 322:760 – 782, 2016.
- [94] Chaoyu Quan and Benjamin Stamm. Meshing molecular surfaces based on analytical implicit representation. *Journal of Molecular Graphics and Modelling*, 71:200–210, 2017.
- [95] Chaoyu Quan, Benjamin Stamm, and Yvon Maday. Polarizable continuum model based on the solvent excluded surface. HAL-01489262, March 2017.
- [96] Alfio Quarteroni and Alberto Valli. *Domain decomposition methods for partial differential equations*. Number CMCS-BOOK-2009-019. Oxford University Press, 1999.
- [97] Anthony K Rappé, Carla J Casewit, KS Colwell, WA Goddard Iii, and WM Skiff. UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American Chemical Society*, 114(25):10024–10035, 1992.
- [98] Tanya M Raschke, Jerry Tsai, and Michael Levitt. Quantification of the hydrophobic interaction by simulations of the aggregation of small hydrophobic solutes in water. *Proceedings of the National Academy of Sciences*, 98(11):5965–5969, 2001.
- [99] Frederic M Richards. Areas, volumes, packing and protein structure. *Annual Review of Biophysics and Bioengineering*, 6:151–176, 1977.
- [100] Timothy J Richmond. Solvent accessible surface area and excluded volume in proteins: Analytical equations for overlapping spheres and implications for the hydrophobic effect. *Journal of Molecular Biology*, 178(1):63–89, 1984.
- [101] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- [102] Michel F Sanner, Arthur J Olson, and Jean-Claude Spehner. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320, 1996.
- [103] Stefan A Sauter and Christoph Schwab. Boundary element methods. In *Boundary Element Methods*, pages 183–287. Springer, 2010.
- [104] Stefan A. Sauter and Christoph Schwab. *Elliptic Boundary Integral Equations*, pages 101–181. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [105] Joachim Schöberl. NETGEN An advancing front 2D/3D-mesh generator based on abstract rules. *Computing and visualization in science*, 1(1):41–52, 1997.
- [106] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

- [107] Barry Smith, Petter Bjorstad, William D Gropp, and William Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge University Press, 2004.
- [108] Benjamin Stamm, Eric Cancès, Filippo Lipparini, and Yvon Maday. A new discretization for the polarizable continuum model within the domain decomposition paradigm. *Journal of Chemical Physics*, 144(5):054101, 2016.
- [109] Jacopo Tomasi, Benedetta Mennucci, and Roberto Cammi. Quantum mechanical continuum solvation models. *Chemical Reviews*, 105(8):2999–3094, 2005.
- [110] Jacopo Tomasi and Maurizio Persico. Molecular interactions in solution: An overview of methods based on continuous distributions of the solvent. *Chemical Reviews*, 94(7):2027–2094, 1994.
- [111] Oleg V Tsodikov, M Thomas Record, and Yuri V Sergeev. Novel computer program for fast exact calculation of accessible and molecular surface areas and average surface curvature. *Journal of Computational Chemistry*, 23(6):600–609, 2002.
- [112] Luiz Velho, Jonas Gomes, and Luiz H de Figueiredo. *Implicit objects in computer graphics*. Springer Science & Business Media, 2002.
- [113] Neil R Voss and Mark Gerstein. 3V: cavity, channel and cleft volume calculator and extractor. *Nucleic Acids Research*, page gkq395, 2010.
- [114] Wikipedia. Spherical cap - Wikipedia, the free encyclopedia, 2016. [Online; accessed 25-February-2016].
- [115] Wikipedia. Torus - Wikipedia, the free encyclopedia, 2016. [Online; accessed 3-November-2016].
- [116] Byung Jun Yoon and AM Lenhoff. A boundary element method for molecular electrostatics with electrolyte effects. *Journal of Computational Chemistry*, 11(9):1080–1086, 1990.
- [117] Hongkai Zhao. A fast sweeping method for Eikonal equations. *Mathematics of Computation*, 74(250):603–627, 2005.

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.

Mathematical methods for implicit solvation models in quantum chemistry

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 solvatation 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”, c'est-à-dire la surface exclue du solvant (SES). À partie 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 construisons donc un modèle 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 solvatation de Poisson-Boltzmann, un autre modèle de solvatation 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 solvatation implicite, modèle de continuum polarisable, équation de Poisson-Boltzmann, méthode de décomposition du domaine