1-28-2015

Delaunay-Laguerre Geometry For Macromolecular Modeling And Implicit Solvation

Michelle Hummel

Michelle Hatch Hummel
Candidate

Mathematics and Statistics
Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Evangelos Coutsias, Chairperson

Deborah Sulsky

Scott Mitchell

Tudor Oprea

DELAUNAY-LAGUERRE GEOMETRY FOR MACROMOLECULAR MODELING AND IMPLICIT SOLVATION

by

MICHELLE HATCH HUMMEL

B.S. Applied Mathematics, Utah State University, 2007
Ph.D. Mathematics, University of New Mexico, 2014

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Mathematics

The University of New Mexico
Albuquerque, New Mexico

December, 2014

DELAUNAY-LAGUERRE GEOMETRY FOR MACROMOLECULAR MODELING AND IMPLICIT SOLVATION

by

Michelle Hatch Hummel

B.S. Applied Mathematics, Utah State University

Ph.D. Mathematics, University of New Mexico

ABSTRACT

We develop and implement geometric methods to study three-dimensional structures of proteins, the knowledge of which is critical to the understanding of the molecules and their interactions. Delaunay and Laguerre methods, which concern sets of overlapping spheres and their interrelationships, are well suited to the study of molecules. We discuss and implement algorithms for the calculation of molecular volume, atomic solvent accessible surface areas, their gradients and discontinuities. This is used for a detailed analysis of parameters obtained by the implicit solvation method, Semi-Explicit Assembly (SEA). We introduce the concept of Laguerre-Intersection cells which consist of the intersection of the Laguerre tessellation and space-filling diagram. This method eliminates the need for explicit water molecules to cap infinite Laguerre cells of certain solvent accessible solute atoms. We discuss and implement a quick weighted Delaunay tetrahedrization algorithm which is tailored specifically to the aforementioned algorithms. Finally, we use concepts from continuum mechanics to study the motion of the HIV protease dimer.

DEDICATION

To Albert, my parents Kristine and Robert, and my grandparents for their love, support, and encouragement.

ACKNOWLEDGEMENTS

I would like to thank my dissertation advisor, Dr. Evangelos Coutsias, for his instruction, and all others who helped me on the way to graduation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Proteins are involved in a wide variety of processes in a biological system: enzyme catalysis, cell signaling, regulation of DNA replication and repair, immune response, and transportation. The three-dimensional structure of a protein is related to its function. Active sites of enzymes are often in tunnels, pockets, and cavities. Docking of signal molecules, substrates, and other ligands is influenced by shape complementarity of the ligands and biomolecules. In this dissertation, we develop and implement geometric methods to study three-dimensional structures of proteins, the knowledge of which is critical to the understanding of the molecules and their interactions.

Delaunay and Laguerre methods, which concern sets of overlapping spheres and their interrelationships, are well suited to the study of molecules. The Laguerre diagram generalizes the Voronoi diagram. It tesselates a three-dimensional space by assigning a set of nearest neighbors, via the power distance, to each sphere. Contact surfaces between Laguerre cells are along radical planes which partition the space in physically relevant ways including along the plane of intersection of two spheres [3].

The Laguerre tessellation is dual to the weighted Delaunay tetrahedrization which triangulates sets of spheres. The alpha complex, a subcomplex of the Delaunay tetrahedrization, describes interactions between spheres in varying levels of detail [12]. This complex is an essential component of molecular, atomic, and Laguerre volume and surface calculations which are discussed later in this dissertation. The Laguerre tesselation,

Delaunay tetrahedrization, alpha complex, and an original simplex extraction method for the alpha complex calculation are discussed in Chapters 2, 3, 11.

Molecular volume and atomic solvent accessible surface areas and their gradients are useful quantities for molecular modelling. Accessible surface areas are employed in implicit solvation methods, including the calculation of the free energy of solvation [18], [17], [10], [26]. Area gradients are used to calculate forces on a molecule. These gradients entail discontinuities which are discussed in [36], [4], [11]. Molecular volume is important for nonpolar free energy of solvation [17], and assessment of protein packing [23], [24]. Similar methods have also been used to locate and measure voids, cavities, and pockets [25], [8], [9]. Volume, surface, and gradient calculations are discussed and efficient formulas are presented in Chapters 4 and 5, and a quick and accurate implementation of this calculation is presented. An solvation method called Semi-Explicit Assembly (SEA) is discussed in Chapter 6. Chapter 7 analyzes results of SEA using surface area implementation discussed in Chapter 4.

Laguerre tessellation methods have been used to find volumes of atoms, residues, and molecules which are useful in studying packing and deformation of proteins, and in cavity location. They may also be used to check predicted structures as, for example, residue volume may be considered an "intrinsic" property of amino acid type and is thus a predictable or checkable quantity [14]. Laguerre surfaces have been used as a quick and parameter free way to measure accessibility of atoms and residues for quantifying exposure of a molecule with solvent [14]. Residue contacts, which are important for protein structure and stability, have been studied using Laguerre surfaces. These quantities are useful for protein structure analysis and prediction, and in studying structure-function relationships. Preferential contacts between amino acid species and atom-atom contact frequencies have also been found[15], [28]. Differences in contacts between model and reference structures have been used as a scoring function and for benchmarking protein sturcure prediction methods [30]. Surfaces have also been used to characterize interactions between multiple proteins, or proteins and ligands, which has important applications in protein docking and formation of complexes [32], [29].

The Laguerre cell is the region which is closest to the generator of the cell (atom or residue center, for example) by the power distance. The power distance is an appropriate quantity to use as larger areas are assigned to data points with larger weights (larger atoms or residues, for example). The Laguerre cell of a data point depends on its neighbors which poses a problem for systems that are typically found in a medium such as solvent but for which the surrounding medium is not explicitly known. For such data sets, Laguerre cells of points on the convex hull are unbounded and the cells of points otherwise in contact with the surrouding medium tend to be too large.

Researchers have addressed this difficulty in a variety of ways. Some consider only cells in the bulk of the protein which are not affected by the surrounding (unknown) environment. Others surround the structure by a layer of water or an artificial environment of spheres with size equal to the average amino acid size [19], [33]. However, this causes the problem to balloon. Soyer etal, bounded Laguerre cells by only considering Laguerre facet vertices from tetrahedra of a small enough size [34]. Still cells near the boundary tend to be elongated or have fewer facets than those in the bulk of the protein.

McConkey etal construct Laguerre-like cells by considering the surfaces of extended radical contact planes between neighboring atoms within a cutoff distance and the surface of an expanded sphere [28]. This poses problems since the cells of atoms in the bulk, which should not be affected by the cutoff distance, are affected. Furthermore, cells of small atoms completely disappear for cutoff distances as small as 1.4 A. The algorithm also cannot correctly handle cell volumes when the center of the generator of the cell lies outside its cell (See Section 8.3).

The proposed Laguerre-Intersection cell algorithm, which is presented in Chapter 8, considers the intersection of Laguerre cells with expanded atoms. The contact planes are the radical planes, which means that as the solvent weight is varied, Laguerre cells stay constant. This method simulates the environment better than using the extended radical plane. The algorithm also correctly handles the weighted case when the center of the generator of the cell is outside the cell. This has been been a challenge for previous methods. Chapter 9 finds optimal parameters for the Laguerre-Intersection

method. One of the results of this analysis applied to a set of molecular trajectories of the essential amino acids in explicit water, was a stratification of the amino acids according to increasing order of Laguerre parameter (radius or weight) showing a rough correspondence to commonly used "hydrophobicity scales" [40], [37].

Next we discuss Delaunay tetrahedrization methods [5], [31], [21], as the alpha complex on which previously mentioned algorithms rely is derived from this triangulation. In Chapter 12 we present and implement a fast and original algorithm tailored specifically toward volume, surface, and Laguerre- Intersection cell calculations.

Finally, we study the motion of a molecular machine, the HIV protease dimer, with ideas from continuum mechanics. We present an analysis of a molecular dynamics trajectory [1] using strain tensors in Chapter 14.

Part I

Theory: Laguerre Diagram, Weighted Delaunay Tetrahedrization, Alpha Complex

Chapter 2

Delaunay-Laguerre Geometry

2.1 Laguerre Diagram

Let $\mathcal{A} \subset \mathbb{R}^3 \times \mathbb{R}$ be a finite set of weighted points. For $p \in \mathcal{A}$, write $p = (p', p'')$ where $p' \in \mathbb{R}^3$ is the location or center of the point and $p'' \in \mathbb{R}$ is the weight of the point. The point p can be thought of as a closed ball, B_p, in $\mathbb{R}^3$ centered at p' with radius $\sqrt{p''}$. The boundary of this ball is S_p [7],[9]. In our applications, $\mathcal{A}$ represents atoms in a molecule.

Define the *power distance* (Figure 2.2a) between a weighted point p and an unweighted point x' as

$$\pi(p, x') = |p' - x'|^2 - p''. \tag{2.1.1}$$

The *Laguerre cell* of $p \in \mathcal{A}$ is

$$L_p = \{x' \in \mathbb{R}^3 : \pi(p, x') \leq \pi(q, x') \; \forall \; q \in \mathcal{A}\}. \tag{2.1.2}$$

which is the set of points whose nearest neighbor by the power distance is p. The point p is said to be the generator of the cell L_p which is a convex polytope.

The *Laguerre diagram*, $\mathcal{L}(\mathcal{A})$, is the collection of all Laguerre cells and their faces, which we will call Laguerre facets, segments, and nodes. (Figure 2.1)

We assume all the points in $\mathcal{A}$ are in general position, that is no three points are co-

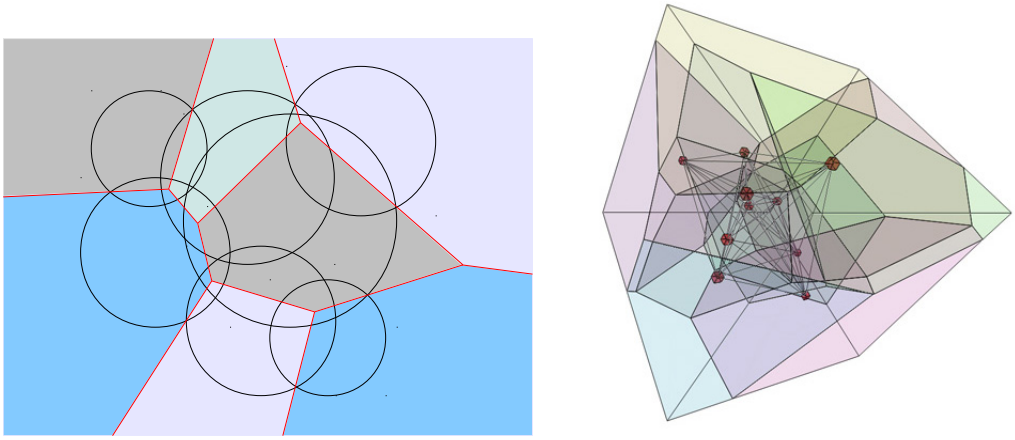


Figure 2.1: Two-dimensional and three-dimensional Laguerre diagrams. The 2d version illustrates how the radical plane partitions space between atoms of different radii. The 3d version is courtesy of F. Vanhoutte [35] .

linear, no four points are cocircular, and no five points are cospherical or are equipowerdistant to a common point. This means that a Laguerre facet is the intersection of two Laguerre cells, a Laguerre segment is the intersection of three Laguerre cells, and a Laguerre node is the intersection of four Laguerre cells. Each Laguerre facet lies on the plane which is equipowerdistant between two points and which is called the radical plane (Figure 2.2b). The points in a Laguerre segment are equipowerdistant to the three points which generate the intersecting Laguerre cells, and a Laguerre node is equipowerdistant to the generators of the four intersecting Laguerre cells [12],[6].

The Laguerre diagram may also be called the *Power diagram* or *weighted Voronoi diagram*. An unweighted point can be thought of weighted point with weight zero. For a set of unweighted points the Voronoi diagram and the Laguerre diagram are identical.

2.2 Delaunay and Regular Tetrahedrizations

We write the set of data point centers as

$$\mathcal{A}' = \{p' \text{ such that } p \in \mathcal{A}\}. \tag{2.2.1}$$

The *weighted Delaunay tetrahedrization* or *regular tetrahedrization*, $\mathcal{T}(\mathcal{A})$, tesselates the convex hull of $\mathcal{A}'$ into tetrahedra. The tetrahedrization consists of the tesselating

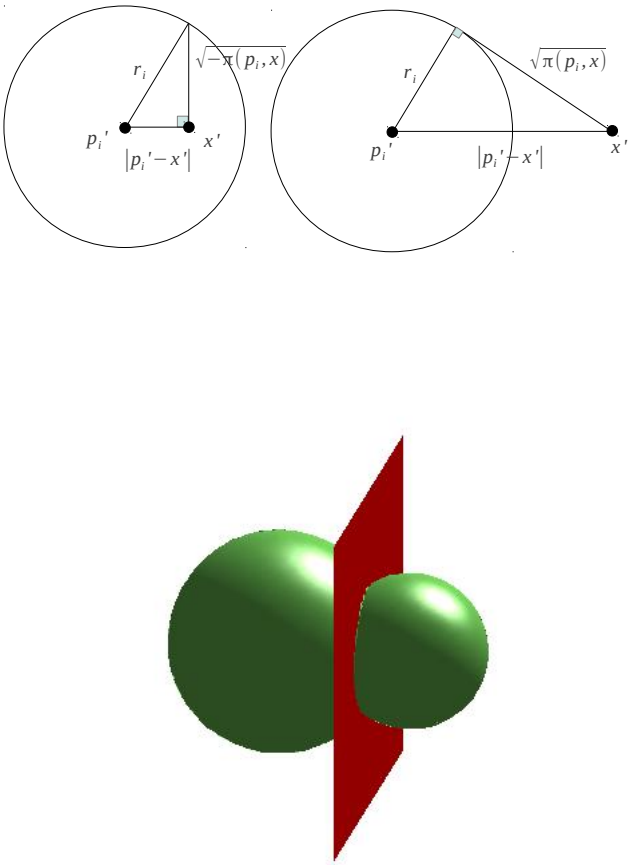


Figure 2.2: a) Power distance b) Radical plane between intersecting atoms

tetrahedra and their faces and thus is a simplicial complex [6]. The regular tetrahedrization of a set of unweighted points is a *Delaunay tetrahedrization*.

A simplex is a generalization of a triangle which is a 2-simplex. Three-dimensional simplices are tetrahedra, 1-dimensional simplices are edges, and 0-dimensional simplices are vertices. We define $T \subset \mathcal{A}$ where $|T| = k$ means that there are k elements in T. In this discussion we only consider $k \leq 4$. We write a simplex which is the convex hull of point centers in T as

$$\sigma_T = \text{conv}(T'). \tag{2.2.2}$$

For $|T| = k$, σ_T is a $(k-1)$-simplex.

2.2.1 Duality

The regular tetrahedrization, $\mathcal{T}(\mathcal{A})$, is dual to the Laguerre diagram. There is a one to one correspondence between the $(3-k)$-faces in $\mathcal{L}(\mathcal{A})$ and the k-simplices in $\mathcal{T}(\mathcal{A})$ (Figure 2.3). Each node in $\mathcal{L}(\mathcal{A})$ corresponds to a tetrahedron in $\mathcal{T}(\mathcal{A})$ whose vertices are equipowerdistant to the node and which we call the center of the characteristic point of the tetrahedron. Each segment in $\mathcal{L}(\mathcal{A})$ corresponds to a triangle in $\mathcal{T}(\mathcal{A})$ whose vertices are equipowerdistant to the power segment. For each facet in $\mathcal{L}(\mathcal{A})$ there is an edge in $\mathcal{T}(\mathcal{A})$ whose vertices are equipowerdistant to the power facet. Each cell in $\mathcal{L}(\mathcal{A})$ corresponds to a point in $\mathcal{T}(\mathcal{A})$, namely the generator of the cell [9],[12],[6]. (Figure 2.4).

LAGUERRE 3D	**DELAUNAY 3D**
cell	vertex
facet	edge
segment	triangle
node	tetrahedron
LAGUERRE 2D	**DELAUNAY 2D**
cell	vertex
facet	edge
node	triangle

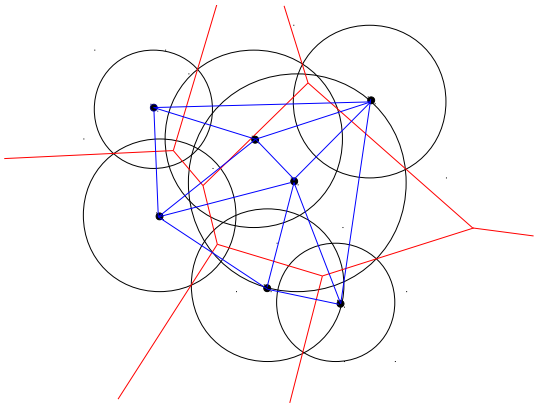


Figure 2.3: 2d version illustrates 1-1 correspondence between triangulation simplices and Laguerre faces.

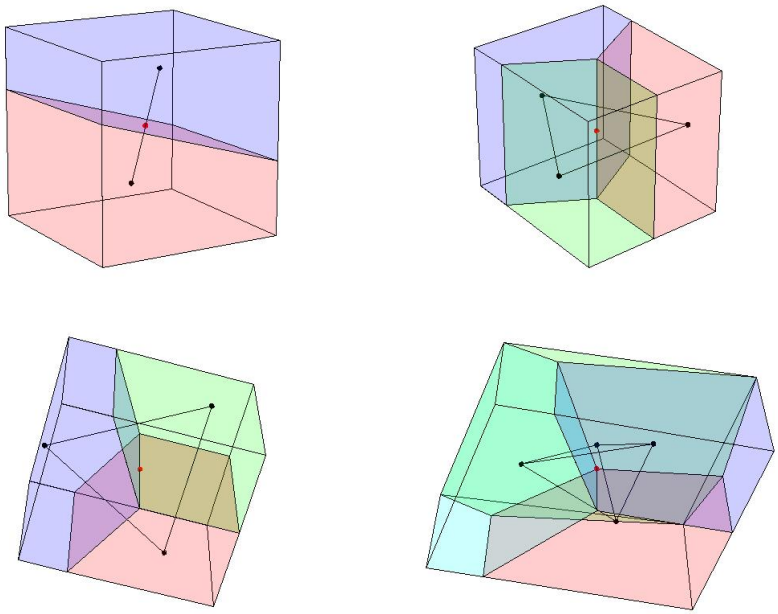


Figure 2.4: Correspondence between Laguerre facets and tetrahedrization simplices. Centers of the characteristic points of simplices are shown in red.

2.2.2 Power Product and Orthogonality

The *power product* of two weighted points p and q is

$$\Pi(p, q) = |p' - q'|^2 - p'' - q''. \tag{2.2.3}$$

Points p and q are said to be *orthogonal* if (See Figure 2.5)

$$\Pi(q, p) = 0. \tag{2.2.4}$$

This means that S_p and S_q intersect at right angles.

We define the *characteristic point* of a simplex σ_T as the point x_T such that

$$\Pi(x_T, p_i) = 0 \text{ for all } p_i \in T \tag{2.2.5}$$

with x_T'' minimal. The characteristic point of a simplex is unique.

Definition (Regular). *The tetrahedron, σ_T, is* regular *if and only if $\Pi(p, x_T) \geq 0$ for all $p \in \mathcal{A}$ (See Figure 2.6).*

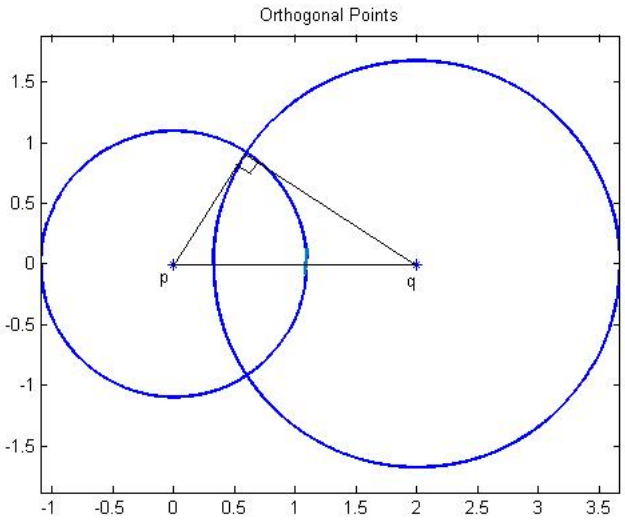


Figure 2.5: Orthogonal points p and q intersect at right angles.

A tetrahedrization is regular if and only if all the tetrahedra are regular.

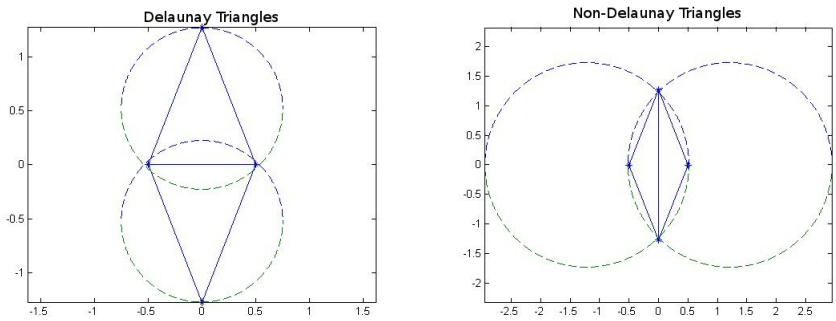


Figure 2.6: Two dimensional illustration of regular and non-regular simplices. In the unweighted case regular and non-regular is equivalent to Delaunay and non-Delaunay. The characteristic point of a Delaunay triangle is its circumcircle.

In the special case when $\mathcal{A}$ is unweighted, the term 'regular' is replaced with 'Delaunay'. The characteristic point, x_T, of a tetrahedron in the unweighted case is the circumsphere of the four points in the tetrahedron. A Delaunay tetrahedrization, $\mathcal{T}(\mathcal{A})$, has the geometric property that for each tetrahedron $\sigma_T \in \mathcal{T}(\mathcal{A})$, the points $p \in T$ lie on the sphere represented by x_T and points $p \in \mathcal{A} \backslash T$ lie on or outside the sphere. Furthermore, if the interior of the circumsphere of four points $p \in T$ is empty then $\sigma_T \in \mathcal{T}(\mathcal{A})$. (See Figure 2.7). This motivates the use of the term 'Empty sphere property' for regular tetrahedrizations [12],[7],[6].

Empty Sphere Property. $\sigma_T \in \mathcal{T}(\mathcal{A}) \iff \Pi(p, x_T) \geq 0$ *for all* $p \in \mathcal{A}$.

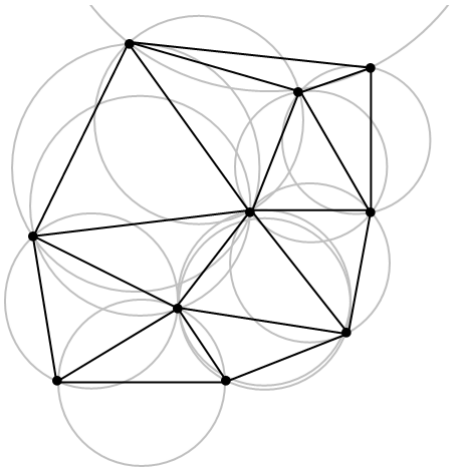


Figure 2.7: Illustration of the Empty Sphere Property for the two-dimensional unweighted case. The circumcircle or characteristic point of each triangle is empty.

For $p = (p', p'')$, define the expanded point

$$p_\alpha = (p', p'' + \alpha) \tag{2.2.6}$$

and let $\mathcal{A}_\alpha$ be the set of expanded points in $\mathcal{A}$. We have

$$\Pi(p_\alpha, q_{-\alpha}) = \Pi(p, q). \tag{2.2.7}$$

This means that the Laguerre tesselation and the regular tetrahedrization is a constant function of α. (Figure 2.8).

$$\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_\alpha) \tag{2.2.8}$$

$$\mathcal{T}(\mathcal{A}) = \mathcal{T}(\mathcal{A}_\alpha) \tag{2.2.9}$$

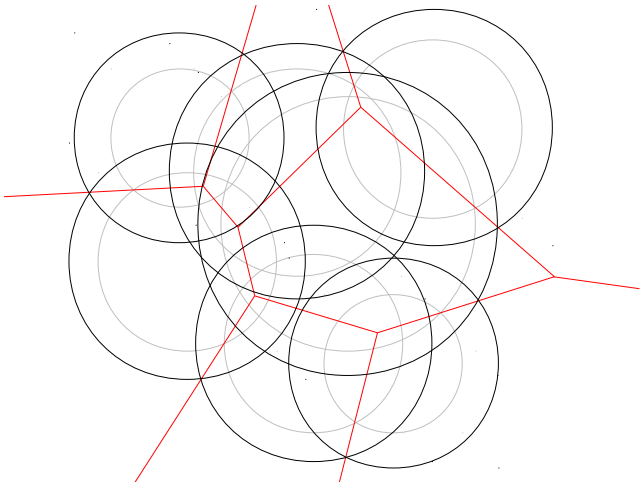


Figure 2.8: Laguerre cells are independent of solvent weight. Red bounds Laguerre cells of $\mathcal{A}_\alpha$ (shown in black) and $\mathcal{A}$ (shown in gray).

Chapter 3

Alpha Complex

In this section, we will develop the idea of the alpha complex, $\mathcal{C}_\alpha(\mathcal{A})$ [9],[12].

3.1 Space Filling Model

Recall that the weighted point $p = (p', p'')$ can be thought of as a sphere B_p centered at p' with radius $\sqrt{p''}$. Define

$$\mathcal{B}_\alpha = \bigcup_{p \in \mathcal{A}} B_{p_\alpha}. \tag{3.1.1}$$

Roughly speaking a simplex, $\sigma_T \in \mathcal{T}(\mathcal{A})$, is in the alpha complex, $\mathcal{C}_\alpha(\mathcal{A})$, if σ_T is interior to $\mathcal{B}_\alpha$ (See Figure 3.1). This is always true for simplices that are called *unattached*. However, it is not necessarily true for *attached* simplices, a concept that will be introduced in section 3.4.

3.2 Size of a Simplex

The size, ρ_T, of a simplex is the weight of the characteristic point. It is defined to be the radius squared of the smallest sphere that is orthogonal to all vertices in the simplex. Define

$$X_T = \{x' \in \mathbb{R}^3 : \quad \pi(p, x') = \pi(q, x') \quad \forall \quad p, q \in T\}. \tag{3.2.1}$$

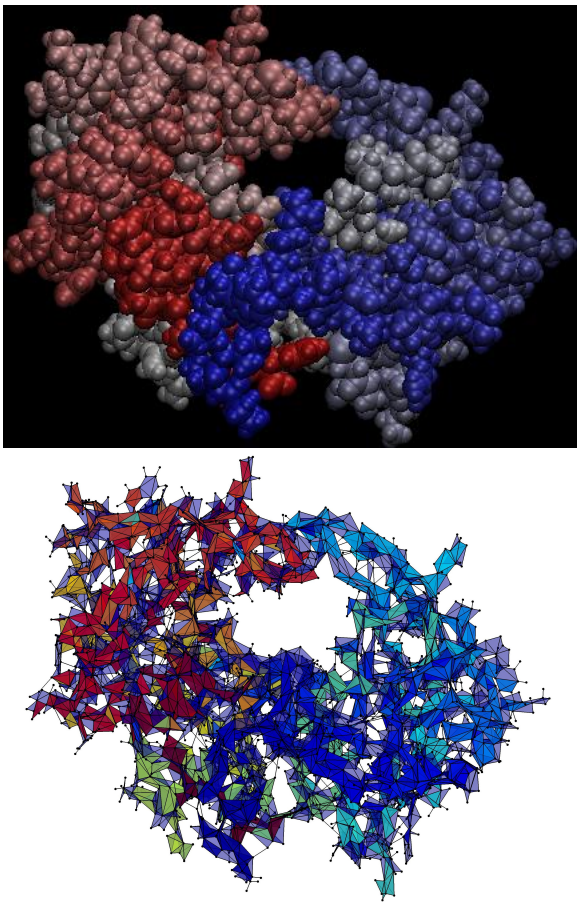


Figure 3.1: A van der Waals representation of HIV protease along with its alpha complex for $\alpha = 0$.

The set X_T is either a point, line, or plane for a simplex which is a tetrahedron, triangle, or edge, respectively. If σ_T is a 0-simplex (i.e. a point), $X_T = \mathbb{R}^3$.

For each $x' \in X_T$, there is a weight, ρ, such that $x = (x', \rho)$ is orthogonal to all $p \in T$. Define ρ_T to be the minimum such weight and $x_T = (x'_T, \rho_T)$ the corresponding point which is unique. The size of the simplex is said to be ρ_T. When the size of the characteristic point is positive, the characteristic point intersects the points in the simplex at right angles.

The characteristic point of an edge for varying configurations is shown in Figure 3.2.

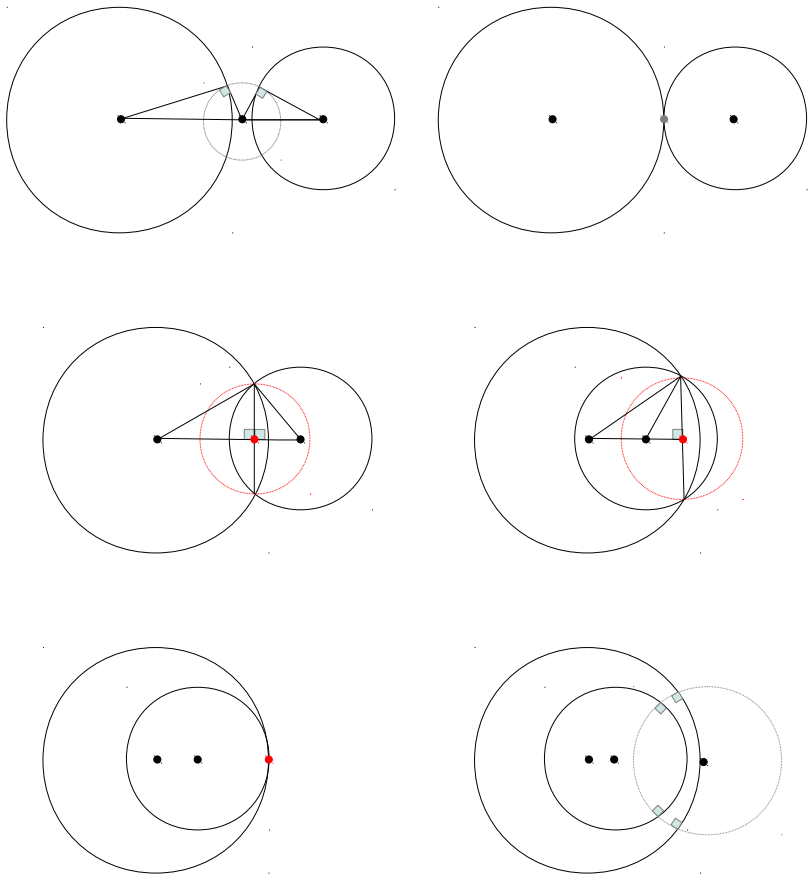


Figure 3.2: Characteristic points of edges in a variety of configurations. An edge connects the two spheres that are represented with black lines. When the weight of a characteristic point is positive it is represented by a gray sphere, otherwise it is represented by a red sphere with radius $\sqrt{-x''}$.

3.3 Geometric Interpretation

The size of a simplex, ρ_T, gives us information about how the balls represented by points in T intersect.

Let $\bigcap Surf_T$ be the intersection of the surface of the balls represented by points in T. We assume that the weights of all points in T are non-negative. We present relationships between ρ_T and $\bigcap Surf_T$.

Spherical Intersection.

For $|T| = 2$ and $|T| = 3$:

1. $\rho_T < 0 \iff |\bigcap Surf_T| > 1$

2. $\rho_T = 0 \iff |\bigcap Surf_T| = 1$

3. $\rho_T > 0 \iff |\bigcap Surf_T| = 0$

For $|T| = 4$:

1. $\rho_T = 0 \iff |\bigcap Surf_T| = 1$

2. $\rho_T \neq 0 \iff |\bigcap Surf_T| = 0$

Proof. For simplicity we write the characteristic point of a simplex $x_T = x = (x', x'')$ where $\rho_T = x''$.

Edge: Let $T = \{p_i, p_j\}$.

Assume $\rho_T = 0$. Then we have $x' \in \bigcap Surf_T$. Now for any $y' \in \bigcap Surf_T$, the point $y = (y', 0)$ satisfies $\Pi(p_m, y) = 0$ for $m = i, j$. By uniqueness of the characteristic point, $y' = x'$. This gives $\rho_T = 0 \implies |\bigcap Surf_T| = 1$.

Now assume $\bigcap Surf_T = y'$. Since $y' = p'_j + t(p'_i - p'_j)$ for some scalar t we have $z'' > 0$ for all z'' such that $\Pi(p_m, z) = 0$, $m = i, j$. By minimality of the weight of the characteristic point we have $y' = x'$ and $x'' = 0$. This gives $|\bigcap Surf_T| = 1 \implies \rho_T = 0$.

Now assume $\rho_T < 0$. By definition, the characteristic point satisfies $\Pi(x, p_m) = 0$ for $m = i, j$ with x'' minimal, which means we can write $x' = p'_j + t(p'_i - p'_j)$ for some scalar t. Let n be an arbitrary unit vector with $n \cdot (p'_i - p'_j) = 0$ and let $y' = x' + \sqrt{-x''}n$.

Then

$$\begin{aligned} |y' - p'_j|^2 &= |x' - p'_j|^2 - x'' + 2(x' - p'_j) \cdot \sqrt{-x''}n \\ &= |x' - p'_j|^2 - x'' \\ &= p''_j \end{aligned} \tag{3.3.1}$$

and by the same algebra

$$|y' - p'_i|^2 = p''_i. \tag{3.3.2}$$

This gives $x' \in \bigcap Surf_T$. Now the point $y' = x' - \sqrt{-x''}n$ is also in $\bigcap Surf_T$. Hence $\rho_T < 0 \implies |\bigcap Surf_T| > 1$.

Now assume y' is on the surface of p_j and p_i,

$$\begin{aligned} |y' - p'_i|^2 &= p''_i \\ |y' - p'_j|^2 &= p''_j. \end{aligned} \tag{3.3.3}$$

The weighted point $y = (y', 0)$ satisfies

$$\begin{aligned} \Pi(y, p_j) &= 0 \\ \Pi(y, p_i) &= 0. \end{aligned} \tag{3.3.4}$$

Since x satisfies $\Pi(x, p_j) = 0$ and $\Pi(x, p_i) = 0$ and x'' is minimal we have $\rho_T \leq 0$. By property 2. we have $\rho_T < 0$.

Property 3. comes from properties 1. and 2.

Triangle: Let $T = \{p_i, p_j, p_k\}$.

Assume $\rho_T = 0$. Then we have $x' \in \bigcap Surf_T$. Now for any $y' \in \bigcap Surf_T$, the point $y = (y', 0)$ satisfies $\Pi(p_m, y) = 0$ for $m = i, j, k$. By uniqueness of the characteristic point, $y' = x'$. This gives $\rho_T = 0 \implies |\bigcap Surf_T| = 1$.

Now assume $\bigcap Surf_T = y'$. Since $y' = p'_j + s(p'_i - p'_j) + t(p'_i - p'_k)$ for some scalars s and t we have $z'' > 0$ for all z'' such that $\Pi(p_m, z) = 0$ for $m = i, j, k$. By minimality of the weight of the characteristic point we have $y' = x'$ and $x'' = 0$. This gives $|\bigcap Surf_T| = 1 \implies \rho_T = 0$.

Assume $\rho_T < 0$. We can write $x' = p'_i + s(p'_i - p'_j) + t(p'_i - p'_k)$ for scalars s and t. Define n such that $n \cdot (p'_i - p'_j) = 0$, $n \cdot (p'_i - p'_k) = 0$, and $|n| = 1$. Let $y' = x' + \sqrt{-x''}n$. Then for $m = i, j, k$,

$$\begin{aligned} |y' - p'_m|^2 &= |x' - p'_m|^2 - x'' + 2(x' - p'_m) \cdot \sqrt{-x''}n \\ &= |x' - p'_m|^2 - x'' \\ &= p''_m \end{aligned} \tag{3.3.5}$$

which shows that the surfaces of p'_i, p'_j, and p'_k have nonempty intersection. Now the point $z' = x' - \sqrt{-x''}n \in \bigcap Surf_T$ which gives $|\bigcap Surf_T| > 1$.

We can show as for $|T| = 2$ that $|\bigcap Surf_T| > 1 \implies \rho_T < 0$.

Property 3. comes from property 1. and 2.

Tetrahedron: We have $\bigcap Surf_T \subset X_T$. Properties 1., 2., and 3. come from the fact that the set X_T contains only one point, x', which is the center of the characteristic point.

□

Let $\bigcap B_T$ be the intersection of balls represented by points in T. We call $\bigcap B_T$ *non-redundant* if $\bigcap B_T \neq \bigcap B_{\widetilde{T}}$ for all $\widetilde{T} \subset T$, otherwise we call $\bigcap B_T$ redundant. Define $V_T = vol(\bigcap B_T)$ and $S_T = surf(\bigcap B_T)$. We present relationships between ρ_T and properties of the set of balls B_T.

Volume Intersection.

1. $\rho_T < 0 \iff V_T > 0$, $S_T > 0$ *and* $\bigcap B_T$ *is non-redundant.*

2. $\rho_T \geq 0 \iff V_T = S_T = 0$ *or* $\bigcap B_T$ *is redundant.*

Proof. **Edge:** Let $T = \{p_i, p_j\}$.

Assume $\rho_T < 0$. By the Spherical Intersection property, we know that $|\bigcap Surf_T| > 1$. This means that the surfaces of p_i and p_j intersect along a circle with positive radius. This gives $V_T > 0$, $S_T > 0$ and $\bigcap B_T$ non-redundant.

Now assume that $V_T > 0$, $S_T > 0$ and $\bigcap B_T$ is non-redundant. Then the surfaces of p_i and p_j intersect along a circle with positive radius which shows that $\rho_T < 0$.

Property 2. comes from property 1.

Triangle: Let $T = \{p_i, p_j, p_k\}$.

Assume $\rho_T < 0$. We know that $|\bigcap Surf_T| > 1$. This gives the configurations shown in Figure 3.3 case 1. Case 2 is violates the assumption of that $\mathcal{A}$ is in general position. Without loss of generality, call the sphere p_i. The red and blue circles on the surface of p_i are the circles of intersection of the surfaces of p_j and p_k with p_i.

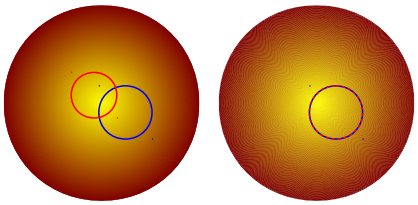


Figure 3.3: Illustration of the possible circles of intersection of the sphere p_i with the spheres p_j and p_k.

In each case, it is easy to see that $\bigcap B_T$ is not redundant. Also, $\bigcap B_T$ is a subset of the ball p_i which is a above both of the planes which cut through the circles of intersection on the sphere. This gives $V_T > 0$ and $S_T > 0$.

Now assume that $V_T > 0$, $S_T > 0$, and $\bigcap B_T$ is non-redundant. Then we have the intersection shown in Figure 3.4. We let i_1 and i_2 be the common points on the two intersection circles. Then $x' = \frac{1}{2}(i_1 + i_2)$ and $x'' = -\frac{|i_1 - i_2|}{4}^2 < 0$ are the components of the characteristic point of σ_T. This gives $\rho_T < 0$.

Property 2. comes from property 1.

Tetrahedron: Let $T = \{p_i, p_j, p_k, p_l\}$.

Assume $\rho_T < 0$. We know that $x' \in \bigcap B_T$. By the Spherical Intersection property

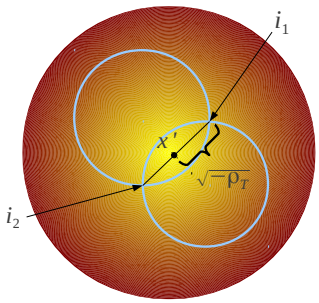


Figure 3.4: Circles of intersection of the sphere p_i with spheres p_j and p_k. The points i_1 and i_2 lie on the intersection of the two circles. The characteristic point of the triangle is $x_T = (x', \rho_T)$.

we have $x' \notin \bigcap Surf_T$ which gives $|\bigcap B_T| > 1$. This means $V_T > 0$ and $S_T > 0$. Now for every triad $\widetilde{T} \subset T$ we have $\rho_{\widetilde{T}} \leq \rho_T$. This along with the Spherical Intersection property gives the intersection shown in Figure 3.5 and $\bigcap B_T$ non-redundant.

Now assume $V_T > 0$, $S_T > 0$, and $\bigcap B_T$ is non-redundant. Then we have the intersection shown in Figure 3.5. Let i_1, i_2, and i_3 be the points of intersection on the surface of the sphere. Then x' is the intersection of the line segments i_1i_2, i_1i_3, and i_2i_3 and $\rho_T = -|x' - i_1|^2 < 0$.

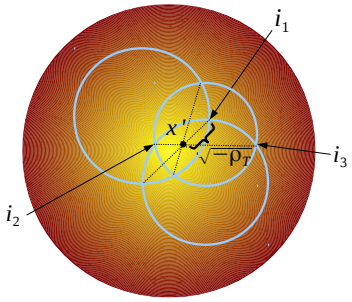


Figure 3.5: Circles of intersection of p_i with p_j, p_k, and p_l. The characteristic point of the tetrahedron is $x_T = (x', \rho_T)$.

□

3.4 Attached and Unattached Simplices

A simplex, σ_T, is called *unattached* if x_T satisfies $\Pi(x_T, p) > 0$ for all $p \in \mathcal{A} \setminus T$; otherwise a simplex is called attached. By the Empty Sphere Property, and assuming general position of data points, all tetrahedra are unattached. We say a simplex $\sigma_{\widetilde{T}}$ is attached to σ_T if $\Pi(x_{\widetilde{T}}, p) \leq 0$ for $p = T \setminus \widetilde{T}$.

Definition 3.4.1. *An unattached simplex σ_T is in $\mathcal{C}_\alpha$ if and only if $\alpha \geq \rho_T$.*

We call the union of the balls in T that are expanded by α, $\bigcup B_{T^\alpha}$, and similarly for $\bigcap B_{T^\alpha}$ and $\bigcap S_{T^\alpha}$.

Equivalence. *If a simplex, σ_T, is unattached then the following are equivalent:*

1. $\sigma_T \in \mathcal{C}_\alpha$
2. $\alpha \geq \rho_T$
3. $\bigcap B_{T^\alpha} \neq \emptyset$ *and is non-redundant*
4. σ_T *is interior to* $\mathcal{B}_\alpha$

Proof.

$(\mathbf{1} \Leftrightarrow \mathbf{2})$: This comes from Definition 3.4.1.

$(\mathbf{2} \Leftrightarrow \mathbf{3})$: This comes from the Spherical Intersection property and Volume Intersection property along with Equation 2.2.7.

$(\mathbf{3} \Leftrightarrow \mathbf{4})$: This is easy to see using a geometric argument. □

Definition 3.4.2 (Incidence). *Two simplices are called incident if one is included in the other and their dimensions differ by one [3].*

For a simplex σ_T with $|T| < 3$, let inc_T be the set of simplices that contain σ_T. That is inc_T is the set of higher dimensional simplices that are incident to σ_T.

Definition 3.4.3. *An attached simplex σ_T is in $\mathcal{C}_\alpha$ if there exists a simplex, $\sigma_{T^+} \in inc_T$, such that $\sigma_{T^+} \in \mathcal{C}_\alpha$.*

3.5 Filtrations and Persistence

As alpha increases from $-\infty$ to ∞ a discrete set of alpha complexes are obtained. By equation 2.2.9,

$$\mathcal{C}_{\alpha i} \subseteq \mathcal{C}_{\alpha j} \text{ for } \alpha_i \leq \alpha_j.$$

The sequence of nested complexes $\emptyset = \mathcal{C}_{\alpha 0} \subseteq \cdots \subseteq \mathcal{C}_{\alpha m} = \mathcal{T}$ is a filtration of the tetrahedrization $\mathcal{T}$. For

$$\alpha < -\max_{p \in \mathcal{A}} p''$$

$\mathcal{C}_\alpha = \emptyset$. As α increases we first obtain vertices in $\mathcal{C}_\alpha$, followed by edges connecting bonded atoms. The first triangles connect an atom with two of its bonded neighbors. Additional edges, triangles, and tetrahedra become part of $\mathcal{C}_\alpha$. Finally for

$$\alpha \geq \max_{\sigma_T \in \mathcal{T}} \rho_T$$

we have the entire tetrahedrization $\mathcal{C}_\alpha = \mathcal{T}$ (See Figure 3.6). By comparing different spatial resolutions one may learn about the topology of the underlying space. See [42] for a more in depth discussion on computational topology.

3.6 Classification of Simplices

A simplex, $\sigma_T \in \mathcal{C}_\alpha$, is called *interior* if it contains interior points of $\mathcal{C}_\alpha$, *singular* if it is incident to no higher dimensional simplex in $\mathcal{C}_\alpha$, *regular* if it is incident to a higher dimensional simplex in $\mathcal{C}_\alpha$ but it is not interior. Note that for any value of α, an attached simplex is never singular, a tetrahedron is never singular nor regular, and simplices on the convex hull of $\mathcal{A}'$ (see equation 2.2.1) are never interior.

For each simplex, σ_T, there exist three disjoint (possibly empty) intervals for which σ_T is singular, regular, or interior. An unattached simplex, σ_T, becomes part of $\mathcal{C}_\alpha$ when $\alpha = \rho_T$. As α increases, this simplex remains singular until $\alpha = \underline{\mu}_T$ when it becomes

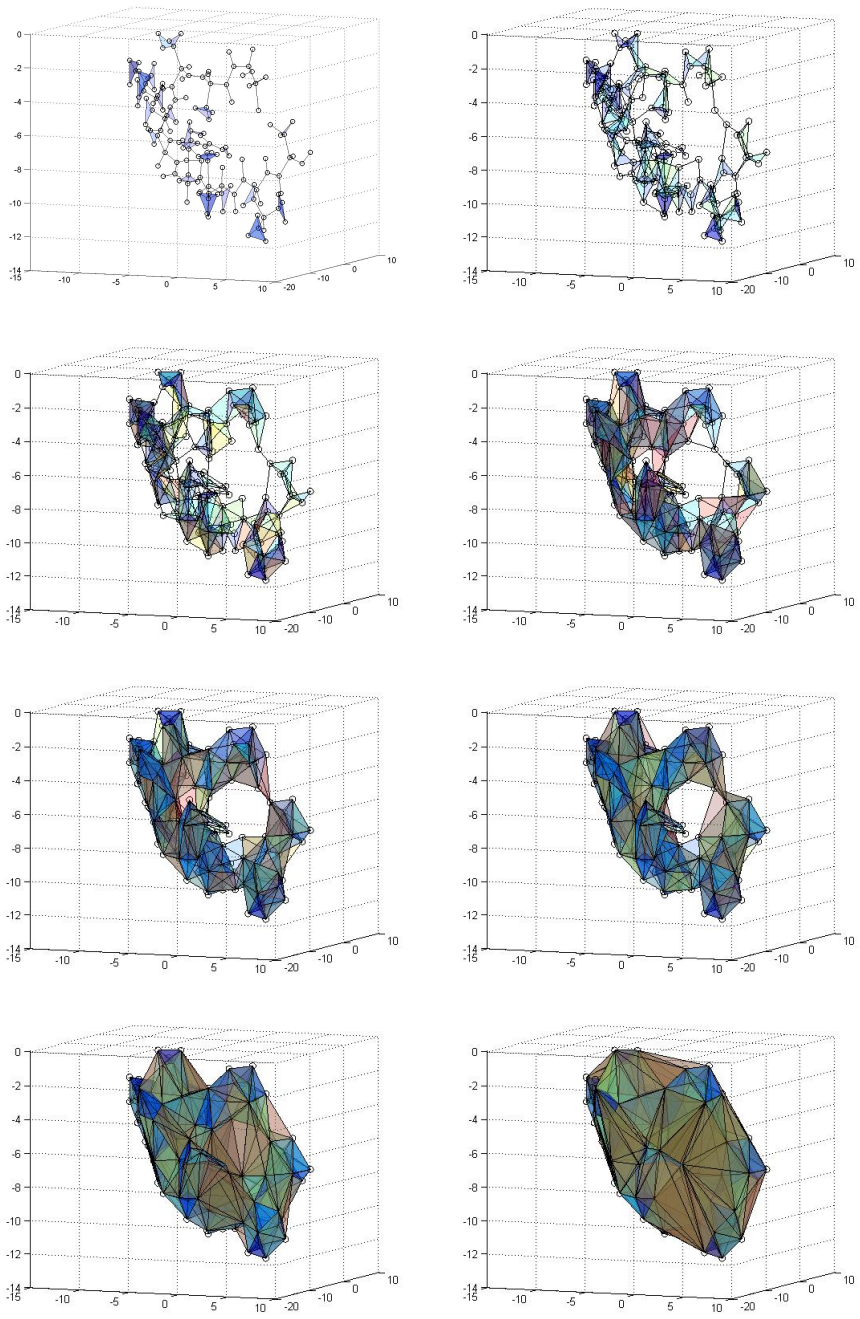


Figure 3.6: A set of alpha complexes for increasing α of a small polypeptide.

regular. The value $\underline{\mu}_T$ is defined as follows [12]:

$$\underline{\mu}_T = \min_{\sigma_{T^+} \in inc_T} \begin{cases} \rho_{T^+} & \text{if } \sigma_{T^+} \text{unattached} \\ \underline{\mu}_{T^+} & \text{if } \sigma_{T^+} \text{attached} \end{cases} \tag{3.6.1}$$

As α increases, σ_T remains regular until $\alpha = \overline{\mu}_T$ when it becomes interior. Note that simplices on the boundary of $conv(\mathcal{A}')$ never become interior. In this case, one can define $\overline{\mu}_T = \infty$. If σ_T is not on $conv(\mathcal{A}')$, then σ_T is interior when all incident tetrahedra are in $\mathcal{C}_\alpha$.

To summarize, let μ_s, μ_r, and μ_i be the α values for which a simplex becomes singular, regular, or interior, respectively. Then these values are given by:

Classification of Simplex	μ_s	μ_r	μ_i
Unattached, interior to convex hull	ρ_T	$\underline{\mu}_T$	$\overline{\mu}_T$
Unattached, on convex hull	ρ_T	$\underline{\mu}_T$	DNE
Attached, interior to convex hull	DNE	$\underline{\mu}_T$	$\overline{\mu}_T$
Attached, on convex hull	DNE	$\underline{\mu}_T$	DNE

Part II

Molecular Volume, Atomic Solvent Accessible Surface Area, Jacobian, and Discontinuities

Chapter 4

Volume, Surface Area, and Gradient

4.1 Overview: Volume and Surface Area Computation Methods

A molecule may be modelled as a union of spheres: each atom is represented by a sphere centered at the nucleus center with van der Waals radius. We associate the set of weighted points $\mathcal{A}$ with a molecule where $p \in \mathcal{A}$ corresponds to an atom in the molecule centered at p' with radius $\sqrt{p''}$.

In the following subsections three algorithms are presented for computing the volume and surface area [26], [7], [12]. The first two algorithms are preludes to the third which is the fastest and the one that is implemented.

4.1.1 Inclusion-Exclusion

The volume and surface area of a molecule can be computed using the inclusion-exclusion formula. The total volume and surface area of the molecule are given by a combinatorial equation,

$$\mathcal{V} = \sum_{T \subset \mathcal{A}} (-1)^{k-1} V_T, \qquad |T| = k \tag{4.1.1}$$

and

$$\mathcal{S} = \sum_{T \subset \mathcal{A}} (-1)^{k-1} S_T, \qquad |T| = k. \tag{4.1.2}$$

The sum is taken over all possible subsets T of $\mathcal{A}$ for $|T| = 1, ..., n-1$ where n is the number of atoms. Here $V_T = vol(\bigcap B_T)$ and $S_T = surf(\bigcap B_T)$ as defined in Section 3.3.

The number of terms in the sums 4.1.1 and 4.1.2 is unwieldy even for moderately large n. The quantities V_T and S_T may be difficult for many T with $|T| \geq 3$, and in the end may even cancel to zero. A possible improvement would be to take the sum over all subets T of $\mathcal{A}$ that have nonempty intersection.

4.1.2 Direct Inclusion-Exclusion

The direct inclusion-exclusion formula is an exact method which omits zero terms and terms with $|T| > 4$ in the sums 4.1.1 and 4.1.2. It has been shown that the intersections of five or more balls may be reduced to a combination of the intersection of four or fewer balls.

The terms needed to compute the volume and surface area of a molecule are encoded in the alpha complex:

$$\mathcal{V} = \sum_{\sigma_T \in \mathcal{C}} (-1)^{k-1} V_T, \qquad |T| = k \tag{4.1.3}$$

and

$$\mathcal{S} = \sum_{\sigma_T \in \mathcal{C}} (-1)^{k-1} S_T, \qquad |T| = k. \tag{4.1.4}$$

4.1.3 Short Inclusion-Exclusion

The short inclusion-exclusion method, like the direct method, relies on the alpha complex. However, with this method the intersections of four balls do not need to be computed. Rather the computation involves the weighted sum of triple, double, and single intersections. The weights, ϕ_T, as well as the other components of the formula will be

discussed in the next section.

$$\mathcal{V} = V + \sum_{\sigma_T \in \partial\mathcal{C}} (-1)^{k-1} \phi_T V_T, \qquad |T| = k < 4 \tag{4.1.5}$$

$$\mathcal{S} = \sum_{\sigma_T \in \partial\mathcal{C}} (-1)^{k-1} \phi_T S_T, \qquad |T| = k < 4. \tag{4.1.6}$$

4.2 Volume and Surface Area Formulas

Consider a set of weighted points $\mathcal{A}$, where $p \in \mathcal{A}$ is written $p = (p', p'')$ with $p' \in \mathbb{R}^3$ the location of the point and $p'' \in \mathbb{R}$ the weight. The point p can be thought of as a ball (or atom) centered at p' with radius $\sqrt{p''}$, and the set $\mathcal{A}$ can be thought of as a set of atoms which describe a molecule. Define $\mathcal{C} = \mathcal{C}_\alpha$ to be the alpha complex of $\mathcal{A}$ for the desired value of α, and $\partial\mathcal{C}$ the set of simplices in the boundary of $\mathcal{C}$. For a subset $T \subset \mathcal{A}$, we write $|T| = k$ if there are k elements in T. Let T' be the set of the centers of points in T. A k-simplex is a k-dimensional polytope which is the convex hull of $k+1$ points. A simplex which is the convex hull of points in T' is written as σ_T. As we are working in three dimensions we only consider $|T| \leq 4$. The terms S_T and V_T are the surface area and volume respectively of the intersection of the balls in T (See Figure 4.1).

The surface area and volume of the union of balls are

$$\mathcal{S} = \sum_{\sigma_T \in \partial\mathcal{C}} (-1)^{k+1} c_T S_T, \qquad |T| = k \tag{4.2.1}$$

$$\mathcal{V} = V + \sum_{\sigma_T \in \partial\mathcal{C}} (-1)^{k+1} c_T V_T, \qquad |T| = k \tag{4.2.2}$$

where V is the volume of the tetrahedra in the alpha complex. The coefficients, c_T, are given for the following simplices,

- $|T| = 1$, i.e. a vertex v_i: $c_T = \Omega_T$ is the fraction of the ball i outside the tetrahedra in the alpha complex. That is Ω_T is the normalized outer solid angle subtended by the union of tetrahedra in $\mathcal{C}$ which contain v_i.

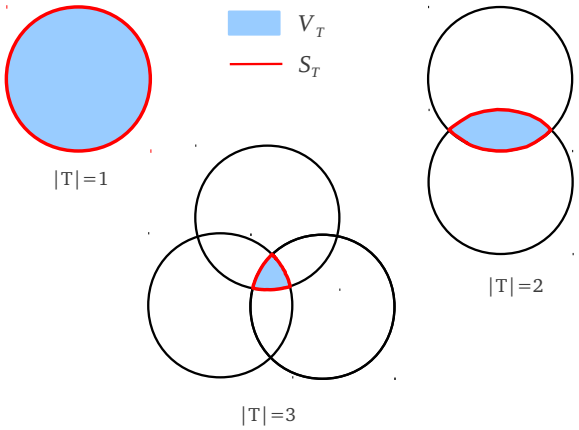


Figure 4.1: Volume (blue) and surface area (red) contributions from simplices of dimensions one, two, and three.

- $|T| = 2$, i.e. an edge e_{ij}: $c_T = \Phi_T$ is normalized outer dihedral angle of the union of tetrahedra in $\mathcal{C}$ which contain the edge e_{ij}.
- $|T| = 3$, i.e. a triangle t_{ijk}: c_T is 1 if the triangle is singular and $\frac{1}{2}$ if the triangle is regular. In other words, c_T is the fraction of V_T and S_T that is outside the union of tetrahedra in the alpha complex.

Here $v_i = p_i'$, $e_{ij} = conv(\{p_i', p_j'\})$, and $t_{ijk} = conv(\{p_i', p_j', p_k'\})$.

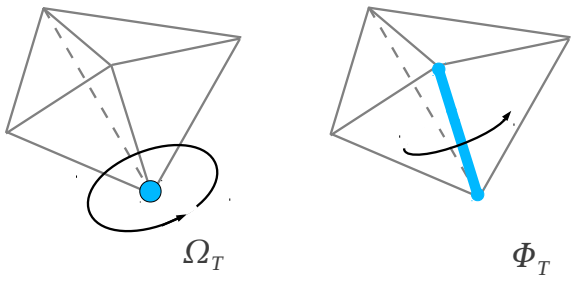


Figure 4.2: Fractional outer solid angle and fractional outer dihedral angle.

The term $\mathcal{S}$ gives the total surface area of the molecule, but we are also interested in the contribution of an individual atom, p_i, to the total surface area. Call this term $\mathcal{S}^i$.

Then

$$\mathcal{S}^i = \sum_{\sigma_T \in \partial\mathcal{C}} (-1)^{k+1} c_T S_T^{(i)}, \qquad |T| = k \tag{4.2.3}$$

where $S_T^{(i)}$ is the contribution of S_T to $\mathcal{S}^i$ with

$$\sum_{i=1}^{n} \mathcal{S}^i = \mathcal{S}. \tag{4.2.4}$$

These formulas will be given in subsequent sections. The sum 4.2.3 may also be taken over all $\sigma_T \in \partial C$ such that $p_i \in T$ since $S_T^{(i)} = 0$ if $p_i \notin T$. Assuming $\mathcal{A}$ has n elements, i.e. the molecule has n atoms, define the n-vector $\vec{\mathcal{S}}$ where the ith element of $\vec{\mathcal{S}}$ is $\mathcal{S}^i$. We will also consider the weighted surface area

$$\mathcal{S}_w = \sum_{i=1}^{n} w_i \mathcal{S}^i. \tag{4.2.5}$$

where the w_i are given constants.

4.3 Derivatives

Consider the radius of each of the balls in $\mathcal{A}$ fixed. Then for $i = 1 \ldots n$, $\mathcal{S}$, $\mathcal{S}^i$, and $\mathcal{V}$ are piecewise differentiable functions of the $3n$ position variables. Discontinuities in the derivatives in $\vec{\mathcal{S}}$, $\mathcal{S}$, $\mathcal{S}^i$, and $\mathcal{V}$ occur at exactly the same points in conformation space.

Let $x_{i,j}$ be the jth coordinate of atom i and define

$$X = \big(x_{1,1}, x_{1,2}, x_{1,3}, \ldots, x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{3(n-1),1}, x_{3(n-1),2}, x_{3(n-1),3}\big). \tag{4.3.1}$$

In a region where the functions are differentiable write

$$\nabla\mathcal{V} = \left(\frac{\partial\mathcal{V}}{\partial X_1}, \quad \dots \quad , \frac{\partial\mathcal{V}}{\partial X_{3n}}\right) \in \mathbb{R}^{3n} \tag{4.3.2}$$

$$\nabla\mathcal{S} = \left(\frac{\partial\mathcal{S}}{\partial X_1}, \quad \dots \quad , \frac{\partial\mathcal{S}}{\partial X_{3n}}\right) \in \mathbb{R}^{3n} \tag{4.3.3}$$

$$\nabla\mathcal{S}_w = \left(\frac{\partial\mathcal{S}_w}{\partial X_1}, \quad \dots \quad , \frac{\partial\mathcal{S}_w}{\partial X_{3n}}\right) \in \mathbb{R}^{3n} \tag{4.3.4}$$

$$\tag{4.3.5}$$

and

$$J(\vec{\mathcal{S}}) = \begin{pmatrix} \nabla\mathcal{S}^1 \\ . \\ . \\ . \\ \nabla\mathcal{S}^n \end{pmatrix} \in \mathbb{R}^{n\times 3n} \tag{4.3.6}$$

where $\frac{\partial}{\partial X_k} = \frac{\partial}{\partial x_{i,j}}$ is the partial derivative with respect to the jth coordinate of atom i and $k = 3(i-1)+j$.

Applying the product rule,

$$\begin{aligned} \nabla\mathcal{V} &= \sum_{\sigma_T\in\partial\mathcal{C}:|T|=1} \nabla\Omega_T V_T \\ &+ \sum_{\sigma_T\in\partial\mathcal{C}:|T|=2} \nabla\Phi_T V_T + \Phi_T\nabla V_T \\ &+ \sum_{\sigma_T\in\partial\mathcal{C}:|T|=3} c_T\nabla V_T \\ &+ \sum_{\sigma_T\in\partial\mathcal{C}:|T|=4} \nabla V_T. \end{aligned} \tag{4.3.7}$$

By writing Ω_T, Φ_T, c_T with $T \in \partial\mathcal{C}$ in the column vectors $\vec{\Omega}$, $\vec{\Phi}$, $\vec{c}$, and V_T with $|T| = 1,2,3,4$ in the column vectors $\vec{V}1$, $\vec{V}2$, $\vec{V}3$, and $\vec{V}4$ respectively, we can write equation 4.3.7 as

$$\nabla \mathcal{V} = \vec{V1}^T J(\vec{\Omega}) + \vec{V2}^T J(\vec{\Phi}) + \vec{\Phi}^T J(\vec{V2}) + \vec{c}^T J(\vec{V3}) + \vec{e}^T J(\vec{V4}). \tag{4.3.8}$$

where $\vec{e}$ is a vector of the appropriate dimensions with each entry equal to one, and T refers to the transpose of the vector.

Equations equivalent to 4.3.7 and 4.3.8 hold for $\mathcal{S}^i$, $\mathcal{S}$, and $\mathcal{S}_w$ but with the $|T| = 4$ terms omitted.

4.4 Equations

This section contains formulas for terms in equation 4.3.7.

4.4.1 $|T| = 1$

Volume and Surface Area

Consider $T = \{p_i\}$. The formulas for the volume and surface area of a ball are

$$V_T = \frac{4}{3}\pi p_i''^{3/2} \tag{4.4.1}$$

$$S_T = 4\pi p_i'' \tag{4.4.2}$$

$$S_T^{(i)} = 4\pi p_i''. \tag{4.4.3}$$

Solid Angle

Let $\mathcal{I}$ be the set of tetrahedra in $\mathcal{C}$ to which the vertex p_i is incident. For $\sigma_I \in \mathcal{I}$ define ω_I as the normalized inner solid angle subtended by the tetrahedron σ_I from the point p_i'. Then

$$\Omega_T = 1 - \sum_{\sigma_I \in \mathcal{I}} \omega_T^I \tag{4.4.4}$$

$$\nabla \Omega_T = -\sum_{\sigma_I \in \mathcal{I}} \nabla \omega_T^I \tag{4.4.5}$$

The normalized inner solid angle, ω, of a tetrahedron subtended by the vectors $\mathbf{a} = p'_j - p'_i$, $\mathbf{b} = p'_k - p'_i$, and $\mathbf{c} = p'_l - p'_i$ (See Figure 4.3) is given by the equation

$$\omega = \frac{1}{2\pi} \arctan\left(\frac{|\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})|}{abc + (\mathbf{a} \cdot \mathbf{b})c + (\mathbf{a} \cdot \mathbf{c})b + (\mathbf{b} \cdot \mathbf{c})a} \right) \tag{4.4.6}$$

where $a = |\mathbf{a}|$ and likewise for b and c (See Figure 4.3).

Define

$$F = \frac{|\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})|}{abc + (\mathbf{a} \cdot \mathbf{b})c + (\mathbf{a} \cdot \mathbf{c})b + (\mathbf{b} \cdot \mathbf{c})a} \tag{4.4.7}$$

and

$$D = abc + (\mathbf{a} \cdot \mathbf{b})c + (\mathbf{a} \cdot \mathbf{c})b + (\mathbf{b} \cdot \mathbf{c})a. \tag{4.4.8}$$

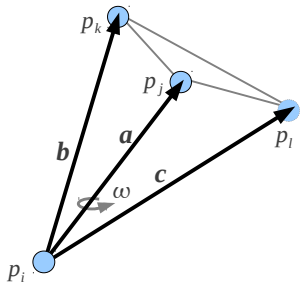


Figure 4.3: Normalized inner solid angle, ω, subtended by $\mathbf{a} = p'_j - p'_i$, $\mathbf{b} = p'_k - p'_i$, and $\mathbf{c} = p'_l - p'_i$.

We have

$$\begin{aligned} \nabla|\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})| &= sgn(\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}))\nabla\Big(\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})\Big) \\ = sgn(\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}))&\Big[(\mathbf{b} \times \mathbf{c})^T J(\mathbf{a}) + (\mathbf{c} \times \mathbf{a})^T J(\mathbf{b}) + (\mathbf{a} \times \mathbf{b})^T J(\mathbf{c})\Big] \end{aligned} \tag{4.4.9}$$

$$\begin{aligned} \nabla D &= (\nabla a)(bc + \mathbf{b} \cdot \mathbf{c}) + (\nabla b)(ac + \mathbf{a} \cdot \mathbf{c}) + (\nabla c)(ab + \mathbf{a} \cdot \mathbf{b}) \\ &+ (b\mathbf{c} + c\mathbf{b})^T J(\mathbf{a}) + (a\mathbf{c} + c\mathbf{a})^T J(\mathbf{b}) + (a\mathbf{b} + b\mathbf{a})^T J(\mathbf{c}) \end{aligned} \tag{4.4.10}$$

$$\nabla F = \frac{(\nabla|\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c})|)D - (\nabla D)|\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c})|}{D^2} \tag{4.4.11}$$

and

$$\nabla\omega = \frac{\nabla F}{2\pi(1+F^2)}. \tag{4.4.12}$$

The Jacobian matrices $J(\mathbf{a})$, $J(\mathbf{b})$, and $J(\mathbf{c})$ are sparse. Let $J(\cdot)_m$ be the mth column of $J(\cdot)$. Then,

$$\begin{aligned} J(\mathbf{a})_{3i-2:3i} &= -\mathbf{I}_3 \\ J(\mathbf{a})_{(3j-2):(3j)} &= \mathbf{I}_3 \\ J(\mathbf{a})_m &= \mathbf{0} \qquad \text{for all other } m. \end{aligned} \tag{4.4.13}$$

The equations for $J(\mathbf{b})$ and $J(\mathbf{c})$ are similar with j replaced by k and l, respectively.

Then gradient of a is,

$$\nabla a = \frac{\mathbf{a}^T J(\mathbf{a})}{a} \tag{4.4.14}$$

which means

$$\begin{aligned} (\nabla a)_{3i-2:3i} &= \frac{-\mathbf{a}^T}{a} \\ (\nabla a)_{3j-2:3j} &= \frac{\mathbf{a}^T}{a} \\ (\nabla a)_m &= 0 \qquad \text{for all other } m. \end{aligned} \tag{4.4.15}$$

where $(\nabla a)_m$ is the mth element of ∇a. Equivalent equations hold for $\mathbf{b}$ and $\mathbf{c}$.

4.4.2 $|T| = 2$

Volume and Surface Area

Consider $T = \{p_i, p_j\}$. The formulas for the volume and surface area of the intersection of the two balls, p_i and p_j are

$$V_T = \pi h_i^2 \left(\sqrt{p_i''} - \frac{h_i}{3} \right) + \pi h_j^2 \left(\sqrt{p_j''} - \frac{h_j}{3} \right) \tag{4.4.16}$$

$$S_T^{(i)} = 2\pi \sqrt{p_i''} h_i \tag{4.4.17}$$

$$S_T^{(j)} = 2\pi \sqrt{p_j''} h_j \tag{4.4.18}$$

$$S_T = S_T^{(i)} + S_T^{(j)} \tag{4.4.19}$$

where h_i and h_j are the heights of the spherical caps of p_i and p_j (See Figure 4.4).

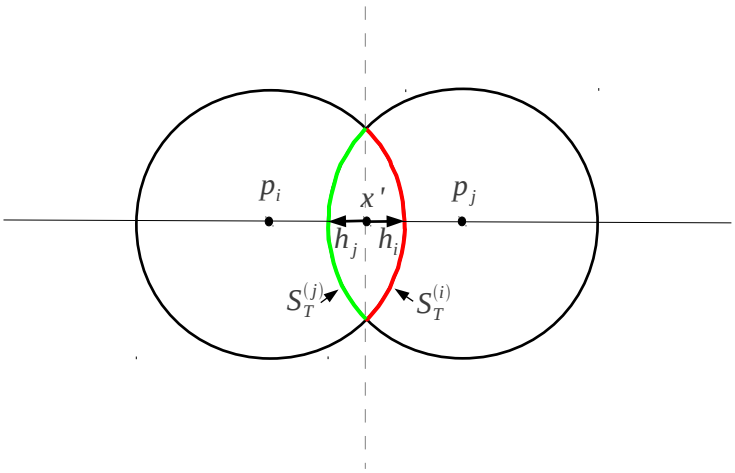


Figure 4.4: Heights of the spherical caps and the partition of the surface area S_T between two atoms.

The characteristic point, $\mathbf{x} \in \mathbb{R}^3 \times \mathbb{R}$, of the edge σ_T satisfies

$$\begin{aligned} \Pi(\mathbf{p}_i, \mathbf{x}) &= 0 \\ \Pi(\mathbf{p}_j, \mathbf{x}) &= 0 \\ \mathbf{x}' &= \mathbf{p}_i' + t(\mathbf{p}_j' - \mathbf{p}_i') \end{aligned} \tag{4.4.20}$$

for some scalar t (See Figure 4.5). Equations 4.4.20 gives

$$p_i'^2 - p_j'^2 - 2\mathbf{x}' \cdot (\mathbf{p}_j' - \mathbf{p}_i') - p_i'' + p_j'' = 0.$$

Define $k = p_i'' - p_j''$ and $\mathbf{a} = \mathbf{p}_j' - \mathbf{p}_i'$. Then

$$\begin{aligned} t &= \frac{k - {p_i'}^2 + {p_j'}^2 - 2{\mathbf{p}'}_i \cdot (\mathbf{p}_j' - \mathbf{p}_i')}{2a^2} \\ &= \frac{1}{2}\left(\frac{k}{a^2} + 1\right) \end{aligned} \tag{4.4.21}$$

and

$$\begin{aligned} h_i &= \sqrt{p_i''} - sgn(t)|\mathbf{x}' - {\mathbf{p}'}_i| \\ h_j &= \sqrt{p_j''} - sgn(1-t)|\mathbf{x}' - {\mathbf{p}'}_j|. \end{aligned} \tag{4.4.22}$$

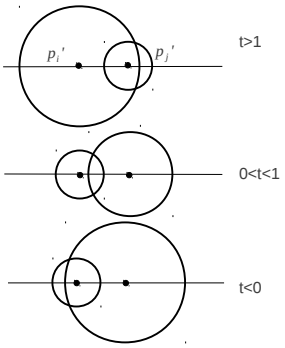


Figure 4.5: The quantity t satisfies $\mathbf{x}' = \mathbf{p}_i' + t(\mathbf{p}_j' - \mathbf{p}_i')$. Configurations which produce t in the three ranges of interest are shown.

Writing $\mathbf{x}' - \mathbf{p}_i' = t\mathbf{a}$ and $\mathbf{x}' - \mathbf{p}_j' = -(1-t)\mathbf{a}$

$$h_i = \sqrt{p_i''} - ta \tag{4.4.23}$$

$$h_j = \sqrt{p_j''} - (1-t)a. \tag{4.4.24}$$

Then

$$\begin{aligned}
\nabla V_T &= \pi\left[2h_i\left(\sqrt{p_i''}-\frac{h_i}{3}\right)\nabla h_i-\frac{h_i^2}{3}\nabla h_i\right] \\
&\quad +\pi\left[2h_j\left(\sqrt{p_j''}-\frac{h_j}{3}\right)\nabla h_j-\frac{h_j^2}{3}\nabla h_j\right] \\
&= \pi\left[\left(2\sqrt{p_i''}-h_i\right)h_i\nabla h_i+\left(2\sqrt{p_j''}-h_j\right)h_j\nabla h_j\right] \qquad (4.4.25)\\
\nabla S_T^{(i)} &= 2\pi\sqrt{p_i''}\nabla h_i \qquad (4.4.26)\\
\nabla S_T^{(j)} &= 2\pi\sqrt{p_j''}\nabla h_j \qquad (4.4.27)\\
\nabla S_T &= \nabla S_T^{(i)}+\nabla S_T^{(j)}. \qquad (4.4.28)
\end{aligned}$$

The gradients of h_i and h_j are

$$\begin{aligned}
\nabla h_i &= -\big(a\nabla t+t\nabla a\big) \qquad (4.4.29)\\
\nabla h_j &= a\nabla t-(1-t)\nabla a
\end{aligned}$$

where ∇a is given in the previous section and

$$\nabla t=-\frac{k\nabla a}{a^3}. \qquad (4.4.30)$$

This gives

$$\begin{aligned}
\nabla h_i &= \frac{\nabla a}{2}\left(\frac{k}{a^2}-1\right) \qquad (4.4.31)\\
\nabla h_j &= -\frac{\nabla a}{2}\left(\frac{k}{a^2}+1\right) \qquad (4.4.32)
\end{aligned}$$

or more specifically

$$\begin{aligned}
(\nabla h_i)_{3i-2,3i} &= -\frac{\mathbf{a}^T}{2a}\left(\frac{k}{a^2}-1\right)\\
(\nabla h_i)_{3j-2,3j} &= -(\nabla h_i)_{3i-1,3i}\\
(\nabla h_i)_m &= 0 \qquad \text{for all other } m. \qquad (4.4.33)
\end{aligned}$$

and

$$\begin{aligned}(\nabla h_j)_{3i-2,3i} &= \frac{\mathbf{a}^T}{2a}\left(\frac{k}{a^2}+1\right)\\ (\nabla h_j)_{3j-2,3j} &= -(\nabla h_j)_{3i-2,3i}\\ (\nabla h_i)_m &= 0 \qquad \text{for all other } m. \end{aligned} \tag{4.4.34}$$

Dihedral Angle

Let $\mathcal{I}$ be the set of tetrahedra in $\mathcal{C}$ to which the edge σ_T is incident. For $\sigma_I \in \mathcal{I}$ define ϕ_I as the normalized inner dihedral angle of σ_I along σ_T. Then

$$\Phi_T = 1 - \sum_{\sigma_I \in \mathcal{I}} \phi_T^I \tag{4.4.35}$$

The normalized dihedral angle between planes with normals $\mathbf{n}_k$ and $\mathbf{n}_l$ is

$$\phi = \frac{\arccos(\mathbf{n}_k \cdot \mathbf{n}_l)}{2\pi} \tag{4.4.36}$$

which gives

$$\nabla\phi = \frac{-\nabla(\mathbf{n}_k \cdot \mathbf{n}_l)}{2\pi\sqrt{1-(\mathbf{n}_k \cdot \mathbf{n}_l)^2}}. \tag{4.4.37}$$

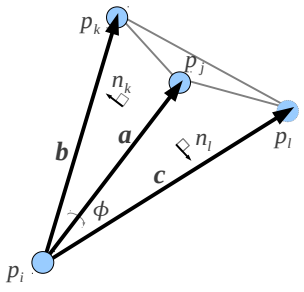


Figure 4.6: Normalized dihedral angle ϕ between planes with normals $\mathbf{n}_k$ and $\mathbf{n}_l$.

Assume plane k is defined by the vectors $\mathbf{a} = \mathbf{p'}_j - \mathbf{p'}_i$ and $\mathbf{b} = \mathbf{p'}_k - \mathbf{p'}_i$ and the

plane l is defined by the vectors $\mathbf{a}$ and $\mathbf{c} = \mathbf{p'}_l - \mathbf{p'}_i$. Then

$$\begin{aligned} \mathbf{n}_k &= \frac{\mathbf{a} \times \mathbf{b}}{|\mathbf{a} \times \mathbf{b}|} \\ \mathbf{n}_l &= \frac{\mathbf{a} \times \mathbf{c}}{|\mathbf{a} \times \mathbf{c}|}. \end{aligned} \tag{4.4.38}$$

Write $\mathbf{n}_k \cdot \mathbf{n}_l = \frac{N}{D}$ where $N = (\mathbf{a} \times \mathbf{b}) \cdot (\mathbf{a} \times \mathbf{c})$ and $D = |\mathbf{a} \times \mathbf{b}||\mathbf{a} \times \mathbf{c}|$. Using vector identities

$$N = (\mathbf{a} \cdot \mathbf{a})(\mathbf{b} \cdot \mathbf{c}) - (\mathbf{b} \cdot \mathbf{a})(\mathbf{a} \cdot \mathbf{c}) \tag{4.4.39}$$

$$D = \sqrt{a^2 b^2 - (\mathbf{a} \cdot \mathbf{b})^2}\sqrt{a^2 c^2 - (\mathbf{a} \cdot \mathbf{c})^2}. \tag{4.4.40}$$

Then

$$\begin{aligned} \nabla N = {} & 2(\mathbf{b} \cdot \mathbf{c})\mathbf{a}^T J(\mathbf{a}) + a^2\Big(\mathbf{b}^T J(\mathbf{c}) + \mathbf{c}^T J(\mathbf{b})\Big) \\ & -(\mathbf{a} \cdot \mathbf{c})\Big(\mathbf{b}^T J(\mathbf{a}) + \mathbf{a}^T J(\mathbf{b})\Big) \\ & -(\mathbf{a} \cdot \mathbf{b})\Big(\mathbf{c}^T J(\mathbf{a}) + \mathbf{a}^T J(\mathbf{c})\Big). \end{aligned} \tag{4.4.41}$$

That is

$$\begin{aligned} \nabla N_{3i-2:3i} &= \Big(\mathbf{a} \cdot (\mathbf{b} + \mathbf{c}) - 2(\mathbf{b} \cdot \mathbf{c})\Big)\mathbf{a}^T \\ &\quad + \Big(\mathbf{a} \cdot \mathbf{c} - a^2\Big)\mathbf{b}^T \\ &\quad + \Big(\mathbf{a} \cdot \mathbf{b} - a^2\Big)\mathbf{c}^T \\ \nabla N_{3j-2:3j} &= 2(\mathbf{b} \cdot \mathbf{c})\mathbf{a}^T - (\mathbf{a} \cdot \mathbf{c})\mathbf{b}^T - (\mathbf{a} \cdot \mathbf{b})\mathbf{c}^T \\ \nabla N_{3k-2:3k} &= a^2\mathbf{c}^T - (\mathbf{a} \cdot \mathbf{c})\mathbf{a}^T \\ \nabla N_{3l-2:3l} &= a^2\mathbf{b}^T - (\mathbf{a} \cdot \mathbf{b})\mathbf{a}^T \\ \nabla N_m &= 0 \qquad \text{for all other } m \end{aligned} \tag{4.4.42}$$

Define $A = \frac{|\mathbf{a}\times\mathbf{c}|}{|\mathbf{a}\times\mathbf{b}|}$. Then

$$\begin{aligned}\nabla D &= A\left(\left(b^2\mathbf{a}^T - (\mathbf{a}\cdot\mathbf{b})\mathbf{b}^T\right)J(\mathbf{a}) + \left(a^2\mathbf{b}^T - (\mathbf{a}\cdot\mathbf{b})\mathbf{a}^T\right)J(\mathbf{b})\right) \\ &+ \frac{1}{A}\left(\left(c^2\mathbf{a}^T - (\mathbf{a}\cdot\mathbf{c})\mathbf{c}^T\right)J(\mathbf{a}) + \left(a^2\mathbf{c}^T - (\mathbf{a}\cdot\mathbf{c})\mathbf{a}^T\right)J(\mathbf{c})\right)\end{aligned} \tag{4.4.43}$$

which gives

$$\begin{aligned}\nabla D_{3i-2:3i} &= -\nabla D_{3j-2:3j} - \nabla D_{3k-2:3k} - \nabla D_{3l-2:3l} \\ \nabla D_{3j-2:3j} &= \left(Ab^2 + \frac{c^2}{A}\right)\mathbf{a}^T - A(\mathbf{a}\cdot\mathbf{b})\mathbf{b}^T - \frac{1}{A}(\mathbf{a}\cdot\mathbf{c})\mathbf{c}^T \\ \nabla D_{3k-2:3k} &= A\left(a^2\mathbf{b}^T - (\mathbf{a}\cdot\mathbf{b})\mathbf{a}^T\right) \\ \nabla D_{3l-2:3l} &= \frac{1}{A}\left(a^2\mathbf{c}^T - (\mathbf{a}\cdot\mathbf{c})\mathbf{a}^T\right) \\ \nabla D_m &= 0 \qquad \text{for all other } m\end{aligned} \tag{4.4.44}$$

From equations 4.4.41 and 4.4.43 we can find $\nabla(\mathbf{n}_k\cdot\mathbf{n}_l) = \frac{D\nabla N - N\nabla D}{D^2}$.

4.4.3 $|T| = 3$

Volume and Surface Area

Consider $T = \{p_i, p_j, p_k\}$. The volume and surface area of the common intersection of three balls can be written as a weighted sum of the surface area of the single and the double intersections. If p_i, p_j, and p_k have a non-empty intersection then there are two points in common with the surfaces of all three balls. Call one of these points $\mathbf{x}'$, define $p_x = (\mathbf{x}', 0) \in \mathbb{R}^3 \times \mathbb{R}$, and let $T_x = \{p_i, p_j, p_k, p_x\}$ (See Figure 4.7). Let S_2 be the set of edges defined by σ_{T_x} and S_1 the set of vertices in σ_{T_x}.

The volume and surface areas (See Figure 4.8) of the intersection of p_i, p_j, and p_k

are given by [20]

$$\frac{1}{2}V = V_{T_c} + \sum_{\sigma_t \in S_2} \Phi_t V_t - \sum_{\sigma_t \in S_1} \Omega_t V_t \quad k,l = (1,2,3) \tag{4.4.45}$$

$$\frac{1}{2}\mathcal{S}_T^{(i)} = \Phi_{\{i,j\}} S_{\{i,j\}}^{(i)} + \Phi_{\{i,k\}} S_{\{i,k\}}^{(i)} - \Omega_{\{i\}} S_{\{i\}} \tag{4.4.46}$$

$$\frac{1}{2}\mathcal{S}_T^{(j)} = \Phi_{\{j,k\}} S_{\{j,k\}}^{(j)} + \Phi_{\{j,i\}} S_{\{j,i\}}^{(j)} - \Omega_{\{j\}} S_{\{j\}} \tag{4.4.47}$$

$$\frac{1}{2}\mathcal{S}_T^{(k)} = \Phi_{\{k,i\}} S_{\{k,i\}}^{(k)} + \Phi_{\{k,j\}} S_{\{k,j\}}^{(k)} - \Omega_{\{k\}} S_{\{k\}} \tag{4.4.48}$$

$$\frac{1}{2}\mathcal{S}_T = \mathcal{S}_T^{(i)} + \mathcal{S}_T^{(j)} + \mathcal{S}_T^{(k)} \tag{4.4.49}$$

where $\Phi_{\{i,j\}}$ is the normalized dihedral angle of σ_{T_C} along the edge $\sigma_{\{i,j\}}$, Ω_i is the normalized solid angle of σ_{T_c} subtended from p_i', and similarly for other combinations i, j, and k.

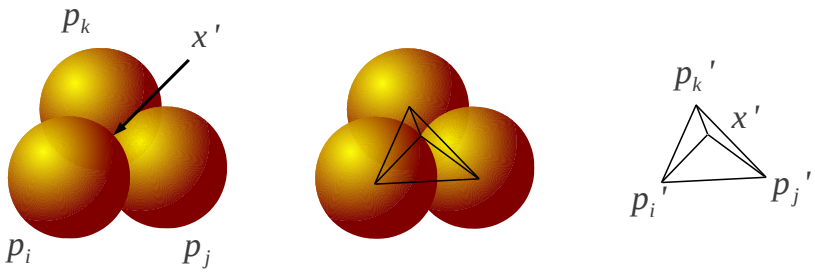


Figure 4.7: Center of characteristic point x' and resulting tetrahedron.

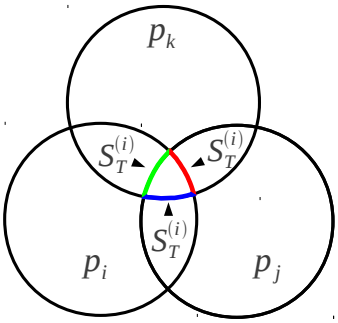


Figure 4.8: Partition of S_T between the three atoms.

We have

$$\begin{aligned}
\nabla V &= \nabla V_{T_c} + \sum_{\sigma_t \in S_2} (\nabla \Phi_t) V_t \\
&\quad + \sum_{\sigma_t \in S_2} \Phi_t (\nabla V_t) - \sum_{\sigma_t \in S_1} (\nabla \Omega_t) V_t \qquad (4.4.50) \\
\nabla \mathcal{S}_T^{(i)} &= (\nabla \Phi_{\{i,j\}}) S_{\{i,j\}}^{(i)} + \Phi_{\{i,j\}} \nabla (S_{\{i,j\}}^{(i)}) \\
&\quad + (\nabla \Phi_{\{i,k\}}) S_{\{i,k\}}^{(i)} + \Phi_{\{i,k\}} (\nabla S_{\{i,k\}}^{(i)}) \\
&\quad - (\nabla \Omega_{\{i\}}) S_{\{i\}}) \qquad (4.4.51)
\end{aligned}$$

similarly for $\mathcal{S}_T^{(j)}$ and $\mathcal{S}_T^{(k)}$, and

$$\nabla \mathcal{S}_T = \nabla \mathcal{S}_T^{(i)} + \nabla \mathcal{S}_T^{(j)} + \nabla \mathcal{S}_T^{(k)} \qquad (4.4.52)$$

The point $\mathbf{x}$ satisfies the following equations

$$|\mathbf{p}'_i - \mathbf{x}'|^2 - p_i'' = 0 \qquad (4.4.53)$$

$$|\mathbf{p}'_j - \mathbf{x}'|^2 - p_j'' = 0 \qquad (4.4.54)$$

$$|\mathbf{p}'_k - \mathbf{x}'|^2 - p_k'' = 0 \qquad (4.4.55)$$

Let $\mathbf{a} = \mathbf{p_j}' - \mathbf{p}'_i$ and $\mathbf{b} = \mathbf{p}'_k - \mathbf{p}'_i$, which gives the normal to the plane that contains σ_T as $\mathbf{n} = \mathbf{a} \times \mathbf{b}$. The characteristic point, $\mathbf{x}_c$, of the triangle σ_T was found in the computation of the alpha complex and satisfies $\mathbf{x} = \mathbf{x}_c + h\mathbf{n}$ where h is the height of $\mathbf{x}$ above the plane containing σ_T. Plugging this into equation 4.4.53 gives

$$|\mathbf{p}'_i - \mathbf{x}_c|^2 - 2h(\mathbf{p}'_i - \mathbf{x}_c) \cdot \mathbf{n} + h^2 n^2 - p''_i = 0 \tag{4.4.56}$$

The vector $(\mathbf{p}'_i - \mathbf{x}_c)$ is orthogonal to $\mathbf{n}$ which gives

$$h = \frac{\sqrt{p''_i - |\mathbf{p}'_i - \mathbf{x}_c|^2}}{n} = \frac{\sqrt{-\alpha_{\sigma_T}}}{n} \tag{4.4.57}$$

where α_{σ_T} is the size of the triangle σ_T.

Euqations 4.4.53, 4.4.54, and 4.4.55 can be differentiated implicitly to obtain $J(\mathbf{x}) = \mathbf{M}^{-1}Y$ where

$$\mathbf{M} = \begin{bmatrix} (\mathbf{p}_i - \mathbf{x})^T \\ (\mathbf{p}_j - \mathbf{x})^T \\ (\mathbf{p}_k - \mathbf{x})^T \end{bmatrix}, \qquad Y = \begin{bmatrix} (\mathbf{p}_i - \mathbf{x})^T J(\mathbf{p}_i) \\ (\mathbf{p}_j - \mathbf{x})^T J(\mathbf{p}_j) \\ (\mathbf{p}_k - \mathbf{x})^T J(\mathbf{p}_k). \end{bmatrix} \tag{4.4.58}$$

This gives

$$J(\mathbf{x})_{3i-2:3i} = \mathbf{M}^{-1} \begin{bmatrix} (\mathbf{p}_i - \mathbf{x})^T \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{4.4.59}$$

$$J(\mathbf{x})_{3j-2:3j} = \mathbf{M}^{-1} \begin{bmatrix} \mathbf{0} \\ (\mathbf{p}_j - \mathbf{x})^T \\ \mathbf{0} \end{bmatrix} \tag{4.4.60}$$

$$J(\mathbf{x})_{3k-2:3k} = \mathbf{M}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ (\mathbf{p}_k - \mathbf{x})^T \end{bmatrix} \tag{4.4.61}$$

$$J(\mathbf{x})_m = \mathbf{0} \qquad \text{for all other } m \tag{4.4.62}$$

The terms in (4.50) and (4.51) may be found by substituting $\mathbf{x}$ and $J(\mathbf{x})$ into the appropriate equations which can be found in previous sections.

Coefficients

Given a triangle, σ_T, its coefficient is given by

$$c_T = \begin{cases} 1, & \text{if } \sigma_T \text{ is singular} \\ \frac{1}{2} & \text{if } \sigma_T \text{ is regular} \end{cases} \tag{4.4.63}$$

The triangle, σ_T, transitions from singular to regular when an incident tetrahedron becomes part of the alpha complex.

4.4.4 $|T| = 4$

Volume

Consider $T = \{p_i, p_j, p_k, p_l\}$. Let $\mathbf{a} = p'_j - p'_i$, $\mathbf{b} = p'_k - p'_i$, and $\mathbf{c} = p'_l - p'_i$. The volume of the tetrahedron σ_T is given by

$$V_{tetra} = \frac{|\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})|}{6}. \tag{4.4.64}$$

Then ∇V is one-sixth equation 4.4.9.

Chapter 5

Gradient Discontinities

5.1 Continuity and Regionwise Differentiability

Intuitively, the surface area and volume functions, $\vec{\mathcal{S}}$, $\mathcal{S}$, $\mathcal{S}_w$, and $\mathcal{V}$, are continuous functions of the positions vectors of the atoms. These functions do not intrinsically depend on the alpha complex; the alpha complex is merely a useful tool to to compute these quantities. Although the alpha complex, $\mathcal{C}$, is only regionwise continuous, at a discontinuity of $\sigma_T \in \mathcal{C}$ the corresponding surface and volume contributions V_T, and $S_T^{(i)}$ vanish. This preserves the continuity of $\vec{\mathcal{S}}$, $\mathcal{S}$, $\mathcal{S}_w$, and $\mathcal{V}$ as expected.

The functions $\vec{\mathcal{S}}$, $\mathcal{S}$, $\mathcal{S}_w$, and $\mathcal{V}$ are regionwise differentiable with respect to the coordinates of the atom's centers. Note that continuity of $J(\vec{\mathcal{S}})$ at a given configuration implies the continuity of $\nabla\mathcal{S}$ and $\nabla\mathcal{S}_w$. Also, if $\nabla\mathcal{S}_w$ is continuous for arbitrary coefficients, then $J(\vec{\mathcal{S}})$ and $\nabla\mathcal{S}$ are continuous. However, continuity of $\nabla\mathcal{S}$ at a given configuration does not ensure the continuity of $J(\vec{\mathcal{S}})$ and $\mathcal{S}_w$ as will be seen.

We say the alpha complex, $\mathcal{C}$, is discontinuous at a given configuration, X, if and only if there exists a simplex, σ_T, such that in any neighborhood of X there exists configurations where $\sigma_T \in \mathcal{C}$ and $\sigma_T \notin \mathcal{C}$. This happens if and only if $size(\sigma_T) = 0$.

Discontinuous derivatives of the surface area and volume functions occur only at configurations where $\mathcal{C}$ is discontinuous. However a discontinuous alpha complex does not necessarily imply discontinuous derivatives. We shall examine the types of disconti-

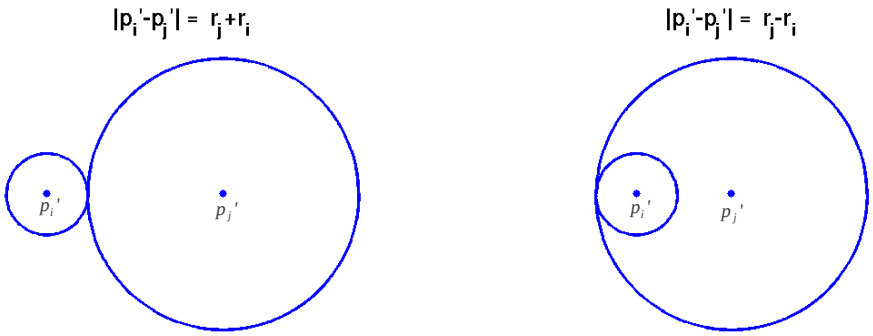
nuities that are encountered in the alpha complex and determine which ones correspond to discontinuous derivatives of the surface area and volume functions.

5.1.1 Binary Discontinuities

Two types of discontinuities occur between pairs of atoms (See Figure 5.1).

B 1. Spheres are externally tangent

B 2. Spheres are internally tangent



a) b)

Figure 5.1: Two types of binary discontinuities: a) External tangency, b) Internal tangency

Given an edge $e_{ij} \in \mathcal{C}$ the gradients of the surface areas $S_{ij}^{(i)}$ and $S_{ij}^{(j)}$ are given by,

$$\begin{aligned} \nabla S_{ij}^{(i)} &= 2\pi r_i \nabla h_i \\ &= \pi \nabla a r_i \Big(\frac{k}{a^2} - 1 \Big) \end{aligned} \tag{5.1.1}$$

and

$$\begin{aligned} \nabla S_{ij}^{(j)} &= 2\pi r_j \nabla h_j \\ &= -\pi \nabla a r_j \Big(\frac{k}{a^2} + 1 \Big) \end{aligned} \tag{5.1.2}$$

with

$$\begin{aligned}\nabla S_{ij} &= 2\pi\Big(r_i\nabla h_i + r_j\nabla h_j\Big) \\ &= \pi\nabla a\left[r_i\Big(\frac{k}{a^2}-1\Big) - r_j\Big(\frac{k}{a^2}+1\Big)\right]. \end{aligned} \tag{5.1.3}$$

In the case where the two spheres are externally tangent, $a = |r_j + r_i|$. As $a \to |r_j + r_i|^-$, ∇S_{ij}^i, ∇S_{ij}^j, and ∇S_{ij} are nonzero. In the case where the two spheres are internally tangent, $a = |r_j - r_i|$, and ∇S_{ij}^i, ∇S_{ij}^j are discontinuous. However ∇S_{ij} is continuous. We have

$$r_i(k - a^2) - r_j(k + a^2) = 0. \tag{5.1.4}$$

Then assuming the two spheres are not identical,

$$\lim_{a \to |r_j - r_i|^+} \nabla S_{ij} = 0 \tag{5.1.5}$$

Thus externally tangent spheres produce discontinuities in $\mathcal{S}$, $\mathcal{S}_w$, and $\vec{\mathcal{S}}$. However internally tangent spheres only produce discontinuities in $\mathcal{S}_w$, and $\vec{\mathcal{S}}$. At such a configuration $\mathcal{S}$ is continuous. Note that if the binary tangency is completely interior to the union of the remaining balls, a derivative discontinuity is not produced.

5.1.2 Tertiary Discontinuities

First we will classify interactions with three atoms and determine which such configurations give continuous surface area gradients. For this we define an atom to be redundant if it is completely interior to another atom.

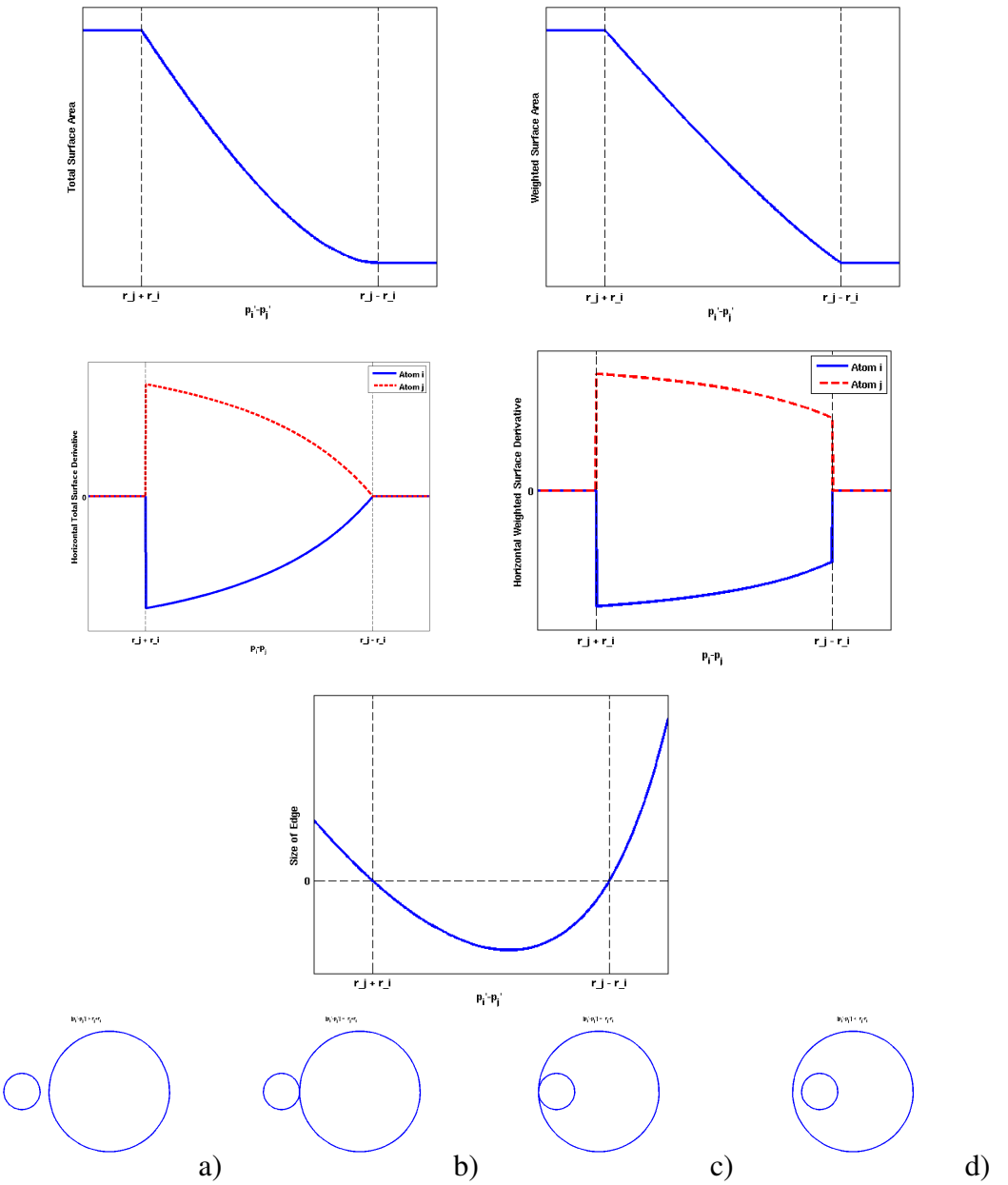


Figure 5.2: Spheres along sample trajectory: a) initial position of atoms, b) external tangency, c) internal tangency, d) final position of atoms

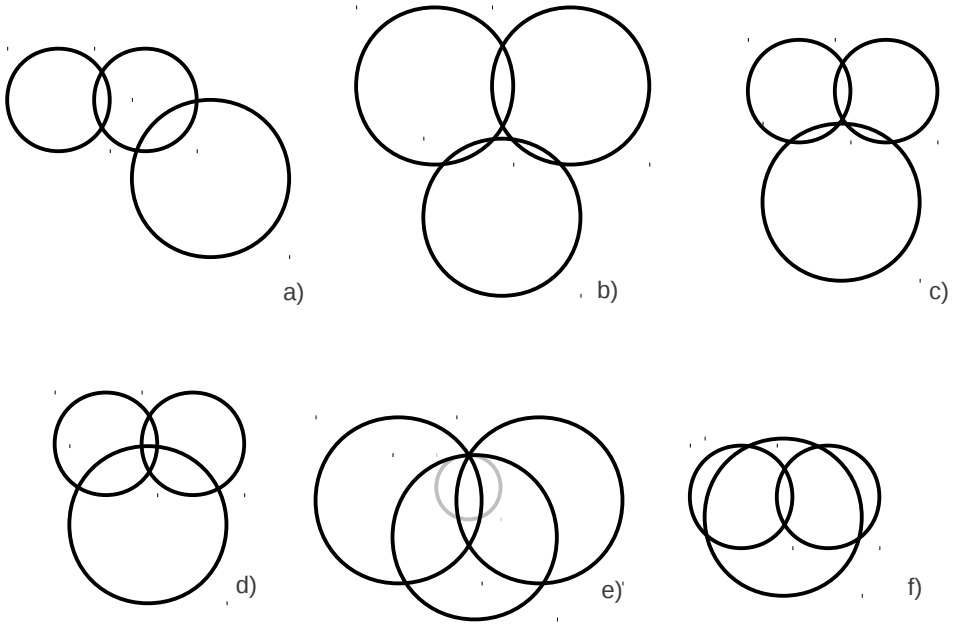


Figure 5.3: Tertiary interactions with no tangencies or redundant atoms.

Tertiary Interactions with No Tangencies or Redundant Atoms

Refer to Figure 5.3. Cases a), b), and f) can be treated as continuous binary interactions and case d) gives continuous derivatives as given by formulas in section 4.4.3. The corresponding triangles given by cases c) and e) have size 0. At such a configuration the alpha complex is discontinous. However, as it will be shown, $\nabla\mathcal{S}$, $\nabla\mathcal{S}_w$, and $J(\vec{\mathcal{S}})$ are continuous.

Continuity of $\nabla\mathcal{S}$, $\nabla\mathcal{S}_w$, and $J(\vec{\mathcal{S}})$ in c) depend on the smoothness of $S_{ijk}^{(l)}$, for $l = i, j, k$, as $S_{ijk} \to 0$. In this case, each of the atoms i, j, and k may be treated identically. We will consider the surface area contribution, $S^{(i)}$, from atom i and the results may be generalized to area contributions from atoms j and k.

In any neighborhood of configuration c), the surface area, $S^{(i)}$, is given by

$$S^{(i)} = \begin{cases} S_i - S_{ij}^{(i)} - S_{ik}^{(i)} & \text{(Case b)} \\ S_i - S_{ij}^{(i)} - S_{ik}^{(i)} + S_{ijk}^{(i)} & \text{(Case d).} \end{cases} \tag{5.1.6}$$

It can be shown, (see Appendix A), that

$$\lim_{S_{ijk}\to 0} S^{(i)}_{ijk} \to 0. \tag{5.1.7}$$

Given equation 5.1.7 and continuity of the involved edges, configuration c) gives a continuous gradient with

$$\nabla S^{(i)} = -\nabla\Big(S^{(i)}_{ij} + S^{(i)}_{ik}\Big). \tag{5.1.8}$$

Similar equations hold for $\nabla S^{(j)}$ and $\nabla S^{(i)}$. This shows continuity of $\nabla\mathcal{S}$, $\nabla\mathcal{S}_w$, and $J(\vec{\mathcal{S}})$ at such a conformation.

Now we will consider case e). To this end we will label the atoms p_i, p_j, and p_k as shown Figure 5.4. Note that in the following arguments, p_i and p_j may be treated equivalently, whereas surface areas involving p_k must be treated differently. As will be shown, case e) and case e) alternate are qualitatively the same. However, we will begin with the first case. In any neighborhood of configuration e) (See Figure 5.4), the surface area contributions are given by

$$S^{(i)} = \begin{cases} S_i - S^{(i)}_{ik} & \text{(Case f)} \\ S_i - S^{(i)}_{ij} - S^{(i)}_{ik} + S^{(i)}_{ijk} & \text{(Case d),} \end{cases} \tag{5.1.9}$$

$$S^{(j)} = \begin{cases} S_j - S^{(j)}_{jk} & \text{(Case f)} \\ S_{(j)} - S^{(j)}_{ij} - S^{(j)}_{jk} + S^{(j)}_{ijk} & \text{(Case d),} \end{cases} \tag{5.1.10}$$

$$S^{(k)} = \begin{cases} S_k - S^{(k)}_{ik} - S^{(k)}_{jk} & \text{(Case f)} \\ S_k - S^{(k)}_{ik} - S^{(k)}_{jk} + S^{(k)}_{ijk} & \text{(Case d).} \end{cases} \tag{5.1.11}$$

We want to consider the case as $S_{ijk} \to S_{ij}$. That is, when case d) approaches case e), or equivalently as case d) approaches case f) since any transition from one to the other requires a passage through case e). As shown in the appendix by equations A.0.8

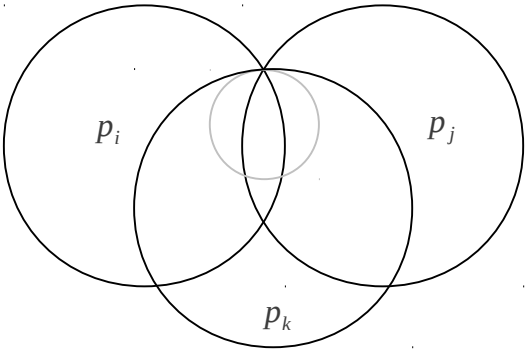


Figure 5.4: Case e.

we have

$$\lim_{S_{ijk}\to S_{ij}^-} \nabla\left(S_{ij}^{(i)} - S_{ijk}^{(i)}\right) = 0 \tag{5.1.12}$$

$$\lim_{S_{ijk}\to S_{ij}^-} \nabla\left(S_{ij}^{(j)} - S_{ijk}^{(j)}\right) = 0 \tag{5.1.13}$$

$$\lim_{S_{ijk}\to S_{ij}^-} \nabla S_{ijk}^{(k)} = 0. \tag{5.1.14}$$

Since the gradient of the remaining terms in equations 5.1.9, 5.1.10, and 5.1.11 are continuous this shows the continuity of $\nabla S^{(i)}$, $\nabla S^{(j)}$, $\nabla S^{(k)}$ and thus $\nabla \mathcal{S}$ and $\nabla \mathcal{S}_w$ at conformation e).

We will now consider case e) alternate. In any neighborhood of configuration e) alt., the surface area contributions are given by (See Figure 5.5),

$$S^{(i)} = \begin{cases} S_i - S_{ij}^{(i)} & \text{(Case g)} \\ S_i - S_{ij}^{(i)} - S_{ik}^{(i)} + S_{ijk}^{(i)} & \text{(Case d)}, \end{cases} \tag{5.1.15}$$

$$S^{(j)} = \begin{cases} S_j - S_{ij}^{(j)} & \text{(Case g)} \\ S_{(j)} - S_{ij}^{(j)} - S_{jk}^{(j)} + S_{ijk}^{(j)} & \text{(Case d)}, \end{cases} \tag{5.1.16}$$

$$S^{(k)} = \begin{cases} 0 & \text{(Case g)} \\ S_k - S_{ik}^{(k)} - S_{jk}^{(k)} + S_{ijk}^{(k)} & \text{(Case d)}. \end{cases} \tag{5.1.17}$$

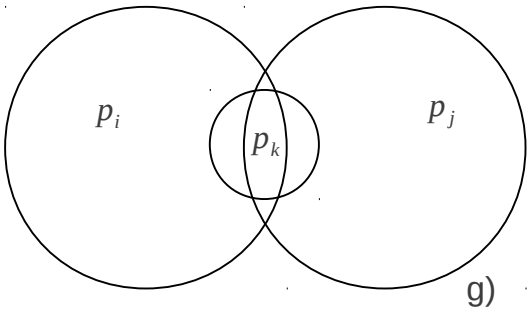


Figure 5.5: Case g.

We want to condider the case as $S^{(k)} \to 0$. That is, when case d) approaches case e), or equivalently as case d) approaches case g) since any transition from one to the other requires a passage through case e). As shown in the appendix, we have

$$\lim_{S^{(k)}\to 0} \quad \nabla\left(S_{ik}^{(i)} - S_{ijk}^{(i)}\right) \quad = 0 \tag{5.1.18}$$

$$\lim_{S^{(k)}\to 0} \quad \nabla\left(S_{jk}^{(j)} - S_{ijk}^{(j)}\right) \quad = 0. \tag{5.1.19}$$

Since the gradient of the remaining terms in equations 5.1.15 and 5.1.16 are continuous this shows the continuity of $\nabla S^{(i)}$ and $\nabla S^{(j)}$. Also shown in the appendix,

$$\lim_{S^{(k)}\to 0} \nabla S_{ijk}^{(k)} = 0. \tag{5.1.20}$$

which shows continuity of $\nabla S^{(k)}$. Thus $\nabla \mathcal{S}$, $\nabla \mathcal{S}_w$, and $J(\vec{\mathcal{S}})$ are continuous at conformation e) alternate.

Figures 5.6 and 5.7 illustrate how a given sphere is cut by the two remaining spheres both case e) and case c).

As seen both types of case e) are qualitatively the same, and the surface of p_k in case e) is qualitatively the same as the surfaces in case c).

Tertiary Interactions with a Binary External Tangency and No Redundant Atoms

It is easy to see that cases shown in Figure 5.8 can be reduced to binary cases with possibly the exception of case ext d) (See Figure 5.9). Consider atoms p_i, p_j, and p_k at

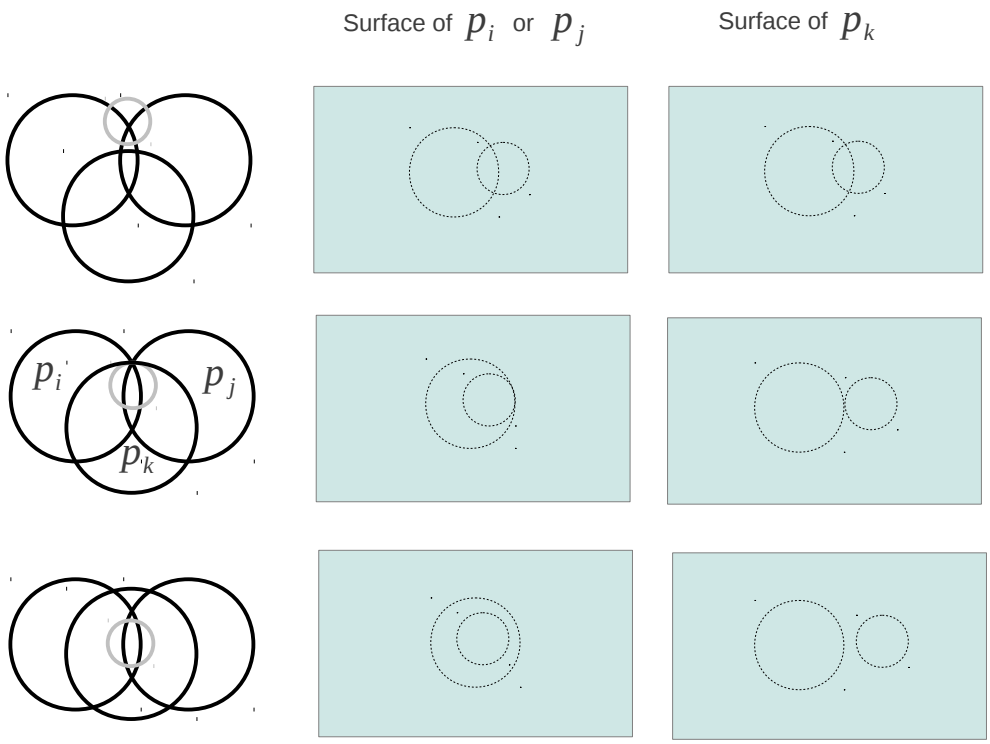


Figure 5.6: Illustration of how a given sphere is cut by two remaining spheres in case e).

a discontinuity of type ext d.

In any neigbhorhood of interactions of type ext d) we have interactions of type ext c), ext e), a), or d), with surface areas given by

$$S = S_i + S_j + S_k - S_{ik} - S_{jk} \quad \text{(cases ext c), ext e), a)}$$

$$S = S_i + S_j + S_k - S_{ik} - S_{jk} - Sij + S_{ijk} \quad \text{(case d).}$$

We have

$$S^i = \begin{cases} S_i - S^i_{ik} & \text{(Cases ext c), ext e), a)} \\ S_i - S^i_{ij} - S^i_{ik} - S^i_{ijk} & \text{(Case d),} \end{cases}$$

$$S^j = \begin{cases} S_j - S^j_{jk} & \text{(Cases ext c), ext e), a)} \\ S_j - S^j_{ij} - S^j_{jk} - S^j_{ijk} & \text{(Case d),} \end{cases}$$

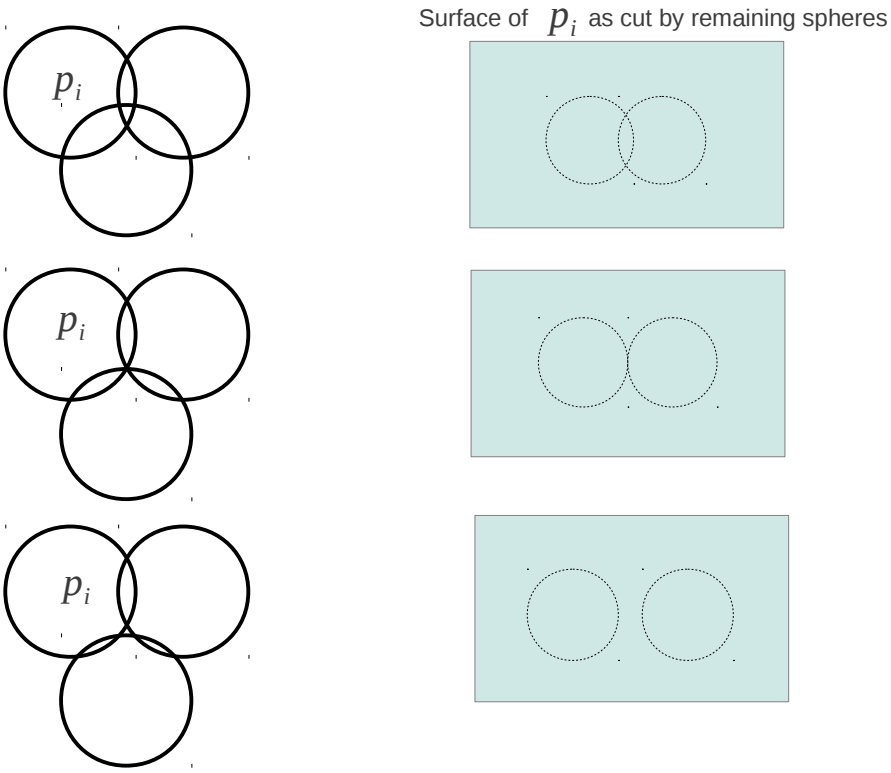


Figure 5.7: Illustration of how a given sphere is cut by two remaining spheres in case c).

and

$$S^k = \begin{cases} S_k - S^k_{ik} - S^k_{jk} & \text{(Cases ext c), ext e), a))} \\ S_k - S^k_{ik} - S^k_{jk} + S^k_{ijk} & \text{(Case d).} \end{cases}$$

Since the binary discontinuity does not affect the surface area, $S^{(k)}$, of atom k and by equation A.0.7 we get a continuous gradient with

$$\nabla S^k = -\nabla(S_{ik} + S_{jk}). \tag{5.1.21}$$

Furthermore, by equation A.0.7, the only terms in 5.1.21 and 5.1.21 that cause discontinuities in the ∇S^i and ∇S^j are S^i_{ij} and S^j_{ij}, respectively. This type of discontinuity has already been discussed.

Tertiary Interactions with a Binary Internal Tangency, No Redundant Atoms, or Intersection on a Circle

Refer to Figure 5.10. Cases a) and b) can be reduced to cases already discussed.

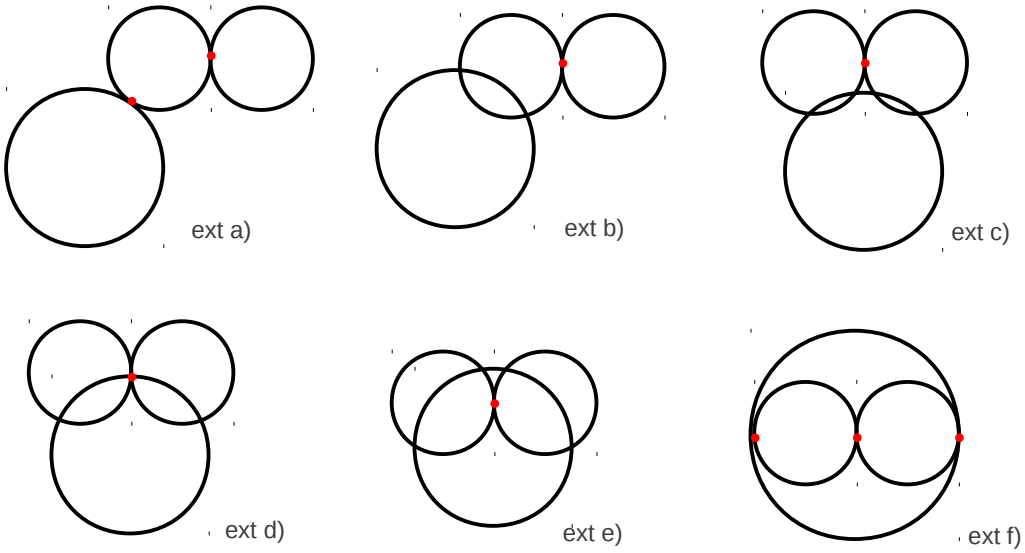


Figure 5.8: Tertiary interactions with a binary external tangency and no redundant atoms

Intersection on a Common Circle

If three or more spheres intersect on a common circle then a discontinuous surface area gradient is produced (See Figure 5.11).

The forces on each atom for a sample trajectory of three atoms that pass through a common circle of intersection are shown in Figure 5.12.

5.1.3 Quaternary interactions

We want to consider the case when a tetrahedron is discontininuous, that is when the surface of the four balls, p_i, p_j p_k, and p_l, intersect at a point. In this case, the three circles of intersection on the surface of the fourth ball have a common point (See Figure 5.13).

In any neighborhood of discontinuity a), (see Figure 5.14), the surface area $S^{(m)}$ with $m = i, j, k, l$ is given by

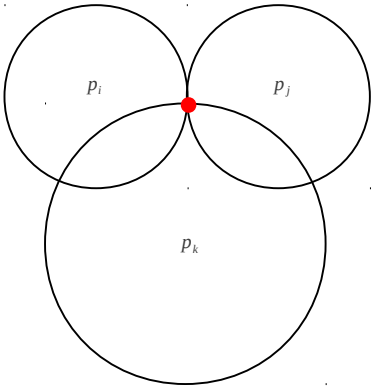


Figure 5.9: Case ext d).

$$S^{(m)} = S_m - \sum_{n=i,j,k,l \neq m} S^{(m)}_{mn} + \sum_{\substack{p,n=i,j,k,l \\ n \neq p; n,p \neq m}} S^{(m)}_{mnp} - S^{(m)}_{ijkl} \qquad \text{case a) (5.1.22)}$$

$$S^{(m)} = S_m - \sum_{n=i,j,k,l \neq m} S^{(m)}_{mn} + \sum_{\substack{p,n=i,j,k,l \\ n \neq p; n,p \neq m}} S^{(m)}_{mnp} \qquad \text{case c).}$$

As shown in the appendix,

$$\lim_{S_{ijkl} \to 0^+} \nabla S^{(m)}_{ijkl} = 0 \qquad (5.1.23)$$

which shows $\nabla \mathcal{S}$ is continuous.

Figure 5.15 shows the forces acting on a four atom trajectory and summarizes qualitative effect of discontinuous simplices on the calculated force. The forces on each atom are represented by a different color.

5.2 Near Discontinuities

As shown in the previous section, the surface area gradients are discontinuous if and only if an edge exterior to the alpha complex has size zero or three or more spheres are cocircular. We will call the first type a generic discontinuity and the second type a nongeneric discontinuity. Since nongeneric discontinuities are very rarely found in practice we ignore this type for the remainder of this section.

We want to know when a configuration is near discontinuous. That is, given ϵ, there exists an atom, p_i, and a direction, d, such that if p_i moves distance ϵ in the direction d the surface area gradient is discontinuous.

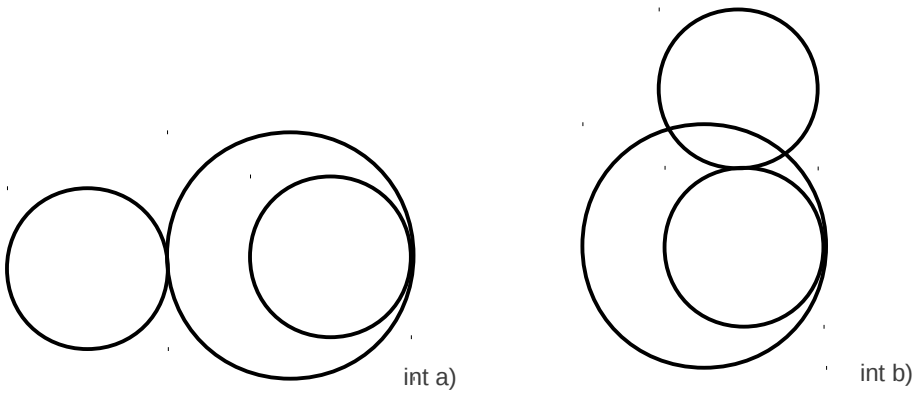


Figure 5.10: Tertiary interactions with a binary internal tangency, no redundant atoms, or intersection on a circle

5.3 Implementation

In practice is is very unlikely that an edge will be exactly at a discontinuity. Even assuming that analytically the edge is exactly at a discontinuity, the size of an edge may be calculated as $O(10^{-16})$ due to inexact arithmetic.

The program accepts a tolerance, ϵ, such that if an edge is "ϵ-near" a dicontinuity a discontinuous force and direction will be returned. An atom is ϵ-near to a discontinuity if it can move any distance less than ϵ with out encountering a binary discontinuity.

An edge, e_{ij}, is at a discontinuity if the size of the edge is equal to 0. However, the size of the edge does not tell us exactly how far either atom can move without encountering a tangency. When the size of an edge is 0 either $d_e = |p'_i - p'_j| - r_i - r_j = 0$ (external tangency), or $d_i = |p'_i - p'_j| - |r_i - r_j| = 0$ (internal tangency). Atoms in the edge with either $|d_e| \leq \epsilon$ or $|d_i| \leq \epsilon$ are marked as giving discontinuous forces. For each such atom a base force is returned. The base force for an atom with $d_e \neq 0$ and $d_i \neq 0$ is the force calculated exactly at the given configuration. If d_e or d_i is exactly

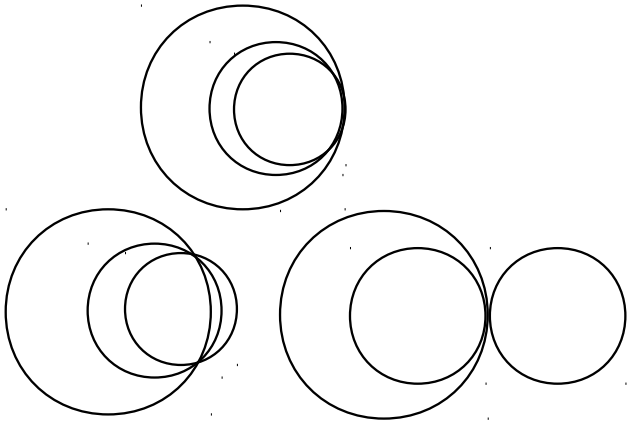


Figure 5.11: Three spheres intersect along a common circle.

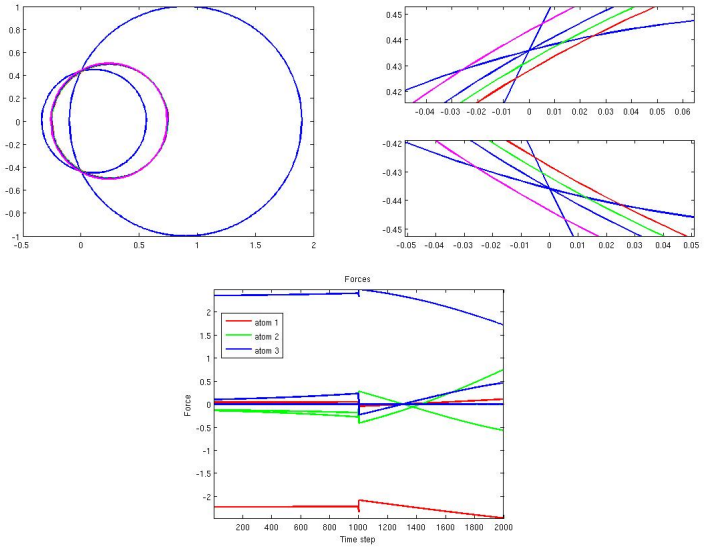


Figure 5.12: Forces as three spheres pass through a common circle of intersection.

zero, the force at that point is undefined. In any neighborhood, a force is well defined and one of these is arbitrarily taken to be the "base" force. For each edge a direction is also returned. This is the direction vector between the two atom centers. It means that for the given atom, any movement in the $p_i' - p_j'$ direction with distance less than d_e (or when appropriate d_i) does not introduce a discontinuous force. A "correction" vector is also returned. That is, if the atom moves over the plane of discontinuity the resulting force on the atom is the "base" force plus the "correction" force. Another option is to return a derivative smoothed by a spline.

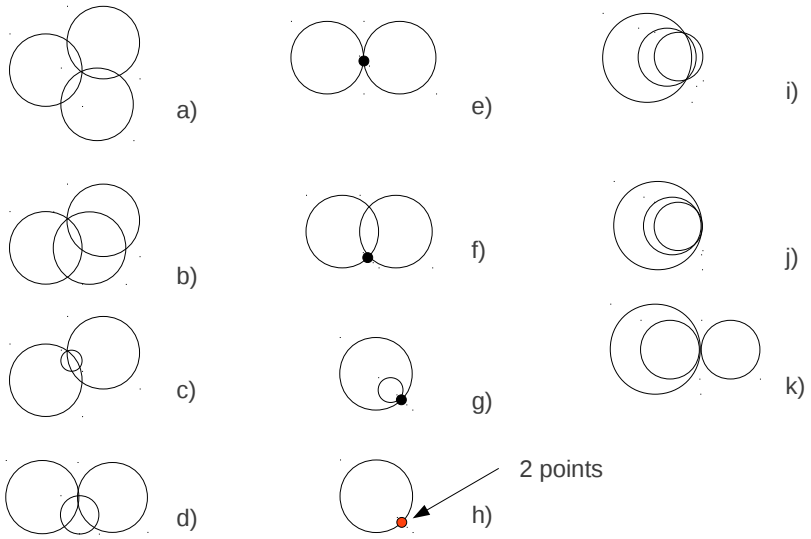


Figure 5.13: Tetrahedron is discontinuous when the surface of four spheres have a common intersection.

5.4 Statistics

A set of 4500 conformations of simulated HIV protease trajectory were examined for near force discontinuities ($\epsilon = 10^{-5}$) with a solvent radius of 0 and 1.4. The total results over all conformations are shown in the following table and Figure 5.16.

Solvent radius	# of edges near disc.	Minimum distance from disc.	# of nonsingular edges near disc.
0	4839	$4.4409e^{-16}$	106
1.4	326	$9.6154e^{-7}$	14

For both solvent radii, certain distances lock into place for the later conformations. Figure 5.17 shows that each of these distances correspond to atom pairs and is an artifact of the calculated trajectory.

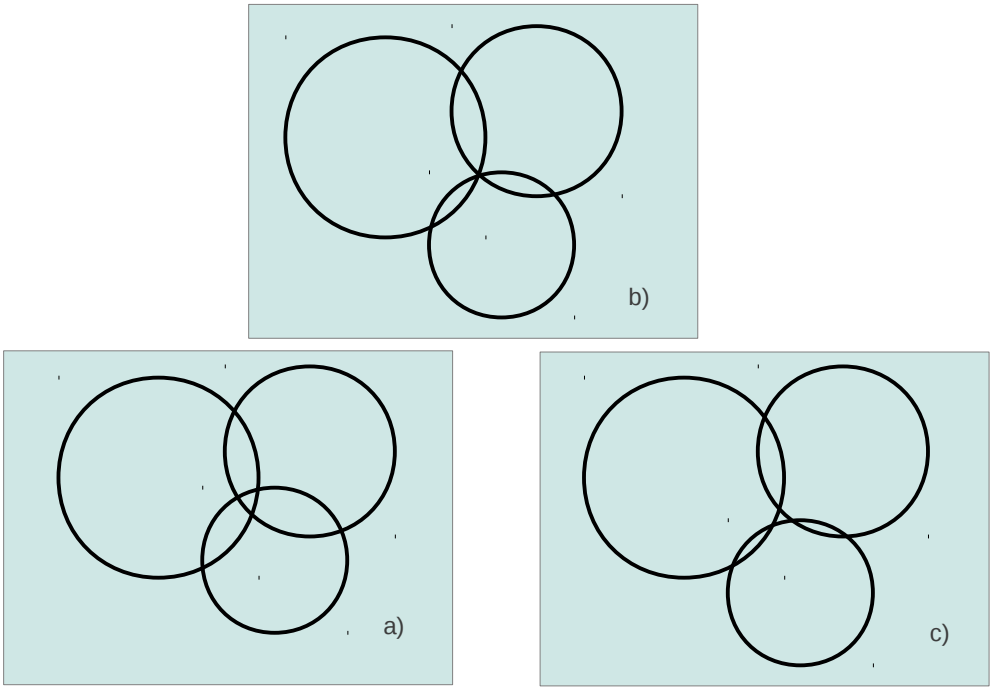


Figure 5.14: Discontinuity and near discontinuities.

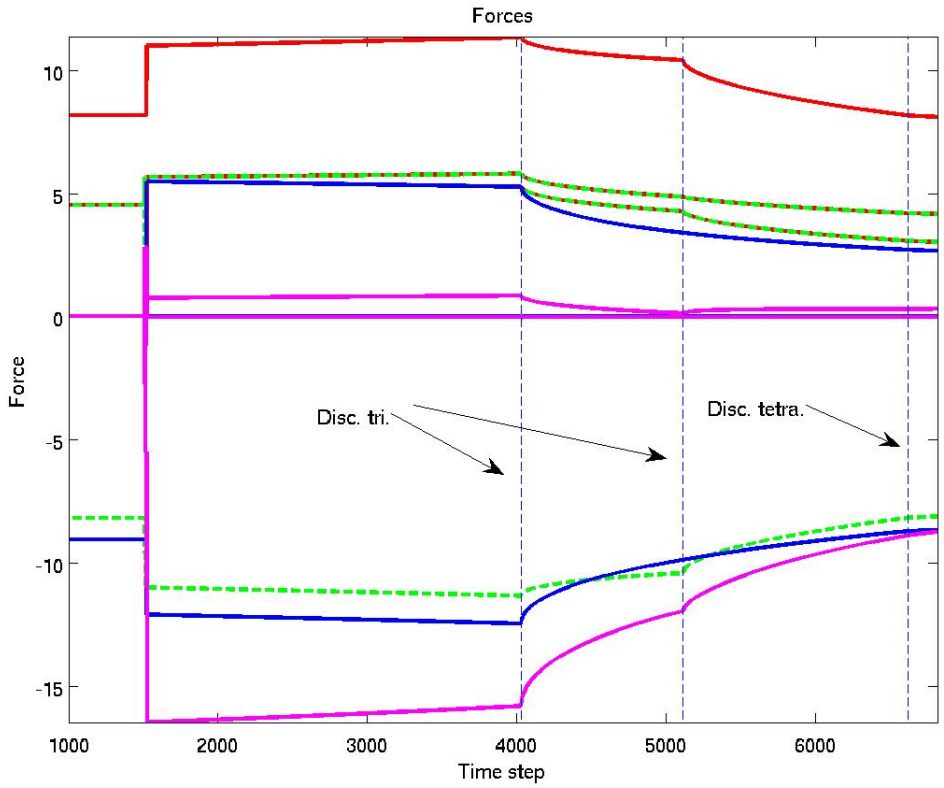


Figure 5.15: Forces on a four atom trajectory summarizes qualitative effect of discontinuous simplices.

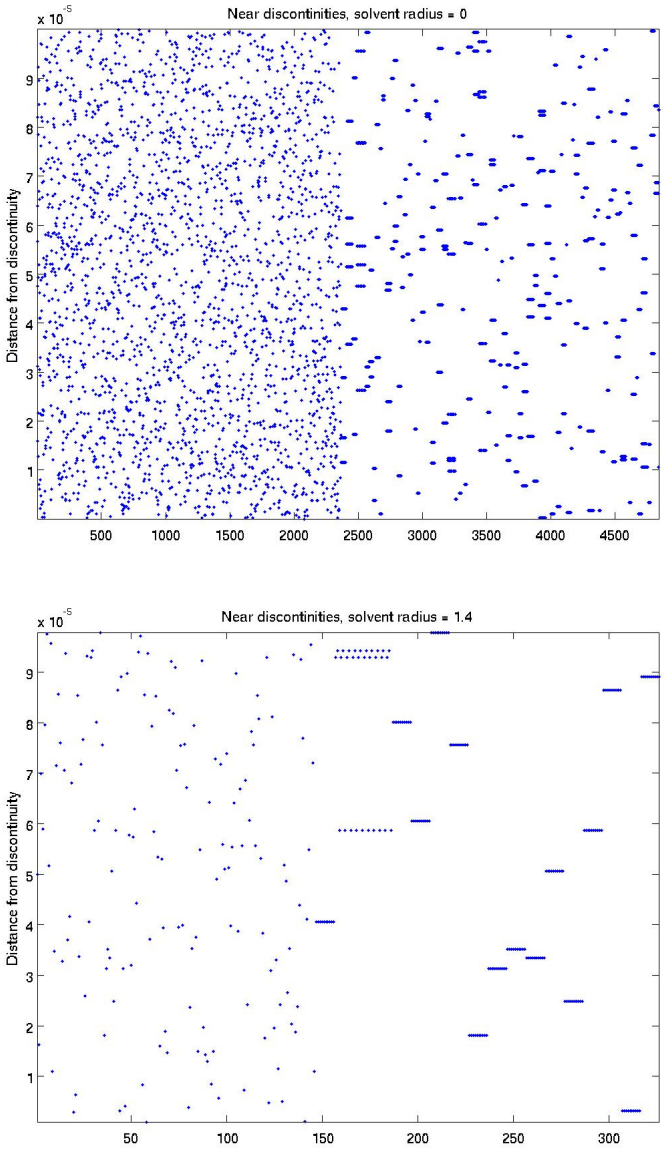


Figure 5.16: Distances from discontinuities for HIV protease trajectory for solvent radii $r = 0$ Angstroms and $r = 1.4$ Angstroms.

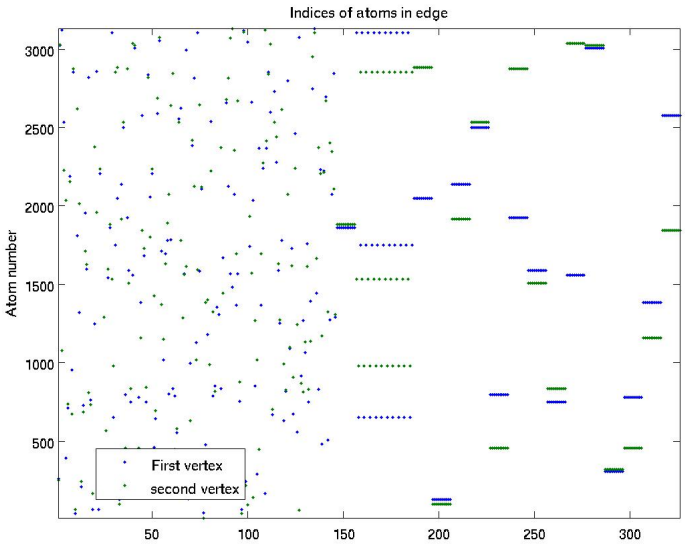


Figure 5.17: Atoms involved in near discontinuities for HIV protease trajectory. Solvent radius is 1.4 Angstroms.

Part III

Solvation Free Energy, Forces, and SEA

Chapter 6

Solvation Model Semi-Explicit Assembly (SEA)

6.1 Modeling Aqueous Solvation by Semi-Explicit Assembly (SEA)

We would like to study the effect of solvent on a solute molecule, in particular we would like to calculate the free energy of solvation, ΔG.

Explicit methods, which surround a solute by explicit water molecules, tend to be slow; thousands to tens of thousands of waters are added to the system when the solute is hydrated. Implicit solvent models typically treat the solvent as a continuous medium and the free energy of solvation is commonly calculated as

$$\Delta G = \sum_{i=1}^{N} \gamma_i A_i \tag{6.1.1}$$

where N is the number of solute atoms, A_i is the individual atomic surface area of atom i and γ_i is a parameter that depends on atom type [13]. While implicit methods are generally faster than explicit methods, accuracy decreases due to the neglect of certain physical properties of the system, such as discrete microscopic effects of the water molecules, dispersive interactions, and the effect of the specific shape of the solute on

the free energy.

A method called *Semi-Explicit Assembly* or SEA, developed by Fennell et al. [17], [18], [22], better captures the physics of the system and gives accuracy close to explicit models but is computationally fast like implicit models. The method will be visited briefly here. Refer to [17], [18], and [22] for a more in-depth discussion.

The free energy of solvation may be modeled as

$$\Delta G = \Delta G_{np} + \Delta G_{pol} \tag{6.1.2}$$

where ΔG_{np} is due to cavity formation (nonpolar interactions), and ΔG_{pol} is due to electrostatic interactions. We first discuss the nonpolar free energy of solvation followed by the polar free energy.

6.1.1 Nonpolar Free Energy of Solvation

Steric repulsion and attractive dispersion interactions are handled using the Lennard-Jones potential

$$V_{LJ}(r_{ij}) = \begin{cases} 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] & r_{ij} \leq r_c \\ 0 & r_{ij} > r_c \end{cases} \tag{6.1.3}$$

where r_{ij} is the distance between particles, r_c is a cut-off distance, ϵ_{ij} is the well-depth parameter, and σ_{ij} is the size parameter. A one-time precomputation step consists of using explicit solvent simulations to compute behaviors of waters around nonpolar solute spheres with different Lennard-Jones parameters σ and ϵ. Nonpolar solvation free energies (See Figure 6.1), and water distances r_w (See Figure 6.2 a), are calculated for a wide range of σ and ϵ. This step only needs to be performed once for a given temperature, pressure, or solvent model.

Next,the free energy of solvation is approximated for an arbitrary solute. For solute atom i, r_w is interpolated from precalculated values and the solvent-accessible

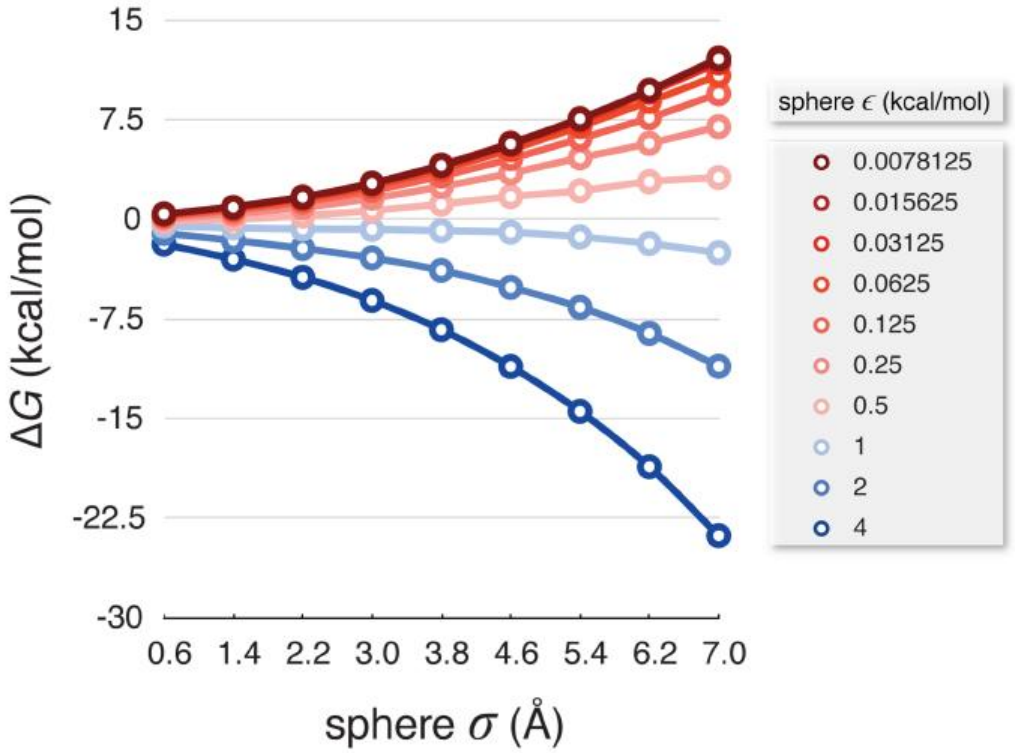


Figure 6.1: Nonpolar solvation free energy (ΔG) of single LJ spheres in TIP3P water at 300 K as a function of their σ and ϵ parameters. Unfavorable ΔG values are red. Favorable ΔG values are blue. Figure from [17].

surface area is computed (See Figure 6.2 b). Dot sites are generated on the atom's solvent-accessible surface and Lennard-Jones potential fields are probed along each atom center-dot site vector (See Figure 6.2 c). Potentials are averaged over the dot sites which generates a region-averaged dispersion potential, $V_{ra,i}$. This potential is fitted to a Lennard-Jones curve to extract region-averaged parameters $\sigma_{ra,i}$ and $\epsilon_{ra,i}$. Given these parameters, the atomic free energy, ΔG_i, is interpolated from Figure 6.1.

Total nonpolar energy of solvation is related to a weighted sum of atomic free energies

$$\Delta G_{np} = pV_v + \sum_{i=1}^{N} f_i \Delta G_i(\sigma_{ra,i}, \epsilon_{ra,i}) \tag{6.1.4}$$

where f_i is the fractional solvent-accessible surface area of atom i. The pV_v term is due to buried particles in the molecule.

6.1.2 Polar Free Energy of Solvation

The ΔG_{pol} term is found by a procedure similar to that of the ΔG_{np} term. Explicit solvent free energy calculations are performed on a series of charged Lennard-Jones spheres. The resulting energies are collected as a function of the curvature, C_i, and electric field, $\mathbf{E}_i$, at the solvent-accessible surface. Total free energy is calculated as a weighted sum over all solute atoms

$$\Delta G_{pol} = \sum_{i=1}^{N} f_i \Delta G_{pol}(\mathbf{E}_i, C_i). \tag{6.1.5}$$

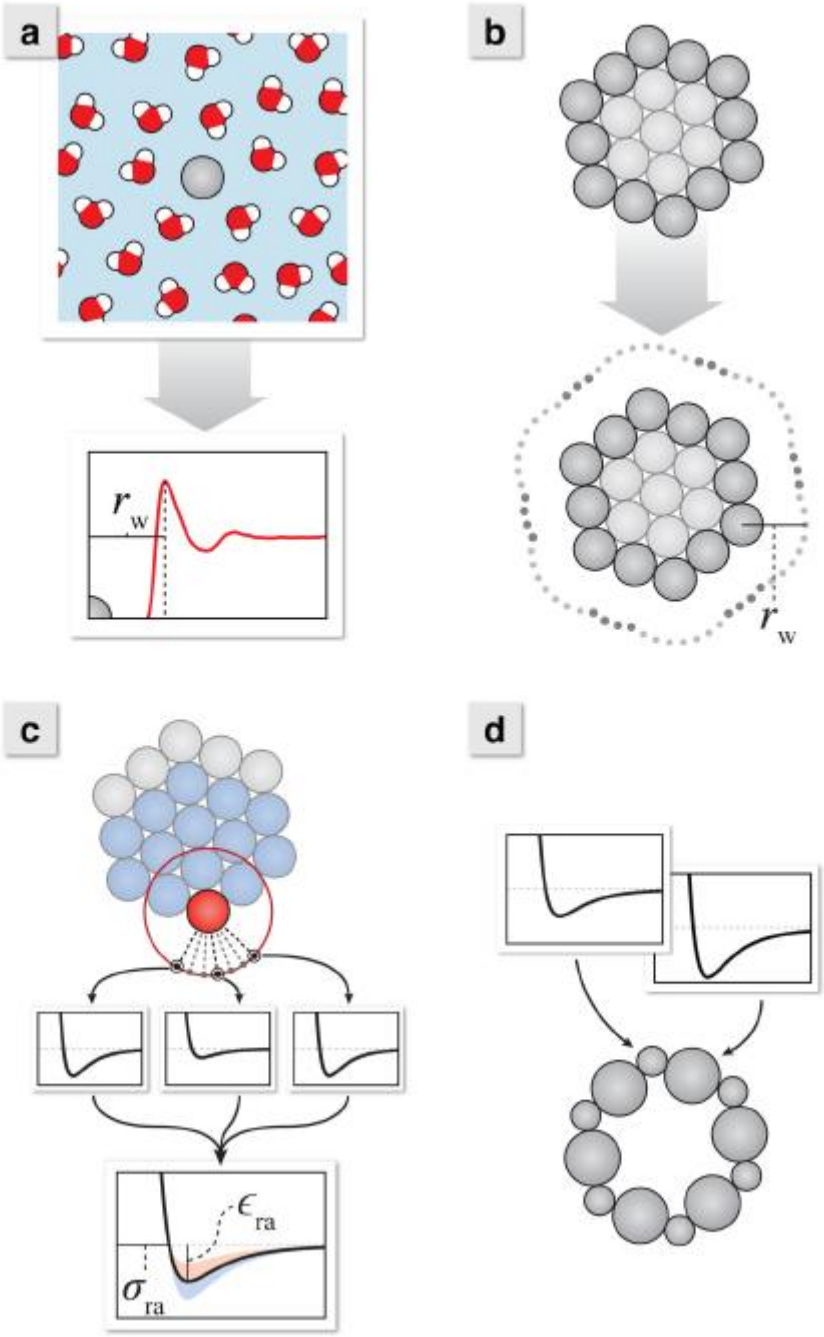


Figure 6.2: The process for incorporating non-additive environmental effects on the solute surface atoms. (a) Sample LJ spheres in explicit water and build a map of water distances (r_w) as a function of σ and ϵ. (b) Construct the solvent accessible surface (SAS) using the distances from the explicit solvent map. (c) Probe the LJ potential of the solute along the line connecting each SAS dot to its surface atom. Average these potentials for each surface atom, and extract new "effective" LJ parameters (σ_{ra} and ϵ_{ra}) from this curve. (d) Use these effective potential parameters when calculating the solvation free energy. Note that edge atoms will have more attractive ϵ_{ra} values than corner atoms because of the greater number of atoms near to the probe particle. Figure from [17].

Chapter 7

Interpolation of Nonpolar SEA Coefficients

The nonpolar free energy of solvation calculated by the SEA method may be written

$$\begin{aligned} \Delta G &= pV_v + \sum_{i=1}^{N} f_i \Delta G_i \\ &= pV_v + \sum_{i=1}^{N} \gamma_i A_i. \end{aligned} \tag{7.0.1}$$

Here $A_i = f_i \cdot A_{i,total}$ is the atomic solvent-accessible surface area. The SEA coefficient $\gamma_i = \frac{\Delta G_i}{A_{i,total}}$ depends on the specific conformation of the solute molecule and not just the atom type as in Equation 6.1.1.

We would like to determine an interpolation method for γ_i. To this end, the nonpolar coefficients for 10001 conformations of each of the 22 dipeptides were found using the SEA algorithm.

7.1 Dipeptide Trajectories

For a given dipeptide, let $\gamma_{i,j}$ and $A_{i,j}$ be the nonpolar SEA coefficient and atomic accessible surface area for atom i in conformation j. Then the nonpolar free energy of

hydration of conformation j is

$$\Delta G_{np,j} = \sum_{i=1}^{N} \gamma_{i,j} A_{i,j} \tag{7.1.1}$$

7.1.1 Average Coefficients

For a given dipeptide, we define the average coefficient of atom i as

$$\overline{\gamma_i} = \frac{1}{10001} \sum_{j=1}^{10001} \gamma_{i,j} \tag{7.1.2}$$

The nonpolar free energy of hydration of conformation j using average coefficients is

$$\overline{\Delta G_{np,j}} = \sum_{i=1}^{N} \overline{\gamma_i} A_{i,j}. \tag{7.1.3}$$

We calculate the "error due to averaging" by the relative difference

$$E_{av} = \sum_{j=1}^{10001} \left| \frac{\overline{\Delta G_{np,j}} - \Delta G_{np,j}}{\Delta G_{np,j}} \right| \tag{7.1.4}$$

Errors due to average for each dipeptide are given in Figure 7.2. Errors range from about 8% for arginine to less than 3% for alanine.

7.1.2 Coefficient Linear Estimate

We found that for a given atom in a given dipeptide, there is a linear relationship between $\gamma_{i,j}$ and $A_{i,j}$. (See Figure 7.1). The coefficient data was fitted to a line via least squares

$$\hat{\gamma}_i = \alpha_i A_i + \beta_i \tag{7.1.5}$$

Certain atoms, such as the alpha carbons, in each dipeptide exhibited bimodal behavior. However, the bimodality was only displayed for extremely small ($< 10^{-3}$) values of the area; due to the γA form of ΔG this did not pose a problem. The average

coefficients were used in these cases.

The nonpolar solvation free energy of conformation j using linear coefficients is

$$\begin{aligned} \Delta\hat{G}_{np,j} &= \sum_{i=1}^{N} \hat{\gamma_{i,j}} A_{i,j} \\ &= \sum_{i=1}^{N} (\alpha_i A_{i,j} + \beta_i) A_{i,j} \end{aligned} \tag{7.1.6}$$

We calculate the "error due to the linear approximation" by the relative difference

$$E_{lin} = \sum_{j=1}^{10001} \left| \frac{\Delta\hat{G}_{np,j} - \Delta G_{np,j}}{\Delta G_{np,j}} \right| \tag{7.1.7}$$

Errors due to the linear approximation for each dipeptide are given in Figure 7.2. As expected, the E_{lin} is less than E_{av}. Errors range from less that 5% for histidine to about 2% for valine.

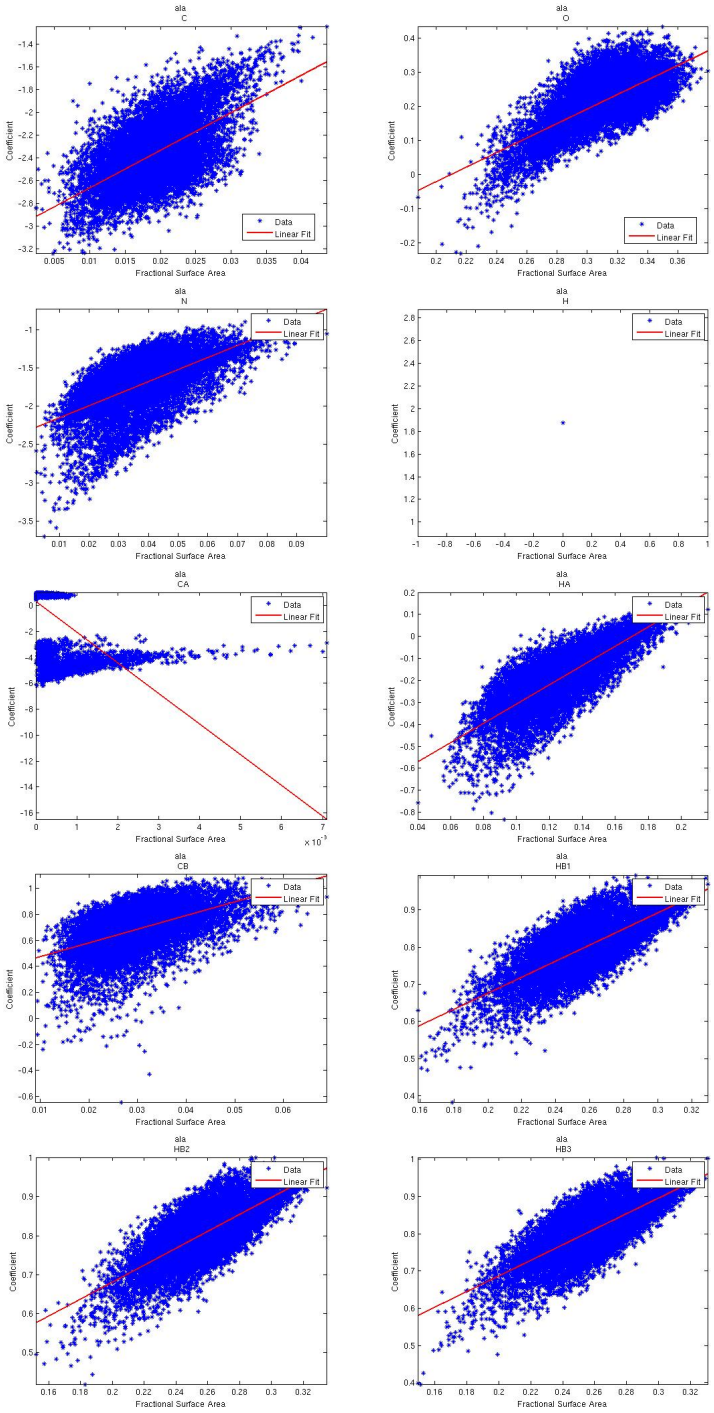


Figure 7.1: Example of linear approximation of nonpolar SEA coefficients for the Alanine dipeptide trajectory. The H atom is interior, which gives zero surface area. The C-alpha carbon exhibits bimodal behavior for conformations with extremely small surface area ($< 10^{-3}$). The average coefficient was used in this case. All other atoms display a clear linear relationship between γ_i and A_i.

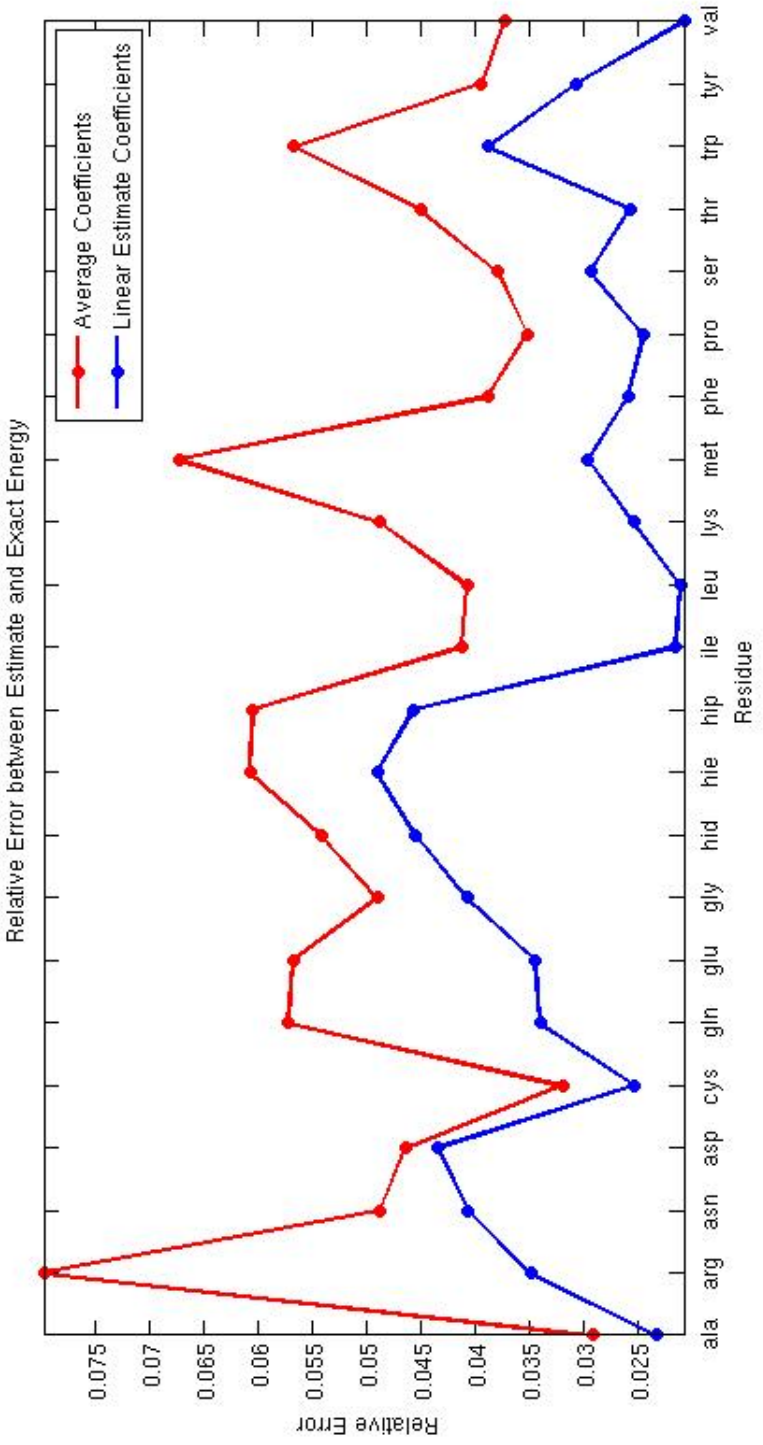


Figure 7.2: Relative errors due to interpolation of nonpolar SEA coefficients. As expected E_{lin} (blue) is less than E_{av} (red).

Part IV

Laguerre-Intersection Cells

Chapter 8

Laguerre-Intersection Cells

Motivation for and a preliminary discussion of Laguerre-Intersection cells were given in the Introduction (Chapter 1). For continuity in the text, they will be reiterated here. Laguerre tessellation methods have been used to find volumes of atoms, residues, and molecules which are useful in studying packing and deformation of proteins, and in cavity location. They may also be used to check predicted structures as, for example, residue volume may be considered an "intrinsic" property of amino acid type and is thus a predictable or checkable quantity [14]. Laguerre surfaces have been used as a quick and parameter free way to measure accessibility of atoms and residues for quantifying exposure of a molecule with solvent [14]. Residue contacts, which are important for protein structure and stability, have been studied using Laguerre surfaces. These quantities are useful for protein structure analysis and prediction, and in studying structure-function relationships. Preferential contacts between amino acid species and atom-atom contact frequencies have also been found[15], [28]. Differences in contacts between model and reference structures have been used as a scoring function and for benchmarking protein sturcure prediction methods [30]. Surfaces have also been used to characterize interactions between multiple proteins, or proteins and ligands, which has important applications in protein docking and formation of complexes [32], [29].

The Laguerre cell is the region which is closest to the generator of the cell (atom or residue center, for example) by the power distance. The power distance is an ap-

propriate quantity to use as larger areas are assigned to data points with larger weights (larger atoms or residues, for example). The Laguerre cell of a data point depends on its neighbors which poses a problem for systems that are typically found in a medium such as solvent but for which the surrounding medium is not explicitly known. For such data sets, Laguerre cells of points on the convex hull are unbounded and the cells of points otherwise in contact with the surrouding medium tend to be too large.

Researchers have addressed this difficulty in a variety of ways. Some consider only cells in the bulk of the protein which are not affected by the surrounding (unknown) environment. Others surround the structure by a layer of water or an artificial environment of spheres with size equal to the average amino acid size [19], [33]. However, this causes the problem to balloon. Soyer etal, bounded Laguerre cells by only considering Laguerre facet vertices from tetrahedra of a small enough size [34]. Still cells near the boundary tend to be elongated or have fewer facets than those in the bulk of the protein.

McConkey etal construct Laguerre-like cells by considering the surfaces of extended radical contact planes between neighboring atoms within a cutoff distance and the surface of an expanded sphere [28]. This poses problems since the cells of atoms in the bulk, which should not be affected by the cutoff distance, are affected. Furthermore, cells of small atoms completely disappear for cutoff distances as small as 1.4 A. The algorithm also cannot correctly handle cell volumes when the center of the generator of the cell lies outside its cell (See Section 8.3).

The proposed Laguerre-Intersection cell algorithm, which is presented in Chapter 8, considers the intersection of Laguerre cells with expanded atoms. The contact planes are the radical planes, which means that as the solvent weight is varied, Laguerre cells stay constant. This method simulates the environment better than using the extended radical plane. The algorithm also correctly handles the weighted case when the center of the generator of the cell is outside the cell. This has been been a challenge for previous methods. Chapter 9 finds optimal parameters for the Laguerre-Intersection method. One of the results of this analysis applied to a set of molecular trajectories of the essential amino acids in explicit water, was a stratification of the amino acids

according to increasing order of Laguerre parameter (radius or weight) showing a rough correspondence to commonly used "hydrophobicity scales" [40], [37].

8.1 Laguerre Tessellation

8.1.1 Laguerre and Laguerre Intersection Cells

Consider a molecule represented by a set of spheres or atoms $\mathcal{A} \subset \mathbb{R}^3 \times \mathbb{R}$. For $p_i \in \mathcal{A}$ we write $p_i = (p_i', w_i)$ where p_i' is the center of the atom and $w_i = r_i^2 = p_i''$ is the weight or squared radius of the atom. The Laguerre cell, L_i, of atom i is defined to be

$$L_i = \{x' \in \mathbb{R}^3 \quad : \quad |p_i' - x'|^2 - w_i \quad \leq \quad |p_j' - x'|^2 - w_j \quad \text{for all} \quad p_j \in \mathcal{A}\} \tag{8.1.1}$$

and is a convex polytope (Figure 2.1).

The quantity $|p_i' - x'|^2 - w_i$ is called the power distance (Figure 2.2a) between p_i and x' and is written as

$$\pi(p_i, x') = |p_i' - x'|^2 - w_i. \tag{8.1.2}$$

A Laguerre cell is the set of points whose nearest neighbor by the power distance is p_i, and each Laguerre facet lies on the plane which is equi-powerdistant between two points and which is called the radical plane (Figure 2.2b). The weights of the atoms in $\mathcal{A}$ may be increased or decreased by a certain amount w which we call the solvent weight. In this section, we call this modified set $\mathcal{A}(w)$ where $p_i(w) \in \mathcal{A}(w)$ has $p_i'(w) = p_i'$ and $w_i(w) = w_i + w$. The Laguerre cell of $p_i(w)$ is defined as

$$L_i(w) = \{x' \in \mathbb{R}^3 : |p_i(w) - x'|^2 - w_i(w) \leq |p_j(w) - x'|^2 - w_j(w) \quad \forall p_j(w) \in \mathcal{A}(w)\}. \tag{8.1.3}$$

Since $L_i(w) = L_i(0)$ for all w we simply write the Laguerre cell of atom i as L_i (Figure 2.8).

The Laguerre diagram, $\mathcal{L}(\mathcal{A})$, of $\mathcal{A}$ is the collection of all Laguerre cells and their faces which we call Laguerre facets, segments, and nodes. A Laguerre facet is the

intersection of two Laguerre cells and is a subset of the plane which is equi-powerdistant from the generator of the two cells. A Laguerre segment is the intersection of at least three Laguerre cells, and a Laguerre node is the intersection of at least four Laguerre cells.

For $p_i \in \mathcal{A}$ define

$$B_i(w) = \{x \in \mathbb{R}^3 : |p_i' - x|^2 - w_i(w) \leq 0\}. \tag{8.1.4}$$

The space filling model of $\mathcal{A}$ with solvent weight w is defined as

$$\mathcal{B}_w = \bigcup B_i(w) \tag{8.1.5}$$

We also define the following quantities:

- LV_i: Volume of the Laguerre cell of atom i.
- LS_i: Surface area of the Laguerre facets of atom i.

A residue is the collection of atoms in a single amino acid unit in the protein. The Laguerre volume of residue j is the sum of Laguerre volumes of atoms in residue j. The interresidue Laguerre surface area between residues j and k is the sum of areas of Laguerre facets which lie between atoms in residue j and atoms in residue k.

8.1.2 Laguerre Diagram and Regular Tetrahedrization

Recall that the regular tetrahedrization is dual to the Laguerre diagram $\mathcal{L}(\mathcal{A})$. There is a one to one correspondence between the $(3-k)$-faces in $\mathcal{L}(\mathcal{A})$ and the k-simplices in $\mathcal{T}(\mathcal{A})$. Each node in $\mathcal{T}(\mathcal{A})$ corresponds to a tetrahedron in $\mathcal{T}(\mathcal{A})$ whose vertices are equipowerdistant to the node, which we call the characteristic point of the tetrahedron. Each segment in $\mathcal{L}(\mathcal{A})$ corresponds to a triangle in $\mathcal{T}(\mathcal{A})$ whose vertices are equipowerdistant to the power segment. For each facet in $\mathcal{L}(\mathcal{A})$ there is an edge in $\mathcal{T}(\mathcal{A})$ whose vertices are equipowerdistant to the power facet. Each cell in $\mathcal{L}(\mathcal{A})$ corresponds to a point in $\mathcal{T}(\mathcal{A})$, namely the generator of the cell (Figure 2.4).

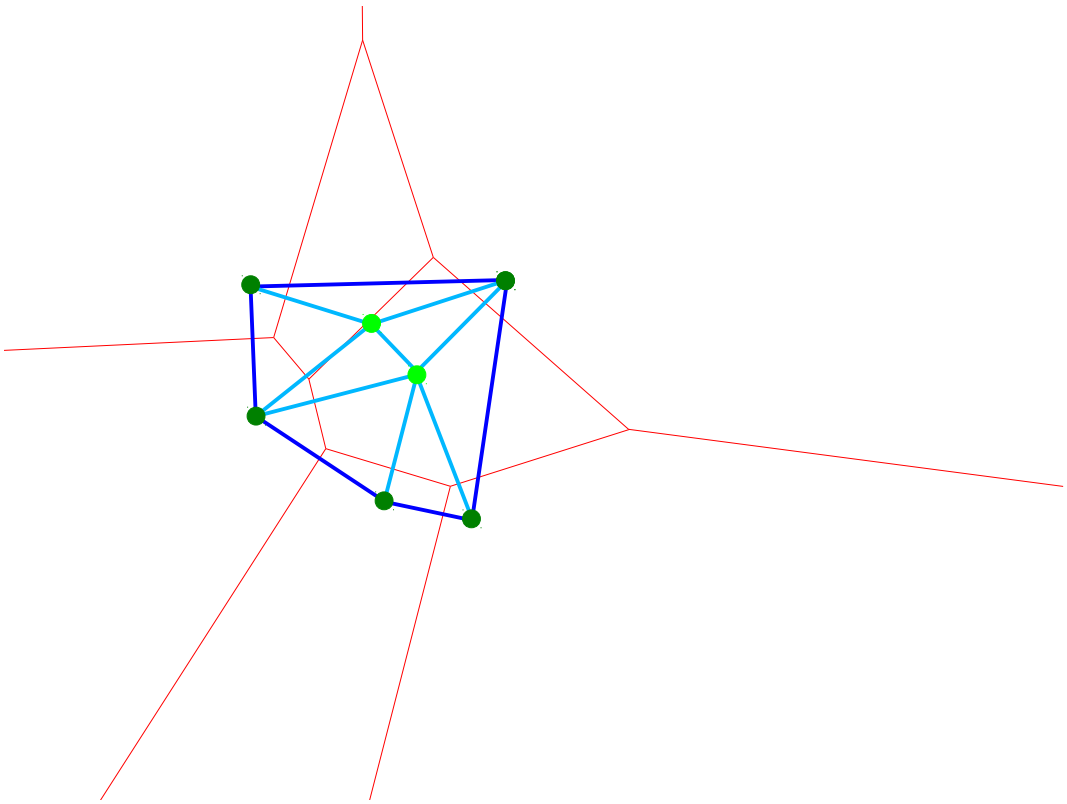


Figure 8.1: Exterior points (edges) shown in dark green (blue), interior points (edges) shown in light green (blue). Red represents boundaries of Laguerre cells. Note that the cells of interior (exterior) points are bounded (unbounded), and facets corresponding to interior (exterior) edges are bounded (unbounded).

We call a point whose center is on the convex hull of $\mathcal{T}(\mathcal{A})$ exterior to the tetrahedrization. Otherwise a point is called interior to $\mathcal{T}(\mathcal{A})$. An edge is called exterior if both of its vertices are exterior to $\mathcal{T}(\mathcal{A})$. An edge is called interior if at least one of its vertices is interior. Note that the Laguerre cell of an exterior point is unbounded whereas the Laguerre cells of interior points are bounded. The Laguerre facets corresponding to exterior edges are unbounded whereas Laguerre facets of interior edges are bounded (Figure 8.1).

8.2 Computation of Laguerre Surface Areas of Interior Facets

Interior Laguerre cells are bounded by Laguerre facets. Each of these facets correspond to an edge in the regular tetrahedrization. We call an edge e_{ij} if the vertices of that

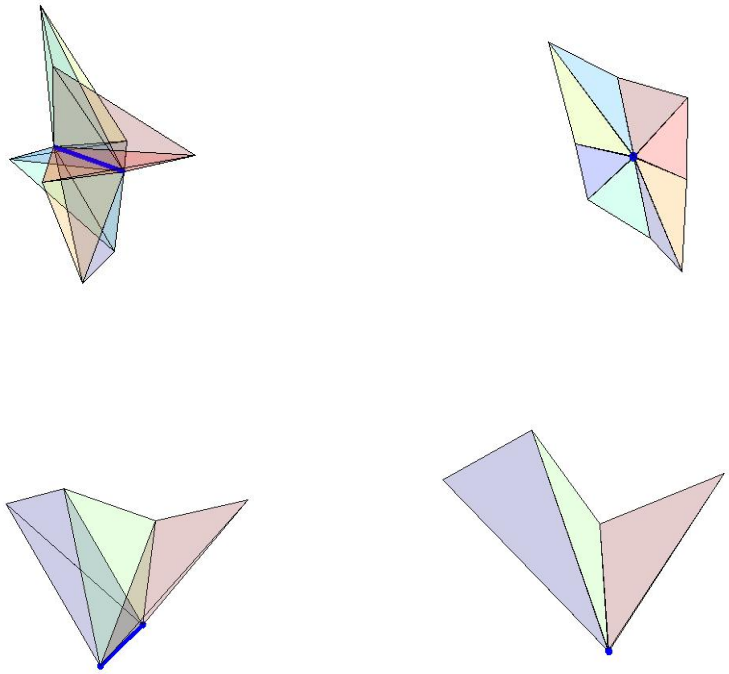


Figure 8.2: Top row: Side and perpendicular views of the complete tetraring of an interior edge (shown in dark blue). Bottom row: Side and perpendicular views of an incomplete tetraring of an exterior edge (shown in dark blue).

edge are p'_i and p'_j and call the Laguerre facet corresponding to that edge L_{ij}. The facets are convex polygons whose vertices are nodes in the Laguerre diagram, namely the characteristic points of the tetrahedra which surround that given edge. If a Laguerre cell has n facets, the volume may be divided into n pieces, one for each of its facets. The volume of the Laguerre cell of i corresponding to the edge e_{ij} is written $LV_{ij}^{(i)}$ and the volume of the Laguerre cell of j corresponding to the edge is written $LV_{ij}^{(j)}$.

We call the ordered list of tetrahedra which surround a given edge, i.e. the ordered list of tetrahedra in the edge's star, a *tetrahedra ring* or *tetraring*. A tetraring is called complete if the ring closes, otherwise it is called incomplete. Note that the tetraring of an interior edge is complete, whereas the tetraring of an exterior edge is incomplete (Figure 8.2).

In this section, we assume that the protein is surrounded by a layer of solvent. This means that all nonsolvent atoms are interior to the regular tetrahedrization as well as all edges which contain a nonsolvent atom as a vertex. We call these "nonsolvent edges".

The computation of the Laguerre volumes and surfaces of each nonsolvent atom proceeds as follows:

- Regular tetrahedrization of all atoms is computed
- Characteristic points of tetrahedra are found
- For each nonsolvent edge
    1. Surface area of the corresponding Laguerre facet is computed and assigned to appropriate atoms
    2. Using the calculated area, corresponding Laguerre volumes are found and assigned to appropriate atoms
    3. Interresidual Laguerre surface areas are assigned
- Laguerre residue volumes are summed

8.2.1 Surface Areas

When considering edge e_{ij}, we compute the surface area, LS_{ij}, of the facet, L_{ij}, between atoms i and j. The ordered vertices in the Laguerre facet are the characteristic points of the tetrahedra in the edge's tetraring. Note that the edge does not always intersect its Laguerre facet, but it is always perpendicular to it (Figure 8.3).

Next we define the characteristic point of an edge. The characteristic point, $x_{ij} = (x'_{ij}, x''_{ij})$ of an edge, e_{ij} satisfies

$$\pi(p_i, x'_{ij}) = \pi(p_j, x'_{ij}) \tag{8.2.1}$$

with x''_{ij} minimal. The center of this point lies on the intersection of the line containing p'_i and p'_j and the plane of points which are equipowerdistant to p_i and p_j (Figure 8.7).

The surface area of a Laguerre facet may be computed in one of two ways. The unsigned method computes the surface area as the sum of the areas of the triangles as shown in Figure 8.4 and the following pseudocode.

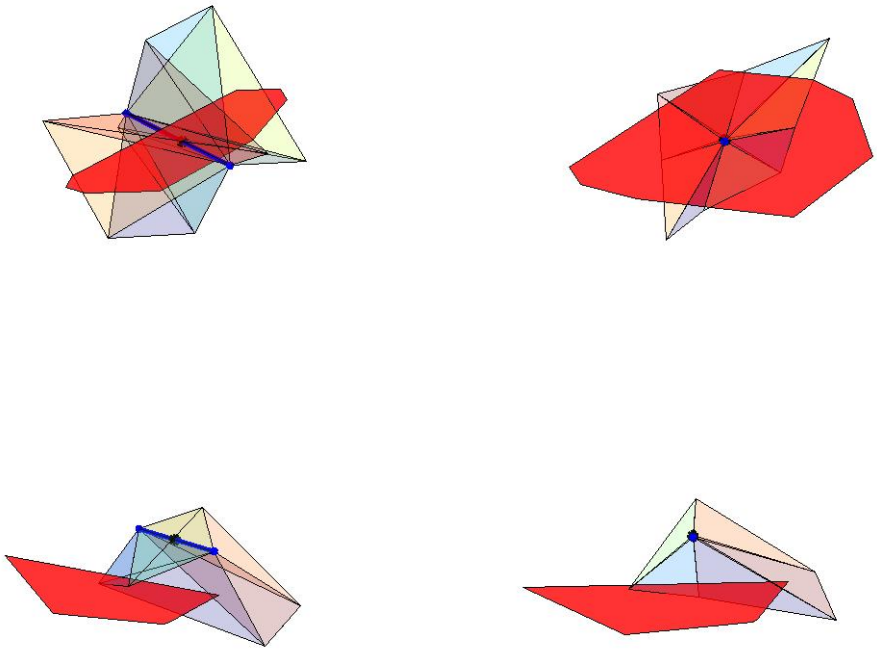


Figure 8.3: Laguerre facet of edge represented in red. The vertices of the facet are the characteristic points of the tetrahedra in the edge's tetraring. Top row: Two views of a facet which is intersected by its corresponding edge. Bottom row: Two views of a facet which is not intersected by its corresponding edge.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!COMPUTE AREA OF LAGUERRE FACET: UNSIGNED METHOD
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!surf     =   surface area of the Laguerre facet
!num_ring   =   number of nodes in facet
!nodes      =   num_ring x 3 matrix
!               contains coords of ordered Laguerre nodes
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

surf=0
a=nodes(2,:)-nodes(1,:)
do i=3,num_ring
   b=nodes(i,:)-nodes(1,:)
   axb=cross_product(a,b)
   area_tri=.5*sqrt(dot_product(axb,axb))
   surf=surf+area_tri
   a=b
end do
```

Signed areas and the characteristic point of the edge may be used to compute the surface area as well. If x'_{ij} is contained in the facet, when traversing its Laguerre nodes, all triangles shown have the same orientation (Figure 8.5). However, when the characteristic point of the edge does not lie in the facet, the triangles do not have the same orientation (Figure 8.6). The sum of these signed areas, as computed in the pseudocode below, gives the desired quantity.

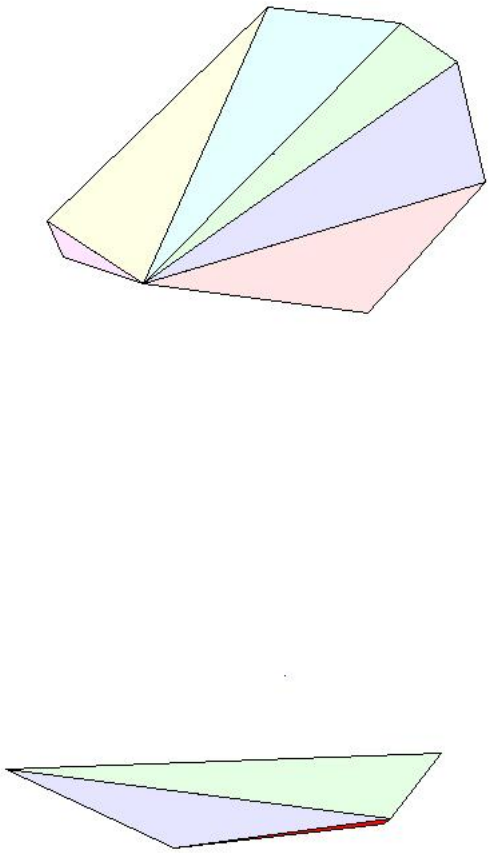


Figure 8.4: Unsigned subdivision method for the two facets shown in Figure 8.3.

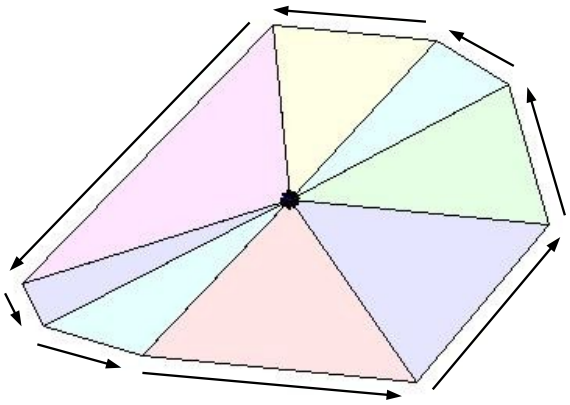


Figure 8.5: Characteristic point (black vertex) of edge is contained in facet and all triangles have the same orientation

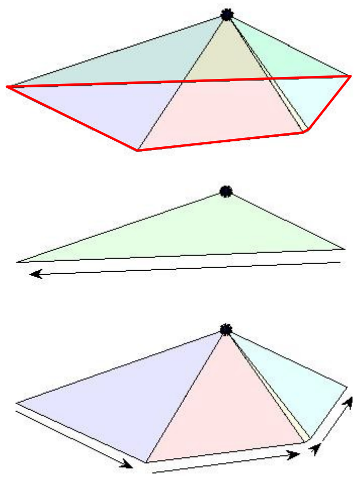


Figure 8.6: Characteristic point (black vertex) of edge is not contained in facet (bounded by red). Triangles have different orientation, and signed areas give desired quantity.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!COMPUTE AREA OF LAGUERRE FACET: SIGNED METHOD
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!surf      =   surface area of the Laguerre facet
!num_ring    =   number of nodes in facet
!nodes       =   num_ring x 3 matrix
!                contains coords of ordered Laguerre nodes
!x      =   center of characteristic point of edge
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

temp_vec=0
a=nodes(1,:)-x
do i=2,num_ring
   b=nodes(i,:)-x
   temp_vec=temp_vec+cross_product(a,b)
   a=b
end do

b=nodes(1,:)-x
temp_vec=temp_vec+cross_product(a,b)

surf=.5*sqrt(dot_product(temp_vec,temp_vec))
```

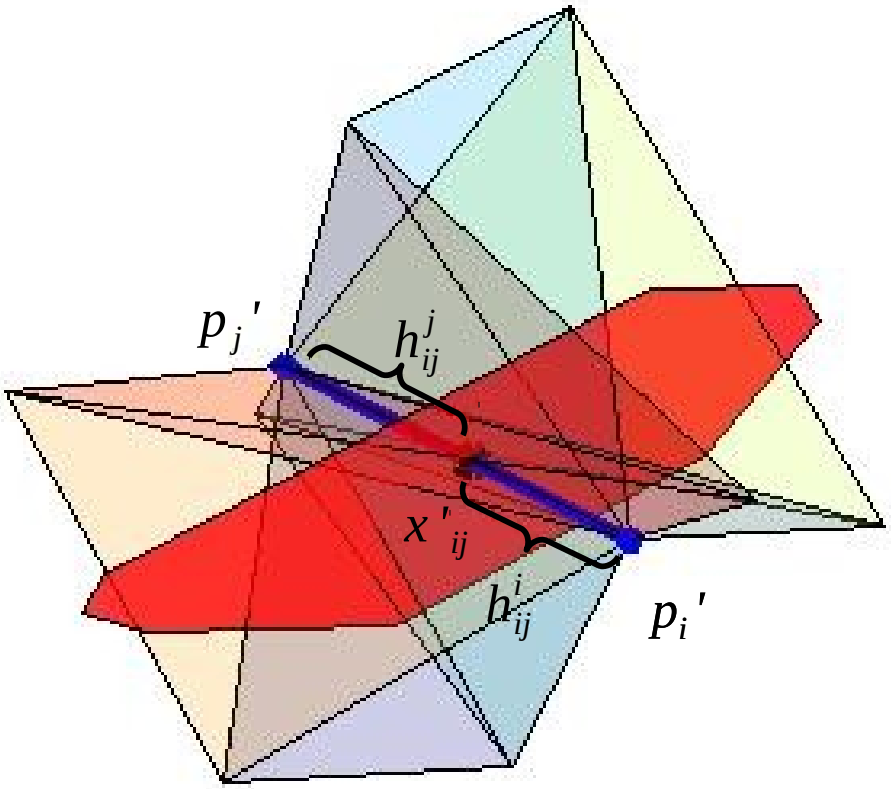


Figure 8.7: Heights of pyramidal volumes

8.2.2 Volumes

The Laguerre volume contributions, $LV_{ij}^{(i)}$ and $LV_{ij}^{(j)}$, are pyramids with base L_{ij}, and heights h_{ij}^i and h_{ij}^j respectively (Figure 8.7), with

$$h_{ij}^i = |x'_{ij} - p'_i| \tag{8.2.2}$$

$$h_{ij}^j = |x'_{ij} - p'_j|. \tag{8.2.3}$$

We can write

$$x'_{ij} = p'_j + t(p'_i - p'_j). \tag{8.2.4}$$

Thus the position of the plane containing the edge's facet with respect to p'_i and p'_j is

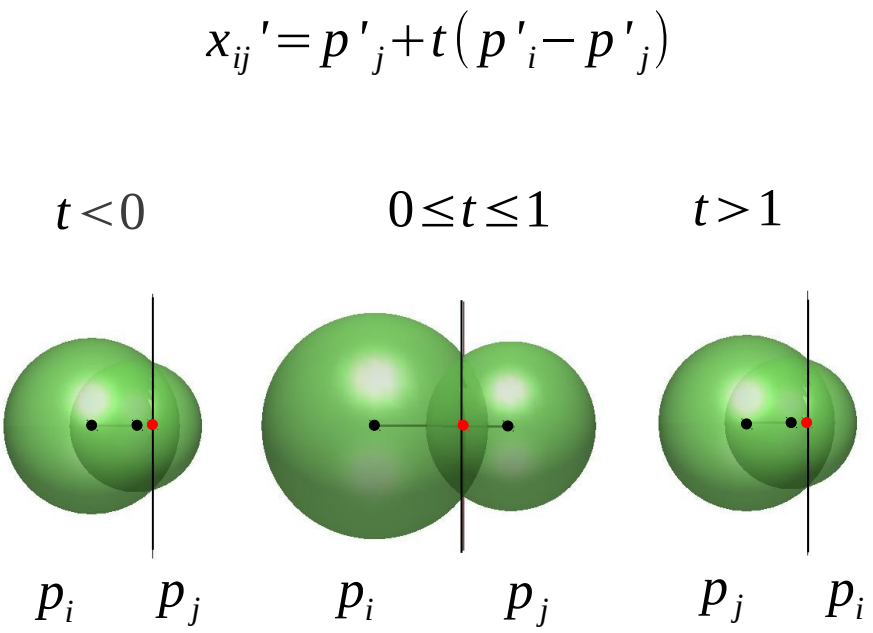


Figure 8.8: The value of t is determined by the relationship of the equi-powerdistant plane with respect to the two edge points and indicates if the center of the generator of the cell is interior or exterior to its cell

encoded in the value of t (Figure 8.8). This gives

$$LV_{ij}^{(i)} = sign(1-t)\frac{h_{ij}^{i} LS_{ij}}{3} \tag{8.2.5}$$

$$LV_{ij}^{(j)} = sign(t)\frac{h_{ij}^{j} LS_{ij}}{3} \tag{8.2.6}$$

$$\tag{8.2.7}$$

The term $LV_{ij}^{(i)}$ is positive if the corresponding pyramid lies in L_i and negative otherwise (Figures 8.9, 8.10).

.

8.3 Removing Solvent

We would like to compute Laguerre-like volumes and surfaces of a molecule that is not surrounded by solvent, so that these quantities are as close as possible to the Laguerre

Figure 8.9: Left: Laguerre cell that contains generator's center. Right: Pyramidal decomposition of Laguerre cell. All volumes are positive.

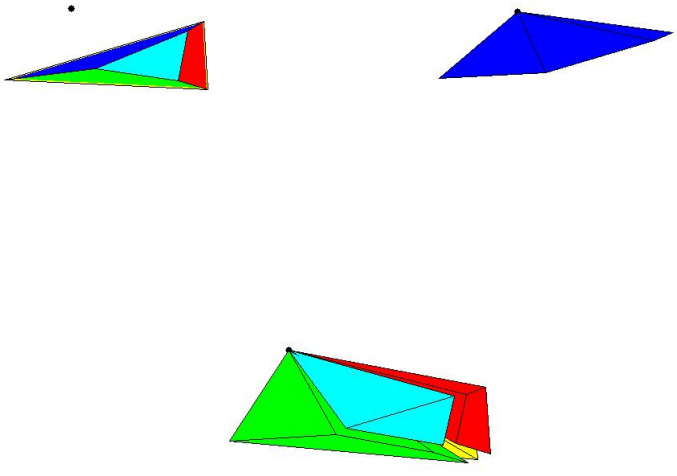


Figure 8.10: Left: Laguerre cell that does not contain generator's center (black vertex). Center: Negative pyramidal volume. Right: Positive pyramidal volumes.

volumes and surfaces that are found when the molecule is in its typical environment. In doing this, we come across two problems:

- The Laguerre cells are unbounded for certain atoms on the convex hull of $\mathcal{A}'$.
- The Laguerre cells of atoms that would otherwise be in contact with solvent are too large

This may be overcome by using Laguerre intersection cells, $LI(w)$. The Laguerre intersection cell of an atom is the intersection of the Laguerre cell of that atom with the expanded atom.

$$LI_i(w) = L_i \bigcap B_i(w). \tag{8.3.1}$$

Exterior cells are bounded in a realistic way, and the cells of atoms which would otherwise be in contact with solvent are shrunk to a more appropriate size (Figure 8.11). Laguerre cells are constant as a function of w, which means this parameter can be tuned to generate appropriate exterior Laguerre intersection cells while interior cells remain unchanged (Figure 8.12).

Let $LIS_i(w)$ and $LIV_i(w)$, be the surface area and volume of $LI_i(w)$. Like the individual atomic surface area and molecular volume, LIS and LIV can be split into contributions from simplices in the alpha complex which is a subset of the Delaunay tetrahedrization.

Define $\mathcal{C}(w)$ to be the alpha complex of $\mathcal{A}(w)$ with $\partial\mathcal{C}(w)$ the set of simplices in the boundary of $\mathcal{C}(w)$. For $T \subset \mathcal{A}(w)$, σ_T represents the simplex which is the convex hull of the centers of points in T. As we are working in three dimensions we only consider $|T| \leq 4$, where $|T|$ is the number of elements in T. Recall that $e_{ij} = \sigma_T$ for $T = \{p_i, p_j\}$. We also represent vertices and triangles as v_i and t_{ijk} in a similar manner.

The terms $LIS_T^{(i)}(w)$ and $LIV_T^{(i)}(w)$ are the surface area and volume contributions of the simplex σ_T to $LIV_i(w)$ and $LIS_i(w)$. We can also write $LI_T^{(i)}(w) = LI_i^{(i)}(w)$ for $T = \{p_i(w)\}$, $LI_T^{(i)}(w) = LI_{ij}^{(i)}(w)$ for $T = \{p_i(w), p_j(w)\}$, etc., and likewise for the volumes and surface areas (See Figure 8.13). The terms $P_T^{(i)}$ and F_T for $|T| = 2$ are pyramidal volumes and facet surface areas that will be discussed shortly.

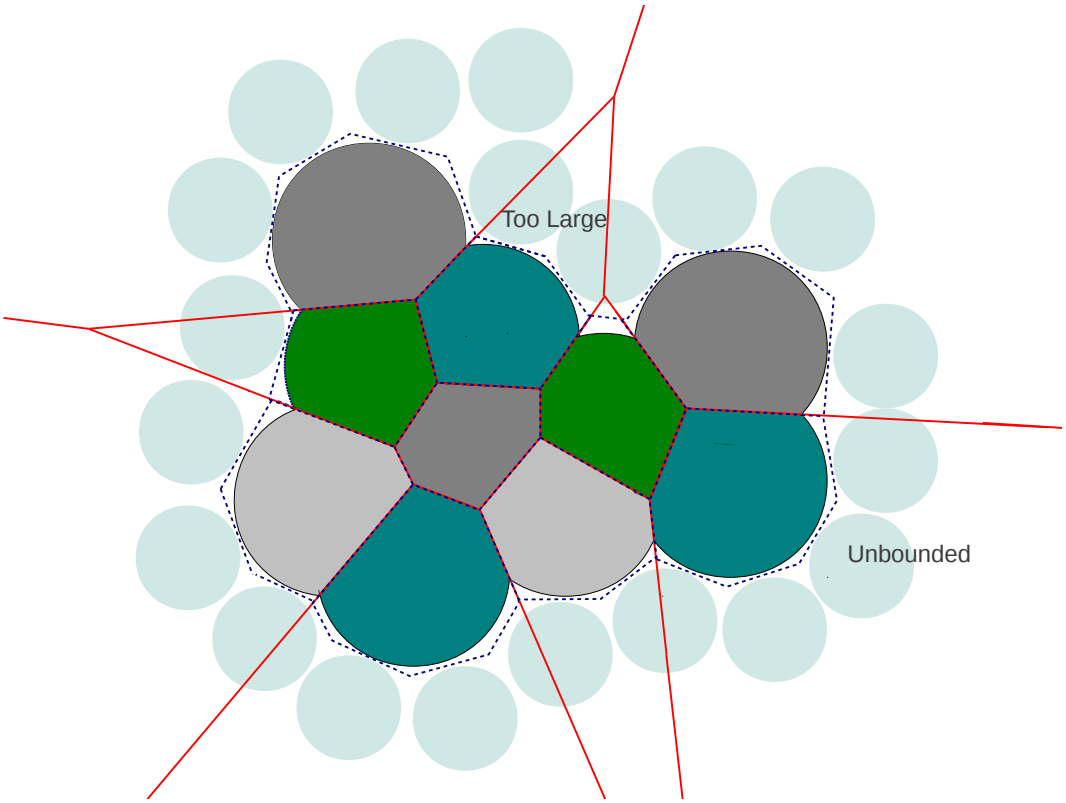


Figure 8.11: Example of Laguerre intersection cells (shown in grays and greens). Solvent is represented by light blue spheres, boundary of Laguerre cells of molecule (minus solvent) are shown in red, and boundary of Laguerre cells of molecule in solvent is represented by the dotted line.

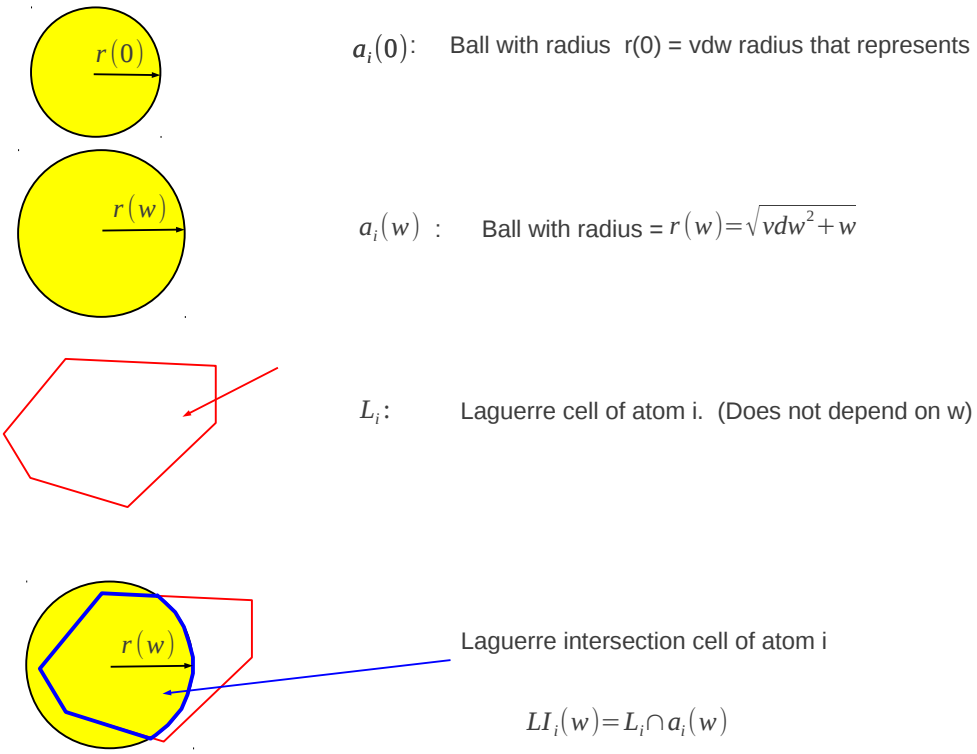


Figure 8.12: Laguerre cell does not depend on w while the Laguerre-Intersection cell does depend on w.

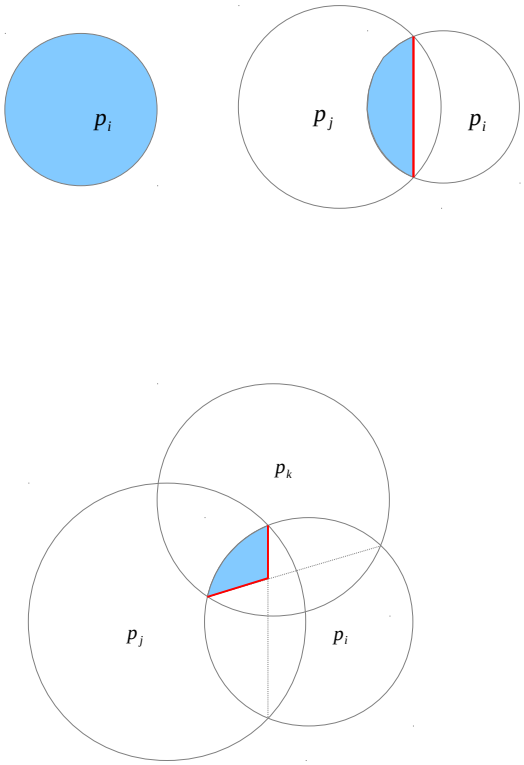


Figure 8.13: Laguerre-Intersection cell volumes (blue), $LIV_i^{(i)}$, $LIV_{ij}^{(i)}$, $LIV_{ijk}^{(i)}$, and surfaces (red) LIS_{ij}, and LIS_{ijk}. to atom i from $\partial\mathcal{C}(w)$ simplices.

We have

$$\begin{aligned} LIV_i(w) &= \sum_{\sigma_T \in \partial\mathcal{C}(w)} (-1)^{k+1} c_T LIV_T^{(i)}(w) \qquad |T| = k \\ &\quad + \sum_{\sigma_T \in \mathcal{C}(w)} P_T^{(i)}(w) \qquad |T| = 2 \end{aligned} \tag{8.3.2}$$

and

$$\begin{aligned} LIS_i(w) &= S_i(w) + \sum_{\sigma_T \in \partial\mathcal{C}(w), k>1} (-1)^k c_T LIS_T^{(i)}(w) \qquad |T| = k \\ &\quad + \sum_{\sigma_T \in \mathcal{C}(w)} F_T(w) \qquad |T| = 2 \end{aligned} \tag{8.3.3}$$

where $S_i(w)$ is the accessible surface area of atom i with solvent radius w, and c_Ts are coefficients that will be defined shortly.

Since

$$S_i(w) = \sum_{\sigma_T \in \partial\mathcal{C}(w)} (-1)^{k+1} c_T S_T^{(i)}(w), \qquad |T| = k \tag{8.3.4}$$

where $S_T^{(i)}(w)$ is the contribution of σ_T to $S_i(w)$, we can write

$$\begin{aligned} LIS_i(w) &= c_i S_i^{(i)}(w) \\ &\quad + \sum_{\sigma_T \in \partial\mathcal{C}(w), k>1} (-1)^k c_T (LIS_T^{(i)}(w) - S_T^{(i)}(w)) \qquad |T| = k \\ &\quad + \sum_{\sigma_T \in \mathcal{C}(w)} F_T \qquad |T| = 2 \end{aligned} \tag{8.3.5}$$

The coefficients, c_T, are given for the following simplices,

- $|T| = 1$, i.e. a vertex v_i: $c_T = \Omega_T = \Omega_i$ is the fraction of a_i outside the tetrahedra in the alpha complex. That is Ω_T is the normalized outer solid angle subtended by the union of tetrahedra in $\mathcal{C}(w)$ which contain p_i' (Figure 4.2).

- $|T| = 2$, i.e. an edge e_{ij}: $c_T = \Phi_T = \Phi_{ij}$ is normalized outer dihedral angle of the union of tetrahedra in $\mathcal{C}(w)$ which contain the edge e_{ij} (Figure 4.2).

- $|T| = 3$, i.e. a triangle t_{ijk}: c_T is 1 if the triangle is singular and $\frac{1}{2}$ if the triangle is regular. In other words, c_T is the fraction of S^T that is outside the union of tetrahedra in the alpha complex.

8.3.1 Equations

Vertices

The Laguerre volume and surface contributions from a vertex, v_i, are the volume and surface area of the corresponding weighted point $p_i(w)$. That is

$$S_i^{(i)}(w) = 4\pi r_i^2(w) \tag{8.3.6}$$

$$LIV_i^{(i)}(w) = \frac{4}{3}\pi r_i^3(w). \tag{8.3.7}$$

The solid angle calculations are given in previously.

This gives

$$c_i S_i^{(i)}(w) = \Omega_i r_i^2(w) \tag{8.3.8}$$

$$c_i LIV_i^{(i)}(w) = \frac{\Omega_i}{3} r_i^3(w). \tag{8.3.9}$$

Edges

The formulas for the volume and surface area of the intersection of the two balls, p_i and p_j, are

$$LIV_T^{(i)}(w) = \pi h_i^2 \left(r_i(w) - \frac{h_i}{3} \right) \tag{8.3.10}$$

$$LIV_T^{(j)}(w) = \pi h_j^2 \left(r_j(w) - \frac{h_j}{3} \right) \tag{8.3.11}$$

$$S_T^{(i)}(w) = 2\pi r_i(w) h_i \tag{8.3.12}$$

$$S_T^{(j)}(w) = 2\pi r_j(w) h_j \tag{8.3.13}$$

$$LIS_T^{(i)}(w) = \pi h_i (2r_i(w) - h_i) \tag{8.3.14}$$

$$LIS_T^{(j)}(w) = LS_T^{(i)}(w) \tag{8.3.15}$$

and

$$c_T LIV_T^{(i)}(w) = \frac{\Phi_{ij}}{2} h_i^2 \left(r_i(w) - \frac{h_i}{3} \right) \tag{8.3.16}$$

$$c_T LIV_T^{(j)}(w) = \frac{\Phi_{ij}}{2} h_j^2 \left(r_j(w) - \frac{h_j}{3} \right) \tag{8.3.17}$$

$$c_T S_T^{(i)}(w) = \Phi_{ij} r_i(w) h_i \tag{8.3.18}$$

$$c_T S_T^{(j)}(w) = \Phi_{ij} r_j(w) h_j \tag{8.3.19}$$

$$c_T LIS_T^{(i)}(w) = \frac{\Phi_{ij}}{2} h_i (2r_i(w) - h_i) \tag{8.3.20}$$

$$c_T LIS_T^{(j)}(w) = c_T LIS_T^{(i)}(w). \tag{8.3.21}$$

The computation of the dihedral angles is outlined previously.

Triangles

Let $x_T = x_{ijk} = (x'_{ijk}, x''_{ijk})$ be the characterstic point of the triangle t_{ijk}. Let p be one of the two points of intersection on the spheres $p_i(w)$, $p_j(w)$, and $p_k(w)$. We can write $p = x'_{ijk} + \mathbf{n}x''_{ijk}$ where $\mathbf{n}$ is the normal to the plane containing the triangle,

$$\mathbf{n} = \frac{(p'_j - p'_i) \times (p'_k - p'_i)}{|(p'_j - p'_i) \times (p'_k - p'_i)|}. \tag{8.3.22}$$

Let σ_{Tc} the tetrahedron defined by the centers of T and x_T. Then

$$\frac{1}{2}S_T^{(i)} = \phi_{ij}S_{ij}^{(i)} + \phi_{ik}S_{ik}^{(i)} - \omega_i S_i^{(i)} \tag{8.3.23}$$

$$\frac{1}{2}S_T^{(j)} = \phi_{jk}S_{jk}^{(j)} + \phi_{ji}S_{ji}^{(j)} - \omega_j S_j^{(j)} \tag{8.3.24}$$

$$\frac{1}{2}S_T^{(k)} = \phi_{ki}S_{ki}^{(k)} + \phi_{kj}S_{kj}^{(k)} - \omega_k S_k^{(k)} \tag{8.3.25}$$

where ϕ_{ij} is the fractional inner dihedral angle of σ_{Tc} along edge σ_{ij}, ω_i is the fractional (inner) solid angle of σ_{Tc} subtended from p'_i, etc.

Let x'_{ij}, x'_{ik}, x'_{jk} be the centers of the characteristic points of the triangle's edges. Then $\frac{1}{2}LS_{ijk}^{(i)}$ is the sum of the areas of the triangles given by x'_{ijk}, x'_{ij}, p, and x'_{ijk}, x'_{ik}, p. We have

$$\frac{1}{2}LIV_T^{(i)} = \widetilde{V}_i - \omega_i V_i^{(i)} + \phi_{ij}LIV_{ij}^{(i)} + \phi_{ik}LIV_{ik}^{(i)} \tag{8.3.26}$$

where $\widetilde{V}_i$ is the signed volume of the solid with vertices at p'_i, x_{ij}, x_{ijk}, x_{ik}, and p. More specifically, let

$$\mathbf{a} = x'_{ij} - p'_i \tag{8.3.27}$$

$$\mathbf{b} = x'_{ijk} - x'_{ij} \tag{8.3.28}$$

$$\mathbf{c} = x'_{ik} - x'_{ijk} \tag{8.3.29}$$

$$\mathbf{d} = p'_i - x'_{ik} \tag{8.3.30}$$

$$\tag{8.3.31}$$

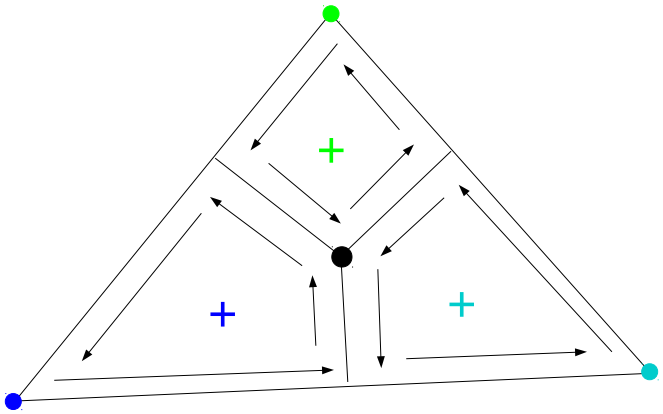


Figure 8.14: p'_i, p'_j, p'_k, and x'_{ijk} represented by dark blue, light blue, green, and black dots, respectively. The point p is out of the page. The black arrows represent positive contributions to the volumes $\widetilde{V}_i$, $\widetilde{V}_j$, and $\widetilde{V}_k$.

then

$$2\widetilde{V}_i = \frac{h}{3}\Big((\mathbf{a}\times\mathbf{b}) + (\mathbf{c}\times\mathbf{d})\Big)\cdot\mathbf{n} \tag{8.3.32}$$

Equivalent equations hold for $LIV_T^{(j)}$, $LIV_T^{(k)}$, $\widetilde{V}_j$, and $\widetilde{V}_k$. See Figures 8.14, 8.15, 8.16 for possible configurations.

Other Contributions

We get additional contributions, F_T and $P_T^{(i)}$, from all edges in $\mathcal{C}(w)$. There are two types of these edges:

- edges interior to $\mathcal{C}(w)$
- edges exterior to $\mathcal{C}(w)$.

Contributions from interior edges of $\mathcal{C}(w)$ are computed as in section 8.2. That is

$$P_{ij}^{(i)} = LV_{ij}^{(i)} \tag{8.3.33}$$

and

$$F_{ij} = LS_{ij}. \tag{8.3.34}$$

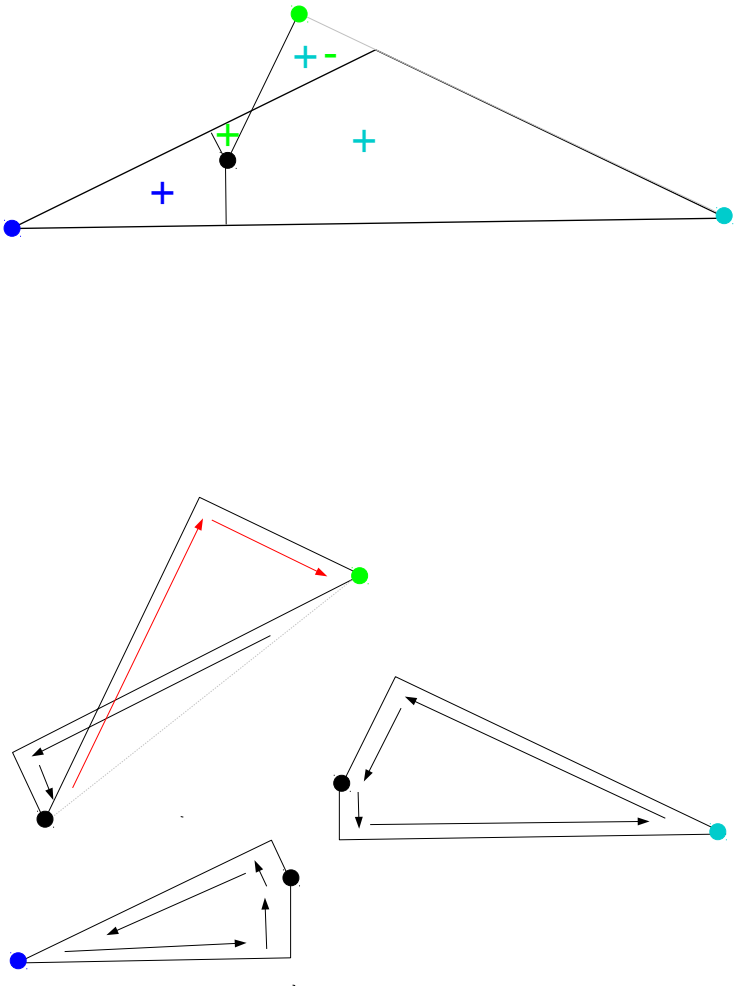


Figure 8.15: Above: Sample configuration. Below: Black arrows represent positive contributions to $\widetilde{V_i}$, $\widetilde{V_j}$, and $\widetilde{V_k}$, while red arrows represent negative contributions.

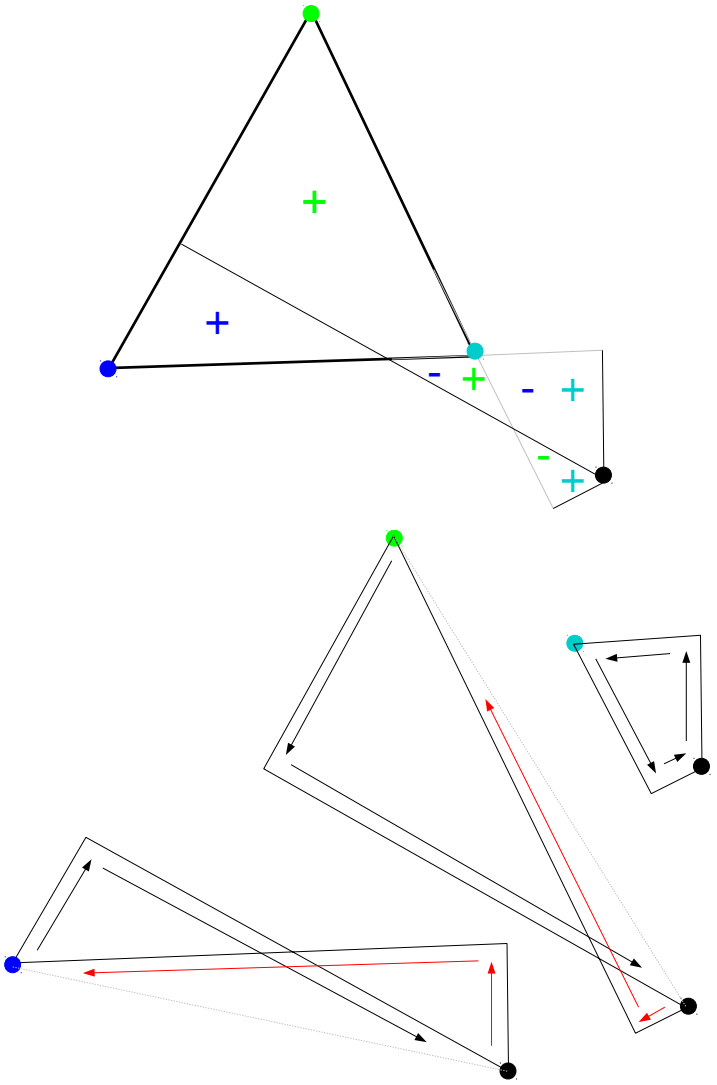


Figure 8.16: Above: Sample configuration. Below: Black arrows represent positive contributions to $\widetilde{V_i}$, $\widetilde{V_j}$, and $\widetilde{V_k}$, while red arrows represent negative contributions.

The terms $P_T^{(i)}$ and F_T are zero for all edges that are not part of at least one tetrahedron in $\mathcal{C}(w)$. To calculate the contribution from all other exterior $\partial\mathcal{C}(w)$ edges, we must know the characteristic points of the triangles on the boundary of the $\mathcal{C}(w)$ tetraring. The characteristic point of the boundary triangle may or may not be on the Laguerre facet, but it always indicates the direction of one of the facet's bounday segments (Figure 8.17). For each exterior edge, the counterclockwise list of Laguerre nodes corresponding to tetrahedra in the alpha complex is already known. The surface area is computed by using signed surface areas as shown in the following pseudocode. The volume contributions to LI_i and LI_j are found by multiplying the resulting surface by h_{ij}^i and h_{ij}^j, respectively.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!COMPUTE AREA OF LAGUERRE INTERSECTION FACET OF EXTERIOR EDGE
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!surf       =   area of the Laguerre intersection facet
!num_ring   =   number of nodes in Laguerre intersection facet
!nodes      =   num_ring x 3 matrix
!               contains coords of counterclockwise
!               Laguerre nodes of tetrahedra in alpha complex
!               with characteristic points of appropriate
!               boundary triangles inserted
!x          =   center of characteristic point of edge
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

surf=0
temp_vec=0
a=nodes(2,:)-x
do i=2,num_ring
   b=nodes(i,:)-x
   temp_vec=cross_product(b,a)
   if (dot_product(temp_vec,edge)>=0 ) then
      surf=surf+.5d0*sqrt(temp_vec)
   else
      surf=surf-.5d0*sqrt(temp_vec)
   end if
   a=b
end do

b=nodes(1,:)-x
temp_vec=cross_product(a,b)
```

```
if (dot_product(temp_vec,edge)>=0 ) then
   surf=surf+.5d0*sqrt(temp_vec)
else
   surf=surf-.5d0*sqrt(temp_vec)
end if
```

Note that the $\mathcal{C}(w)$ tetraring of an edge may not be connected. The pseudocode does not take this into account for simplicity reasons, while the actual implementation does.

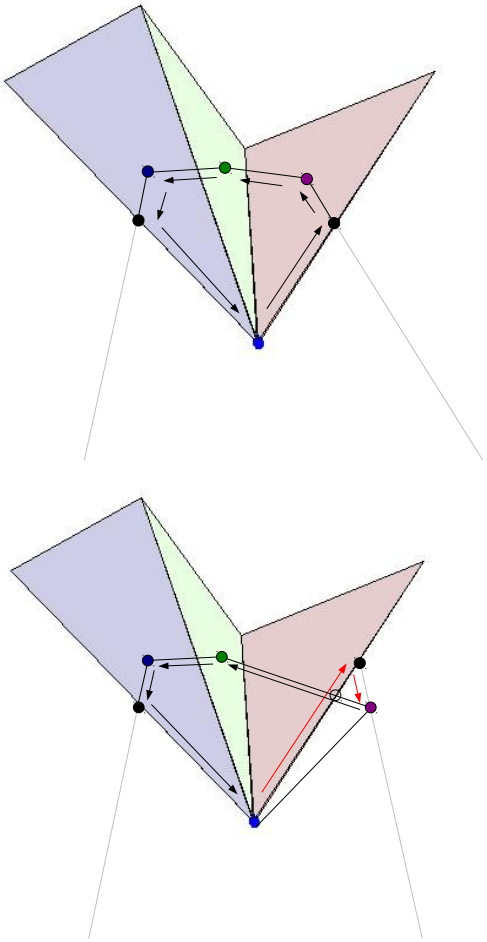


Figure 8.17: Two examples of additional contributions from edges which are regular and exterior with respect to $\mathcal{C}(w)$. The royal blue points are the characteristic points of edges which are exterior to $\mathcal{C}(w)$ and which points directly out of the paper. The triangles represent tetrahedra in the tetrahedra ring which are also in $\mathcal{C}(w)$. Dark blue, green, and purple dots represent the Laguerre nodes corresponding to the tetrahedra of the same color in the $\mathcal{C}(w)$ tetraring. The black points are characteristic points of the boundary triangles. In the first case, both triangle characteristic points lie in the Laguerre facet. In the second case, one of the triangle characteristic points is not in the facet but is still needed to determine a line segment (shown in gray). In the first case all Laguerre nodes lie in the union of tetrahedra and all corresponding surface contributions are positive. In the second case, a Laguerre node lies outside the union of tetrahedra in $\mathcal{C}(w)$. The triangular surface that is bounded by the right black, purple, and clear dots was originally assigned during contributions from the boundary triangle, and must be subtracted to obtain the correct area.

Chapter 9

Optimization of Solvent Parameter

We would like to compute Laguerre-like volumes and surfaces of an unsolvated molecule so that these quantities are as close as possible to the Laguerre volumes and surfaces that are found when the molecule is in its typical solvated environment.

To this end we equilibrated explicit water around various solute molecules and ran a molecular dynamics simulation for each. We recorded N conformations from each trajectory. Atomic and residual Laguerre volumes and surfaces as well as interresidual surfaces were calculated. For various solvent parameters, w_k and r_k,

$$\begin{aligned} w_k &= k \cdot dw \\ r_k &= k \cdot dr \end{aligned}$$

Laguerre-Intersection quantities were computed for each structure. Laguerre and Laguerre-Intersection quantities were compared and "optimal" solvent parameters were found. These will be defined shortly.

We call $LV(i,j)$ and $LS(i,j)$ the Laguerre volume and surface area, respectively, of residue j in structure number i. We call $LIVw(i,j,k)$ and $LIVr(i,j,k)$ the Laguerre-Intersection volumes of residue j in structure i for solvent values w_k and r_k, respectively. Similarly for $LISw(i,j,k)$ and $LISr(i,j,k)$.

We define the 1-norm error and the 2-norm error for $LIVw(k)$ as

$$E_1(LIVw(k)) = \frac{1}{N \cdot num\_res} \sum_{i=1}^{N} \sum_{j=1}^{num\_res} |LV(i,j) - LIVw(i,j,k)| \quad (9.0.1)$$

and

$$E_2(LIVw(k)) = \sqrt{\frac{1}{N \cdot num\_res} \sum_{i=1}^{N} \sum_{j=1}^{num\_res} (LV(i,j) - LIVw(i,j,k))^2} \quad (9.0.2)$$

respectively. In this chapter we consider both residual and atomic volume and surface areas. Similary definitions and equations hold for all other quantities.

The optimal solvent weight (radius) is the value w_k (r_k) that minimizes the error.

9.1 Buckminster Fullerene, Benzene, and Dipeptide Trajectories

9.1.1 Fullerene and Benzene

We begin by studying Buckminster Fullerene and Benzene trajectories. (See Figure 9.1). For each molecule we have $N = 1001$ structures. Both molecules are rigid in the sense that they have no torsional degrees of freedom, assuming constant bond lengths and constant bond angles. During the molecular dynamics simulation, bond lengths and angles are relatively stable and only fluctuate slightly around equilibrium. This means that Laguerre-Intersection volumes and surface areas are near-constant through the entire trajectory. (See Figures 9.2 and 9.3). Variations in Laguerre volumes and surfaces are mostly due to random fluctuations in the water which gives us an estimate for how close Laguerre quantities and Laguerre-Intersection quantities can be in the best case. This will give us a minimum "expected error" for trajectories of molecules with more degrees of freedom.

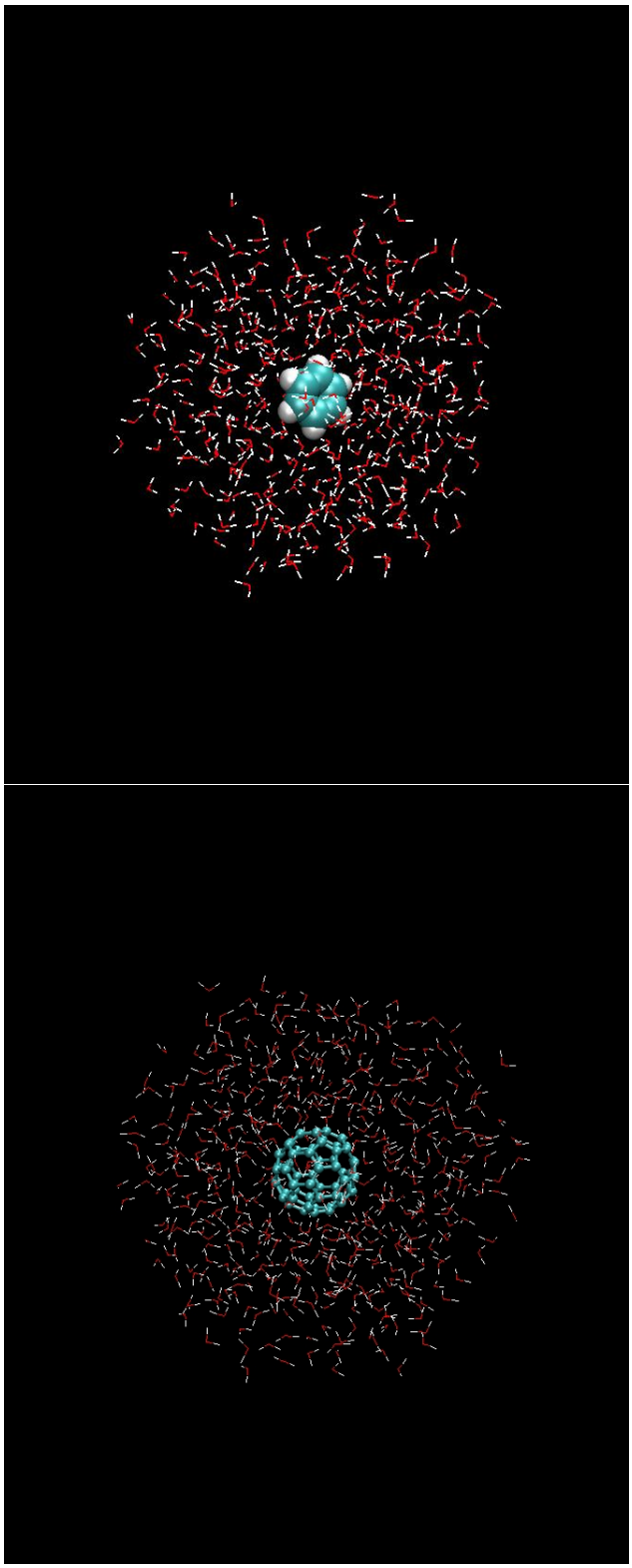


Figure 9.1: Sample conformations from the solvated Benzene (left) and Fullerene (right) trajectories. In this figure, Benzene is represented by van der Waals spheres whereas Fullerene is shown in the ball and stick representation to better illustrate the molecule's spherical structure. Waters are represented using the line model.

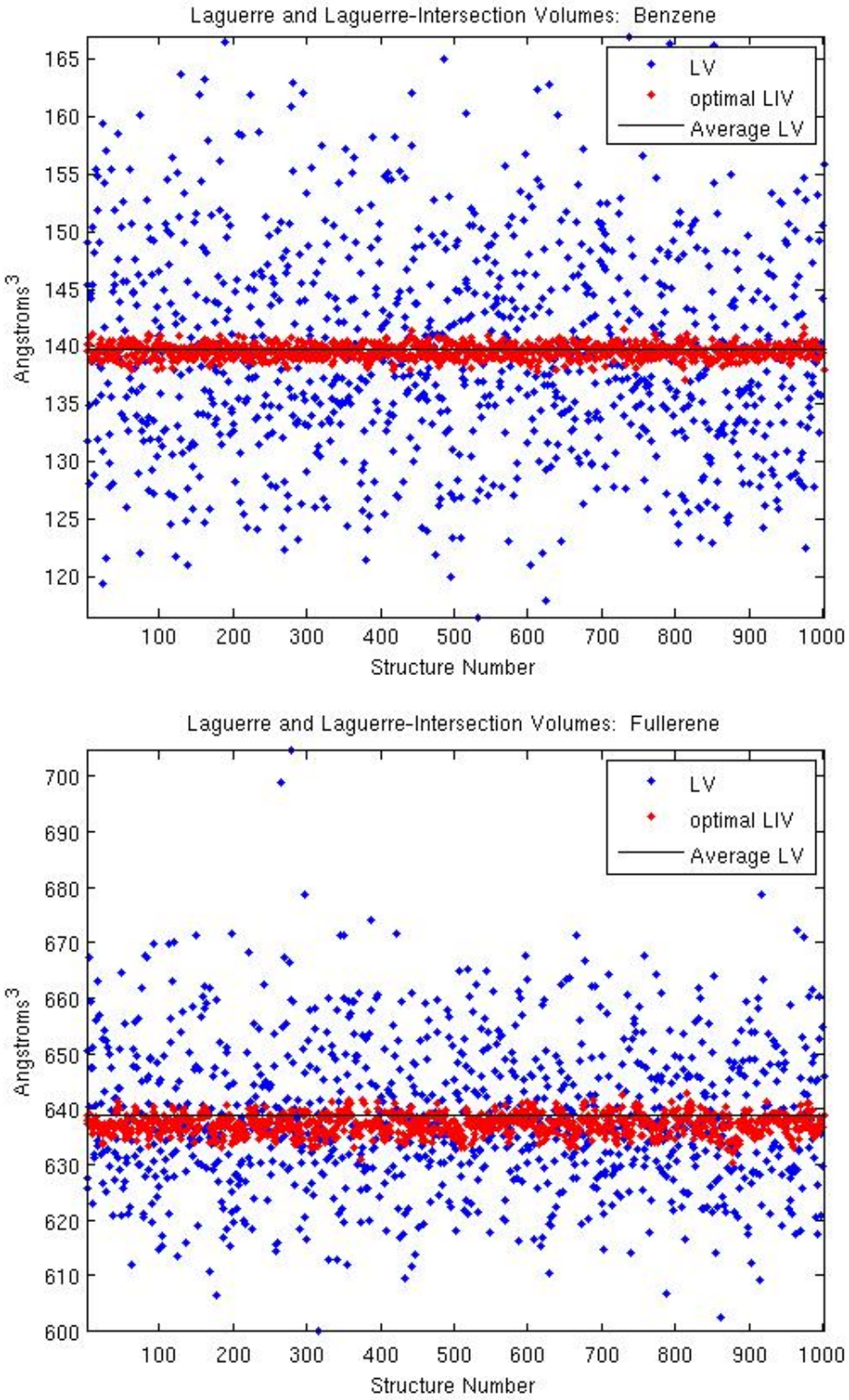


Figure 9.2: Laguerre volumes (blue) vary more than Laguerre-Intersection volumes (red) particularly for benzene and fullerene which are near-rigid. Laguerre-Intersection quantities are shown for the optimal solvent weight.

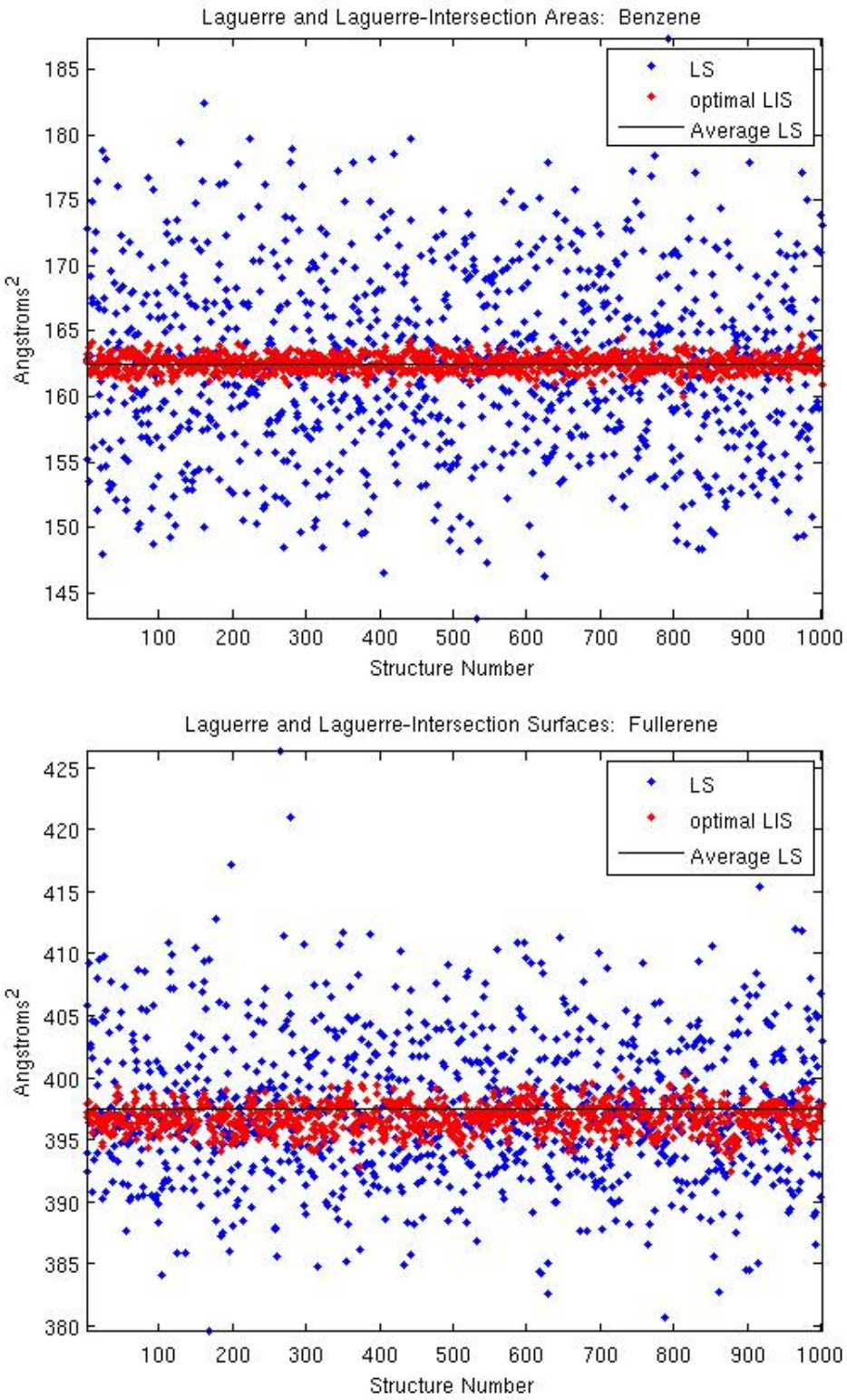


Figure 9.3: Laguerre surface areas (blue) vary more than Laguerre-Intersection surface areas (red) particularly for benzene and fullerene which are near-rigid. Laguerre-Intersection areas are shown for the optimal solvent weight.

The values of dw and dr were chosen so that

$$1.7^2 + (k\_max) \cdot dw = (1.7 + (k\_max) \cdot dr)^2. \tag{9.1.1}$$

That is, for an atom with van der Waals radius 1.7, which is the typical carbon radius, the atom's final total weight is the same regardless of the method used to expand the atom. This means that dw and dr are in a sense "equal in size".

Errors for $LIVw, r$ and $LISw, r$ for Benzene and Fullerene are shown in the appendix, Figures D.1 and D.2. The curves are smooth with distinct minima.

The following table shows the "relative" 1-norm and 2-norm errors which are the errors for the optimal solvent parameter divided by the average quantity over all conformations.

Laguerre quantity	r/w	BNZ Rel. E_1	BNZ Rel. E_2	FLN Rel. E_1	FLN Rel. E_2
LV_res	r	.0519	.0645	.0158	.0202
	w	.0518	.0645	.0158	.0203
LV_atom	r	.2236	.2885	.0772	.1001
	w	.1264	.1739	.0772	.1001
LS_atom	r	.1535	.1855	.0494	.0634
	w	.0855	.1108	.0495	.0633
SAS_res	r	.0340	.0417	.0114	.0145
	w	.0337	.0415	.0114	.0145

We see that we can expect an error of 1% to 6% for residual Laguerre volumes, 7% to 28% for atomic Laguerre volumes, 4% to 18% for atomic Laguerre surfaces, and 1% to 4% for exterior residual Laguerre surfaces.

We see that E_1 and E_2 are smaller for residual quantities than atomic quantities. We also see that $E(r) \approx E(w)$ for residual quantities while $E(r) > E(w)$ for atomic quantities. All errors are smaller for Fullerene, which has spherical symmetry, than for Benzene which has planar symmetry.

9.1.2 Dipeptides

The optimal solvent parameters and associated errors were calculated for dipeptide molecular dynamics trajectories with $N = 1001$. The errors for each dipeptide, Ben-

zene, and Fullerene are shown in Figures 9.4 and 9.5. We see that the errors for each dipeptide are close to the expected bounds.

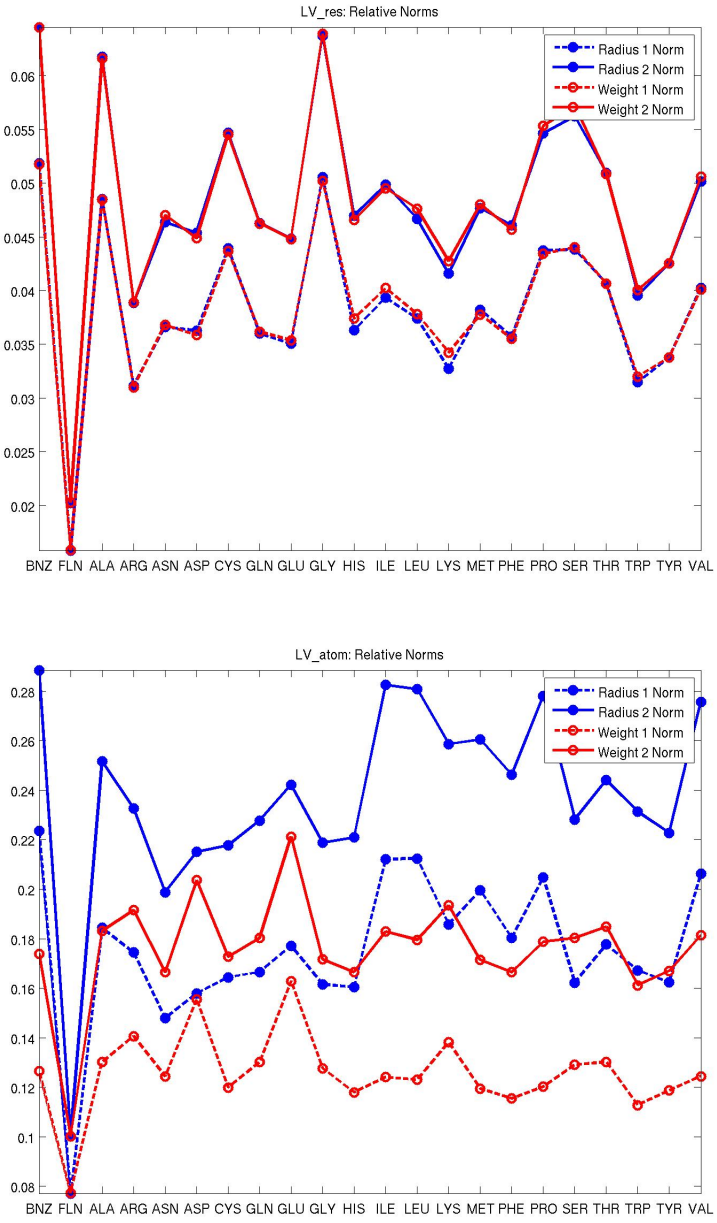


Figure 9.4: Relative E_1 and E_2 for residual (above) and atomic (below) Laguerre volumes over all dipeptides, BNZ, and FLN.

We also plot the optimal solvent parameters for all quantities. Solutes are plotted in order of increasing optimal solvent parameter in Figures 9.6 and 9.7. Residues are plotted by increasing hydrophobicity given by [37] in Figures 9.8 and 9.9. We note that there is a rough correlation between the size of the optimal sovent parameters and residue hydrophobicity.

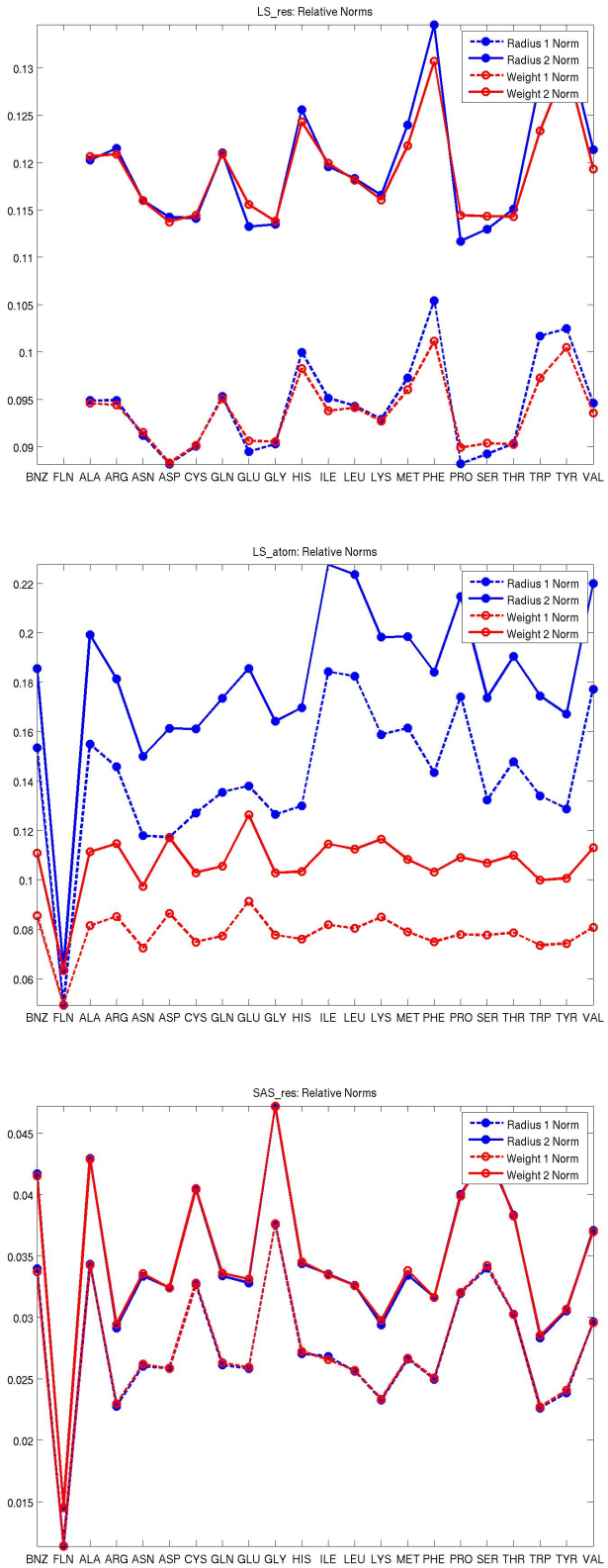


Figure 9.5: Relative E_1 and E_2 for residual (above), atomic (middle), and solvent-accessible (below) Laguerre surfaces over all dipeptides, BNZ, and FLN.

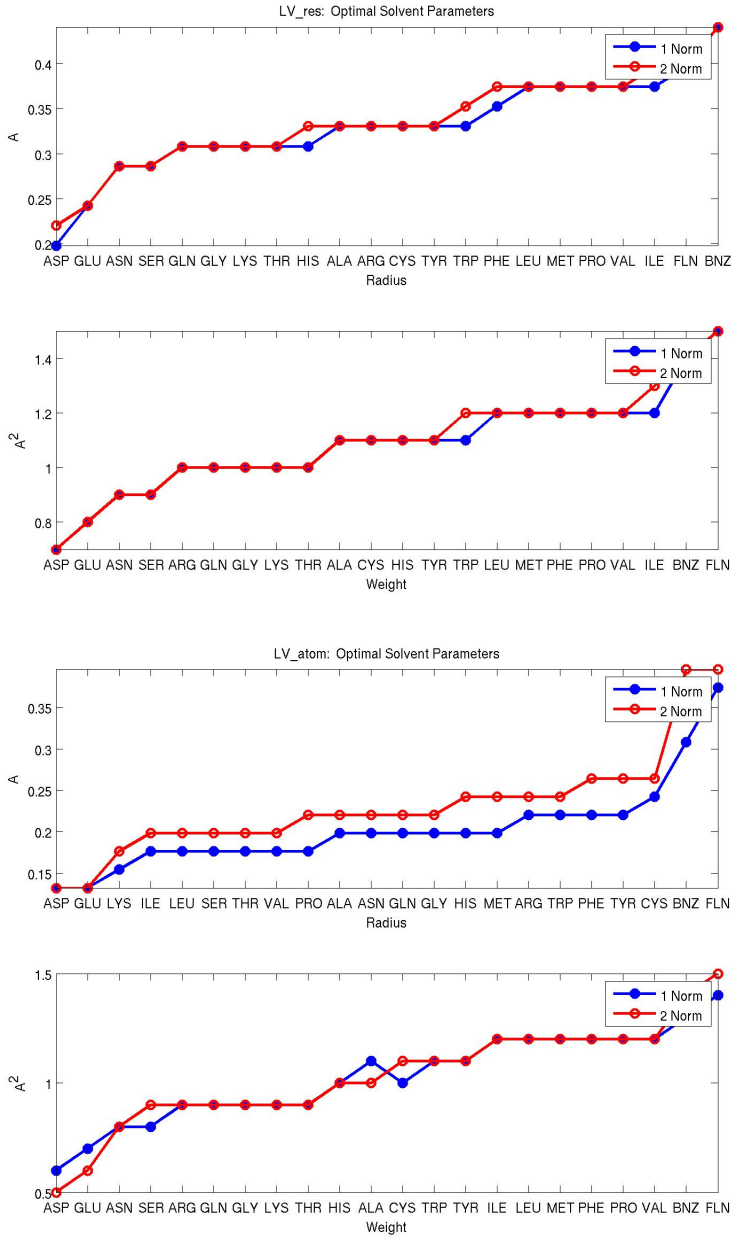


Figure 9.6: Optimal solvent parameters plotted for Laguerre volume quantities in order of increasing optimal solvent parameter. Note the rough correlation between sizes of the optimal solvent parameters and residue hydrophobicity.

We also calculated the optimal solvent parameter over all dipeptides. That is, we found w and r such that

$$E_1(LIVw(k)) = \sum_{all\_dipeptides} \frac{1}{N \cdot num\_res} \sum_{i=1}^{N} \sum_{j=1}^{num\_res} |LV(i,j) - LIVw(i,j,k)| \tag{9.1.2}$$

and

$$E_2(LIVw(k)) = \sum_{all\_dipeptides} \sqrt{\frac{1}{N \cdot num\_res} \sum_{i=1}^{N} \sum_{j=1}^{num\_res} (LV(i,j) - LIVw(i,j,k))^2} \tag{9.1.3}$$

were minimized and similarly for other Laguerre quantities.

Optimal solvent parameters are shown in Figure 9.10, and the corresponding relative errors are shown in Figure 9.11.

9.2 Optimization of the Solvent Parameter from HIV Protease Trajectory

For the HIV Protease trajectory, we had 4500 conformations. Figure 9.12 shows that the Laguerre-Intersection volumes are more stable than Laguerre volumes. Optimal solvent parameters for all Laguerre quantities are shown in Figure 9.13. We see that optimal solvent values for the HIV protease trajectory and the dipeptide trajectories are relatively close for volume quantities, whereas optimal solvent parameters for surface quantities are larger for HIV protease. Corresponding errors and relative errors (Ratios) are shown in the following table. The E_1 ratios are as expected, while E_2 ratios are large which indicate outliers.

Laguerre quantity	Average	r/w	1-Norm	Ratio	2-Norm	Ratio
LV_res	146.5596	r	8.6076	.05873	20.1539	.13751
		w	8.6203	.05881	20.1751	.1376
LV_atom	9.2949	r	1.7816	.1916	4.8569	.52253
		w	1.0550	.1135	4.5628	.4908
LS_atom	27.0871	r	4.4878	.1656	6.6875	.2468
		w	1.9637	.0724	4.5037	.1662
LS_res	.2890	r	.0012	.0041	.1295	.4481
		w	.0014	.0048	.1590	.5507
SAS_res	57.3415	r	7.2040	.1256	16.5567	.2887
		w	7.8420	.1367	17.0548	.2974

9.3 Analytic Estimate

We would like to see if the optimal solvent values found experimentally have a physical interpretation and if they make sense analytically. To this end, we first try to predict optimal solvent values for LV_atom and LS_atom. We inscribe spheres of radii r in a regular dodecahedron and octahedron. The inscribed spheres represent solute atoms and the platonic solids represent the Laguerre cells of the atoms. It was found that the average number of Delaunay neighbors for a given solute atom in HIV protease is about 16.8 which means the average number of Laguerre facets for a cell is about 16.8. However, since the Laguerre faces are irregular its surface areas and volumes are close to that of regular solids with fewer faces (i.e. octahedra and dodecahedra) (see Figure 9.14) as opposed to a regular solid with more faces (icosahedron).

The volumes and surfaces areas of the dodecahedron and octahedron are $Vd(r)$, $Sd(r)$, $Vo(r)$, and $So(r)$, respectively. We inflate the radius and weights of the spheres as follows $R = r + r_s$ and $W = r^2 + w_s$ and calculate the resulting volumes and surface areas $Vs(r, r_s)$, $Ss(r, r_s)$, $Vs(r, w_s)$, $Ss(r, w_s)$. For a given radius, r, we say that the solvent radius, r_s, or solvent weight w_s is optimal with respect to $Vd(r)$, $Sd(r)$, $Vo(r)$, or $So(r)$ if the differences of the spherical volumes or surfaces is minimized.

We see that even in this simple model, the optimal solvent values are not the same for the volume and surface area. In the more complex case of actual Laguerre cells we can hardly expect the optimal solvent values to be the same for both quantities. In the HIV protease molecule, the average van der Waals radius is 1.5 Angstroms and the average van der Waals radius of heavy atoms is about 1.8. We use both of these as values for r and compare the analytic quantities to experimental quantities in the Table 9.1.

We see that the optimal experimental solvent values for LV_atom and LS_atom fall within the range predicted analytically. Furthermore, since LV_res is additive over all atoms in a given residue, the optimal analytic solvent is the same as the optimal analytic solvent value for LV_atom which gives an experimental value in the expected range. However, LS_res and SAS_res are not additive over LS_atom. The surfaces

Table 9.1: Optimal experimental and analytical solvent values.

Laguerre Qty	Opt. Exp. Radius (1 norm / 2 norm)	Platonic Qty	Opt. Anlyt. Radius
LV_atom	.2/.2	Vo(1.8)	.3287
		Vo(1.5)	.2739
		Vd(1.8)	.1770
		Vd(1.5)	.1475
LS_atom	.3/.4	So(1.8)	.5149
		So(1.5)	.4291
		Sd(1.8)	.2720
		Sd(1.5)	.2267
Laguerre Qty	Opt. Exp. Weight (1 norm / 2 norm)	Platonic Qty	Opt. Anlyt. Weight
LV_atom	1.3/1.1	Vo(1.8)	1.291
		Vo(1.5)	.8968
		Vd(1.8)	.6687
		Vd(1.5)	.4644
LS_atom	1.7/1.6	So(1.8)	2.119
		So(1.5)	1.4710
		Sd(1.8)	1.053
		Sd(1.5)	.7313

that produce these quantities are not convex and may be very jagged, which means the platonic solid analysis does not apply. The complexity of these surfaces causes the optimal solvent values of LS_res and SAS_res to be larger than those of LS_atom.

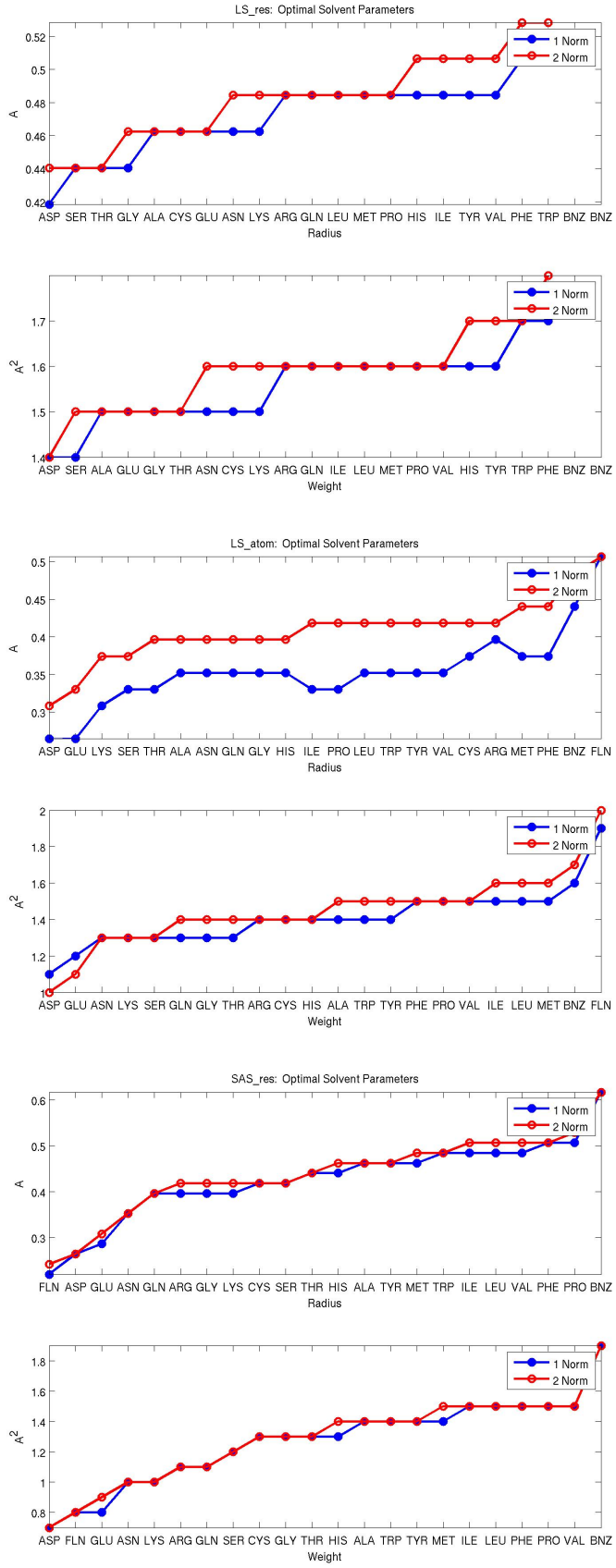


Figure 9.7: Optimal solvent parameters for Laguerre surface quantities plotted in order of increasing optimal solvent parameter. Note the rough correlation between sizes of the optimal solvent parameters and residue hydrophobicity.

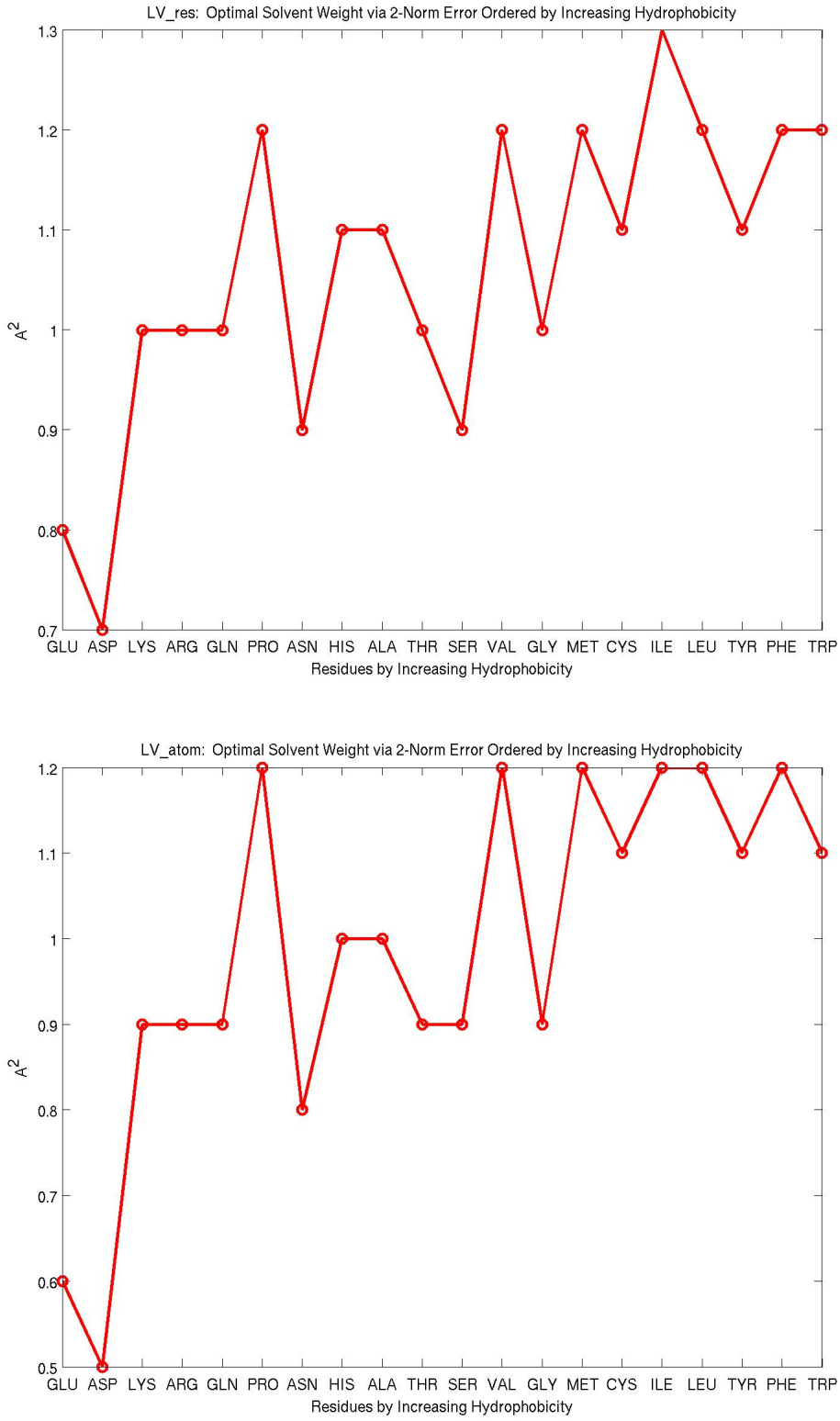
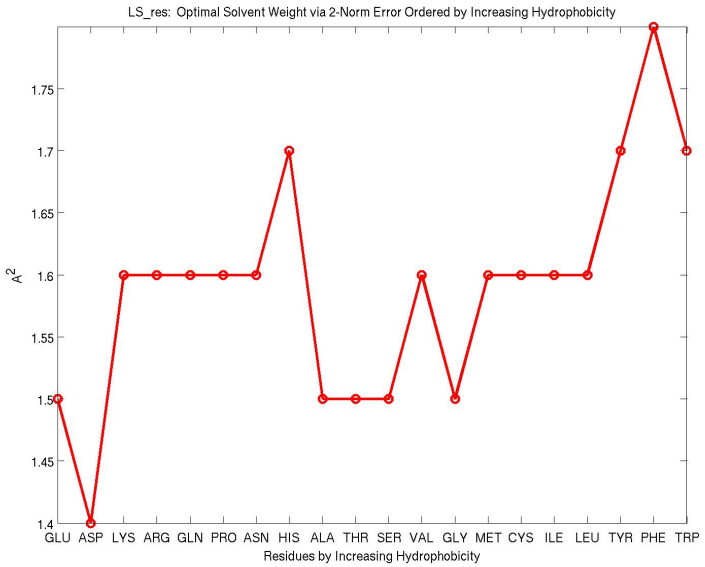
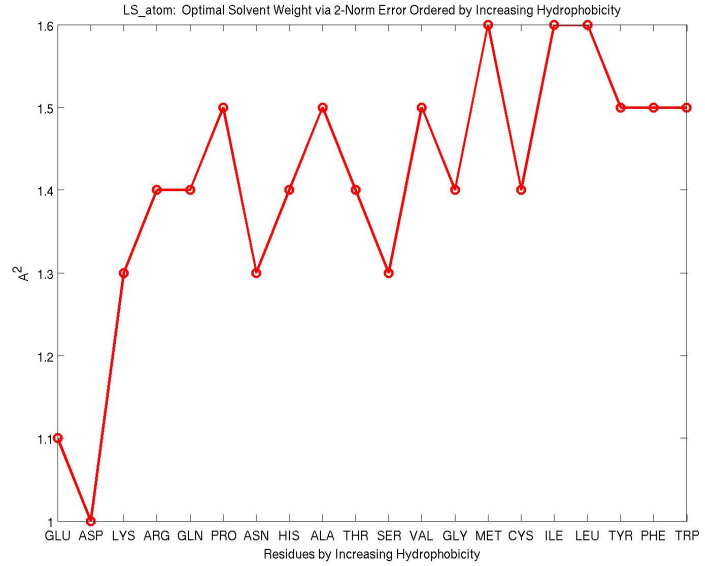


Figure 9.8: Optimal solvent parameters for Laguerre volume quantities plotted by order of increasing hydrophobicity [37].





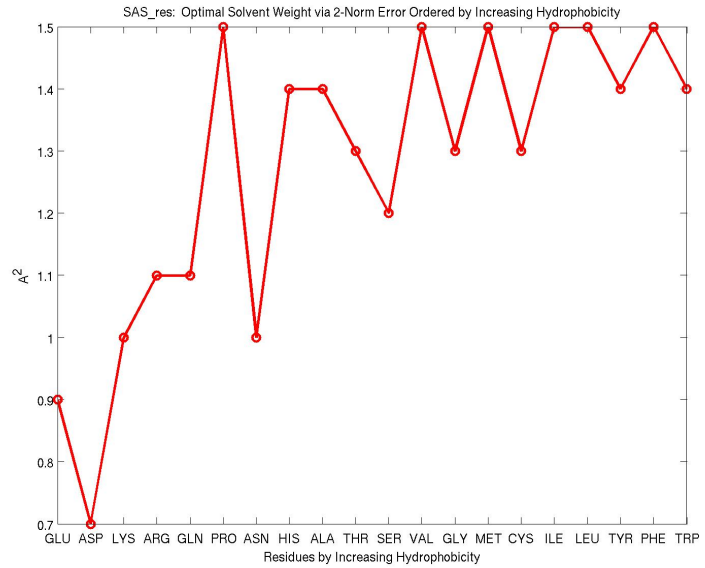


Figure 9.9: Optimal solvent parameters for Laguerre surface quantities plotted by order of increasing hydrophobicity [37].

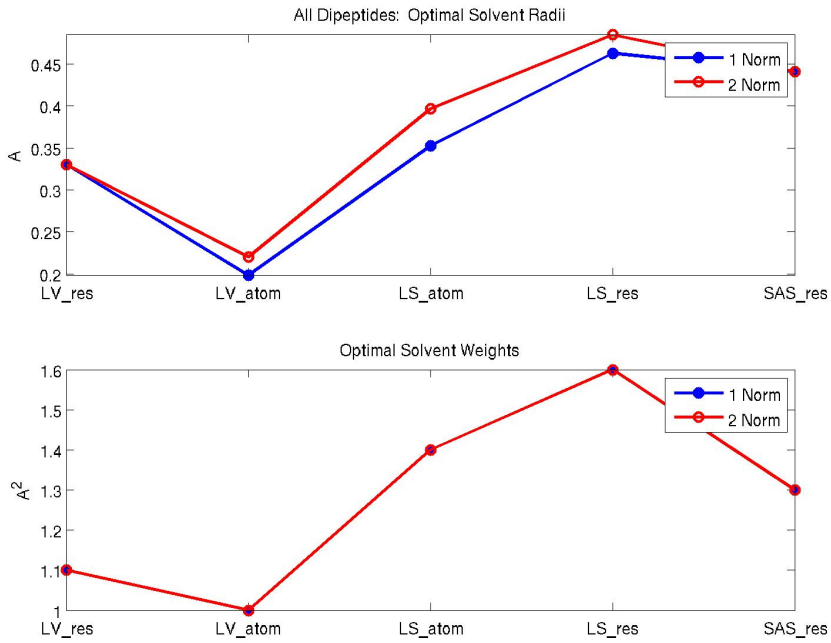


Figure 9.10: Optimal solvent parameters over all dipeptides.

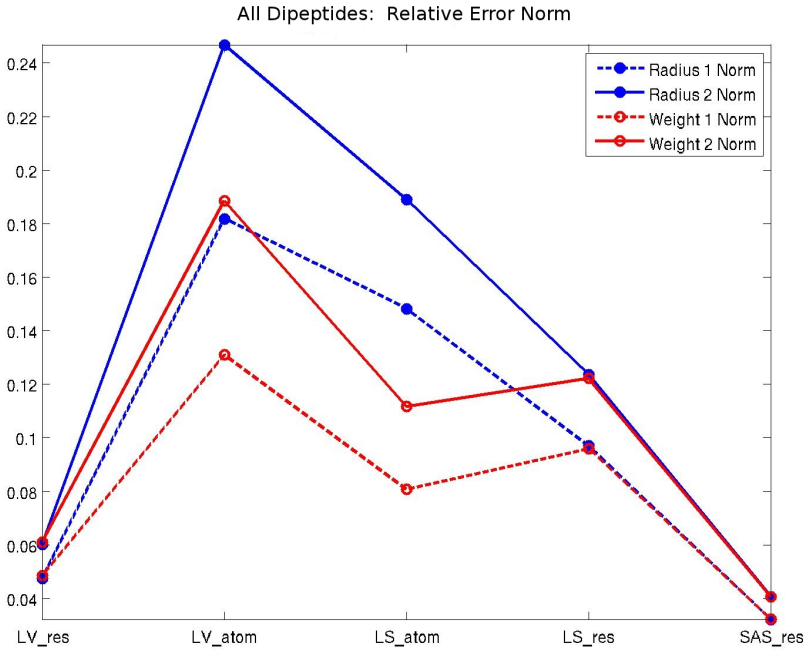


Figure 9.11: Relative errors corresponding to optimal solvent parameters over all dipeptides.

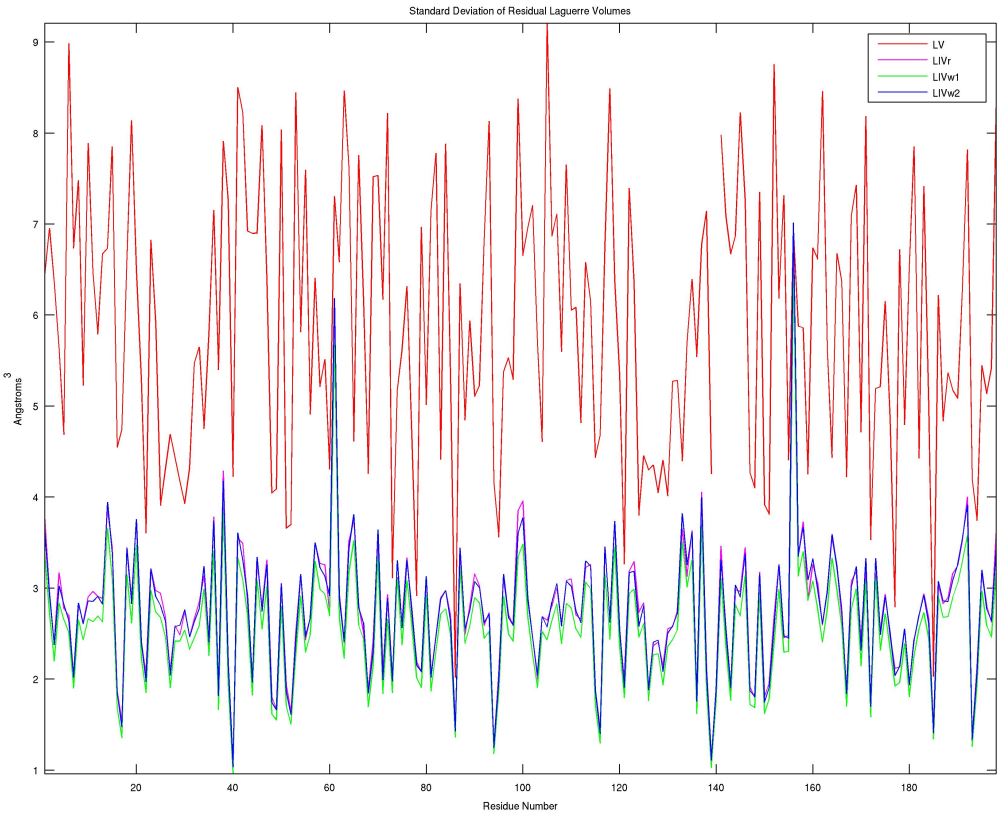


Figure 9.12: Standard deviation of residual Laguerre (red) and Laguerre-Intersection volumes. Note that the standard deviations for Laguerre-Intersection volumes are about four times smaller than the standard deviation for Laguerre volumes.

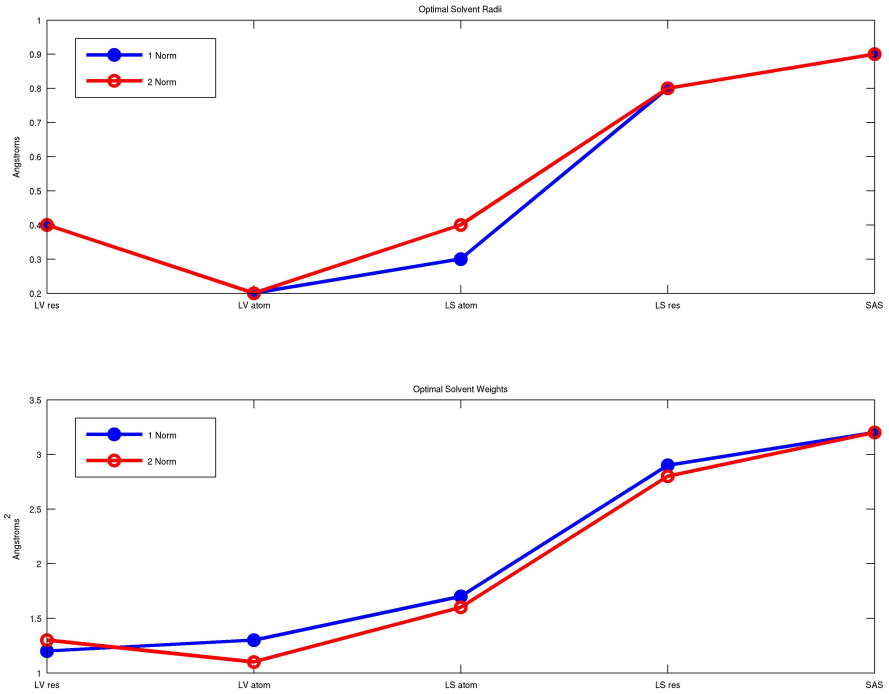


Figure 9.13: Optimal solvent parameters for the HIV Protease trajectory.

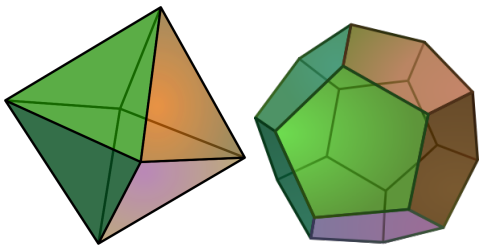


Figure 9.14: Octahedron and Dodecahedron [39], [38].

Part V

Meshing Algorithms

Chapter 10

Motivation

The alpha complex is required for the calculation of molecular and atomic volumes, surfaces, and their gradients, as well as for the calculation of Laguerre cells, Laguerre-Intersection cells and their volumes and surfaces. For typical values of α, the alpha complex is only a small subset of the Delaunay tetrahedrization. Ideally, we would like to compute this complex while bypassing the calculation of the entire tetrahedrization. In addition to the alpha complex, the aforementioned algorithms also require knowledge of the characteristic points of the simplices, incidences, tetrarings, and tetrahedra that contain a given vertex.

The Active-Edge-Face weighted Delaunay tetrahedrization algorithm, which is presented in Chapter 12, is a good basis for such an algorithm as it concurrently extracts simplices, calculates characteristic points, finds incidences, and tetrarings. This algorithm takes ideas from the Tetraring/Simplex Extraction Algorithm in Chapter 11 and the Active-Face algorithm in Section 12.3.

Chapter 11

Tetraring and Simplex Extraction Algorithm

The alpha complex is a subcomplex of the Delaunay tetrahedrization. Computation of this complex requires knowledge of the subsimplices in the tetrahedrization. Chapter 12 presents an original meshing algorithm which extracts subsimplices and "tetrarings", (the ordered sets of tetrahedra surrounding edges which is necessary for computation of Laguerre cells), concurrently during the tetrahedrization process. This method gleans ideas from the extraction algorithm presented here, which works on a precomputed tetrahedrization. In particular, simplices and tetrarings are extracted from the output of the Delaunay tetrahedization program 'regtet' [2].

11.1 Initial Data and Notation

The weighted Delaunay (Regular) tetrahedrization program 'regtet' returns the following information: 1) a list of tetrahedra, and 2) for each tetrahedron a list of neighboring tetrahedra, i.e., tetrahedra that share a facet with the given tetrahedron.

Tetra #	Vertices	Neighbors
1	$v_{1_1}\ v_{2_1}\ v_{3_1}\ v_{4_1}$	$T_{1_1}\ T_{2_1}\ T_{3_1}\ T_{4_1}$
2	$v_{1_2}\ v_{2_2}\ v_{3_2}\ v_{4_2}$	$T_{1_2}\ T_{2_2}\ T_{3_2}\ T_{4_2}$
.	.	.
.	.	.
.	.	.

For the ith tetrahedron, T^i, with vertices $v_{1_i}\ v_{2_i}\ v_{3_i}\ v_{4_i}$ we call its facets t_{1_i}, t_{2_i}, t_{3_i}, t_{4_i} with

$$\begin{aligned} t_{1_i} &= \text{conv}(v_{2_i}, v_{3_i}, v_{4_i}) \\ t_{2_i} &= \text{conv}(v_{1_i}, v_{3_i}, v_{4_i}) \\ t_{3_i} &= \text{conv}(v_{1_i}, v_{2_i}, v_{4_i}) \\ t_{4_i} &= \text{conv}(v_{1_i}, v_{2_i}, v_{3_i}). \end{aligned} \tag{11.1.1}$$

Then the neighboring tetrahedron T_{j_i} shares the facet, t_{j_i}, with T^i. Here the superscript, i, corresponds to the tetrahedron number, and the subscripts correspond the internal numbering of that simplex with respect to tetrahedron i. In some cases, $T_{j_i} = 0$, which means that the facet, $t_{j,i}$, is exterior to the convex hull, conv(V), of vertices and thus only has one neighboring tetrahedron.

For example, a sample data set with six atoms returns the following data:

Tetra #	Vertices	Neighbors
1	5 1 2 3	2 4 0 6
2	1 2 3 4	4 0 3 1
3	2 6 4 1	0 2 6 5
4	2 3 4 5	0 5 1 2
5	2 5 4 6	0 3 6 4
6	6 5 1 2	1 3 5 0

We can see that the sixth tetrahedron, T^6, shares the facet, conv(v^5, v^1, v^2), with its first neighbor $T_{6_1} = T^1$. We also see that the facet conv(v^6, v^5, v^1) is on the convex hull of data points since $T_{6_4} = 0$.

We assign internal edge numbers to the edges in a tetrahedron, T^i with vertices v_{1_i} v_{2_i} v_{3_i} v_{4_i} by

$$\begin{aligned} e_{1_i} &= \text{conv}(v_{1_i}, v_{2_i}) \\ e_{2_i} &= \text{conv}(v_{1_i}, v_{3_i}) \\ e_{3_i} &= \text{conv}(v_{1_i}, v_{4_i}) \\ e_{4_i} &= \text{conv}(v_{2_i}, v_{3_i}) \\ e_{5_i} &= \text{conv}(v_{2_i}, v_{4_i}) \\ e_{6_i} &= \text{conv}(v_{3_i}, v_{4_i}). \end{aligned} \tag{11.1.2}$$

In the tetrahedron, T^i, two internal triangles and two neighboring tetrahedra contain each edge.

$$\begin{array}{lll} e_{1_i} & t_{3_i}, t_{4,i} & T_{3_i}, T_{4_i} \\ e_{2_i} & t_{2_i}, t_{4,i} & T_{2_i}, T_{4_i} \\ e_{3_i} & t_{2_i}, t_{3,i} & T_{2_i}, T_{3_i} \\ e_{4_i} & t_{1_i}, t_{4,i} & T_{1_i}, T_{4_i} \\ e_{5_i} & t_{1_i}, t_{3,i} & T_{1_i}, T_{3_i} \\ e_{6_i} & t_{1_i}, t_{2,i} & T_{1_i}, T_{2_i} \end{array} \tag{11.1.3}$$

11.2 Extracting Simplices

Due to the properties of the Delaunay tetrahedrization, each facet interior to conv(V) is encoded twice in the tetrahedron list; it has two neighboring tetrahedra. Each facet on conv(V) is encoded once in the tetrahedron list since it only has one neighboring tetrahedron. Each edge which is a face of a tetrahedron is encoded at least once and, in the worst case, $n-1$ times where n is the number of vertices in the data set.

11.2.1 Brute Force Method

A brute force simplex extraction method is to loop through all the tetrahedra and list their faces in a global triangle or edge list. If there are N tetrahedra, lists of $4N$ triangles and $6N$ edges are produced. The triangle and edge lists are then examined and those simplices that are listed twice are removed. This usually involves sorting the lists first to accomodate the comparisons.

A drawback to this method is that, in practice, huge lists are created and the sorting and deleting of simplices is slow. Furthermore, one may find the tetrahedra that contain a given edge but the ordering of the tetrahedra around the edge is unknown. Since the edges and triangles are extracted separately, information about which triangle contains which edge requires additional work. Furthermore, due to the sorting and deletion process the indices of the edges and triangles incident to a tetrahedron are not immediately known.

11.2.2 Tetraring Method

Using the tetraring method, each simplex is extracted only once and no sorting or deletion is required. Triangles' neighboring tetrahedra and edges' tetrarings and incident triangles are found simultaneously. The indices of the subsimplices of a tetrahedron are found at no additional cost during the extraction process.

Roughly the method proceeds as follows: The program loops through each tetrahedron and extracts edges that have not been visited yet. As each edge is extracted its

tetrahedra ring is found and the triangle that are incident to the edge are extracted. As each edge and triangle is extracted, the list of higher dimensional simplices that contain the edge or triangle is easily found. Furthermore, the list of incident simplices to the tetrahedra is efficiently updated as part of the bookkeeping process needed for the extraction of the simplices.

To begin each tetrahedron is assigned four triangle and six edge slots which are initialized to 0.

Tetra #	Facets	Incident edges
1	0 0 0 0	0 0 0 0 0 0
2	0 0 0 0	0 0 0 0 0 0
.	.	.
.	.	.
.	.	.

A zero in triangle or edge slot means that the corresponding simplex has not been extracted yet. Once the corresponding simplex has been extracted it is set to the index of the new simplex. These lists are called tetra%tri and tetra%edg respectively. An empty global edge list and an empty global triangle list is initialized. The list edge(i)%tetra_ring lists the tetrahedra around edge i. In our application we also want to know the triangles of which edge i is a face. This is stored in edge(i)%tri.

PSEUDOCODE:

subroutine extract_simplices

for *i=1,num_tetra* **do**

for *j=1,6* **do**

if *tetra(i)%edg(j)==0* **then**

num_edg = num_edg +1

tetra(i)%edg(j)=num_edg

call tetra_ring(num_edg,j,i)

end

end

end

Algorithm 1: extract_simplices

```
PSEUDOCODE:

subroutine     tetra_ring(edg_ind,int_edg_ind,T_ind)

%initialize data
current_edg_int=internal_edg_index
current_tetra=T_ind
num_zero=0
previous_tetra=-1
num_while=0

%set first tetra in tetra_ring
edge(edg_ind)%tetra_ring(1)=T_ind
num_in_ring=1

do while(1==1)
   num_while=num_while+1
   if (num_while>20) then
      print *, 'num_while>20:  exit'
      exit
   end if
   tag=0 !reset at beginning of every loop
   if(current_edg_int==1) then
      !possible current_tris:  1,2
      if (Tnbr(4,current_tetra)/=previous_tetra) then
         current_tri=1
         num_adj_tri=num_adj_tri+1
         mark_tri=tetra(current_tetra)%tri(1)
         if (mark_tri==0) then !extract tri
            !put the new triangle in global triangle list
         end if
```

```
         previous_tetra=current_tetra
         current_tetra=Tnbr(5-current_tri,current_tetra) !advance tetra
         if(current_tetra/=0) then
            !find internal index of edge w.r.t. current tetra
            !reset current_edg_int
            !find internal index tri w.r.t. current tetra
            !mark tetra%facet(.), tetra%edge(.)
         end if
      else
         !do same for tetra(3,current_tetra)
      end if
   elseif(elseif(current_edg_int==2) then
   ...
   elseif(current_edg_int==3) then
   ...
   elseif(current_edg_int==4) then
   ...
   elseif(current_edg_int==5) then
   ...
   elseif(current_edg_int==6) then
   ...
   end if


   !set edg_ring info
   !decide whether to exit loop or set info for next loop
   if (current_tetra/=T_ind) then
      num_in_ring=num_in_ring+1
      edge(edg_ind)%tetra_ring(num_in_ring)=current_tetra
   end if
   if (current_tetra==0) then
      !loop around edge in other direction or exit
      num_zero=num_zero+1
      if (num_zero==1) then
         current_tetra=T_ind
         previous_tetra=refedg(edg_ind)%tetra_ring(2)
         current_edg_int=edg_num_int
      elseif (num_zero==2) then
         exit
      end if
   end if
end do while
```

Example

Consider the sample tetrahedrization given in Section 11.1. We initialize the arrays.

Tetra #	Facets	Edges
1	0 0 0 0	0 0 0 0 0 0
2	0 0 0 0	0 0 0 0 0 0
.	.	.
.	.	.
.	.	.

Edge #	Indices	tetra_ring
...		

Tri #	Indices
...	

We begin by considering the first edge of the first tetra. Since tetra(1)%edge(1)=0 we need to extract and mark the edge.

Tetra #	Facets	Edges
1	0 0 0 0	1 0 0 0 0 0
2	0 0 0 0	0 0 0 0 0 0
.	.	.
.	.	.
.	.	.

Edge #	Indices	tetra_ring
1	5 1	1 ...

Tri #	Indices
...	

The pertinent neighboring tetra in this case are $T_{4_1} = 6$ and $T_{3_1} = 0$. Since there is only one tetrahedron in the edge's tetra_ring, arbitrarily choose $T_{4_1} = 6$ as the next current tetrahedron. The facet between the two tetra is $t_{4_1} = \text{conv}(5, 1, 2)$. Since tetra(1)%tri(4)=0, this triangle needs to be extracted and marked.

Tetra #	Facets	Edges
1	0 0 0 1	1 0 0 0 0 0
2	0 0 0 0	0 0 0 0 0 0
.	.	.
.	.	.
.	.	.

Edge #	Indices	tetra_ring
1	5 1	1 6...

Tri #	Indices
1	5 1 2

It is not immediately obvious what the internal edge and facet numbers of the current edge and current tri are in tetrahedron 6. These internal numbers are found using brute force, which doesn't take very long since there are only four possible triangles and six possible edges. We see that the internal edge number of $e^1 = \text{conv}(5, 1)$ is 4 and the internal triangle number of $t^1 = \text{conv}(5, 1, 2)$ is 1. We now mark these.

Tetra #	Facets	Edges
1	0 0 0 1	1 0 0 0 0 0
2	0 0 0 0	0 0 0 0 0 0
3	0 0 0 0	0 0 0 0 0 0
4	0 0 0 0	0 0 0 0 0 0
5	0 0 0 0	0 0 0 0 0 0
6	1 0 0 0	0 0 0 1 0 0

Edge #	Indices	tetra_ring
1	5 1	1 6...

Tri #	Indices
1	5 1 2

The new internal edge number is 4 which corresponds to tetra neighbors $T_{6_1} = 1$ and $T_{6_4} = 0$. The previous tetrahedron is 1, we choose $T_{6_4} = 0$ as our next tetrahedron. The corresponding facet is $t_{6_4} = \text{conv}(6, 5, 1)$. Since tetra(6)%tri(4)=0 this is a new triangle which needs to be listed and marked.

Tetra #	Facets	Edges
1	0 0 0 1	1 0 0 0 0 0
2	0 0 0 0	0 0 0 0 0 0
3	0 0 0 0	0 0 0 0 0 0
4	0 0 0 0	0 0 0 0 0 0
5	0 0 0 0	0 0 0 0 0 0
6	1 0 0 2	0 0 0 1 0 0

Edge #	Indices	tetra_ring
1	5 1	1 6 0...

Tri #	Indices
1	5 1 2
2	6 5 1

Since the new current tetra is now zero, this means we need to start looping around the tetra ring in the opposite direction. We reset the current tetra as edge(1)%tetra_ring(1)=1, the current edge number to be 1, and the previous tetra as 6. With current edge number 1, we consider $T_{4_1} = 6$ and $T_{3_1} = 0$. We choose $T_{3_1} = 0$ as the next tetra with corresponding facet $t_{3_1} = \text{conv}(5, 1, 3)$ is needs to be extracted and marked since tetra(1)%tri(3)=0.

Tetra #	Facets	Edges
1	0 0 3 1	1 0 0 0 0 0
2	0 0 0 0	0 0 0 0 0 0
3	0 0 0 0	0 0 0 0 0 0
4	0 0 0 0	0 0 0 0 0 0
5	0 0 0 0	0 0 0 0 0 0
6	1 0 0 2	0 0 0 1 0 0

Edge #	Indices	tetra_ring
1	5 1	1 6 0 0

Tri #	Indices
1	5 1 2
2	6 5 1
3	5 1 3

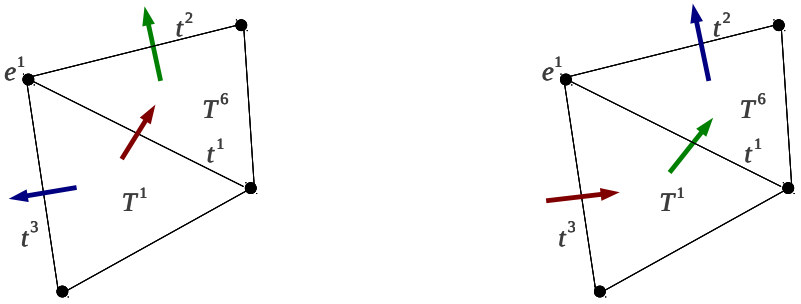
We have now encountered two zeros in the tetra ring of edge one, which means we have extracted all simplices pertaining to that edge.

If needed, tetra ring may be easily reorded as follows which gives one direction of traversal.

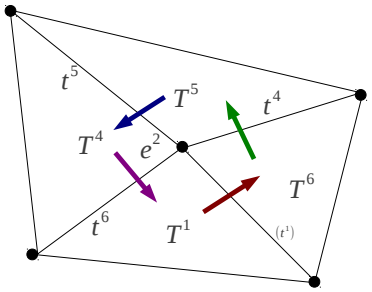
Tetra #	Facets	Edges
1	0 0 3 1	1 0 0 0 0 0
2	0 0 0 0	0 0 0 0 0 0
3	0 0 0 0	0 0 0 0 0 0
4	0 0 0 0	0 0 0 0 0 0
5	0 0 0 0	0 0 0 0 0 0
6	1 0 0 2	0 0 0 1 0 0

Edge #	Indices	tetra_ring
1	5 1	1 6 0 0 $\rightarrow$ 0 1 6 0

Tri #	Indices
1	5 1 2
2	6 5 1
3	5 1 3



We move onto the second edge of the first tetra. As we will see, this edge is interior to the convex hull of points which means no zeros are encountered in the edge's tetra ring and there is one direction of traversal around the edge. The extraction method proceeds with few variations. As we travel from tetra 1 to tetra 6 we see that tetra(1)%tri(4) is nonzero. This means that the corresponding triangle has been extracted and need not be extracted again. We continue through the tetra loop until we reach tetra 4. In this case we see the candidate next tetra is tetra 1 which means the entire ring has been traversed. Before moving onto the next edge, we first extract and mark the common facet, conv$(2, 3, 5)$, between the two tetra.



Tetra #	Facets	Edges
1	0 6 3 1	1 2 0 0 0 0
2	0 0 0 0	0 0 0 0 0 0
3	0 0 0 0	0 0 0 0 0 0
4	0 5 6 0	0 0 2 0 0 0
5	0 0 4 5	2 0 0 0 0 0
6	1 0 4 2	0 0 0 1 2 0

Tri #	Indices
1	5 1 2
2	6 5 1
3	5 1 3
4	6 5 2
5	2 5 4
6	2 3 5

Edge #	Indices	tetra_ring
1	5 1	1 6 0 0 $\rightarrow$ 0 1 6 0
2	5 2	1 6 5 4

We extract edges 3, 4, 5, and 6 from tetra 1, the involved triangles, and populate the neighbor and tetra ring lists.

Tetra #	Facets	Edges
1	10 6 3 1	1 2 3 4 5 6
2	12 11 9 10	4 5 0 6 0 0
3	0 9 8 0	0 0 4 0 0 0
4	0 7 5 6 12	6 0 2 3 0 0
5	0 0 4 5	2 0 0 0 0 0
6	1 8 4 2	0 0 0 1 2 4

Tri #	Indices
1	5 1 2
2	6 5 1
3	5 1 3
4	6 5 2
5	2 5 4
6	2 3 5
7	3 4 5
8	6 1 2
9	2 4 1
10	1 2 3
11	1 3 4
12	2 3 4

Edge #	Indices	tetra_ring
1	5 1	1 6 0 0 $\rightarrow$ 0 1 6 0
2	5 2	1 6 5 4
3	5 3	1 0 4 0 $\rightarrow$ 0 1 4 0
4	1 2	1 6 3 2
5	1 3	1 0 2 0 $\rightarrow$ 0 2 1 0
6	2 3	1 4 2

Having finished with the first tetrahedron, we move onto the second tetrahedron. We see that the first and second edges of this tetrahedron have been extracted since tetra(2)%edge(1) and tetra(2)%edge(2) are nonzero. We extract the third edge of tetra 2 and its corresponding triangles and tetra ring. We continue until all tetrahedra have been visited. Here are the final lists.

Tetra #	Facets	Edges
1	10 6 3 1	1 2 3 4 5 6
2	12 11 9 10	4 5 7 6 8 9
3	13 9 8 14	10 8 4 11 12 7
4	7 5 6 12	6 8 2 9 3 13
5	15 14 4 5	2 8 10 13 14 11
6	1 8 4 2	14 12 10 1 2 4

Edge #	Indices	tetra_ring
1	5 1	1 6 0 0 $\rightarrow$ 0 1 6 0
2	5 2	1 6 5 4
3	5 3	1 0 4 0 $\rightarrow$ 0 1 4 0
4	1 2	1 6 3 2
5	1 3	1 0 2 0 $\rightarrow$ 0 2 1 0
6	2 3	1 4 2
7	1 4	2 3 0 0 $\rightarrow$ 0 2 3 0
8	2 4	2 3 5 4
9	3 4	2 0 4 0 $\rightarrow$ 0 4 2 0
10	2 6	3 5 6
11	6 4	3 5 0 0 $\rightarrow$ 0 3 5 0
12	6 1	3 6 0 0 $\rightarrow$ 0 3 6 0
13	4 5	4 5 0 0 $\rightarrow$ 0 4 5 0
14	5 6	5 6 0 0 $\rightarrow$ 0 5 6 0

Tri #	Indices
1	5 1 2
2	6 5 1
3	5 1 3
4	6 5 2
5	2 5 4
6	2 3 5
7	3 4 5
8	6 1 2
9	2 4 1
10	1 2 3
11	1 3 4
12	2 3 4
13	6 4 1
14	2 6 4
15	5 4 6

Chapter 12

Weighted Delaunay Tetrahedrization Algorithm

12.1 Preliminaries

12.1.1 Spherical Representation

Consider a molecule represented by a set of spheres or weighted points $\mathcal{A} \subset \mathbb{R}^3 \times \mathbb{R}$. For $p_i \in \mathcal{A}$ we write $p_i = (p_i', w_i)$, where p_i' is the center of the atom and $w_i = r_i^2$ is the weight or squared radius of the atom. We may also write $w_i = p_i''$.

For $x' \in \mathbb{R}^3$ we define the power distance between x' and p_i as

$$\pi(p_i, x') = |p_i' - x'|^2 - w_i. \tag{12.1.1}$$

Similarly, we define the power product between two weighted points as

$$\Pi(p_i, p_j) = |p_i' - p_j'|^2 - w_i - w_j. \tag{12.1.2}$$

The points p_i and p_j are said to be orthogonal if $\Pi(p_i, p_j) = 0$.

Let $T = \{p_i, p_j, p_k, p_l\}$ represent a tetrad of points in $\mathcal{A}$. Assuming the points in T are non-coplanar, there exists a unique point $x_T = (x_T', x_T'')$ such that $\Pi(p_m, x_T) = 0$

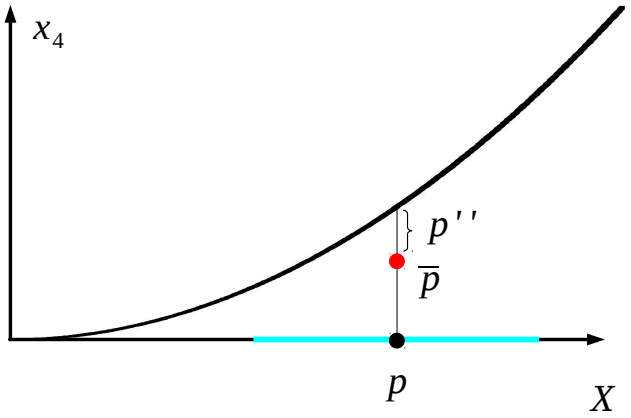


Figure 12.1: The horizontal axis represents they hyperplane $X = \mathbb{R}^3$ that contains the first three coordinates of $\overline{p} \in \mathbb{R}^4$. The vertical axis represents the fourth coordinate, x_4. The sphere p is represented by the cyan colored segment in the X-axis with the center of p represented by the black dot. The lifted point $\overline{p}$, which is a vertical distance p'' from the parabola, $x_4 = X^2$, is represented by the red dot.

for $m = i, j, k, l$. This is equivalent to saying that the points in T are equi-powerdistant from the point x'_T with power distance x''_T. We call x_T the characteristic point of the tetrahedron, with x'_T the center of the characteristic point, and x''_T the size of the tetrahedron.

A tetrahedrization, $\mathcal{T}'$, is a weighted Delaunay tetrahedrization or regular tetrahedrization, $\mathcal{T}$, under the following conditions

$$\mathcal{T}' \text{ is regular} \iff \Pi(x_T, p_n) \geq 0 \text{ for all } T \in \mathcal{T}' \text{ and for all } p_n \in \mathcal{A} \tag{12.1.3}$$

Furthermore, for a regular tetrahedrization $\mathcal{T}$,

$$T \in \mathcal{T} \iff \Pi(x_T, p_n) \geq 0 \text{ for all } p_n \in \mathcal{A}. \tag{12.1.4}$$

12.1.2 Representation in $\mathbb{R}^4$

We may represent spheres in $\mathbb{R}^3$ by points in $\mathbb{R}^4$ under the following map (See Figure 12.1):

$$\phi : \mathbb{R}^3 \times \mathbb{R} \longrightarrow \mathbb{R}^4, \quad \phi(p) = \overline{p} = (p', |p'|^2 - p''). \tag{12.1.5}$$

The set $\mathcal{A}$ under the map ϕ is written $\overline{\mathcal{A}}$.

Polar Hyperplane

Define the *polar hyperplane* ,

$$H(\overline{p}) = \overline{p}^* = \{\overline{x} \in \mathbb{R}^4 : 2x' \cdot p' - x_4 - p_4 = 0\}. \tag{12.1.6}$$

The plane H contains all points $\phi(x)$ such $\Pi(x,p) = 0$. That is, $\phi^{-1}(H(\overline{p}))$ is the orthogonal complement of p,

$$\begin{aligned} 0 &= \Pi(p,x) \\ &= |p' - x'|^2 - p'' - x'' \\ &= |p'|^2 - 2x' \cdot p' + |x'|^2 - |p'|^2 + p_4 - |x'|^2 + x_4 \\ &= -2x' \cdot p' + x_4 + p_4. \end{aligned} \tag{12.1.7}$$

The pole, H^*, of the hyperplane H is defined to be $\overline{p}$.

It is easy to see that

$$(\overline{p}^*)^* = \overline{p} \tag{12.1.8}$$

and

$$(H^*)^* = H. \tag{12.1.9}$$

Let $r(\overline{p})$ be the vertical reflection of $\overline{p}$ over the parabola X^2. That is $r(\overline{p}) = \overline{q}$ where $q = (p', -p'')$. Since $\Pi(p,q) = 0$, we have $r(\overline{p}) \in H(\overline{p})$. Also for any point $s = (s', 0)$ on the sphere $|s' - p'|^2 = p''$, $\overline{s} \in H(\overline{p})$.

This polarity preserves inclusions. Let H^+ and H^- be the spaces above and below the hyperplane H. If

$$\overline{q} \in H \iff H^* \in \overline{q}^* \tag{12.1.10}$$

$$\overline{q} \in H^+ \iff H^* \in q^{*+} \tag{12.1.11}$$

$$\overline{q} \in H^- \iff H^* \in q^{*-}. \tag{12.1.12}$$

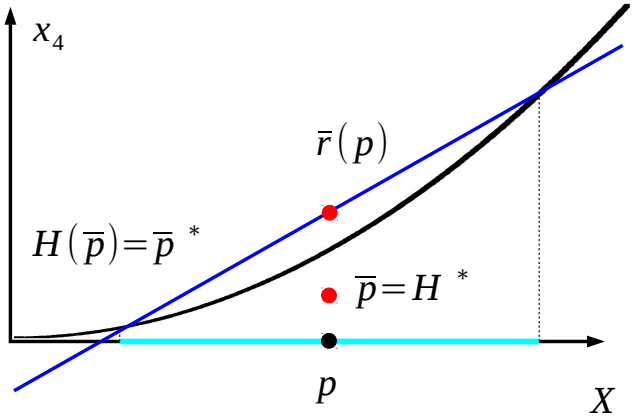


Figure 12.2: The polar hyperplane $H(\overline{p})$ is represented by the blue line. The pole of H is $\overline{p}$ and $r(\overline{p}) \in H(\overline{p})$. The intersection of the H with the parabola projects vertically onto the boundary of the sphere represented by p.

These equations state that if $\overline{q}$ lies on/above/below the hyperplane H, then the pole H^* of H lies on/above/below the polar hyperplane $\overline{q}^*$ of q.

Equation (12.1.10) follows from (12.1.7). To show (12.1.11), let $H = \{\overline{x} \in \mathbb{R}^4 : 2x' \cdot p' - x_4 - p_4 = 0\}$ and $H^* = \overline{p}$. The polar hyperplane of $\overline{q}$ is $\overline{q}^* = \{\overline{x} \in \mathbb{R}^4 : 2x' \cdot q' - x_4 - q_4 = 0\}$. Then

$$\overline{q} \in H^+ \Leftrightarrow q_4 > 2q' \cdot p' - p_4 \Leftrightarrow p_4 > 2q' \cdot p' - q_4 \Leftrightarrow H^* \in q^{*+}. \tag{12.1.13}$$

Equation (12.1.12) can be shown in a similar manner.

Power Product and the Polar Hyperplane

As previously noted, a point $\overline{p}$ is on the hyperplane $\overline{q}^*$ if and only if p and q are orthogonal, i.e. $\Pi(p, x) = 0$. Furthermore, $\overline{p}$ is above the hyperplane $\overline{q}^*$ if and only if $\Pi(p, q) > 0$, and $\overline{p}$ is below $\overline{q}^*$ if and only if $\Pi(p, q) < 0$. That is,

$$\overline{p} \in \overline{q}^* \iff \Pi(p, q) = 0 \tag{12.1.14}$$

$$\overline{p} \in \overline{q}^{*+} \iff \Pi(p, q) > 0 \tag{12.1.15}$$

$$\overline{p} \in \overline{q}^{*+} \iff \Pi(p, q) < 0. \tag{12.1.16}$$

These equations are easy to show using (12.1.6) and $p_4 = |p'|^2 - p''$.

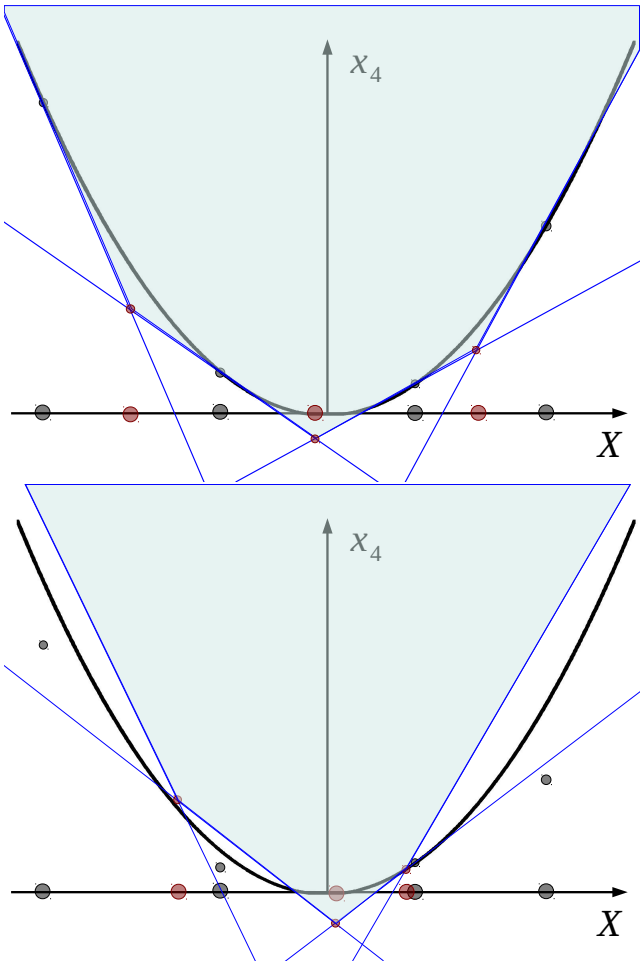
Characteristic Points and Polar Hyperplanes

Recall that the characteristic point, x_T, of a set of four points, T, satisfies

$$\Pi(x_T, p_j) = 0 \text{ for all } p_j \in T \tag{12.1.17}$$

This means that the lifted point $\overline{x_T}$ is the point of intersection of the planes $H(\overline{p_j})$ for p_j in T. From this we see that

$$T \in \mathcal{T} \iff \overline{p} \in H(\overline{x_T})^+ \text{ for all } \overline{p} \in \overline{\mathcal{A}} \tag{12.1.18}$$



12.1.3 Facets of $\mathcal{T}$

Given a triangle t and a point p_a, define $T_a = \{t, p_a\}$. We write the characteristic point of T_a as $x_{T_a} = x_a$. From equation 12.1.4 we see that

$$t \text{ is triangle in } \mathcal{T} \iff \exists p_a \in \mathcal{A} \text{ such that } \Pi(x_a, p) \geq 0 \; \forall \; p \in \mathcal{A} \tag{12.1.19}$$

Define P as the plane that contains t. We may assign an orientation to the triangle t and define an upper half space, P^+, and a lower halfspace P^-. Then we may extend equation 12.1.19 in the following manner

$$t \in \mathcal{T} \text{ and there exists points in } P^{+(-)} \iff \exists p_a \in P^{+(-)} \text{ such that } \Pi(x_a, p) \geq 0 \; \forall \; p \in \mathcal{A} \tag{12.1.20}$$

Equivalently, $t \in \mathcal{T}$ and there exists points in $P^{+(-)}$ if and only if $\exists p_a \in P^{+(-)}$ such that for $T_a = \{t, p_a\}$, $\overline{p}$ is on or above $H(\overline{x_a})$ for all $p \in \mathcal{A}$.

Set of orthogonal spheres

For three non-colinear points t in the plane P, there exists a set of spheres or weighted points, $S(t) = l(t)$, which are orthogonal to all $p_m \in t$. The centers of points in this set lie on the line $l'(t)$ which is perpendicular to P. The characteristic point, x_t, of t is the sphere in $S(t)$ which has minimum weight. The center of x_t is the intersection of $l'(t)$ with P:

$$x'_t = l'(t) \bigcap P. \tag{12.1.21}$$

The line $\overline{S(t)} = \overline{l(t)} \subset \mathbb{R}^4$ is the intersection of the hyperplanes $H(\overline{p_m})$ for $p_m \in t$. That is,

$$\overline{l(t)} = \bigcap_{p_m \in t} H(\overline{p_m}) \tag{12.1.22}$$

This means that for $p_i \notin P$, the characteristic point of $T_i = \{t, p_i\}$ satisfies $x_i \in S(t)$ and $\overline{x_i} \in \overline{l(t)}$ (See Figure 12.3). Furthermore,

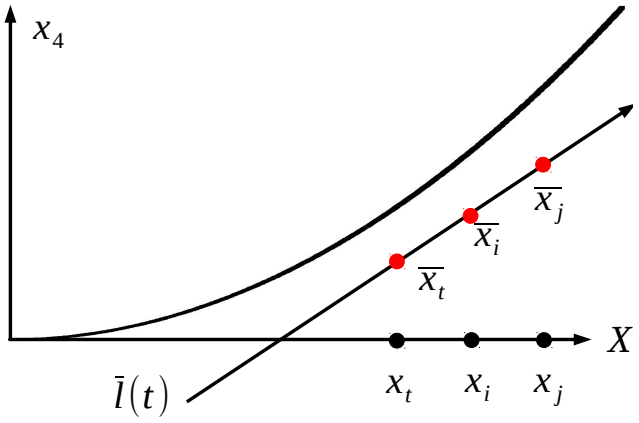


Figure 12.3: The characteristic points of all $T = \{t, p\}$ belong to the set $S(t)$. This set maps to a line $\overline{l(t)} \subset \mathbb{R}^4$ which goes through the point $\overline{x_t}$. The plane P which contains t lies in X, contains the point x_t, and is perpendicular to the page.

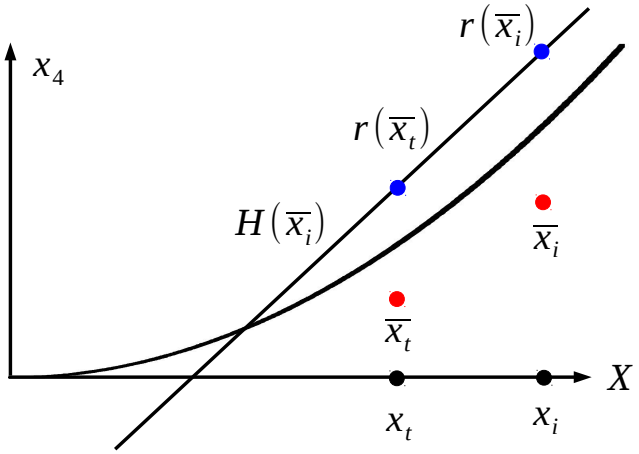


Figure 12.4: The hyperplane $H(\overline{x_i})$ contains the points $r(\overline{x_i})$, $r(\overline{x_t})$ and $\overline{p_m}$ for $p_m \in t$. The points $\overline{p_m}$ line in a plane which contains $r(\overline{x_t})$ and projects vertically onto P.

$$\overline{x_i} \;=\; \bar{l}(t) \bigcap H(\overline{p_i}). \tag{12.1.23}$$

Next, we examine the hyperplane $H(\overline{x_i})$. Recall that $r(\overline{x_i})$ is the vertical reflection of $\overline{x_i}$ over the parabola. This gives $r(\overline{x_i}) \in H(\overline{x_i})$. Also, for $p_m \in t$ we have $\overline{p_m} \in H(\overline{x_i})$ which shows $r(\overline{x_t}) \in H(\overline{x_i})$ (See Figure 12.4). This means that $H(\overline{x_i})$ and $H(\overline{x_j})$ intersect along the 2-dimensional plane $\overline{P(t)}$ that contains the points $\overline{p_m}$. The vertical projection of $\overline{P(t)}$ onto X is P (See Figure 12.5).

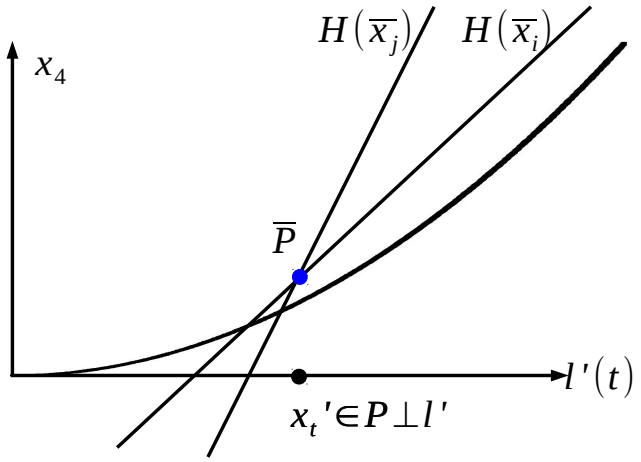


Figure 12.5: The planes $H(\overline{x_i})$ and $H(\overline{x_j})$ intersect along the 2-dimensional plane $\overline{P}$.

Weighted Delaunay Distance

Let $T = \{t, p\}$. The weighted Delaunay distance between t and p is defined as

$$wdd(t,p) = \begin{cases} dist(x'_T, P), & \text{if } x'_T \in \text{Halfspace(t,p)} \\ -dist(x'_T, P) & \text{otherwise} \end{cases} \tag{12.1.24}$$

Here $dist(x'_T, p)$ is the orthogonal distance between x'_T and the plane P (See Figure 12.6).

Since $x'_t \in P$ we may also write

$$wdd(t,p) = \begin{cases} |x'_T, x'_t|, & \text{if } x'_T \in \text{Halfspace(t,p)} \\ -|x'_T, x'_t| & \text{otherwise} \end{cases} \tag{12.1.25}$$

For a triangle, t, we may define the upper and lower nearest weighted Delaunay neighbors

$$nnbr\_wdd^+(t) = p_j \in P^+ \text{ such that } wdd(t,p_j) \text{is a minimum over all points in} P^+ \tag{12.1.26}$$

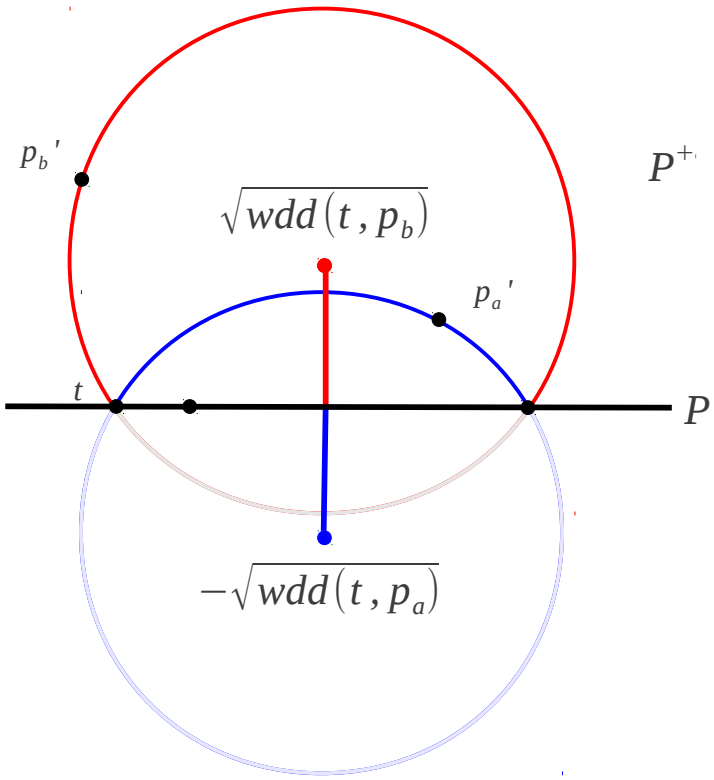


Figure 12.6: Here, the triangle t is represented by the three points on the plane P. The characteristic point of $T_a = \{t, p_a\}$ is represented by the blue circle and the characteristic point of $T_b = \{t, p_b\}$ is represented by the red circle. The centers of the characteristic points x_a and x_b are represented by the red and blue dots respectively. The square roots of the weighted Delaunay distances between t and p_a and p_b are shown by the appropriate red and blue lines.

and

$$nnbr\_wdd^{-}(t) = p_j \in P^{-} \text{ such that } wdd(t, p_j) \text{is a minimum over all points in} P^{-} \tag{12.1.27}$$

If t is not on the convex hull of $\mathcal{A}'$, then $nnbr\_wdd^{+}$ and $nnbr\_wdd^{-}$ both exist. However if t is on the convex hull, then it only has one nearest weighted Delaunay neighbor, $nnbr\_wdd^{+}$ or $nnbr\_wdd^{-}$, depending on the orientation of the triangle.

12.2 Basis of Algorithm

The algorithm is based on the fact that if t is a face in the regular tetrahedrization then, assuming $nnbr\_wdd^{+}$ exists, $\{t, nnbr\_wdd^{+}\} \in \mathcal{T}$, and similarly for $nnbr\_wdd^{-}$.

Nearest Weighted Delaunay Distance Neighbor Theorem 1. *If t is a face in $\mathcal{T}$ and $nnbr\_wdd^{+}$ exists, then $T = \{t, nnbr\_wdd^{+}\} \in \mathcal{T}$.*

An equivalent result holds for $nnbr\_wdd^{-}$.

Before we prove Theorem 1, we present some properties of Π and wdd. As before, the triangle, t, is in the plane P, and for a point p_i not in t, we write $T_i = \{t, p_i\}$. The tetrahedron T_i has characterstic point x_i.

Theorem 2. $\Pi(x_i, p_j) = 0 \iff \Pi(x_j, p_i) = 0$

Proof. (Spherical)

Assume that $\Pi(x_i, p_j) = 0$. We also have $\Pi(x_i, p_m) = 0$ for all $p_m \in t$. Then by definition, x_i is the characteristic point of T_j. That is $x_i = x_j$. Then we have $\Pi(x_j, p_i) = \Pi(x_i, p_i) = 0$.

A similar argument holds for the converse. □

As a warm up for proofs of following theorems, we will also argue Theorem 2 using raised points.

Proof. (In $\mathbb{R}^4$)

Assume that $\Pi(x_i, p_j) = 0$. Then we have $\overline{p_j} \in H(\overline{x_i})$. We also have $\overline{p_m} \in H(\overline{x_i})$ which gives $H(\overline{x_i}) = H(\overline{x_j})$. Then $\overline{p_i} \in H(\overline{x_j})$ gives $\Pi(x_j, p_i) = 0$.

Again, a similar argument holds for the converse. □

Theorem 3. *For p_i and p_j in the same half spaces, $\Pi(x_i, p_j) > 0 \iff \Pi(x_j, p_i) < 0$ (See Figure 12.5 and Figure 12.10).*

Proof. Assume $\Pi(x_i, p_j) > 0$. Then $\overline{p_j} \in H(\overline{x_i})^+$. We have $H(\overline{x_i}$ and $H(\overline{x_j})$ intersecting along the plane $\overline{P}$ whose vertical projection onto X is P. Since $\overline{p_i} \in H(\overline{x_i})$ and $\overline{p_j} \in H(\overline{x_j})$ this shows that $\overline{p_i} \in H(\overline{x_j})^-$ which implies $\Pi(x_j, p_i) < 0$.

A similar argument holds for the converse.

□

Theorem 4. *For p_i and p_j in the same halfspace, $wdd(t, p_i) = wdd(t, p_j) \iff \Pi(x_i, p_j) = 0$*

Proof. (Spherical)

Assume pi, pj are in the same halfspace and $wdd(t, p_i) = wdd(t, p_j)$. Then by definition of the weighted Delaunay distance we have,

$$x'_i = x'_j. \tag{12.2.1}$$

From the definition of the characteristic point we have

$$|x'_i - p'_m| - x''_i - w_m = 0 \text{ for } p_m \in t \tag{12.2.2}$$

$$|x'_j - p'_m| - x''_j - w_m = 0 \text{ for } p_m \in t. \tag{12.2.3}$$

This gives

$$x''_i = x''_j \tag{12.2.4}$$

and

$$\begin{aligned} \Pi(x_i, p_j) &= |x'_i - p'_j|^2 - x''_i - p''_j \\ &= |x'_j - p'_j|^2 - x''_j - p''_j = 0. \end{aligned} \tag{12.2.5}$$

Now assume that $\Pi(x_i, p_j) = 0$. Since $\Pi(x_i, p_m) = 0$ for all $p_m \in t$ this implies that x_i is the characteristic point of $T_j = \{t, p_j\}$ which gives $wdd(t, p_i) = wdd(t, p_j)$. □

Proof. (In $\mathbb{R}^4$)

Assume $wdd(t, p_i) = wdd(t, p_j)$ which gives $x_i = x_j$. We also have $\overline{p_j} \in H(\overline{x_j})$ which gives $\overline{p_j} \in H(\overline{x_i})$ and $\Pi(x_i, p_j) = 0$.

Now assume $\Pi(x_i, p_j) = 0$. Then $\overline{p_j} \in H(\overline{x_i})$. We also have $\overline{p} \in H(\overline{x_i})$ for $p \in t$. This gives $x_i = x_j$ and $wdd(t, p_i) = wdd(t, p_j)$.

□

Theorem 5. *For p_i and p_j in the same halfspace, $wdd(t, p_i) < wdd(t, p_j) \iff \Pi(x_i, p_j) > 0$ (See Figure 12.10).*

Proof. Assume that $p_i, p_j \in P^+$ and that $wdd(t, p_i) < wdd(t, p_j)$. There are three cases:

1. $x_i', x_j' \in P^+$

2. $x_i' \in P^-$ and $x_j' \in P^+$

3. $x_i', x_j' \in P^-$.

Case 1 and 2: (See Figures 12.7 and 12.8)

$wdd(t, p_i) < wdd(t, p_j)$ gives $r(\overline{x_j}) \in H(\overline{x_i})^+$. Since $H(\overline{x_i})$ and $H(\overline{x_j})$ intersect along $\overline{P}$ we have $\overline{x} \in H(\overline{x_i})^+$ for all $\overline{x} \in H(\overline{x_j})$ such that $x \in P^+$. This gives $\overline{p_j} \in H(\overline{x_i})^+$ which shows $\Pi(x_i, p_j) > 0$.

Case 3: (See Figure 12.9)

x_i' and $x_j' \in P^-$ with $wdd(t, p_i) < wdd(t, p_j)$ gives $r(\overline{x_j}) \in H(\overline{x_i})^-$. Since $H(\overline{x_i})$ and $H(\overline{x_j})$ intersect along $\overline{P}$ we have $\overline{x} \in H(\overline{x_i})^-$ for all $\overline{x} \in H(\overline{x_j})$ such that $x \in P^-$. Furthermore $\overline{x} \in H(\overline{x_i})^+$ for all $\overline{x} \in H(\overline{x_j})$ such that $\overline{x} \in P^+$. This gives $\overline{p_j} \in H(\overline{x_i})^+$ which shows $\Pi(x_i, p_j) > 0$.

The converse is shown in a similar manner.

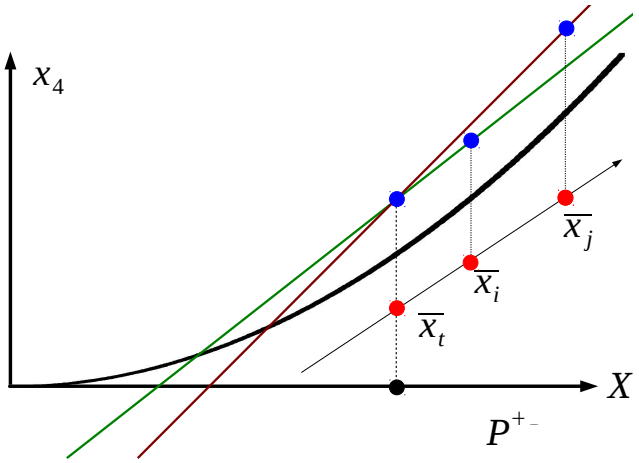


Figure 12.7: x_i and x_j are in P^+ with $\overline{x_i}$ and $\overline{x_j}$ on the line $\overline{l(t)}$ which is represented by the diagonal arrow. The planes $H(\overline{x_i})$ and $H(\overline{x_j})$ are represented by the green and red line segments, respectively. These hyperplanes intersect through the plane $\overline{P(t)}$ which projects vertically onto x_t which is shown by the black dot.

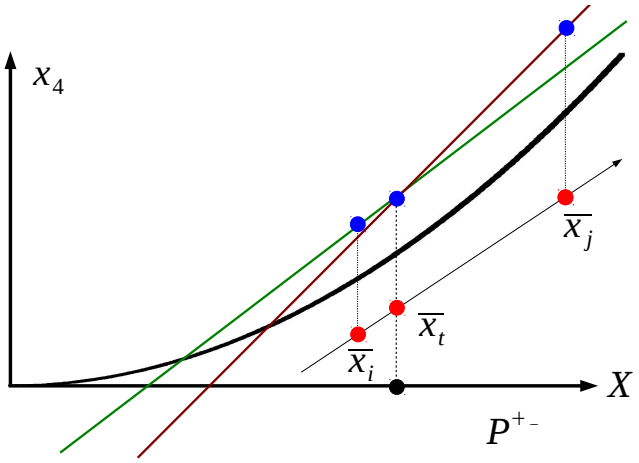


Figure 12.8: $x_i \in P^-$ and $x_j \in P^+$.

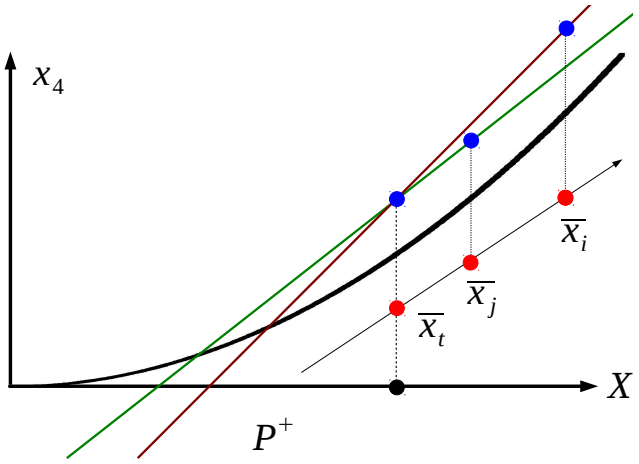


Figure 12.9: x_i and $x_j \in P^-$.

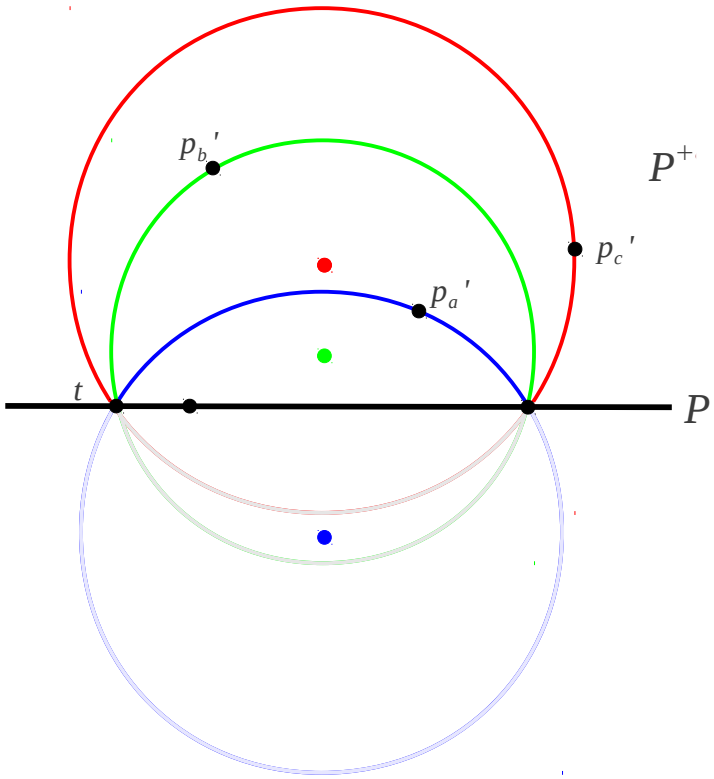


Figure 12.10: Unweighted illustration of Theorem 5. Given the characteristic point, x_a, represented by the blue circle, the point p'_a lies anywhere on the dark blue arc above the plane P. Similarly for x_b (green) and x_{t_c} (red). Here we see that $wdd(t, p_a) < wdd(t, p_b) < wdd(t, p_c)$. The points p'_b and p'_c are outside the sphere represented by x_a which gives $\Pi(x_a, p_{a,b}) > 0$. Furthermore, we see that sphere x_c contains p'_b and p'_a ($\Pi(x_c, p_{a,b}) \leq 0$), and the sphere x_b contains p'_a but not p'_c.

□

From theorem 1 we get the following corollary:

Corollary 6. *If* $p_a = nnbr\_wwd^+(t)$ *then* $\Pi(x_a, p) \geq 0$ *for all* $p \in P^+$.

We will now prove the Nearest Weighted Delaunay Distance Neighbor Theorem 1 .

Proof. Assume that t is a face in $\mathcal{T}$, and that there exist points in P^+. This means there also exists $nnbr\_wdd^+$ to t. By (1.6) there also exists a point $p_i \in P^+$, such that $\Pi(x_i, p) \geq 0$ for all $p \in P$ and $T_i \in \mathcal{T}$. Since $\Pi(x_j, nnbr\_wdd^+) \leq 0$ for all $p_j \neq p_i$, this implies that $p_i = nnbr\_wdd^+$. □

12.3 Simple Active-Face (AF) Algorithm

Each $t \in \mathcal{T}$ is incident to either one or two tetrahedra which we call $T^+(t)$ and $T^-(t)$ depending on the orientation of t. That is, t is incident to

$$T^+(t) = \{t, nnbr\_wdd^+(t)\} \tag{12.3.1}$$

$$T^-(t) = \{t, nnbr\_wdd^-(t)\} \tag{12.3.2}$$

assuming $nnbr\_wdd^+$ and $nnbr\_wdd^-$ exist. For $t \in conv(\mathcal{A})$, the facet has only one $nnbr\_wdd$ and incident tetrahedron in the traditional sense. One may place a symbolic point, p^∞, at inifinity, and set p^∞ as a $nnbr\_wdd$ for each $t \in conv(\mathcal{A})$. We call the resulting tetrahedron the "infinite tetrahedron", to which we assign the index of "0".

For $t \in \mathcal{T}$, its incident tetrahedra, $T(t) \in \mathcal{T}$, may be constructed if $nnbr\_wdd^+(t)$ and $nnbr\_wdd^-(t)$ are known. Adjacent tetrahedra of $T(t)$ may then be constructed by finding the nearest-weighted-Delaunay neighbors of appropriate facets, and so on. Thus, the entire tetrahedrization, $\mathcal{T}$, may be found given an initial triangle $t \in \mathcal{T}$ and the capability of finding nearest-weighted-Delaunay neighbors. Additionally, $\mathcal{T}$ may be found given an initial $t \in \mathcal{T}$ and the capability of finding $nnbr\_wdd^+(t)$.

The algorithm begins with an initial tetrahedra list, T_list, which contains the tetrahedra that have been found, and Active Face List (AFL) which contains the triangles whose T^- are in T_list but whose T^+ are not. A facet is called active if it is in the AFL, and a facet is called the active face (AF) if its nearest-weighted-Delaunay neighbor is being found and the resulting tetrahedron is being constructed. The algorithm proceeds as follows [5], [16]:

1. initial population of AFL, T_list
2. for $t =$ AF, locate $nnbr\_wdd(t)^+$ and construct $T^+(t)$
3. update AFL
4. repeat steps 2-3 until AFL is empty

The initial population of AFL and T_list, and the location and contruction of $T(t)$ are the same as the method used in the Active-Edge-Face method which is discussed in the next section. Here we discuss a possible method to update AFL and some of the difficulties that arise when using this method.

We say a triangle has been "visited" if it was or is in the AFL. We say that a vertex has been visited if it is incident to a visited triangle. At each step in the algorithm we keep a list of triangle indices which contain a given vertex.

Given the active face, t, $nnbr\_wdd^{+}(t)$ is located and the corresponding tetrahedron is extracted. We remove t from AFL. At this point there are two cases:

1. nnbr_dd has not been visited

2. nnbr_dd has been visited

If nnbr_dd has not been visited, we know that three incident triangles of the new tetrahedron have not yet been visited. We insert these into AFL.

If nnbr_dd has been visited, there is a possibility that one or all of the incident triangles have been visited. To determine if the triangles had been visited, we compare the given triangles to the list of triangles that contain nnbr_dd. If a match is found, the face is removed from AFL, otherwise it is added to AFL.

This method is slow, because a large number of triangles may contain a given vertex and three indices need to be compared at each step in the triangle matching process. The number of triangle compares may be decreased by modifying the order of the triangles in the AFL. For example, faces may be inserted so that tetrahedra are found along a certain search direction or "wall" similar to the method used in [5] However, we would like to bypass this step as well as customize the algorithm to specifically meet our needs as discussed in the next section.

12.4 Active-Edge-Face Algorithm

Our motivation in computing the Delaunay tetrahedriztion is that it is used to calculate molecular volumes, surface areas, their derivatives, as well as Laguerre and Laguerre-

intersection volumes and surfaces.

We would like to develop a tetrahedrization process that is tailored specifically for these calculations. These algorithms require the following information which is extracted from the Delaunay tetrahedrization:

1. Alpha complex
    a) Simplices in tetrahedrization
    b) Size of simplices
    c) Incident tetrahedra to each triangle
    d) Incident triangles to each edge
    e) Incident edges to each vertex
2. Ordered tetra-rings
3. Tetrahedra that contain each vertex

Ideally this information should be found with little or no additional work during the DT calculation. Furthermore, an algorithm with this property will be an excellent basis for an algorithm that computes the alpha complex from scratch (ideas for this will be discussed later). The Active-Edge-Face algorithm is such an algorithm.

The AEF method combines ideas from the Active-Face Algorithm as well as the Tetra-ring/Simplex Extraction Algorithm (Chapter 11).

Recall that with the Tetraring/Simplex Extraction Algorithm, an edge is extracted and then the tetraring is found by examining the Tnbr data structure for the tetrahedron that is adjacent to the current tetrahedron. Triangles are extracted as the tetraring is traversed. With the AEF algorithm, the next tetrahedron in the tetraring is found concurrently by a $nnbr\_dd$ search. At each step, we also extract simplices from the new tetrahedron and set incidences for both old and new simplices. This eliminates the need for triangle comparisons.

We say that an edge has a partial tetraring if the edge has been extracted, but all tetrahedra in the ring have not been found. Edges with partial tetrarings are in the

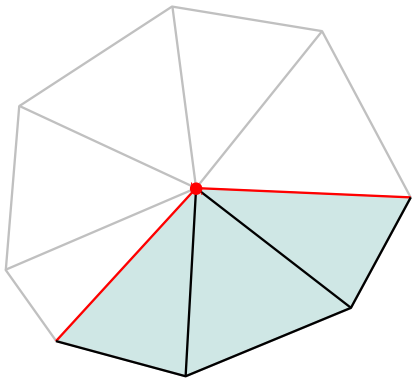


Figure 12.11: An edge (represented by the red dot) with a partial tetraring (represented by the blue triangles). The filled triangles represent tetrahedra that have been extracted and unfilled triangles represent tetrahedra that are yet to be found. The red line segments represent the open faces of the edge.

Active Edge List. Each edge in the Active Edge List contains information about its "open faces" which are the triangles on the edge of the partial tetraring (See Figure 12.11).

The algorithm begins with an initial tetrahedron and proceeds as follows:

1. Extract subsimplices of initial tetrahedron and set incidences and open faces.
2. For $e = active\_edge$ find $nnbr\_dd$ of open face. Extract tetrahedron T_new.
3. Extract subsimplices of T_new. Set incidences and update edges' open faces.
4. If T_new completes $active\_edge$'s tetraring, continue to next edge in Active Edge List. Otherwise repeat step three on new open face.
5. Repeat steps 2-4 until Active Edge List is empty.

12.4.1 Initial Tetrahedron

There are a number of ways an initial tetrahedron, T_init, can be found. Possibilities include:

1. T_init may be constructed via a nearest Delaunay neighbor search on

    a) a convex hull triangle

b) a triangle known *a priori* from properties of molecule. For example, three backbone atoms.

c) a triangle found incrementally from a nearest neighbor search for a Delaunay edge

2. a tetrahedron known *a priori* from the properties of the molecule

12.4.2 Calculation of wdd

The characteristic point, x_t, of the active face t is calculated along with the unit normal $\hat{\mathbf{n}}$. Recall that the Delaunay distance between a triangle with characteristic point x_t and a point p is

$$wdd(t,p) = \begin{cases} |x'_T, x'_t|, & \text{if } x'_T \in \text{Halfspace(t,p)} \\ -|x'_T, x'_t| & \text{otherwise} \end{cases} \tag{12.4.1}$$

where x_T is the characteristic point of $T = \{t, p\}$.

We may compute x_T by solving a 4×4 system of equations, or since we already know x_t and $\hat{\mathbf{n}}$ we may save on computation by writing

$$x'_T = x'_t + d\hat{\mathbf{n}} \tag{12.4.2}$$

and noticing that

$$x''_T = x''_t + d^2 \tag{12.4.3}$$

which gives

$$d = -\frac{x''_t + p'' - |x'_t - p'|^2}{2(x'_t - p') \cdot \hat{\mathbf{n}}} \tag{12.4.4}$$

.

This savings is critical to the time efficiency of the code, since $wdd(t,p)$ must be calculated for all test points.

12.4.3 Nearest neighbor search

At the beginning of the computation, the maximum weight over all atoms is subtracted from each atom's weight in the data set. This means that the radii of all characteristic points are real which makes the nearest neighbor search easier but does not change the triangulation.

Gridding

A box containing all atom centers is divided into bins. We write the atom centers, $x_1, ..., x_n$, in the vector $\mathbf{x}$ where $\mathbf{x}(i,j)$ is the ith element of x_j. We set the bounds on the box by

$$\begin{aligned} x\_min &= min(\mathbf{x}(1,:)) - \epsilon \\ x\_max &= max(\mathbf{x}(1,:)) + \epsilon \\ y\_min &= min(\mathbf{x}(2,:)) - \epsilon \\ y\_max &= max(\mathbf{x}(2,:)) + \epsilon \\ z\_min &= min(\mathbf{x}(3,:)) - \epsilon \\ z\_max &= max(\mathbf{x}(3,:)) + \epsilon. \end{aligned}$$

Then the x, y, and z dimensions of the box are,

$$\begin{aligned} lx &= x\_max - x\_min \\ ly &= y\_max - y\_min \\ lz &= z\_max - z\_min \end{aligned} \tag{12.4.5}$$

Let nx, ny, and nz be the number of bins let dx, dy, and dz be the dimensions of each of the bins in the x, y, and z directions, respectively, There are a couple of ways these values may be chosen.

Method 1:

We would like the total number of bins to equal the number of atoms in the data set, so on average one atom sits in each bin. We would also like the bins to be close to cubic.

$$nx \cdot ny \cdot nz \approx n \tag{12.4.6}$$

$$dx \approx dy \approx dz \tag{12.4.7}$$

We set

$$nx = \left\lceil \left(\frac{lx^2}{ly \cdot lz} n \right)^{\frac{1}{3}} \right\rceil \tag{12.4.8}$$

$$ny = \left\lceil \frac{ly}{lx} nx \right\rceil \tag{12.4.9}$$

$$nz = \left\lceil \frac{lz}{lx} nx \right\rceil \tag{12.4.10}$$

with

$$dx = \frac{lx}{nx} \tag{12.4.11}$$

$$dy = \frac{ly}{ny} \tag{12.4.12}$$

$$dz = \frac{lz}{nz} \tag{12.4.13}$$

$$\tag{12.4.14}$$

.

This gives

$$nx \cdot ny \cdot nz \approx nx \left(\frac{ly}{lx} nx \right) \left(\frac{lz}{lx} nx \right) \tag{12.4.15}$$

$$= (nx)^3 \left(\frac{ly \cdot lz}{lx^2} \right) \tag{12.4.16}$$

$$= n \tag{12.4.17}$$

and

$$dx \approx \left(\frac{lx \cdot ly \cdot lz}{n}\right)^{1/3} \tag{12.4.18}$$

$$dy \approx dx \tag{12.4.19}$$

$$dz \approx dx \tag{12.4.20}$$

Method 2

We may set dx, dy, and dz close to a desired value d. The value of 2 works well. Then

$$\begin{aligned} nx &= \left\lceil \frac{lx}{d} \right\rceil \\ ny &= \left\lceil \frac{ly}{d} \right\rceil \\ nz &= \left\lceil \frac{lz}{d} \right\rceil \end{aligned} \tag{12.4.21}$$

and

$$\begin{aligned} dx &= \frac{lx}{nx} \\ dy &= \frac{ly}{ny} \\ dz &= \frac{lz}{nz} \end{aligned} \tag{12.4.22}$$

The user may decide to use either grid method 1 or grid method 2 in the calculation. The faster version depends on properties of the molecule. In the future, the molecule may be rotated so the principal axes lie along the cartesian coordinates. This will reduce the number of empty bins in the calculation.

Each atom is assigned a bin number.

Grid Traversal

The speed of the search algorithm depends on a fine balance between searching the minimum number of cells, but not taking too much time to determine which these cells are.

First, the characteristic point, x_t, of the active face is computed, along with its containing plane P which has normal $\mathbf{n}$. The characteristic point gives the size of the triangle which is the square of the radius of the sphere that is orthogonal to all points in the triangle,

$$x_t'' = r_t^2 \tag{12.4.23}$$

Initial Search

We call the box of cells which contain the characteristic point of the open face the "initial box" and the traversal of appropriate cells in the initial box the "initial search".

We set the maximum and minimum grid indices for the initial search as:

$$\begin{aligned}
x_max_search &= \min\left(nx, \left\lceil \frac{x_t(1) + r_t - x_min}{dx} \right\rceil\right) \\
y_max_search &= \min\left(ny, \left\lceil \frac{x_t(2) + r_t - y_min}{dy} \right\rceil\right) \\
z_max_search &= \min\left(nz, \left\lceil \frac{x_t(3) + r_t - z_min}{dz} \right\rceil\right) \\
x_min_search &= \max\left(1, \left\lceil \frac{x_t(1) - r_t - x_min}{dx} \right\rceil\right) \\
y_min_search &= \max\left(1, \left\lceil \frac{x_t(2) - r_t - y_min}{dy} \right\rceil\right) \\
z_min_search &= \max\left(1, \left\lceil \frac{x_t(3) - r_t - z_min}{dz} \right\rceil\right)
\end{aligned} \tag{12.4.24}$$

We are looking for $nnbr\_wdd^+$ which we know is in P^+. This means that we need to search all cells in the initial box on the upper side of the plane. However, if the size

of the open face is "small" compared to the size of the grid is is faster to visit all cells in the initial box.

Both sides of plane:

We loop through the cells and visit each test point with the following pseudocode:

```
do i = xmin_search, xmax_search
do j = ymin_search, ymax_search
do k = zmin_search, zmax_search
       do l = 1, num_cell(i,j,k)
              test_pt=grid(i,j,k,l)
              if (test_pt in P^+)
                  calculate current_wdd
                  if (current_wdd<min_wdd)
                       set new nnbr
                  end if
              end if
       end do
end do
end do
end do
```

One side of plane: In this pseudocode, for each pair (i, j) we calculate z-index of the cell that intersects P to only search cells in P^+.

```
do i = xmin_search, xmax_search
   do j = ymin_search, ymax_seach
      z_int_corner =  !z-ind. of P with corners of bin (i,j)
      if (normal(3)<=0) then
         zstart=zmin_search
         zend=max(z_int_corner)
         if (zend>zmax_search)
            zend=zmax_search
         end if
      else !normal(3)>0
         zstart=minval(z_int_corner)
         if (zstart,zmin_search) then
```

```
            zstart=zmin_search
         end if
         zend=zmax_search
      end if
      do k = zstart, zend
      do l = 1, num_cell(i,j,k)
         test_pt=grid(i,j,k,l)
         if (test_pt in P^+) then
            calculate current_wdd
            if (current_wdd<min_wdd) then
               set new nnbr
            end if
         end if
      end do
      end do
   end do
end do
```

The right hand corner of one box may be the upcoming left hand corner of the next box. This means that all entries of z_int_corner do not need to be recomputed each loop.

```
!normal       =  unit normal to plane P
!xy_corner    =  x and y values at corners of cell
!z_corner     =  z values of plane at corners of cell
!z_int_corner =  z-indices of cells that contain z-corner

xr=xmin+(xmin_search-1)*dx
do i = xmin_search, xmax_search
   xy_corner(1,2)=xr
   xr=xmin+i*dx;
   xy_corner(1,1)=xr
   yr=ymin+(ymin_search-1)*dy
   xy_corner(2,1:2)=yr
   do mm = 3, 4
      z_corner=(dot(normal,p1)-dot(normal(1:2), &
                xy_corner(:,mm-2)))/normal(3);
      z_int_corner(mm)=ceiling((z_corner-zmin)/dz)
   end do !mm
   do j = ymin_search, ymax_search
      yr=ymin+j*dy;
      xy_corner(2,1:2)=yr
      z_int_corner(1:2)=z_int_corner(3:4)
      do mm = 3, 4
         z_corner=(dot(normal,p1)-dot(normal(1:2), &
                   xy_corner(:,mm-2)))/normal(3);
         z_int_corner(mm)=ceiling((z_corner-zmin)/dz)
```

```
      end do !mm

      if (normal(3)<=0) then
         zstart=zmin_search
         zend=max(z_int_corner)
         if (zend>zmax_search)
            zend=zmax_search
         end if
      else !normal(3)>0
         zstart=minval(z_int_corner)
         if (zstart,zmin_search) then
            zstart=zmin_search
         end if
         zend=zmax_search
       end if
       do k = zstart, zend
       do l = 1, num_cell(i,j,k)
           test_pt=grid(i,j,k,l)
           if (test_pt on correct side of plane)
              calculate current_wdd
              if (current_wdd<min_wdd)
                 set new nnbr
              end if
           end if
       end do
       end do
   end do
end do
```

We see that for each step in the loop, it takes time to determine zstart and zend. By counting the number of flops needed to compute zstart and zend and comparing it to the number of flops needed to check all cells in initial search box (assuming an average of one data point per cell), we see that it is faster to search all cells in the initial box for $r_t \leq 2.5dx$. However, in practice, r_t satisfies this condition for a small enough number of triangles that it is faster to search the entire initial box rather than comparing the value of r_t for each active face.

If $min\_wdd \leq 0$ then we know $x_T \bigcap P^+$ is contained in the initial box. This means we have located $nnbr\_wdd^+$ and we exit.

Searching Additional Layers: If the exit criterion has not been met after the initial search, we search successive caps of each wall of the previous box. For these additional layers, it is faster to only search cells in P^+.

```
Pseudocode:  Additional Layers
do while
if (xmin_search > 1)
   xmin_search = xmin_search-1
   x_searchvec(1)=xmin_search
else
   x_searchvec(1)=0
end if

if (xmax_search < nx)
   xmax_search=xmax_search+1
   x_searchvec(2)=xmax_search
else
   x_searchvec(2)=0
end if

.
.
.
.  similary for y and z dimensions


do i = x_searchvec
   if (i /= 0) then
      xl=xmin+(i-1)*dx xr=xmin+i*dx
      yr=ymin+(ymin_search-1)*dy
      xy_corner(:,1)=(/xl,yr/)
      xy_corner(:,2)=(/xr,yr/)
      do mm = 3,4
         z_corner=(dot(normal,p1)-dot(normal(1:2),&
                   xy_corner(:,mm-2)))/normal(3)
         z_int_corner(mm)=ceiling((z_corner-zmin)/dz)
      end do !mm
      do j = ymin_search, ymax_search
         yr=ymin+j*dy
         xy_corner(2,:)=yr
         z_int_corner(1:2)=z_int_corner(3:4)
         do mm=3,4
            z_corner=(dot(normal,p1)-dot(normal(1:2),&
                      xy_corner(:,mm-2)))/normal(3)
            z_int_corner(mm)=ceiling((z_corner-zmin)/dz)
         end do !mm

         if (normal(3) <= 0)  then
            zstart=zmin_search
            zend=maxval(z_int_corner)
            if (zend>zmax_search)  then
               zend=zmax_search
```

```
            end if
         else
            zstart=minval(z_int_corner)
            if (zstart < zmin_search) then
                zstart=zmin_search
            end if
            zend=zmax_search
         end if

         do k = zstart, zend
            num_cells_loop=num_cells_loop+1
            do l = 1, num_cell(i,j,k)
               test_pt=grid(i,j,k,l)
               if (test_pt on correct side of plane)
                  calculate current_wdd
                  if (current_wdd<min_wdd)
                     set new nnbr
                  end if
               end if
            end do
         end do
      end do
   end if
end do

.
.
. similary for y and z dimensions

if (exit criterion met) then
   return
end if

end do !while
```

We exit the do loop if one of the following criterion is met:

1. No cells were visited in the last loop
2. Search bounds are maxed out
3. Search box contains x_T

12.4.4 Extraction of Subsimplices and Update of Incidences

Before we discuss how subsimplices of T_new are extracted and incidences updated, we will first present the data structures used in the algorithm.

Data Stucture

Tetrahedra

The array T stores tetrahedra and is allocated to size $(4, max\_num\_T)$. Each column of T will contain the indices of the tetrahedron. That is tetrahedron i is represented by $T(:, i) = (i_1, i_2, i_3, i_4)$. The tetrahedron is oriented such that the point v_4 is above the triangle v_1, v_2, v_3.

The array $aedg$ is of dimension $(6, max\_num\_T)$ where $aedg(:, i)$ contains the indices of the edges which are incident to the tetrahedron i. This array is not needed for volume/surface/Laguerre calculations but is needed to find the indices of edges in the DT calculation.

The array $char\_pt\_T$ contains characteristic points of the tetrahedra.

Triangles

The tri data type has several components.

```
type  tri_type
int, dim(3) :: tri
int, dim(2) :: inc_T
int, dim(2) :: op_ver
double precision, dim(4) :: x_char
end type tri_type
```

The component $tri(i)\%tri$ contains indices of the vertices in the triangle i. The component $tri(i)\%x\_char$ contains the 4-dimensional characteristic point, $tri(i)\%inc\_T$ contains incident tetrahedra, and $tri(i)\%op\_ver$ contains the vertices on the incident tetrahedra that are not part of the triangle. The opposite vertex information is needed in the volume surface Laguerre calculation, and it is faster to store it at this point in the algorithmm rather than calculating it later on.

Edges

The edge data structure is the workhorse of the algorithm.

```
type edge_type
int, dim(2) :: edg
int :: num_inc_tri
int, dim(20) :: inc_tri
int, dim(20) :: op_ver
int, dim(-10:20,max_rings) :: tetra_ring
int, dim(2,max_rings) :: ind_of
int, dim(2,max_rings) :: open_T_ind !ind in T_ring of open tets
int, dim(2,max_rings) :: final_vertex !vert opposite edge in open face
int :: gap !=number of gaps = num_tetra rings -1
int, dim(max_rings) :: order_rings !order of the rings in tetra_ring
int, dim(max_rings) :: num_zero
double precision, dim(max_rings) :: dot
int :: next_ring
end type edge_type
```

The ordered tetrahedra that contain a given edge are stored in the edge's $tetra\_ring$. The first tetrahedron of edge i that is extracted is inserted into $edge(i)\%tetra\_ring(0)$. Tetrahedra are stored in a counterclockwise direction around the edge. This means that the tetrahedron adjacent to $tetra\_ring(0)$ in the counterclockwise direction around the edge is stored in $tetra\_ring(1)$ and the tetrahedron adjacent to $tetra\_ring(0)$ in the clockwise direction is stored in $tetra\_ring(-1)$ and so forth.

Occasionally a newly extracted tetrahedron that contains a given edge is not adjacent to any tetrahedra that are currently in the edge's $tetra\_ring$. In this case we say that a "gap" has formed. We temporarily store this tetrahedron, which is in a currently unconnected segment of the tetra ring, in $tetra\_ring(:, j)$ for some $j > 1$. At the end of the algorithm, when all tetrahedra have been extacted, $tetra\_ring(:, 1)$ contains the entire tetra ring.

Twenty is a good bound for the maximum number of tetrahedra in a tetra ring when working with molecular data sets. The first tetrahedron that contains an edge is inserted in $tetra\_ring(0, 1)$. Five has shown to be a good value for max_num_rings. Since counterclockwise is the default direction of extraction, we set the indices of $tetra\_ring(:, j)$ from $-10 : 20$. We set the size of the array inc_tri, which contains indices of incident triangles, and the size of the array op_ver, which contains the vertex

opposite to the edge in the incident triangle, to twenty. The opposite vertex is needed for the volume surface Laguerre calculations and it is faster to store it here than to recompute it.

In the end, $edg(i)\%tetra\_ring(:,1)$, will contain all the tetrahedra in edge i's tetra ring. However, during the calculation, there may be gaps in the tetra ring. Connected segments are stored in $edg(i)\%tetra\_ring(:,j)$. The component $edg\%open\_T\_ind(:,j)$ contains the indices of the first and last tetrahedra in the tetra ring j. The array ind_of stores the indices of the open faces in the partially completed tetra rings. The component $edg(i)\%final\_vertex(:,j)$ contains the indices of the vertices opposite the edge in the open face of the tetra ring j.

The component $edg(i)\%gap$ is the number of gaps in the tetra ring. This is equal to one less that the number of temporary rings. num_zero counts how many "0" tetrahedra are in the tetra ring. When this number is equal to 2, the edge is finished, i.e. the tetra ring is complete. $order\_rings$ is counter clockwise order of the partial rings around the edge, dot is related to the dot product between rings and is used to set $order\_rings$. $next\_ring$ stores the index of the next avaible tetra ring. $done = 1$ is the tetra ring of that edge is complete and 0 otherwise.

12.4.5 Outline

The tetra rings of the active edges are constructed and at each step, new simplices are extracted and all incidences (including the tetra ring of each subsidiary edge) are updated. As the algorithm proceeds, tetra rings of subsidiary edges may develop gaps.

We begin with an edge which we call the "active edge". An unfinished edge has at most two open faces. We begin, if possible, with the open face that is counterclockwise around the tetra ring. This becomes the "active face", so at each step we have an active edge-tri pair.

For this active edge, the nnbr_wdd is found. Then there are four cases:

1. nnbr_wdd is "new"; i.e. it has not been visited before

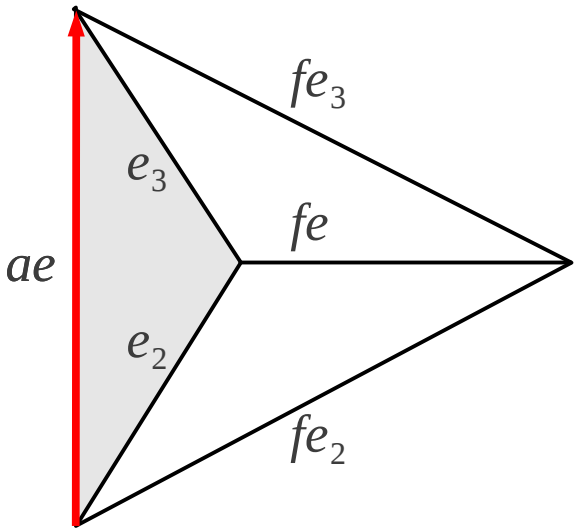


Figure 12.12: The active edge is represented by the red arrow and the active triangle is represented by the gray region. The "opposite vertex" is the point in the active triangle which is not in the active edge. The nnbr is the vertex to the far right. Edge2 is given by the first vertex of the active edge and the opposite vertex. Edge3 is given by the second vertex of the active edge and the opposite vertex. Faredge2 is the first vertex and nnbr, faredge3 is the second vertex and nnbr. Faredg consists of opposite vertex and nnbr. Tri2 is given by the first vertex, opposite vertex, and nnbr. Tri3 is the second vertex, opposite vertex, and nnbr. Far tri is given by the active edge and the nnbr.

2. nnbr_wdd=0; i.e. the active tri is on the convex hull

3. nnbr_wdd=final_vertex; i.e. the tetra ring can be closed

4. nnbr_wdd is not new and not equal to the final vertex

See Figure 12.12 to see how simplices in the new tetrahedron are labeled.

nnbr_wdd is new

If $nnbr\_wdd$ is new then we know that all far_edg2, far_edg3, and far_edg need to be extracted as well as the triangles $tri2$, $tri3$, and far_tri. Incidences are set at this time. The pair $active\_edge/far\_tri$ become the new active edge/face pair.

nnbr_wdd = 0

Incidences are set for $active\_edge$, $active\_tri$, $edge2$, and $edge3$. If the previous search was in the counterclockwise direction, the search proceeds in the clockwise direction with the clockwise open face as the new active_tri.

nnbr_wdd = far_vertex

This means that a tetra ring is completed or a gap is filled. If the tetra ring is completed, the edge is complete and we move on to another active edge. If a gap is filled, we keep the same active edge and move on to the far face of the appropriate partial tetra ring.

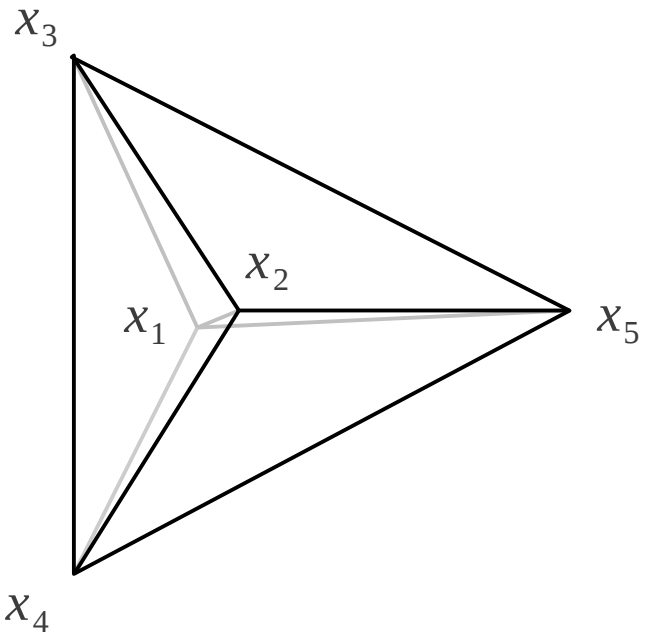
Since fartri has already been extracted, faredge2 and faredge3 have already been extracted.

nnbr_wdd is not new and is not a far vertex

In this case incidences are updated and the new active tri is the far tri.

12.4.6 Examples

Case 1:



We start with the tetrahedron $T_1 = (1, 2, 3, 4)$. As always, the first tetrahedron is oriented so that the vertex v_4 is above the triangle $(v1, v2, v3)$.

We first extract all simplices and set all incidences on this initial tetrahedron.

Tetra #	Vertices
1	1 2 3 4

Tri #	Indices	inc_T
1	1 3 2	1 x
2	1 2 4	1 x
3	1 4 3	1 x
4	2 3 4	1 x

Edge #	Indices	tetra_ring	open_face	final_vertex
1	1 2	1	1 2	3 4
2	1 3	1	3 1	4 2
3	1 4	1	2 3	2 3
4	2 3	1	1 4	1 4
5	2 4	1	4 2	3 1
6	3 4	1	3 4	1 2

Vertices 1 through 4 are marked as visited.

Note that the triangles are oriented so that each points away from the initial tetrahedron. Also note that the first open face of an edge is oriented toward the tetrahedron and the second open face is oriented away from the tetrahedron. This will give a tetraring that is oriented counterclockwise around the edge.

We begin with the first edge = (1,2) as the active edge with open_face(2) as the active tri. The size of the tri is calculated here, and the nearest neighbor x_5 is located. This vertex has not previously been visited, so we know to extract all far_edge, far_edge2, and far_edge3, tri2, tri3, and far_tri. We set incidences for these simplices as well as for the active edge, active tri, edge2, and edge3.

Tetra #	Vertices
1	1 2 3 4

Tri #	Indices	inc_T
1	1 3 2	1 x
2	1 2 4	1 2
3	1 4 3	1 x
4	2 3 4	1 x
5	1 2 5	2 x
6	1 5 4	2 x
7	2 4 5	2 x

Edge #	Indices	tetra_ring	open_face	final_vertex
1	1 2	1 2	1 5	3 5
2	1 3	1	3 1	4 2
3	1 4	2 1	6 3	5 3
4	2 3	1	1 4	1 4
5	2 4	1 2	4 7	3 5
6	3 4	1	3 4	1 2
7	1 5	2	5 6	2 4
8	2 5	2	7 5	4 1
9	4 5	2	7 6	1 2

The triangle 1 2 5 becomes the next active tri. A nearest neighbor search gives x_3. We extract the tetrahedron (1,2,5,3). Since x_3 is the final vertex of the current edge, and the edge has no gaps, the tetra ring is complete.

We know that far_tri, far_edge2, and far_edge3 have already been extracted. We need to update the incidences and tetra ring info for these simplices.

We do not know, by default, whether tri2, tri3, and far_edge have been extracted. It is easy to determine whether the triangles are new by looking at the final vertices of far_edge2 and far_edge3. If either tri2 or tri3 have already been extracted, then we know that far_edg has already been extracted. If neither tri2 nor tri3 have been extracted, then

we need to check the incidences of op_ver and nnbr to determine if far_edge is new. If the edge has already been extracted, then far_edge developed a gap in its tetra ring.

In this case, tri2, tri3, and far_edge are all new. We extract these.

Tetra #	Vertices
1	1 2 3 4
2	1 2 4 5
3	1 2 5 3

Tri #	Indices	inc_T
1	1 3 2	1 3
2	1 2 4	1 2
3	1 4 3	1 x
4	2 3 4	1 x
5	1 2 5	2 3
6	1 5 4	2
7	2 4 5	2 x
8	1 3 5	3 x
9	2 5 3	3 x

Edge #	Indices	tetra_ring	open_face	final_vertex
1	1 2	1 2 3	. .	. .
2	1 3	1 3	3 8	4 5
3	1 4	2 1	6 3	5 3
4	2 3	3 1	9 4	5 4
5	2 4	1 2	4 7	3 5
6	3 4	1	3 4	1 2
7	1 5	3 2	8 6	3 4
8	2 5	2 3	7 9	4 3
9	4 5	2	7 6	2 1
10	5 3	3	8 9	1 2

We move on to the next edge, (1,3), which becomes the active edge. The current tri is open_face(2) = tri 8 = (1 3 5). A nearest neighbor search returns the "0" vertex. This means that the triangle (1,3,5) is on the convex hull of the data set. We update incidences and the tetra rings for edges (1,3) (1,5), (3,5).

Tetra #	Vertices
1	1 2 3 4
2	1 2 4 5
3	1 2 5 3

Tri #	Indices	inc_T
1	1 3 2	1 3
2	1 2 4	1 2
3	1 4 3	1 x
4	2 3 4	1 x
5	1 2 5	2 3
6	1 5 4	2 x
7	2 4 5	2 x
8	1 3 5	3 0
9	2 5 3	3 x

Edge #	Indices	tetra_ring	open_face	final_vertex
1	1 2	1 2 3	. .	. .
2	1 3	1 3 0	3 .	4 .
3	1 4	2 1	6 3	5 3
4	2 3	3 1	9 4	5 4
5	2 4	1 2	4 7	3 5
6	3 4	1	3 4	1 2
7	1 5	0 3 2	. 6	. 4
8	2 5	2 3	7 9	4 3
9	4 5	2	7 6	2 1
10	5 3	0 3	. 9	. 2

Since we hit the convex hull, we continue the nearest neighbor search in the clockwise direction. open_face(1)= (1,4,3) becomes the active tri. Again a nearest neighbor search returns the "0" vertex. Since this is the second 0 for the active edge, this edge has been completed. We update incidences and tetra rings.

Tetra #	Vertices
1	1 2 3 4
2	1 2 4 5
3	1 2 5 3

Tri #	Indices	inc_T
1	1 3 2	1 3
2	1 2 4	1 2
3	1 4 3	1 0
4	2 3 4	1 x
5	1 2 5	2 3
6	1 5 4	2 x
7	2 4 5	2 x
8	1 3 5	3 0
9	2 5 3	3 x

Edge #	Indices	tetra_ring	open_face	final_vertex
1	1 2	1 2 3	. .	. .
2	1 3	0 1 3 0	. .	. .
3	1 4	2 1 0	6 .	5 .
4	2 3	3 1	9 4	5 4
5	2 4	1 2	4 7	3 5
6	3 4	1 0	. 4	. 2
7	1 5	0 3 2	. 6	. 4
8	2 5	2 3 0	7 .	4 .
9	4 5	2	7 6	2 1
10	5 3	0 3	. 9	. 2

Next we move on to edge 3 = (1,4). Since the final tetrahedron in the counterclockwise direction is "0", we begin by searching in the clockwise direction. The active tri is open_face(1)= tri # 6 = (1 5 4). A nearest neighbor search again gives the "0" vertex,

and edge 3 is complete. Incidences and tetra rings are updated. We see that this step completes the tetra ring of edge # 7 = (1,5).

Tetra #	Vertices
1	1 2 3 4
2	1 2 4 5
3	1 2 5 3

Tri #	Indices	inc_T
1	1 3 2	1 3
2	1 2 4	1 2
3	1 4 3	1 0
4	2 3 4	1 x
5	1 2 5	2 3
6	1 5 4	2 0
7	2 4 5	2 x
8	1 3 5	3 0
9	2 5 3	3 x

Edge #	Indices	tetra_ring	open_face	final_vertex
1	1 2	1 2 3	. .	. .
2	1 3	0 1 3 0	. .	. .
3	1 4	0 2 1 0	. .	. .
4	2 3	3 1	9 4	5 4
5	2 4	1 2	4 7	3 5
6	3 4	0 1	. 4	. 2
7	1 5	0 3 2 0	. .	. .
8	2 5	2 3 0	7 .	4 .
9	4 5	0 2	7 .	2 .
10	5 3	0 3	. 9	. 2

We continue through the edge list until all edges are finished.

The elements open_face and final_vertex are temporary variables, and the final result is

Tetra #	Vertices
1	1 2 3 4
2	1 2 4 5
3	1 2 5 3

Tri #	Indices	inc_T
1	1 3 2	1 3
2	1 2 4	1 2
3	1 4 3	1 0
4	2 3 4	1 0
5	1 2 5	2 3
6	1 5 4	2 0
7	2 4 5	2 0
8	1 3 5	3 0
9	2 5 3	3 0

Edge #	Indices	tetra_ring
1	1 2	1 2 3
2	1 3	0 1 3 0
3	1 4	0 2 1 0
4	2 3	0 3 1 0
5	2 4	0 1 2 0
6	3 4	0 1 0
7	1 5	0 3 2 0
8	2 5	0 2 3 0
9	4 5	0 2 0
10	5 3	0 3 0

12.5 Alpha Complex from Scratch

Eventually we would like to compute the alpha complex from scratch; i.e. compute the alpha complex without computing the entire Delaunay tetrahedrization. For a typical size of alpha, the alpha complex is a fraction of the Delaunay tetrahdrization. Also, only simplices less than a certain size are in the alpha complex. This means that the search for the nearest weighted Delaunay distance neighbor only needs to be within a certain cut-off distance.

12.6 Run Times

We compare run times programs 'regtet' [2], 'DAlphaBall' [10], and 'mkDT'. Times were calculated by taking the average over 50 runs. The regtet method uses a the 'regtet' implementation for the tetrahedrization and program we wrote called 'mkalpha' for the alpha complex and surface and volume calculations. The surface and volume calculations in 'mkalpha' are identical to those in 'mkDT'. First we compare the computation of the tetrahedrization and the alpha complex in Figure 12.13. Next we compare the surface and volume calculation times in Figure 12.14 and the total calculation times in Figure 12.15. Note that the run times are linear with respect to the number of data points.

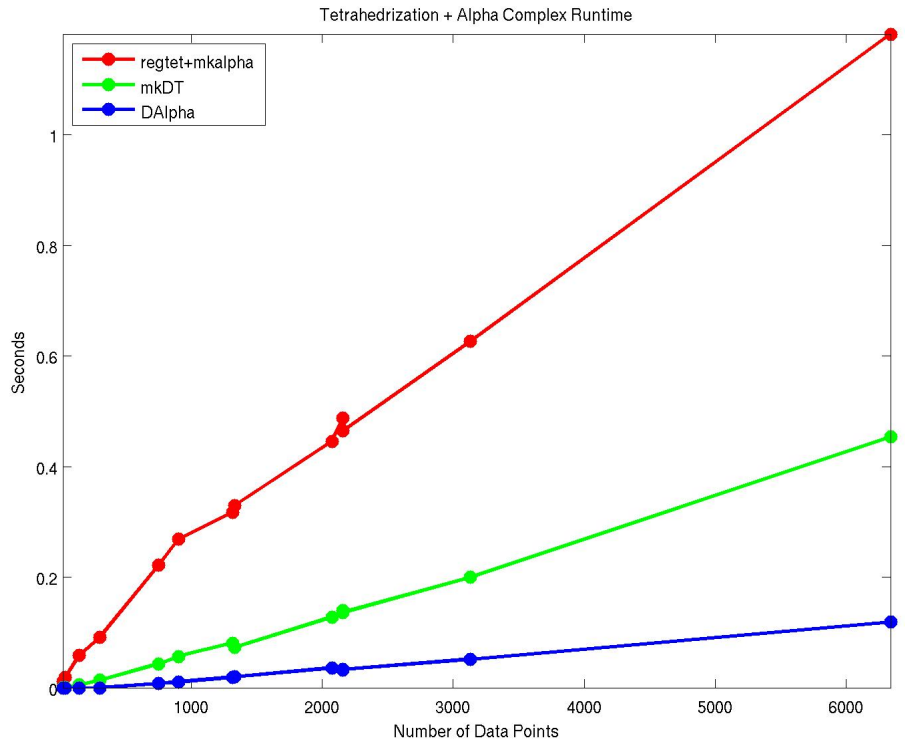


Figure 12.13: Run time for the calculation of Delaunay tetrahedrization and alpha complex.

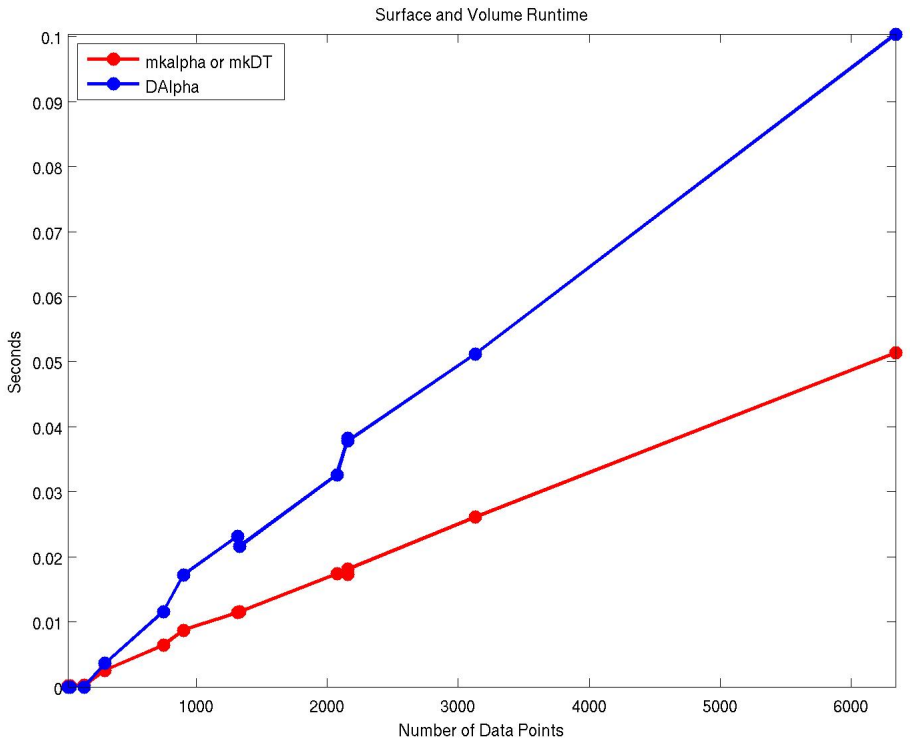


Figure 12.14: Run time for the calculation of surfaces, volumes, and derivatives.

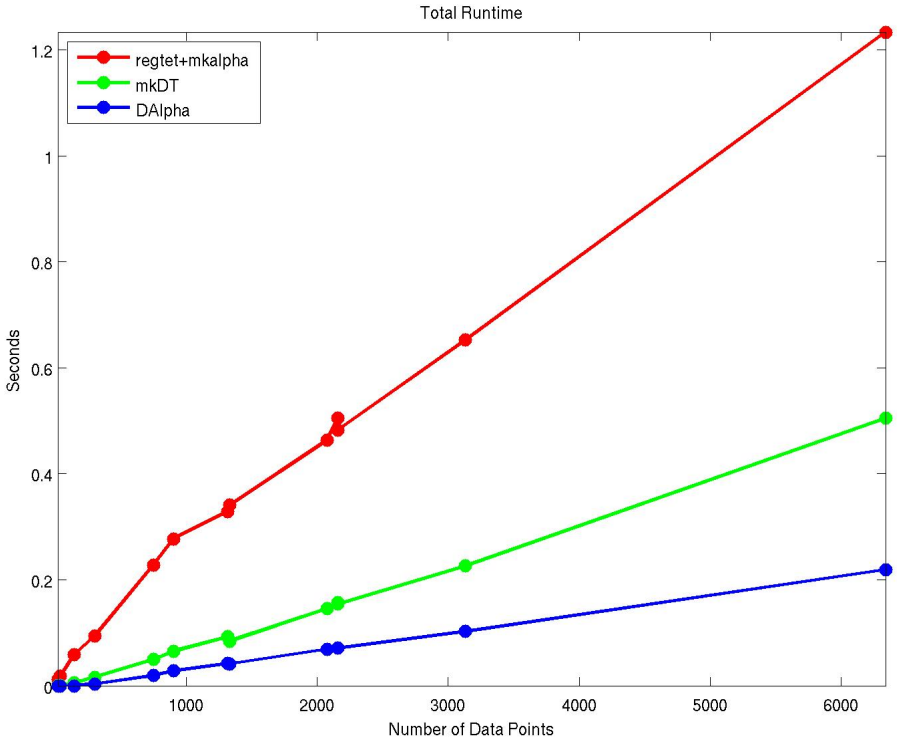


Figure 12.15: Total run time.

Chapter 13

Delaunay Spanning Tree

13.1 Weighted Delaunay Tetrahedrization and Convex Hull

Consider a weighted point $p = (p', p'') \in \mathbb{R}^3 \times \mathbb{R}$ where p' is the location of the point and p'' is the weight of the point. The point p can be thought of as a sphere, B_p in $\mathbb{R}^3$ centered at p with radius $\sqrt{p''}$. In the succeeding sections the sphere p and the weighted point p will be used interchangeably.

Call $\overline{p} = (p', |p'|^2 - p'') \in \mathbb{R}^4$ the "raised point p" and $\phi(\mathcal{A})$ the set of raised points in $\mathcal{A}$. The weighted Delaunay tetrahedrization, $\mathcal{T}$, of $\mathcal{A}$ can be found by computing the convex hull of $\phi(\mathcal{A})$ and projecting the lower facets of $conv(\phi(\mathcal{A}))$ onto $\mathbb{R}^3$.

13.2 Properties of the Weighted Delaunay Tetrahedrization

Consider a set of weighted points $\mathcal{A}$. An edge between points a and b in $\mathcal{A}$ will be written as $\overline{ab}$.

Lemma 1. *The edge $\overline{ab} \in \mathcal{T}(\mathcal{A})$ iff there exists a sphere x mutually orthogonal to a and b with $\Pi(c, x) \geq 0$ for all $c \in \mathcal{A}$.*

Proof. If $\overline{ab}$ is in $\mathcal{T}(\mathcal{A})$ then by the empty sphere property of the Delaunay tetrahedrization there exists an x such that $\Pi(x, c) \geq 0$ for all $c \in \mathcal{A}$.

Now assume there exists a sphere x which is orthogonal to a and b with $\Pi(c, x) \geq 0$ for all $c \in \mathcal{A}$. Define the hyperplane H to be the dual of the lifted point $\phi(x)$. Since $\Pi(x, c) \geq 0$ for all $c \in \mathcal{A}$ the point $\phi(c)$ is on or above the hyperplane H. This means that the hyperplane H supports the convex hull of $\phi(\mathcal{A})$. Since the edge $\overline{\phi(a)\phi(b)}$ is in H the edge $\overline{ab}$ is in $\mathcal{T}(\mathcal{A})$.

□

The equivalent of Lemma 1 holds for any k-simplex with $k \leq 3$.

Edge Theorem. *If $\mathcal{A}$ represents a protein, then the weighted Delaunay triangulation of $\mathcal{A}$ contains the edges between bonded atoms.*

Proof. By Lemma 1, we need to show that if a and b are bonded atoms then for all $c \in \mathcal{A}$, $\Pi(x, c) \geq 0$ where x is the characteristic point of the edge $\overline{ab}$.

The atoms carbon, hydrogen, oxygen, nitrogen, and sulfur comprise amino acids. Following is a table of the atoms and their van der Waals' radii as well as bond lengths between pairs of atoms. The weight of an atom is the van der Waals radius squared.

Atom	Van der Waals radius
C	1.7
H	1.09-1.20
N	1.55
O	1.52
S	1.8

Bond	Bond length
C-C	1.54
C=C	1.34
C-H	1.09-1.11
C-O	1.43
C=O	1.20-1.23
C-N	1.43-1.47
C=N	1.29-1.38
C-S	1.82
N-H	1.01-1.45
O-H	.96

In the case where c is not bonded to a, and c is not bonded to b we have $\Pi(x,c) \geq 0$ by van der Waal's restraints. That is

$$\begin{aligned} \Pi(x,c) &= |x' - c'|^2 - x'' - c'' \\ &> |x' - c'|^2 - c'' \\ &\geq 0 \end{aligned} \tag{13.2.1}$$

since $x'' < 0$ and x' is contained in the union of a and b.

Next we consider the case when c is bonded to a or b. Without loss of generality, we assume that c is bonded to a which we call the *central atom*. Recall that

$$x' = a' + t(b' - a') \tag{13.2.2}$$

where

$$t = \frac{1}{2}\left(\frac{a'' - b''}{|a' - b'|^2} + 1\right). \tag{13.2.3}$$

.

First we consider the subcase when neither a, b nor c is hydrogen. We have

$$t \geq \frac{1}{2}\left(\frac{1.52^2 - 1.8^2}{1.82^2} + 1\right) > 0 \tag{13.2.4}$$

and

$$t \leq \frac{1}{2}\left(\frac{1.8^2 - 1.52^2}{1.2^2} + 1\right) < 1 \tag{13.2.5}$$

which gives x' on the line segment between a' and b'. This along with the fact that the bond angle between c, a, and b is greater than 90 degrees gives

$$|x' - c'|^2 > |a' - c'|^2 + |a' - x'|^2. \tag{13.2.6}$$

Then

$$\begin{aligned}
\Pi(x,c) &= |x'-c'|^2 - x'' - c'' \\
&> |a'-c'|^2 + |a'-x'|^2 - x'' - c'' \\
&= |a'-c'|^2 + x'' + a'' - x'' - c'' \\
&= |a'-c'|^2 + a'' - c'' \\
&\geq 1.2^2 + 1.52^2 - 1.8^2 \\
&> 0. \qquad (13.2.7)
\end{aligned}$$

Next we consider the subcase that involves hydrogen. We have

$$t \geq \frac{1}{2}\left(\frac{1.09^2 - 1.8^2}{1.82^2} + 1\right) > 0. \qquad (13.2.8)$$

This along with the fact that hydrogen cannot be the central atom and that the bond angle is greater than 90 degrees gives

$$\Pi(x,c) > |a'-c'|^2 + a'' - c'' \qquad (13.2.9)$$

as before. For combinations not involving an O-H bond we have

$$\begin{aligned}
\Pi(x,c) &> |a'-c'|^2 + a'' - c'' \\
&\geq 1.09^2 + 1.52^2 - 1.8^2 \\
&> 0. \qquad (13.2.10)
\end{aligned}$$

The remaining combination is C-O-H. In this situation

$$\begin{aligned}
\Pi(x,c) &> |a'-c'|^2 + a'' - c'' \\
&\geq .96^2 + 1.52^2 - 1.7^2 \\
&> 0. \qquad (13.2.11)
\end{aligned}$$

□

Spanning Tree Theorem. *The set of edges between bonded atoms is a weighted Delaunay spanning tree*

Proof. This comes from the Edge Theorem. □

Part VI

Collective Motions of a Molecular Dynamics Trajectory via Strain Tensors

Chapter 14

Strain Tensor

In this chapter we define the strain tensor, ϵ, and calculate the strain on an HIV Protease trajectory.

14.1 Definitions

Consider two points close in space,

$$\mathbf{x} \tag{14.1.1}$$

and

$$\mathbf{x} + d\mathbf{x} \tag{14.1.2}$$

which move to the positions

$$\mathbf{x} + \mathbf{u}(\mathbf{x}) \tag{14.1.3}$$

and

$$\mathbf{x} + d\mathbf{x} + \mathbf{u}(\mathbf{x}) + \mathbf{D}d\mathbf{x}. \tag{14.1.4}$$

We call $\mathbf{D}$ the displacement gradient, and define the strain tensor

$$\epsilon = \frac{1}{2}(\mathbf{D} + \mathbf{D}^t + \mathbf{D}^t\mathbf{D}), \tag{14.1.5}$$

which is invariant to overall translations and rotations of the body. If $\mathbf{D}$ is small then strain tensor can be approximated by

$$\epsilon = \frac{1}{2}(\mathbf{D} + \mathbf{D}^t) \tag{14.1.6}$$

which we call the linear strain tensor. The trace of ϵ is the fractional volume change.

The displacement gradient, $\mathbf{D}$, may be approximated by a calculation on tetrahedra [41]. Consider four atoms in close space, $\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$, which are deformed to $\mathbf{a'}_0, \mathbf{a'}_1, \mathbf{a'}_2, \mathbf{a'}_3$. Then for $i = 1, 2, 3$, $\mathbf{D}$ satisfies

$$(\mathbf{a}'_i - \mathbf{a}_i) - (\mathbf{a}'_0 - \mathbf{a}_0) = \mathbf{D}(\mathbf{a}_i - \mathbf{a}_0) \tag{14.1.7}$$

which gives

$$\mathbf{D} = \mathbf{A'}\mathbf{A}^{-1} - \mathbf{I} \tag{14.1.8}$$

where $\mathbf{A} = (\mathbf{a}_1 - \mathbf{a}_0, \mathbf{a}_2 - \mathbf{a}_0, \mathbf{a}_3 - \mathbf{a}_0)$, and $\mathbf{A'} = (\mathbf{a'}_1 - \mathbf{a'}_0, \mathbf{a'}_2 - \mathbf{a'}_0, \mathbf{a'}_3 - \mathbf{a'}_0)$.

14.2 Algorithm

We have a molecular dynamics trajectory of 4500 structures. For structure i we compute the regular tetrahedrization $\mathcal{T}$. We filter tetrahedra by edge length and solid angle and call the set of remaining tetrahedra $\mathcal{T}'$. At times, we add a subscript $\mathcal{T'}_i$ to indicate which structure $\mathcal{T}'$ was derived from. The values for A are taken from structure i and the values of A' are taken from structure $i + 1$. The first step consists of computing the strain for each atom in each structure. For atom j in structure i define

$$\mathcal{T'}_{i,j} = \{T \in \mathcal{T}_i \text{ such that } p_j \in T\}. \tag{14.2.1}$$

with

$$V_{i,j} = \sum_{T \in \mathcal{T'}_{i,j}} V_T \tag{14.2.2}$$

where V_T is the volume of tetrahedron T.

Then the strain on atom j in structure i is approximated by

$$||\epsilon(\hat{i,j})|| = \frac{1}{V_{i,j}} \sum_{T \in \mathcal{T'}_{i,j}} ||\epsilon(T)|| V_T \tag{14.2.3}$$

where $||\epsilon(T)||$ is a norm of the strain tensor derived from T.

In the second step, we compute a weighted average of $||\epsilon(\hat{i,k})||$ over all neighbors p_k of p_j. Let d be a cutoff distance. The weight is a hat function

$$w_{j,k} = \begin{cases} 1 - \frac{1}{d}|p'_j - p'_k|, \text{ if } |p'_j - p'_k| < d \\ 0, \text{ otherwise.} \end{cases} \tag{14.2.4}$$

$$||\epsilon(i,j)|| = \frac{1}{W} \sum_{k \neq j} ||\hat{\epsilon}(i,k)|| w_{j,k} \tag{14.2.5}$$

where

$$W = \sum_{k \neq j} w_{j,k}. \tag{14.2.6}$$

For step three, we take the mean strain over all structures.

$$||\epsilon(j)|| = \frac{1}{N} \sum_{i=1}^{N} ||\epsilon(i,j)||. \tag{14.2.7}$$

We calculate both the trace

$$||\epsilon||_{tr} = trace(|\epsilon|) \tag{14.2.8}$$

and the Frobenius norm,

$$||\epsilon||_F = \sqrt{trace(\epsilon^* \epsilon)} \tag{14.2.9}$$

in this application.

We carry out the calculation on all atoms and backbone atoms only. The all-atom to backbone figures show the strain for backbone atoms only that was calculated using the all-atom method.

14.3 Application to HIV Protease Molecular Dynamics Trajectory

We found that 7.5 Angstroms was a good value for the cut-off distance d. We kept only tetrahedra whose solid angles were between $2\pi + .001$ and $2\pi(1 - .001)$.

The result of the nonlinear Frobenius all-atom method is shown in Figure 14.1. We see higher strain on solvent exposed residues, especially lysine. We compare results from the nonlinear Frobenius all-atom to backbone and backbone only methods in Figure 14.2. We see that the backbone only method captures strain at the flaps of the HIV protease as well as at the ends of the chains.

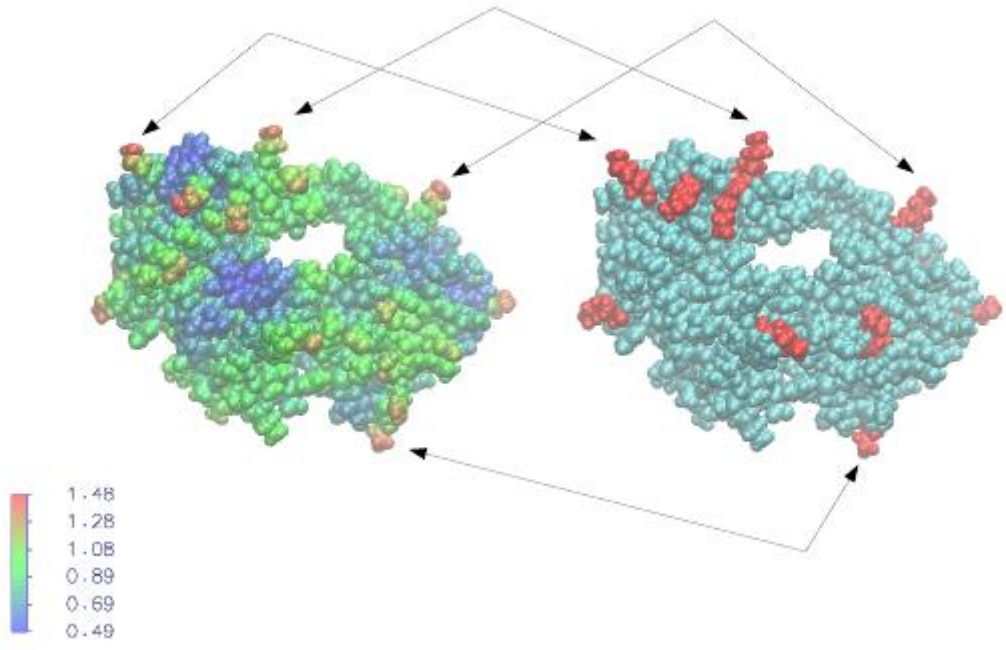
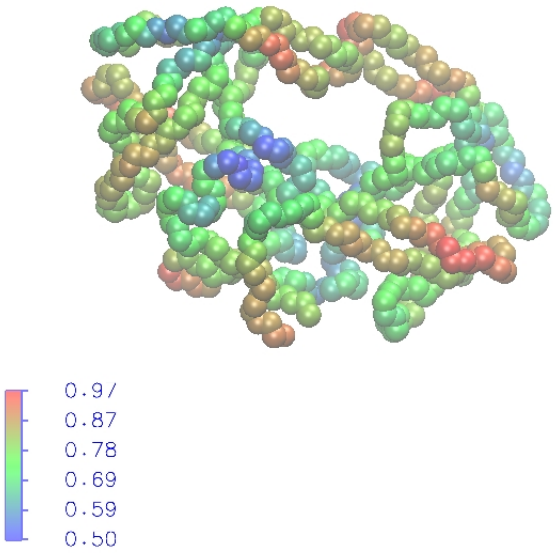


Figure 14.1: Left: Strain as calculated by the nonlinear all-atom Frobenius method. Right: Lysine residues are shown in red.

Results for the backbone calculation, all-atom calculation, and all-atom to backbone calculation are shown in Figures 14.3, 14.4, and 14.5, respectively. From the all-atom and all-atom to backbone methods (Figures 14.4 and 14.5), we see large differences between the Linear Trace and Frobenius norms which indicates shear strain acting on the molecule. Note that this difference is not seen in the backbone only calculation (Figure

14.3). We also compare the results between the backbone and all-atom to backbone calculation in Figures 14.6 and 14.7. It is interesting to note the similarity between the strain and the standard deviation of residual Laguerre volumes in Figure 9.12.



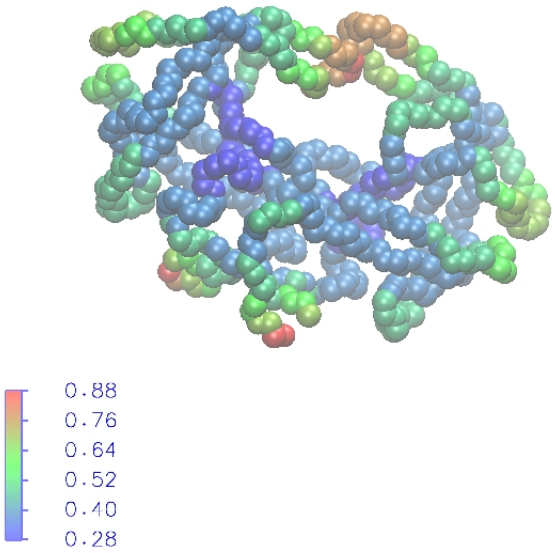


Figure 14.2: Strain calculated via the nonlinear Frobenius method. a) all atom to backbone, b) backbone only.

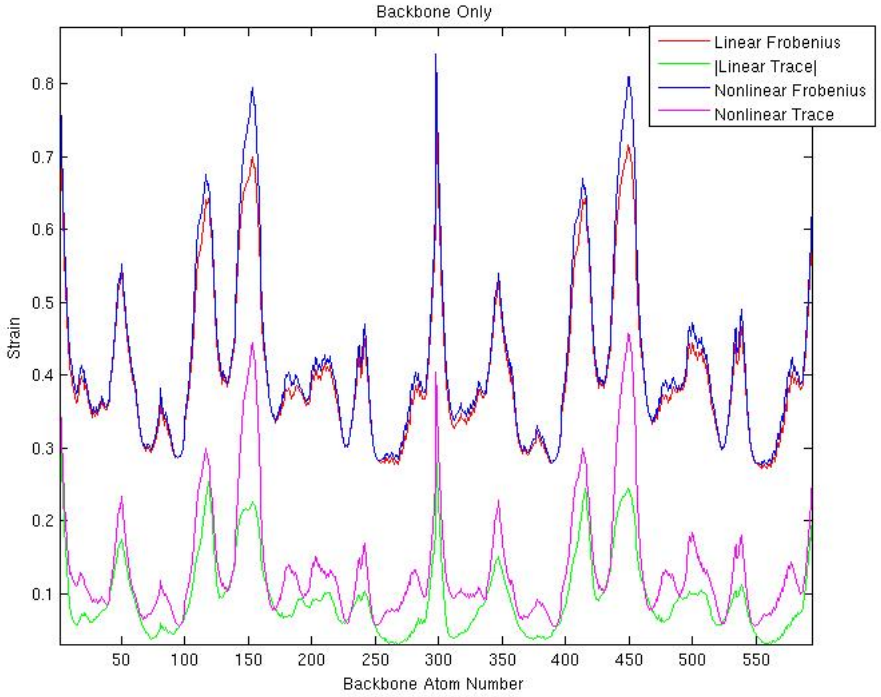


Figure 14.3: Strain from backbone only calculation.

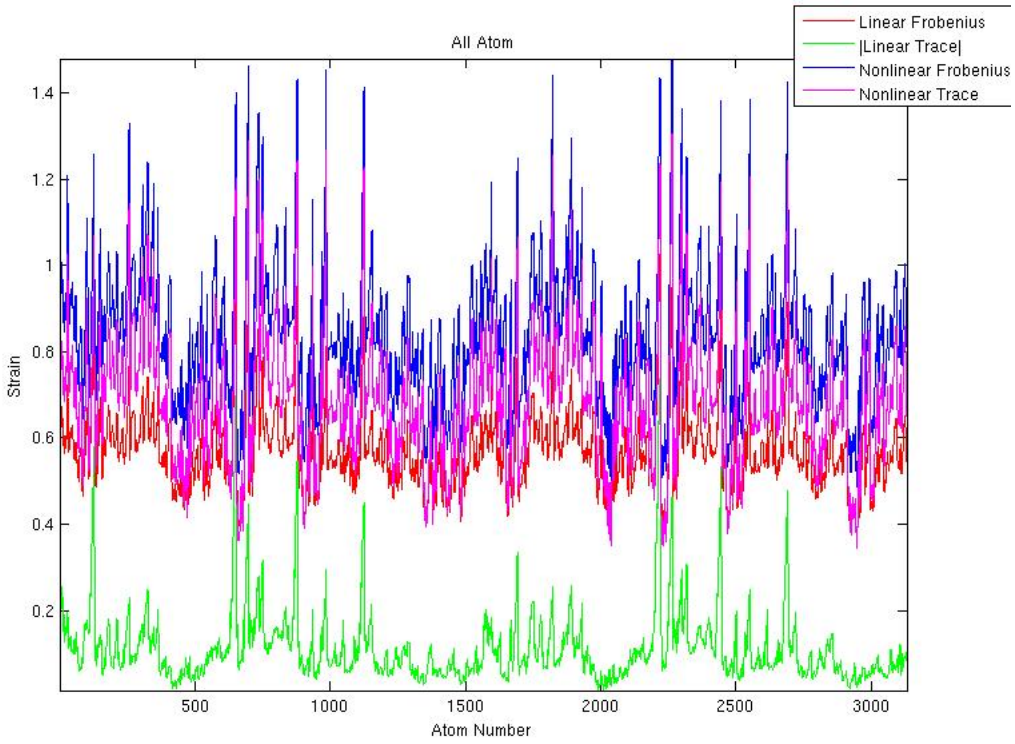


Figure 14.4: Strain from all-atom calculation.

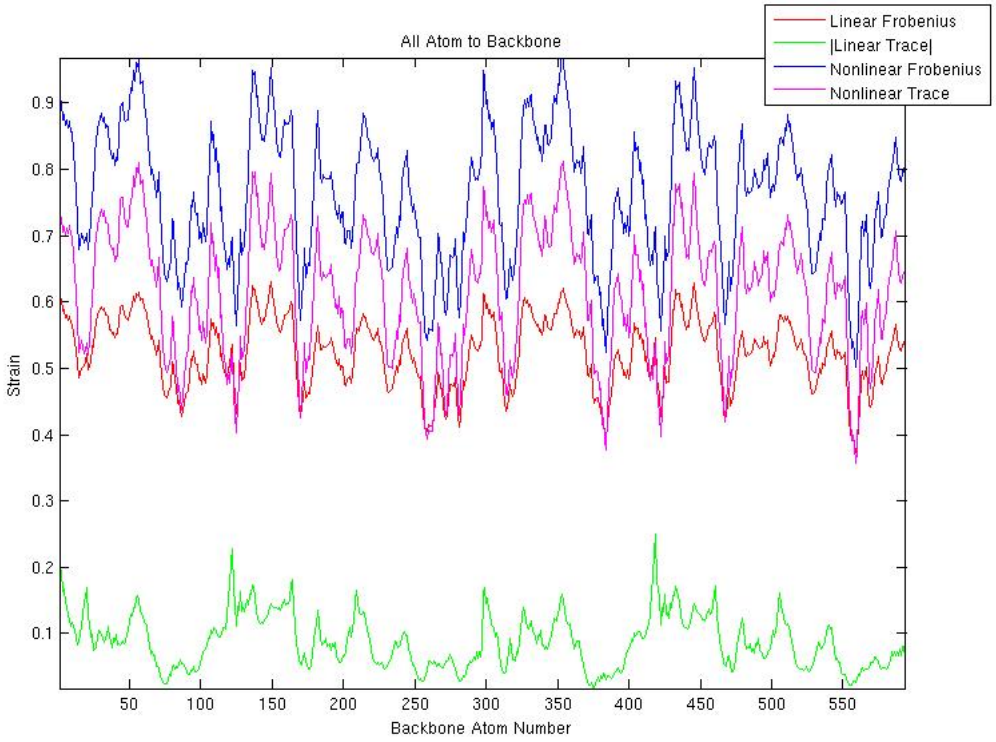


Figure 14.5: Strain from all-atom to backbone calculation.

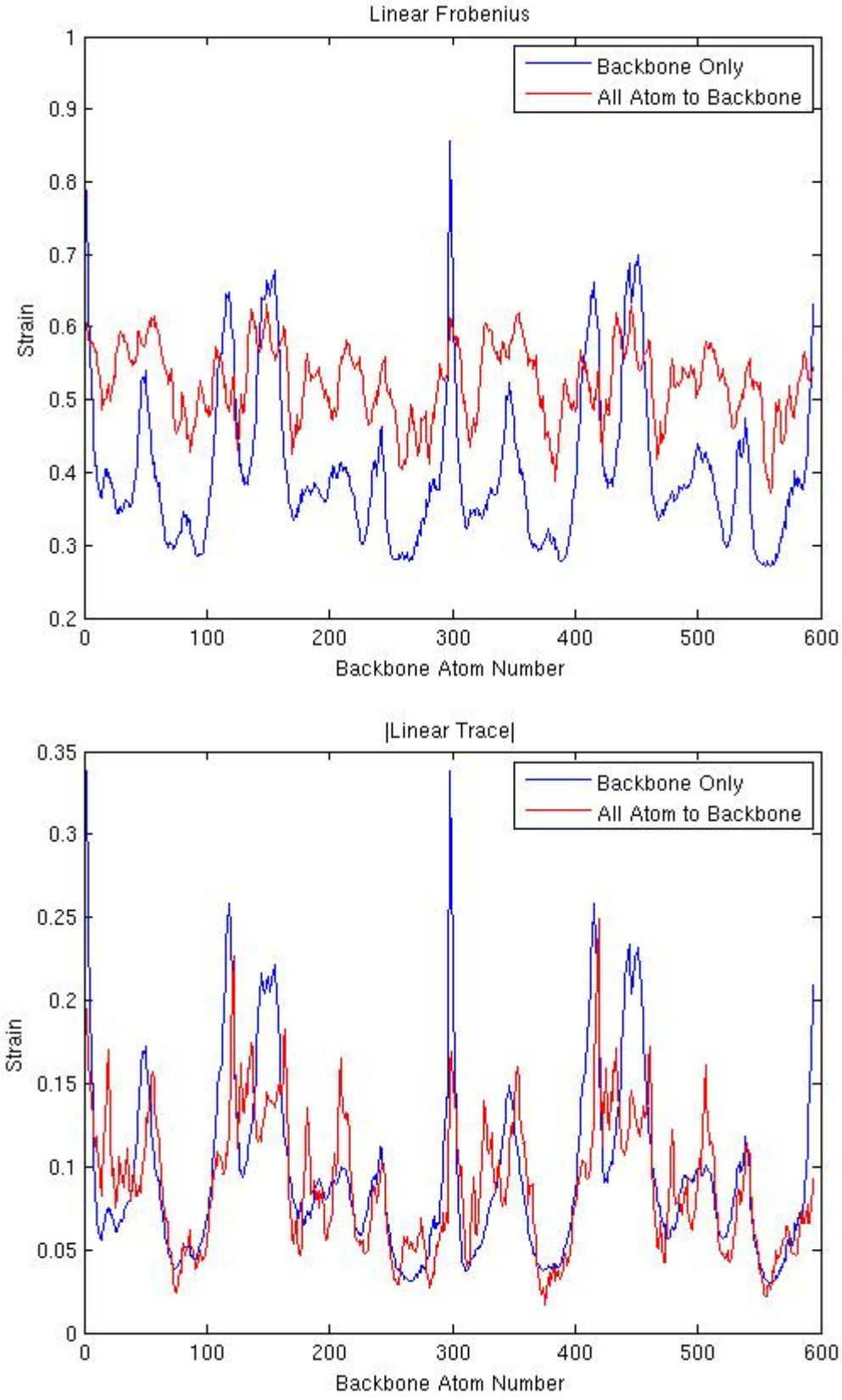


Figure 14.6: Comparison of backbone only and all-atom to backbone strain calculation for linear strain matrix.

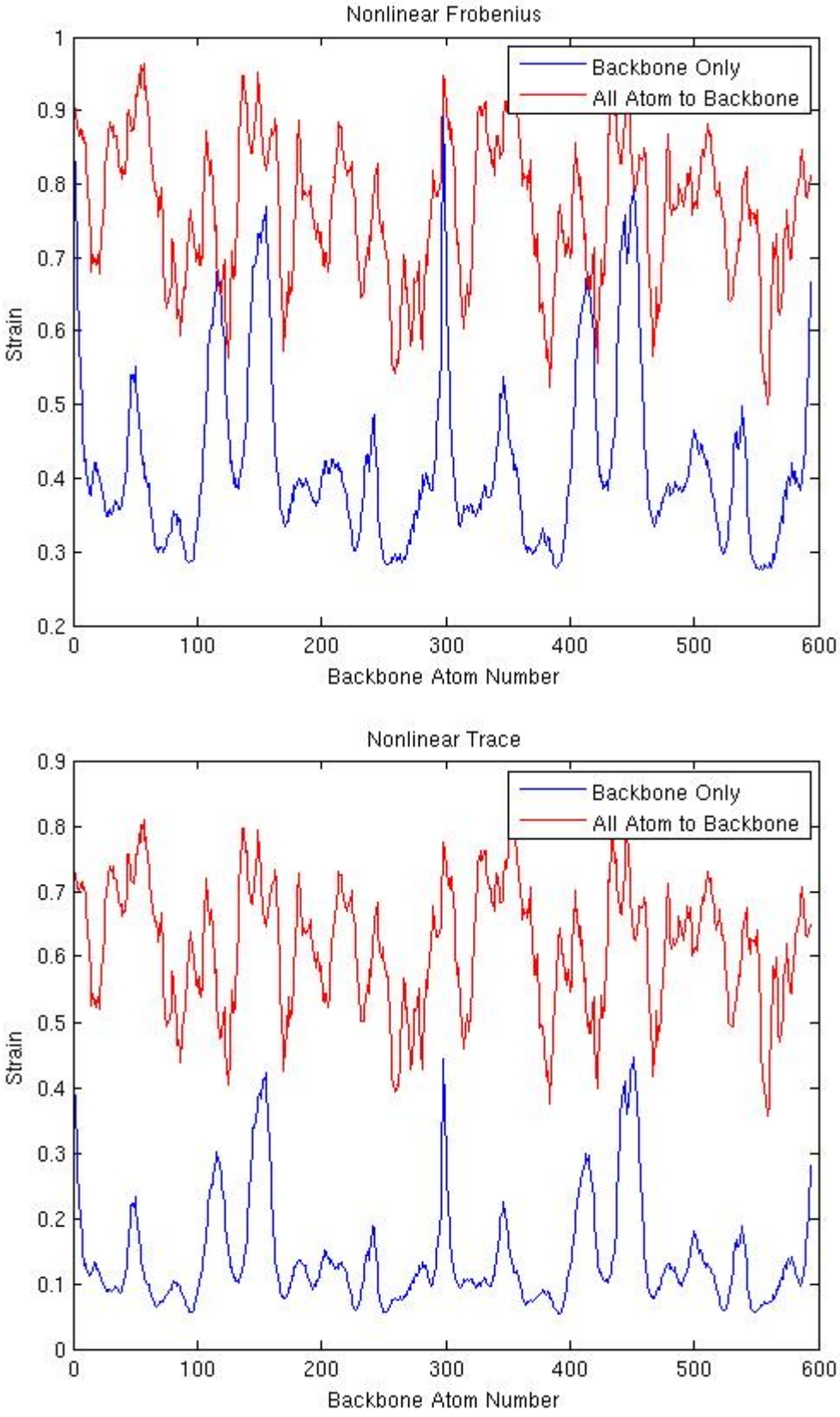


Figure 14.7: Comparison of backbone only and all-atom to backbone strain calculation for the nonlinear strain matrix.

Appendices

Appendix A

Proofs of Continuity

Tertiary proofs:

Consider three balls, p_i, p_j, and p_k, with centers p_i', p_j', and p_k', and radii r_i, r_j, and r_k, respectively. Let P be the plane that contains the centers of the three balls, and define $P + y$ as the plane parallel to and a distance y above P. Let t_{ijk} be the triangle with vertices at p_i', p_j', and p_k' t_{ijk} and define X as in equation 4.3.1.

Define $l_{ijk}\big(y, X\big)$ as the length of the surface of the intersection of the three balls which is cut by the plane $P+y$, and $l_{ijk}^{(m)}\big(y, X\big)$, $m = i, j, k$, as the contribution of atom m to $l_{ijk}\big(y, X\big)$ (See Figure A.1). For y such that there is a nontrivial triple intersection of the three balls in the plane $P + y$ we have

$$l_{ijk}^{(m)}(y) = \frac{1}{2}\Big[(1-\delta_{im})l_{im}^{(m)}(y)+(1-\delta_{jm})l_{jm}^{(m)}(y)+(1-\delta_{km})l_{km}^{(m)}(y)\Big]-\alpha_m r_m(y) \quad \text{(A.0.1)}$$

where δ_{mn} is the Kronecker delta, and $l_{mn}^{(m)}(y)$ is the length of the surface of the intersection of balls m and n in plane $P + y$ (See Figure A.1). The term α_m is the angle of t_{ijk} at point m. That is

$$\alpha_m = \arccos\left(\frac{\mathbf{a}\cdot\mathbf{b}}{ab}\right) \quad \text{(A.0.2)}$$

where $\mathbf{a}$ and $\mathbf{b}$ are two vectors in the triangle t_{ijk} that originate at p_m'. For $n = i, j, k$,

$n \neq m$, we have (See Figure A.2)

$$\begin{aligned} l_{mn}^{(m)}(r_m(y)) &= \theta_{mn} r_m(y) \\ &= 2 \arccos\left(\frac{r_m(y) - h_m(y)}{r_m(y)}\right) r_m \end{aligned} \tag{A.0.3}$$

where θ_{mn} is shown in Figure A.1, $r_m(y) = \sqrt{r_m^2 - y^2}$ is the radius of sphere m at height y and where $h_m(y)$ is given in equation 4.4.23 by substituting $r_m(y)$ and $r_n(y)$ for r_i and r_j. Define h as the height above the plane where the three spheres intersect (see equation 4.4.57).

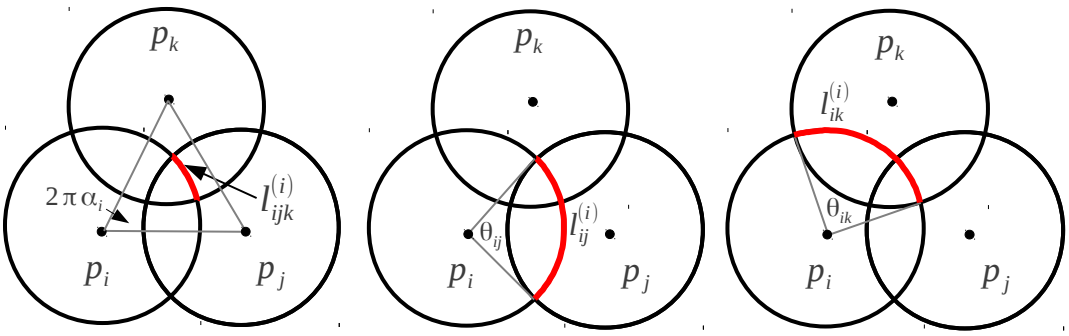


Figure A.1: Illustration of $l_{ijk}^{(i)}$, $l_{ij}^{(i)}$, and $l_{ik}^{(i)}$.

First, we consider the case when $l_{ijk}(0)$ is contained in the triangle t_{ijk}. Then an alternative formula for the surface area, $S_{ijk}^{(m)}$, $m = i, j, k$, is given by

$$S_{ijk}^{(m)}(X) = 2 \int_0^{h(X)} l_{ijk}^{(m)}\big(y, X\big) dy. \tag{A.0.4}$$

Note $l_{ijk}^{(m)}$ is both differentiable and integrable with respect to y and X. Define $L_{ijk}^{(m)}$

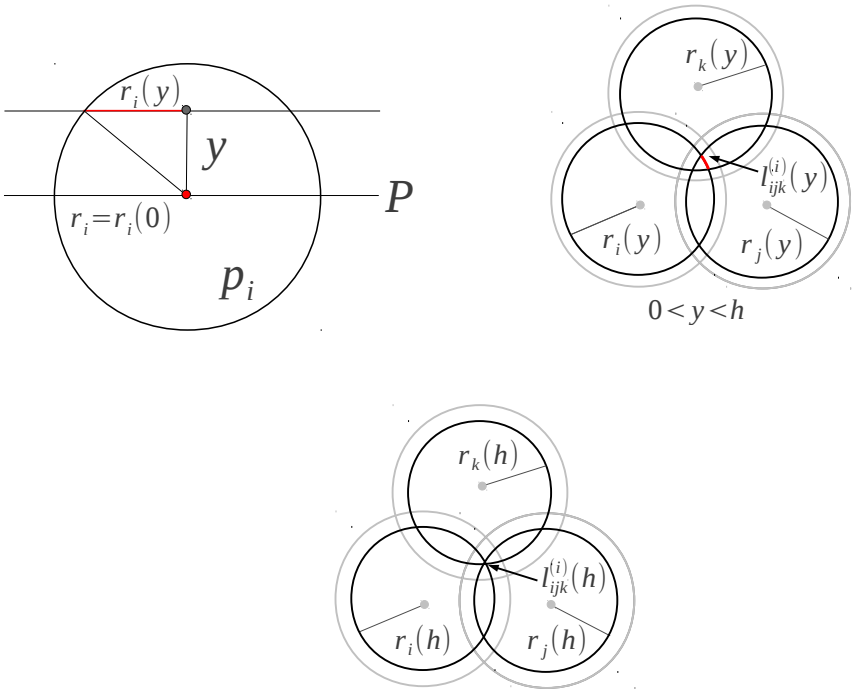


Figure A.2: The quantity $r_i(y)$ is the radius of the circle of intersection of sphere i with a plane y units above P.

such that

$$\frac{\partial L_{ijk}^{(m)}(y,X)}{\partial y} = l_{ijk}^{(m)}(y,X) \tag{A.0.5}$$

.

Let ∇ be the gradient operator with respect to elements in X. Then

$$\begin{aligned} 2\nabla S_{ijk}^{(m)} &= \nabla \int_0^{h(X)} l_{ijk}^{(m)}\big(y,X\big)dy \qquad (A.0.6)\\ &= l_{ijk}^{(m)}(h,X)\nabla h(X) + \nabla L_{ijk}^{(m)}(h,X) - \nabla L_{ijk}^{(m)}(0,X) \\ &= l_{ijk}^{(m)}(h,X)\nabla h(X) + \nabla\Big(L_{ijk}^{(m)}(h,X) - L_{ijk}^{(m)}(0,X)\Big). \end{aligned}$$

where the dependence of h on X in $\nabla L_{ijk}^{(m)}$ is suppressed to show that h is treated as an independent variable in this term. As $S_{ijk} \to 0, h \to 0^+$ which gives $\nabla\Big(L_{ijk}^{(m)}(h(X),X) - L_{ijk}^{(m)}(0,X)\Big) \to 0$ and $l_{ijk}^{(m)} \to 0$. Thus

$$\lim_{S_{ijk}\to 0} \nabla S_{ijk}^{(m)} = 0, \qquad m = i,j,k. \tag{A.0.7}$$

Now we want to consider the surface areas of p_i, p_j, and p_k as shown in d) as the configuration approaches that of e) (See Figure A.3). Define h as before. Since the

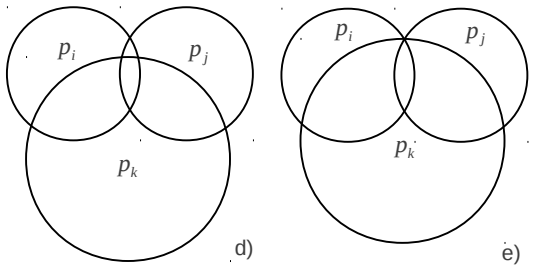


Figure A.3: Tertiary configurations d) and e).

entirety of $\nabla S_{ijk}^{(k)}$ is under the plane $P+h$ we can use A.0.4 and A.0.6 with $m=k$. As $S_{ijk} \to S_{ij}$, the height $h \to 0^+$ and we have

$$\lim_{S_{ijk} \to S_{ij}} S_{ijk}^{(k)} = 0. \tag{A.0.8}$$

Now, above the plane $P+h$, $l_{ijk}^{(m)}(y) = l_{ij}^{(m)}(y)$ for $m = i, j$. This gives

$$2\Big(S_{ij}^{(m)}(X) - S_{ijk}^{(m)}(X)\Big) = \int_0^{h(X)} \Big(l_{ij}^{(m)}(y,X) - l_{ijk}^{(m)}(y,X)\Big)dy. \tag{A.0.9}$$

Define $L_{ij}^{(m)}$ such that

$$\frac{\partial L_{ij}^{(m)}(y,X)}{\partial y} = l_{ij}^{(m)}(y,X). \tag{A.0.10}$$

Then

$$\begin{aligned} 2\nabla\Big(S_{ij}^{(m)}(X) - S_{ijk}^{(m)}(X)\Big) &= l_{ij}^{(m)}\nabla h(X) \\ &\quad +\nabla\Big(L_{ij}^{(m)}(h,X) - L_{ij}^{(m)}(0,X)\Big) \\ &\quad -l_{ijk}^{(m)}(h,X)\nabla h(X) - \nabla\Big(L_{ijk}^{(m)}(h,X) - L_{ijk}^{(m)}(0,X)\Big). \end{aligned} \tag{A.0.11}$$

By sending $S_{ijk} \to S_{ij}$, we have $l_{ij}^{(m)}(h,X) - l_{ijk}^{(m)} \to 0$, and $h \to 0$ sends the ∇L terms

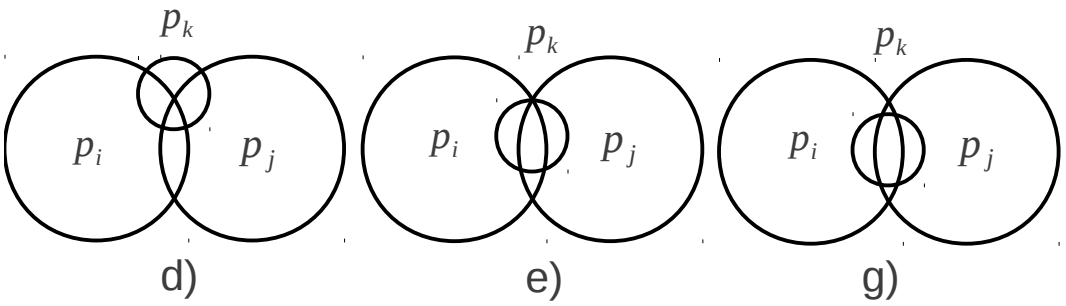


Figure A.4: Tertiary configurations d), e), and g).

to 0. Thus

$$\lim_{S_{ijk}\to S_{ij}} \left(S_{ij}^{(m)} - S_{ijk}^{(m)}\right) = 0 \qquad m = i, j. \tag{A.0.12}$$

Next, we will consider the surfaces of p_i, p_j, and p_k as shown in d) as the configuration approaches case g) through case e) alternate (See Figure A.4).

The surface area, $S^{(k)}$, is given by

$$2S^{(k)} = \int_0^{h(X)} \Big(l_k(y,X) - l_{ik}(y,X) - l_{jk}(y,X) + l_{ijk}(y,X)\Big) \tag{A.0.13}$$

with

$$\begin{aligned} 2\nabla S^{(k)} \;=\; & \Big(l_k(h,X) - l_{ik}(h,X) - l_{jk}(h,X) + l_{ijk}(h,X)\Big)\nabla h(X) \\ & +\nabla\Big(L_k(h,X) - L_k(0,X) + L_{ik}(h,X) - L_{ik}(0,X) \\ & +L_{jk}(h,X) - L_{jk}(0,X) + L_{ijk}(h,X) - L_{ijk}(0,X)\Big) \end{aligned} \tag{A.0.14}$$

As $S_k \to 0$, we have $\Big(l_k(h,X) - l_{ik}(h,X) - l_{jk}(h,X) + l_{ijk}(h,X)\Big) \to 0$, and $h \to 0^+$

gives

$$\lim_{S_k \to 0} S^{(k)} = 0. \tag{A.0.15}$$

Now, above the plane $P + h$, $l_{ijk}^{(m)}(y) = l_{ik}^{(m)}(y)$ for $m = i, j$. This gives

$$2\Big(S_{ik}^{(m)}(X) - S_{ijk}^{(m)}(X)\Big) = \int_0^{h(X)} \Big(l_{ik}^{(m)}(y, X) - l_{ijk}^{(m)}(y, X)\Big) dy. \tag{A.0.16}$$

Then

$$\begin{aligned} 2\nabla\Big(S_{ik}^{(m)}(X) - S_{ijk}^{(m)}(X)\Big) &= l_{ik}^{(m)} \nabla h(X) + \nabla\Big(L_{ik}^{(m)}(h, X) - L_{ik}^{(m)}(0, X)\Big) \\ &\quad - l_{ijk}^{(m)}(h, X)\nabla h(X) - \nabla\Big(L_{ijk}^{(m)}(h, X) - L_{ijk}^{(m)}(0, X)\Big). \end{aligned} \tag{A.0.17}$$

By sending $S^{(k)} \to 0$, we have $l_{ik}^{(m)}(h, X) - l_{ijk}^{(m)} \to 0$, and $h \to 0$ sends the ∇L terms to 0. Thus

$$\lim_{S^{(k)} \to 0} \Big(S_{ik}^{(m)} - S_{ijk}^{(m)}\Big) = 0 \qquad m = i, j. \tag{A.0.18}$$

Quaternary proofs:

We can show $\nabla S_{ijkl}^{(m)} \to 0$ for $m = i, j, k, l$ as $S_{ijkl} \to 0$ in a similar manner as for the tertiary case. Detailed proofs will not be presented here.

Appendix B

Accuracy of Volume and Surface Area Implementation

B.1 Overview

The program 'alphavol' or 'mkalpha' returns the alpha complex, volume, void volume, outer surface area, void surface area, surface area contributions from individual atoms, and the tetrahedra in the voids of a given molecule.

1. Tetrahedra in voids were checked using an alternate (slow and brute force) method to locate the tetrahedra in the voids.

2. Total volume was estimated as follows: The program computes the smallest (with a buffer region) box that contains all atoms in the molecule. Next, 10^n uniformly random points in this box are chosen and tested to see if they lie in the space filling model (union of atoms). The volume of the box is multiplied by the fraction of points in the space filling model to obtain a volume estimate.

3. Total void volume was estimated as follows: The volume of the void tetrahedra are computed and then the "inside fringe" volume is subtracted using an estimate as in 2. The number of test points for each tetrahedron is $10^n/num\_Tvoid$ where num_Tvoid is the number of tetrahedra in the voids.

4. Total surface area is estimated as follows: For a given atom, $10^n/num\_coords$ uniform test points are generated on the surface of the sphere [27]. Each point is tested to see if it is exterior to all other atoms. The surface area of the given atom is computed and multiplied by the fraction of points outside the union of atom interiors. This is done for each atom and the surface areas of each are summed.

5. Individual surface area is also estimated in a similar manner as the total surface area. Further tests are performed on a given atom where the surface area of the given atom is checked using 10^n test points instead of $10^n/num\_coords$ test points.

The alpha complex, outer surface area, and void surface area are not explicitly checked. However, since the alpha complex is used in the volume and surface area calculations, correct volume and surface area imply a correct alpha complex. The outer surface area is found in 'alphavol' by subtracting the surface area of the voids from the total surface area. The same formulas are used in the void surface area and the total surface area calculations, and the same simplices are involved in the void surface area and void volume calculations. This means that correct total surface area and void volumes imply correct void surface area and outer surface area.

B.2 Test 1: Increase Number of Test Points

A polypeptide which we call 'p25' and which has 9 residues and 144 atoms was examined to test the correctness of the 'alphavol' implementation. Each point was assigned a radius using a parameter file which was derived from AMBER, and the weight is the radius squared. The number n was increased from 3 to 11. The convergence of total volume, void volume, and surface area estimates to analytic values are shown in Figure B.1. The relative errors are shown in Figure B.2 The total runtime was about 3 hours and 32 minutes.

B.3 Test 2: Vary α

The test protein was a TRP-Cage construct, PDB identification code '1L2Y', which has 20 residues and 304 atoms. Each point was assigned a radius using parameters derived from AMBER. The weight is the squared radius. First the volume and surface area calculations were found using 'alphavol'. Then α was taken in increments of $.4$ from -4.2 to 5 Angstroms. The number of test points for each file was 10^8. From Figure B.3 we see that volume, surface area, and void volume estimates behave as expected with respect to analytic quantities. The calculations for each value of α took about 15 minutes for a total of about six hours.

B.4 Test 3: Ten Different Structures

Ten structures from a HIV Protease molecular dynamics trajectory next used to text accuracy of the 'alphavol' implementation. Each structure has 198 residues and 3132 atoms and the total number of test points is 10^8. The testing took about three and a half hours per structure for a total of 35 hours. The void volume estimates and 'alphavol' values are close. The estimates and 'alphavol' values for the total volume and total surface area are not as similar as those found in test 2 (See Figure B.4) due to the fact that HIV Protease is a larger molecule than the polypeptide 'p25'. The same data set was studied with 10^9 test points. (See Figure B.5). Each set required about 35 hours of computation.

B.5 Test 4: Atomic Solvent Accessible Surface Area

The individual atomic solvent accessible surface areas of the TRP-Cage construct, PDB identification code '1L2Y', were tested with solvent weights of $w_s = 0$ and $w_s = 10$. For each weight, estimates were calculated for 10^n total test points with $n = 5, 6, 7, 8, 9$. The molecule has 304 atoms which gives a total of $\frac{10^n}{304}$ test points per atom surface. The surface areas for atoms with the largest errors at $\frac{10^n}{304}$ test points per atom were then

estimated using 10^n with $n = 5, 6, 7, 8, 9$ test points per atom.

For $w_s = 0$ each of the atoms were exterior and had nonzero surface area while for $w_s = 10$ there were interior atoms with zero surface areas. For this reason, relative differences are considered for $w_s = 0$ and absolute differences are considered for $w_s = 10$.

The relative differences between analytical and estimated surface areas for 10^n total test points, $n = 5, 6, 7, 8, 9$, and $w_s = 0$ are plotted in Figure B.6. The average, maximum, and total relative differences for $n = 5, 6, 7, 8, 9$ are plotted in Figure B.7. The atom number with the largest difference is plotted in Figure B.8. The fact that the atom with the highest difference changes as a function of the number of test points is a good indication that the error is due to numerical inaccuracies rather than error in the analytical algorithm or implementation. Figure B.9 shows more accurate estimates for the individual surface area of atom 251 which had the largest relative difference in the test run with 10^9 total points. This difference is referred to as the 'previous difference' and the further estimates have 10^n test points, $n = 5, 6, 7, 8, 9$, for the individual atom.

Absolute differences between analytical and estimated surface areas for 10^n total test points with $n = 5, 6, 7, 8, 9$ and solvent weight 10 are shown in Figure B.10. The average, maximum, and total absolute differences are plotted in Figure B.11. Figure B.13 shows additional estimates for the atom with highest absolute error (See Figure B.5).

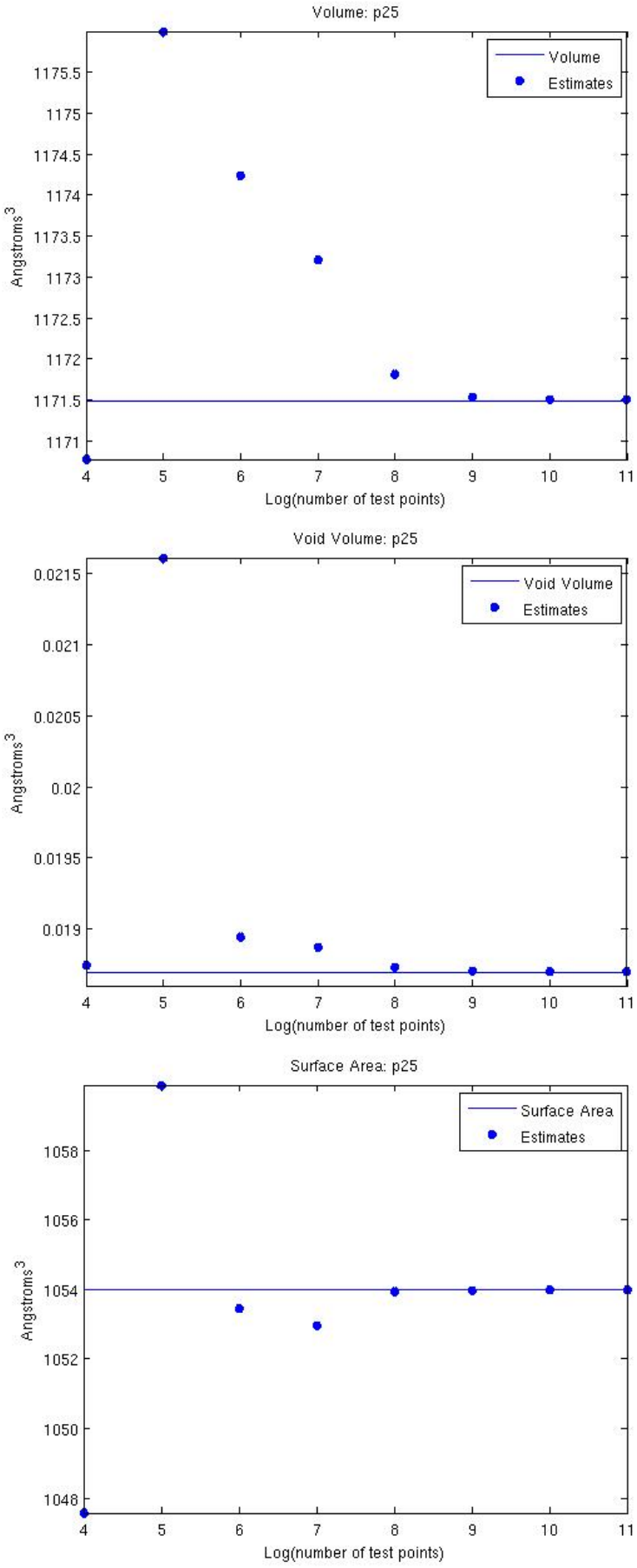


Figure B.1: Convergence of estimates to analytic values for polypeptide 'p25'.

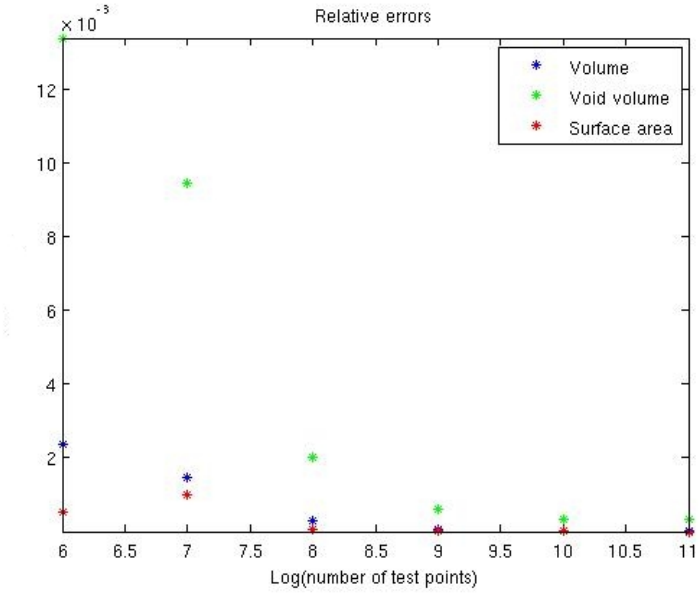


Figure B.2: Relative errors between estimates and analytic quantities.

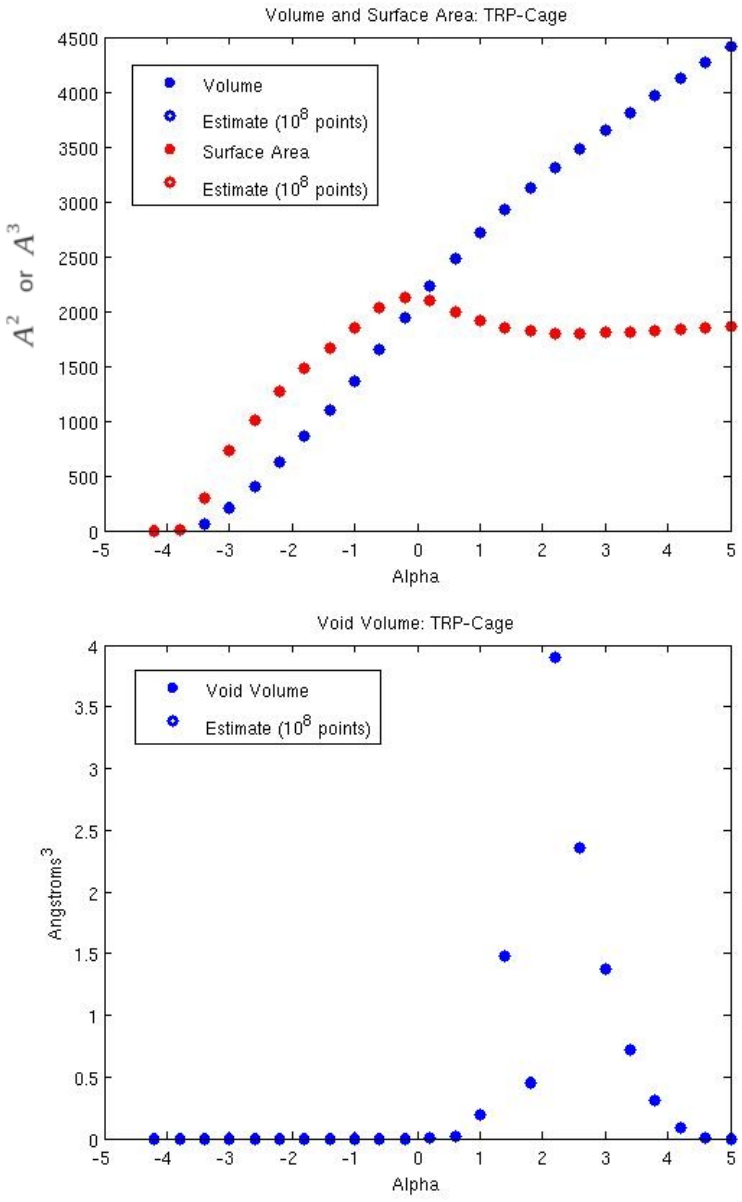


Figure B.3: Estimates and analytic values for various values of α.

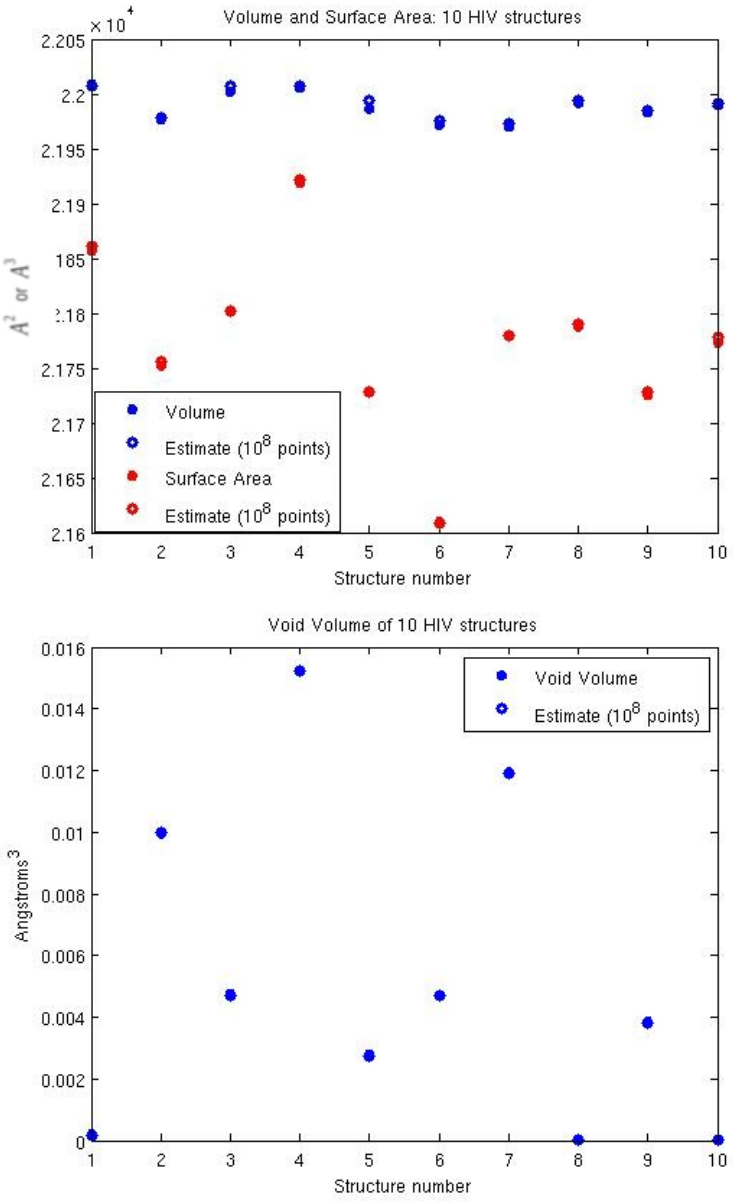


Figure B.4: Estimate and analytic quantities ten HIV protease structures.

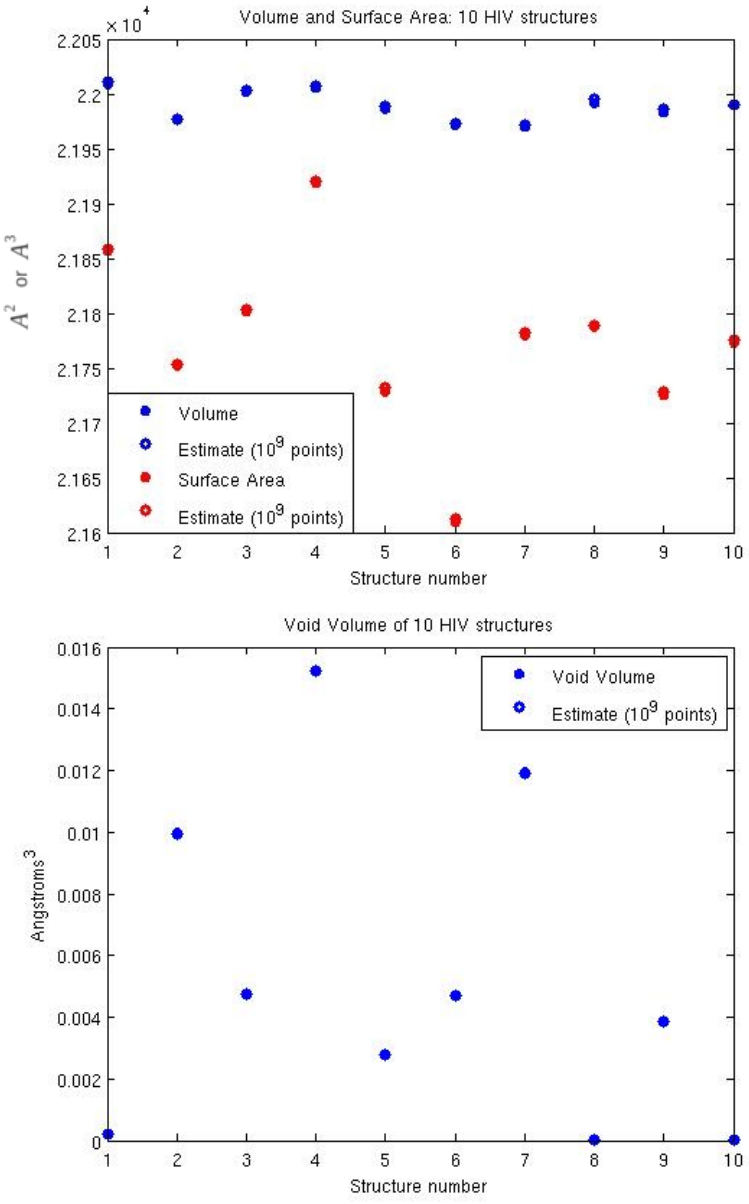


Figure B.5: Estimate versus analytic for 10^9 test points.

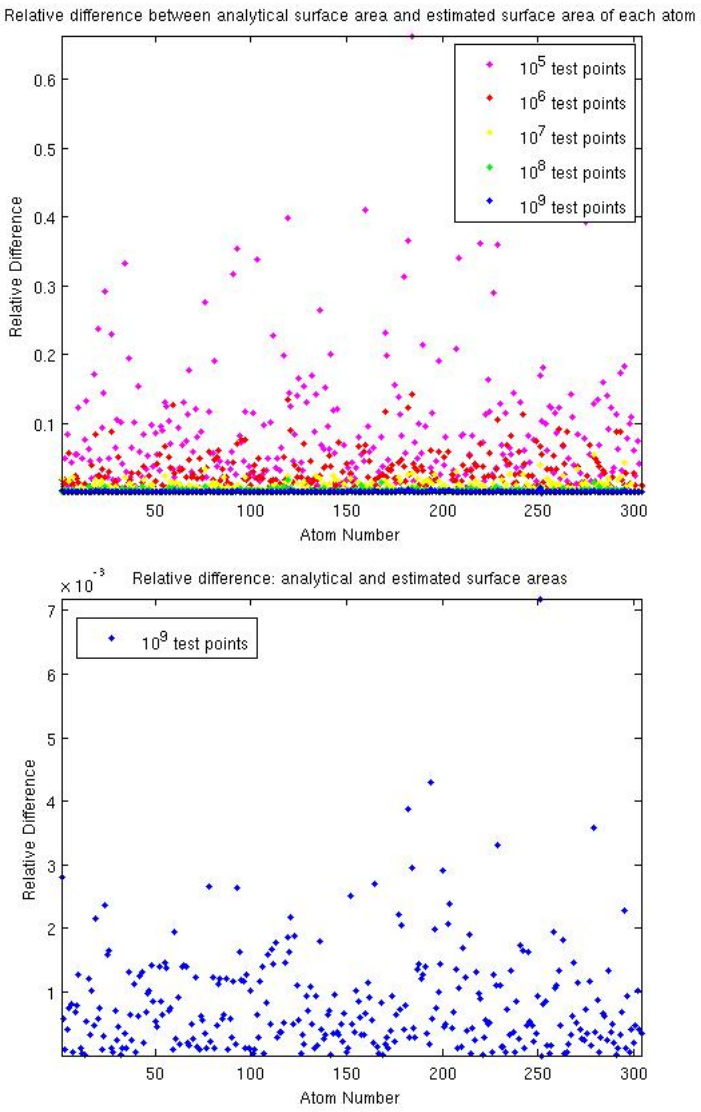


Figure B.6: Relative difference between analytic and estimated surface areas for each atom in molecule '1L2Y' for solvent weight 0.

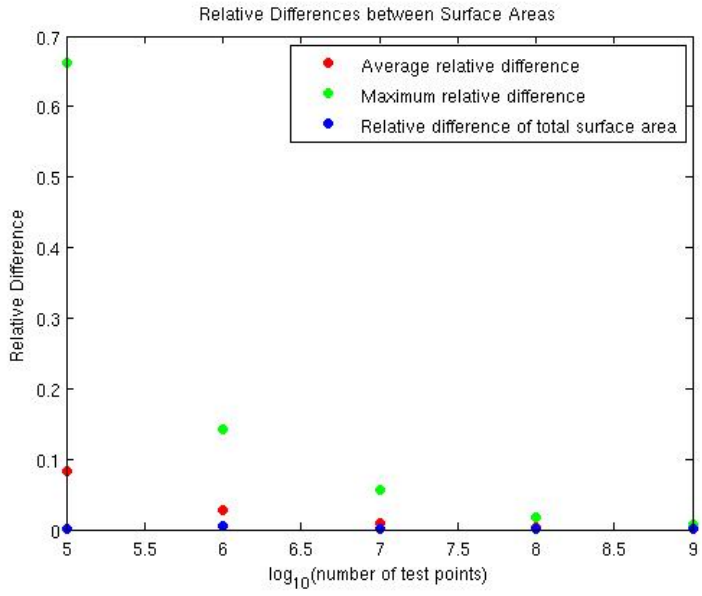


Figure B.7: Average relative differences and maximum relative differences for individual atomic surface areas and relative differences of total surface area for molecule '1L2Y' with solvent weight 0.

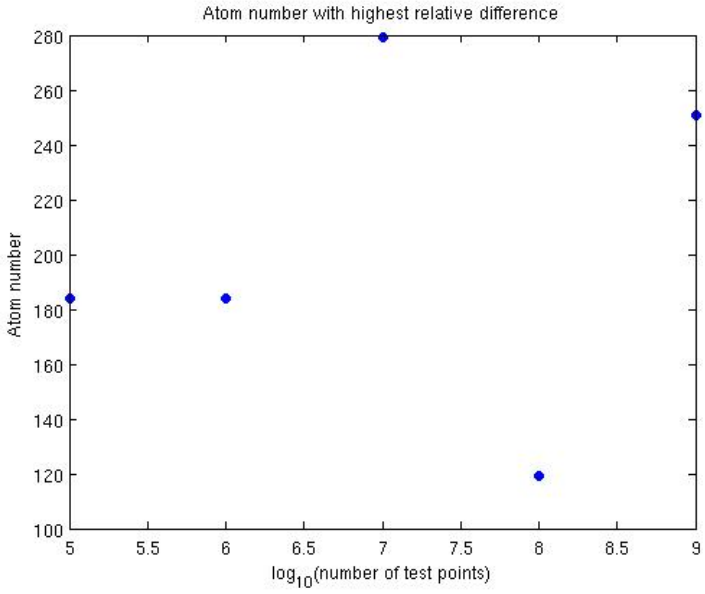


Figure B.8: Atom number with the highest relative difference for '1L2Y' with solvent weight 0.

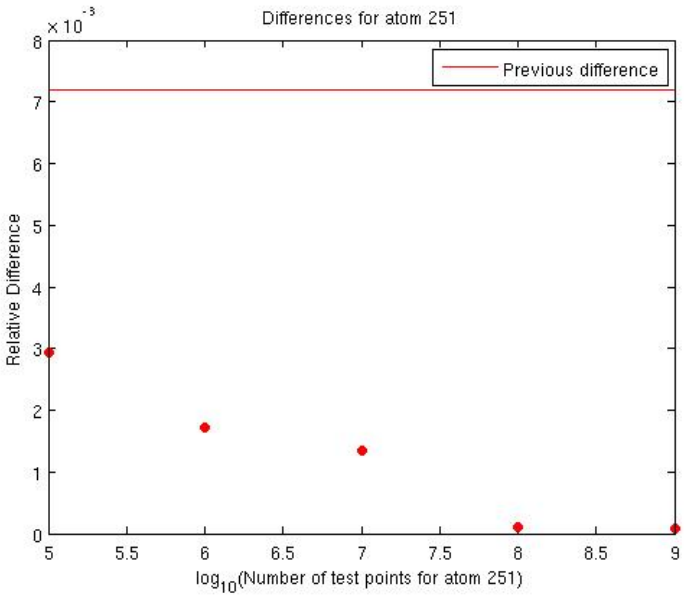


Figure B.9: Relative differences for $n = 5, 6, 7, 8, 9$ of atom with highest relative difference at $n = 10^9$ for molecule '1L2Y' and solvent weight 0.

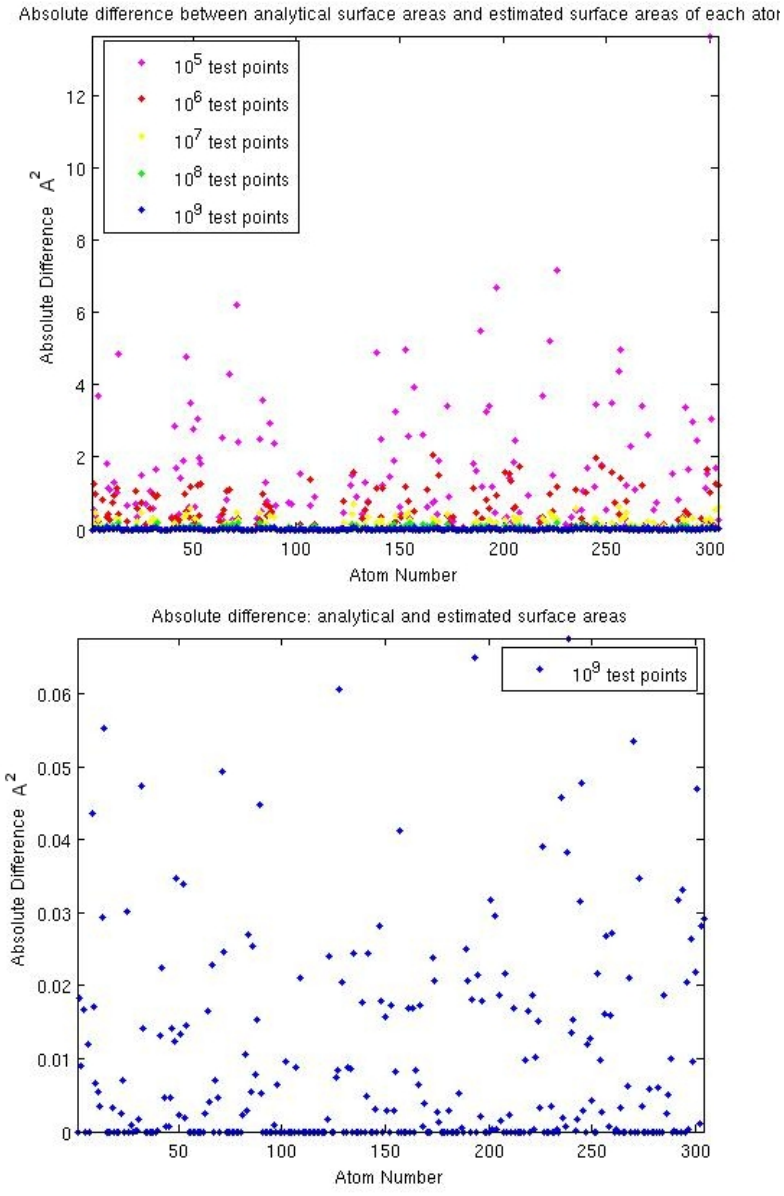


Figure B.10: Absolute differences between analytic and estimated surface areas for each atom in the TRP-Cage for solvent weight 10.

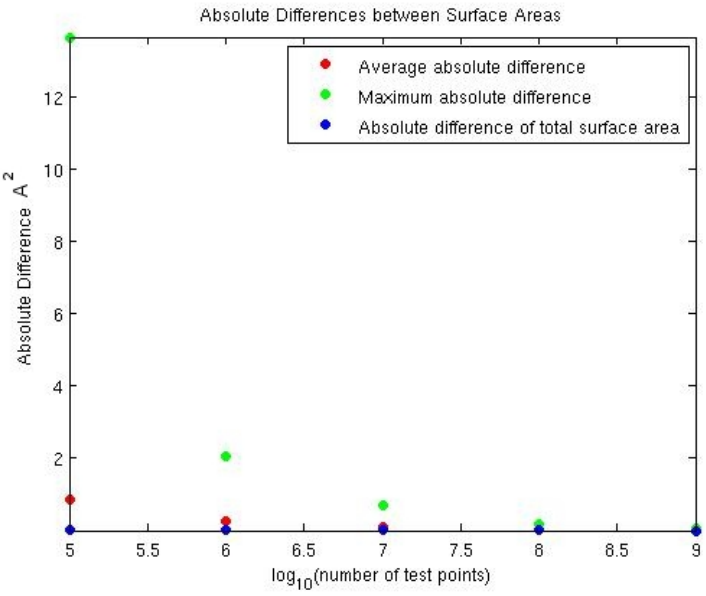


Figure B.11: Average absolute differences and maximum absolute differences for individual atomic surface areas and absolute differences of total surface area for TRP-Cage molecule with solvent weight 10.

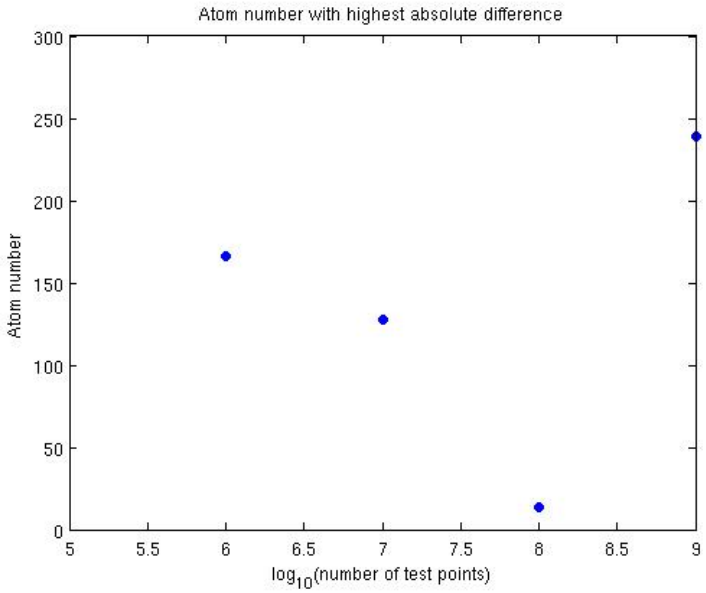


Figure B.12: Atom number with the largest absolute error for 10^n, $n = 5, 6, 7, 8, 9$ total test points.

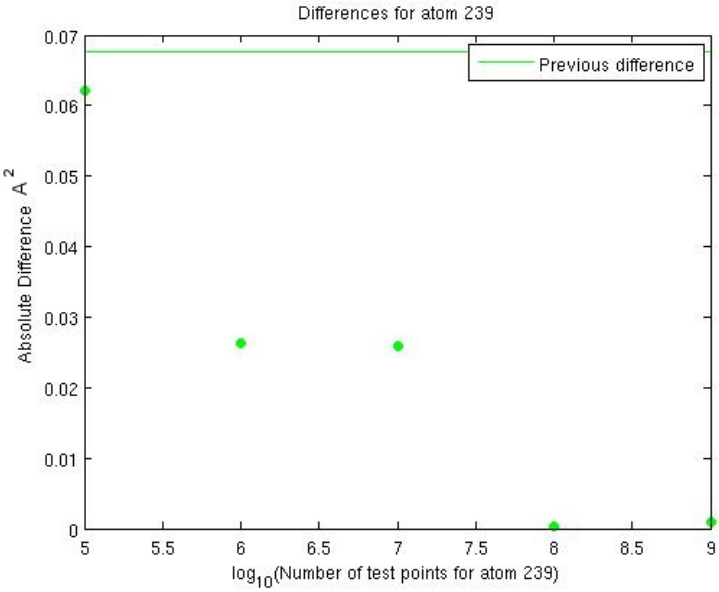


Figure B.13: Absolute differences for 10^n with $n = 5, 6, 7, 8, 9$ test points per atom for atom with highest relative difference at 10^9 total test points for TRP-Cage solvent weight 10.

Appendix C

Accuracy of Laguerre-Intersection Implementation

C.1 Notation

Define the following quantities:

- LV_i: Volume of the Laguerre cell of atom i
- LS_i: Surface area of the Laguerre facets of atom i
- $LIV_i(w)$: Volume of the Laguerre-Intersection cell of atom i with solvent weight w
- $LIS_i(w)$: Surface area of the Laguerre-Intersection cell of atom i with solvent weight w

C.2 Testing Laguerre-Intersection Volumes and Surface Areas

First $LIV(w)$ and $LIS(w)$ were computed for all atoms in a tryptophan residue for $w = 0$ and $w = 1.4$.

C.2.1 LIV

Numerical estimates, $LIV_i\_est(w,n)$, for $LIV_i(w)$ were found as follows:

1. 2^n uniformly random test points were generated in the smallest box containing $\mathcal{B}_w$ (Refer to Equation 3.1.1).
2. $x \notin \mathcal{B}_w$ are thrown out
3. If $x \in \mathcal{B}_w$ its nearest power neighbor is found by looping through all elements in $\mathcal{A}_w$
4. Call n_i the number of test points with atom i as its nearest power neighbor
5. Then $LIV_i\_est(w,n) = \frac{n_i}{2^n}V$ where V is the volume of the containing box

The quanitity $LIV_i\_est(w,n)$, with $w = 0$ and $w = 1.4$, was computed for $n = 17 : 29$, and the maximum relative difference

$$\max_{p_i(w)\in\mathcal{A}_w} \frac{|LIV_i(w) - LIV_i\_est(w,n)|}{LIV_i(w)} \tag{C.2.1}$$

is plotted for all n in Figure C.1. For $n = 29$ the atom with the largest relative error was located. The Laguerre-Intersection volume for this atom, call it $p_i(w)$, was again estimated as follows:

1. 2^n test points were generated in $B_{p_i(w)}$
2. call n_i the number of test points whose nearest neighbor is $p_i(w)$
3. then $LIV_i\_est(w,n) = \frac{n_i}{2^n}V$ where V is the volume of the sphere $B_{p_i(w)}$

The relative differences decreased from about 10^{-2} in the first test to 10^{-4} as shown in the Figure C.2.

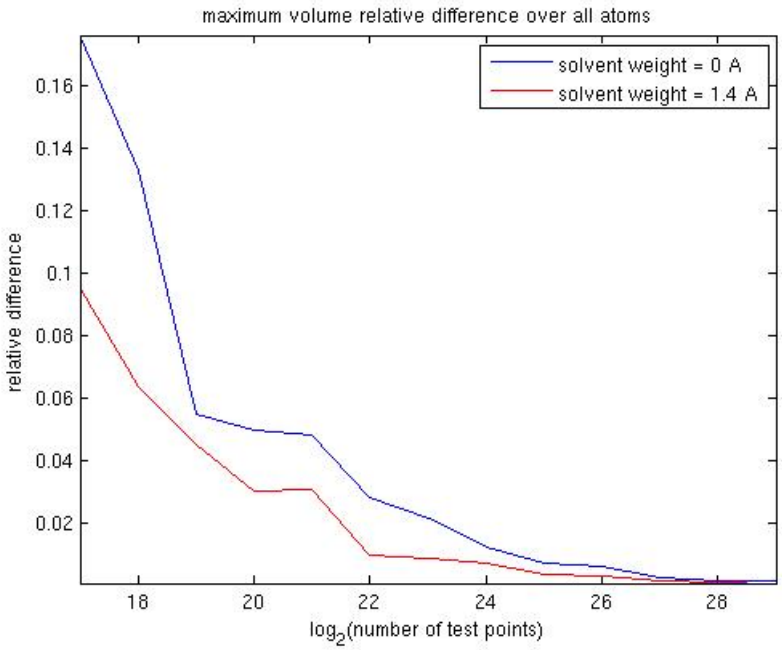


Figure C.1: Maximum relative differences between estimated and analytic Laguerre-Intersection atomic volumes.

C.3 LIS

The union of Laguerre facets of $p_i(w)$ is written as Lf_i. Define

$$LfI_i(w) = Lf_i \bigcap B_{p_i(w)}. \tag{C.3.1}$$

Let LfS_i and $LfIS_i(w)$ be the surface areas of these two sets, and let S_i be the accessible surface area of $B_{p_i(w)}$.

The Laguerre-Intersection surface area is the sum of two quantities:

$$LIS_i(w) = S_i(w) + LfIS_i(w) \tag{C.3.2}$$

The calculation of the accessible surface area, $S_i(w)$, has been tested extensively and has shown to be correct. Our focus will be on checking $LfIS_i(w)$.

As mentioned before, each Laguerre facet corresponds to an edge in the weighted Delaunay tetrahedrization, $\mathcal{T}$. We write the edge between atoms i and j, or more precisely, the line segment between p_i' and p_j' as e_{ij}. We call the corresponding Laguerre

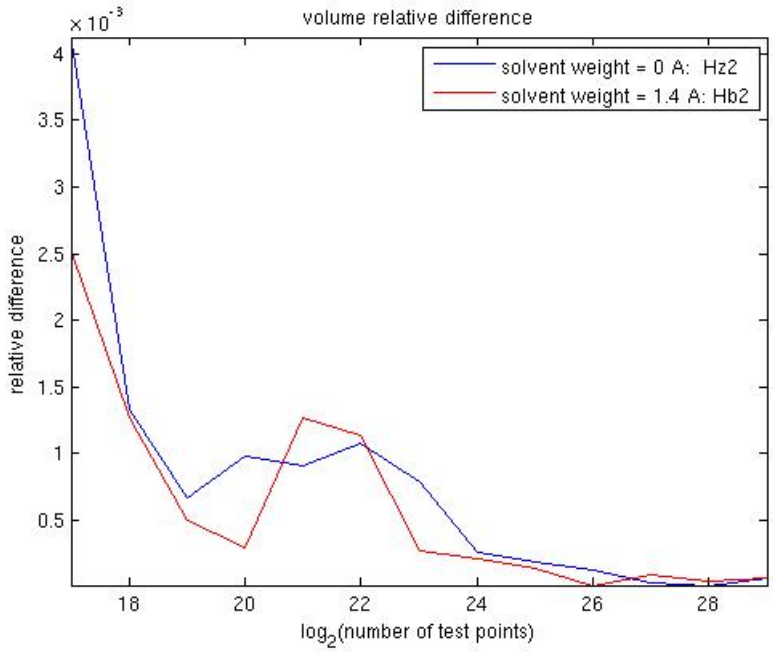


Figure C.2: Maximum relative differences between estimated and analytic Laguerre-Intersection atomic volumes for the Hz2 and Hb2 atoms.

facet L_{ij} which has surface area LS_{ij}. The Laguerre surface area of atom i is the sum of the areas of the Laguerre facets corresponding to edges which contain p_i'. That is

$$LS_i = \sum_{e_{ij} \in \mathcal{T}} LS_{ij} \tag{C.3.3}$$

A similar formula holds for $LfIS$ with

$$LfIS_i(w) = \sum_{e_{ij} \in \mathcal{T}} LfIS_{ij}(w). \tag{C.3.4}$$

Since $LfIS_{ij} = 0$ for $e_{ij} \notin \mathcal{C}_w$ we may write

$$LfIS_i(w) = \sum_{e_{ij} \in \mathcal{C}(w)} LfIS_{ij}(w). \tag{C.3.5}$$

.

Numerical estimates, $LfIS_i\_est(w, n)$, for $LfIS_i(w)$ were found as follows:

1. 2^n uniformly random points were generated on the disc $L_{ij} \bigcap \left(B_{p_i(w)} \bigcup B_{p_j(w)}\right)$.

2. Call n_{ij} the number of test points whose nearest power neighbors are $p_i(w)$ and $p_j(w)$ (recall that all points in L_{ij} are equipowerdistant to p_i and p_j).

3. Then $LfIS_{ij}\_est(w,n) = \frac{n_{ij}}{2^n} D$ where D is the area of the disc.

4. And $LfIS_i\_est(w,n) = \sum_{e_{ij} \in \mathcal{C}(w)} LfIS_{ij}\_est(w,n)$.

The maximum relative difference over all atoms for $n = 17 : 24$ is shown in the Figure C.3 for solvent weights $w = 0$ and $w = 1.4$.

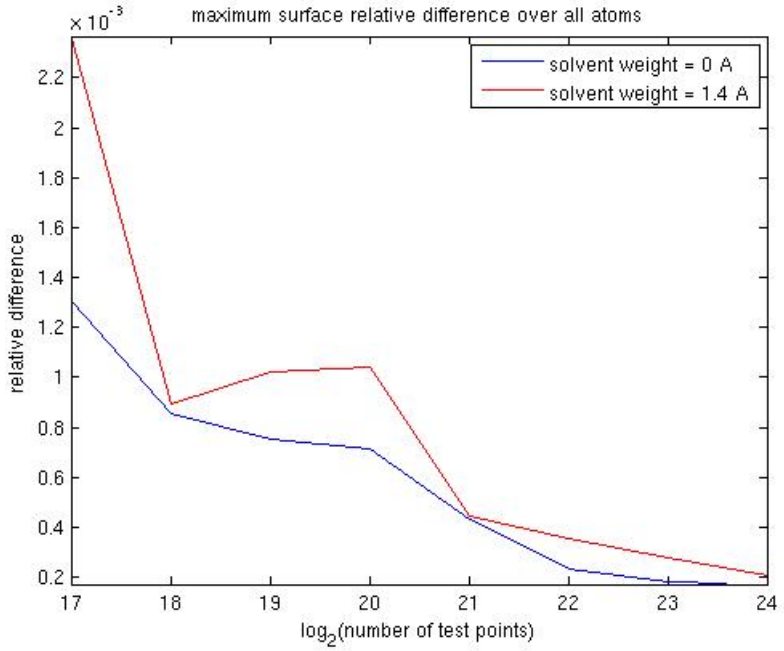


Figure C.3: Maximum relative differences between estimate and analytic $LfIS$ quantities.

For $n = 24$ the atoms with the largest relative error were located. This was the atom N for $w = 0$ and Cg for $w = 1.4$. The Laguerre facet Intersection surfaces for the atoms were again estimated. This time the algorithm only looped over edges in $\mathcal{C}(w)$ that contained the given atom. This allowed the maximum number of test points per disc to be increased from 2^{24} to 2^{29}. The relative differences decrease from about 10^{-4} in the first test to about 10^{-7} (See Figure C.4).

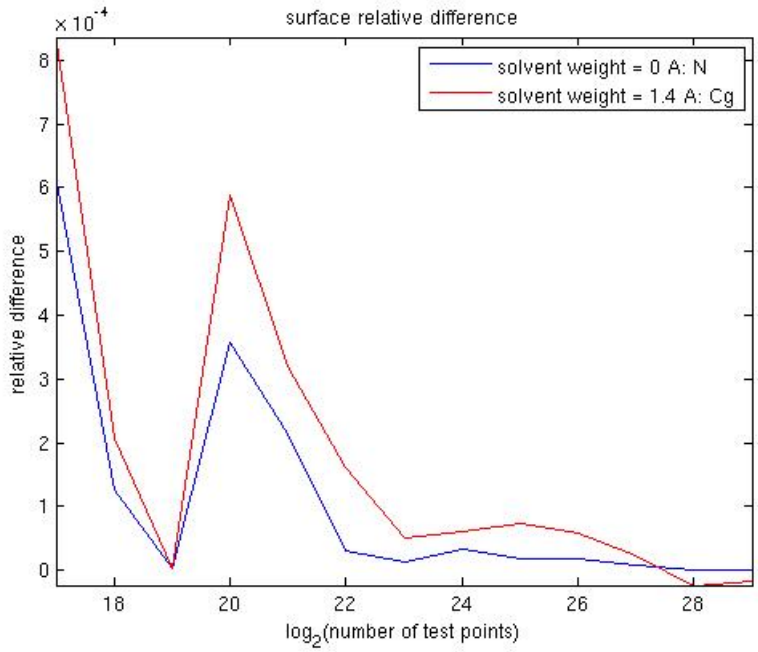


Figure C.4: Maximum relative differences between estimate and analytic $LfIS$ quantities for atoms N and Cg.

C.4 Convergence of LI to L for Interior Atoms

The Laguerre-Intersection cell is a subset of the Laguerre cell,

$$LI_i(w) \subset L_i \tag{C.4.1}$$

Let $\mathcal{A}'$ be the set of p_i' such that $p_i \in \mathcal{A}$. If p_i' does not lie on the convex hull of $\mathcal{A}'$, then there exists a critical weight, W_i, such that

$$LI_i(w) = L_i \quad \forall \quad w \geq W_i. \tag{C.4.2}$$

Let $\mathcal{T}_i$ be the set of tetrahedra in $\mathcal{T}(\mathcal{A})$ that contain p_i' as a vertex. Write the size of tetrahedron T as w_T. Then

$$W_i = \max_{T \in \mathcal{T}_i} w_T. \tag{C.4.3}$$

For points on the convex hull of $\mathcal{A}'$

$$LI_i(w) \to L_i \text{ as } w \to \infty \tag{C.4.4}$$

Atom	Ca	Cb	Cg	Cd1	Ne1	Cd2
W	7.006	11.86	34.75	62.23	68.31	87.36

Table C.1: Atoms in test molecule with smallest critical weights.

The quantities LIV_i and LIS_i are increasing functions of w. These quantities converge to LV_i and LS_i for atoms interior to $\mathcal{A}'$ as w increases.

To test this convergence the six atoms with the smallest critical weights, W, were located (see Table C.1). The solvent weights were increased by increments of $5\ A^2$ from 0 to 95. The weights between which LIV_i and LIS_i became equal to LS_i and LV_i are plotted in Figure C.5. When the weight increment is decreased, the bound on the weights approaches the critical weight. The larger step size of $5\ A^2$ is used to demonstrate the convergence of LI_i to L_i for the atoms which have a relatively large difference in critical weights.

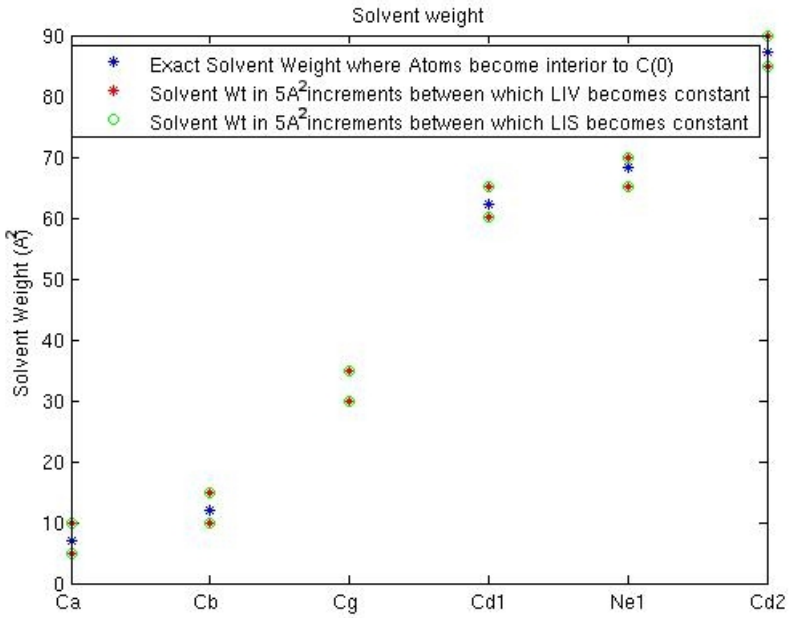


Figure C.5: The critical solvent weight (blue) and solvent weights in 5 A^2 increments between which Laguerre-Intersection volumes and surfaces become constant.

Laguerre-Intersection volumes and surfaces are increasing functions of w. Figure C.6 illustrates the monotonicity and the convergence of Laguerre-Intersection quantities to Laguerre quantities for interior atoms.

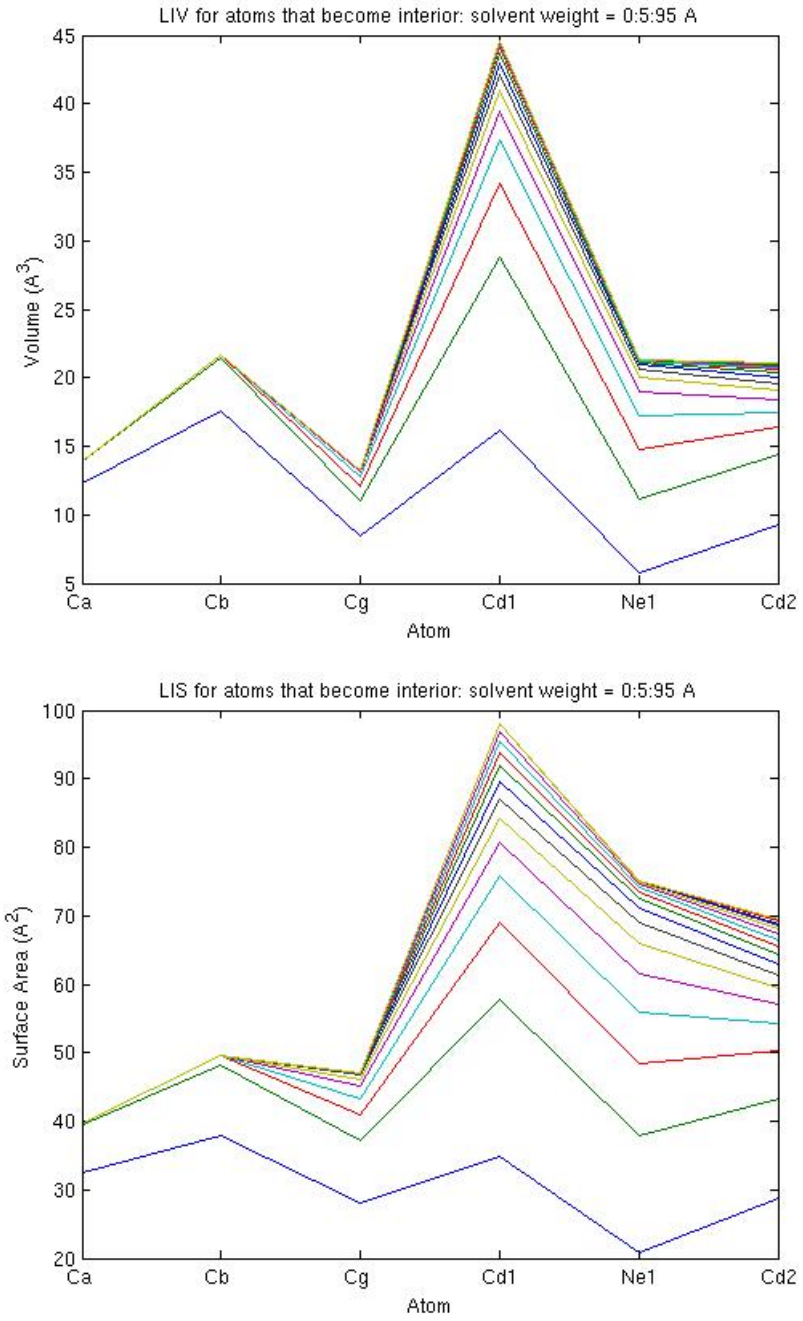


Figure C.6: Laguerre-Intersection volumes and surfaces for increasing values of w. Note that the quantities are monotonic with respect to w and converge to Laguerre volumes and surfaces.

Appendix D

Figures for Optimization of Solvent Parameter

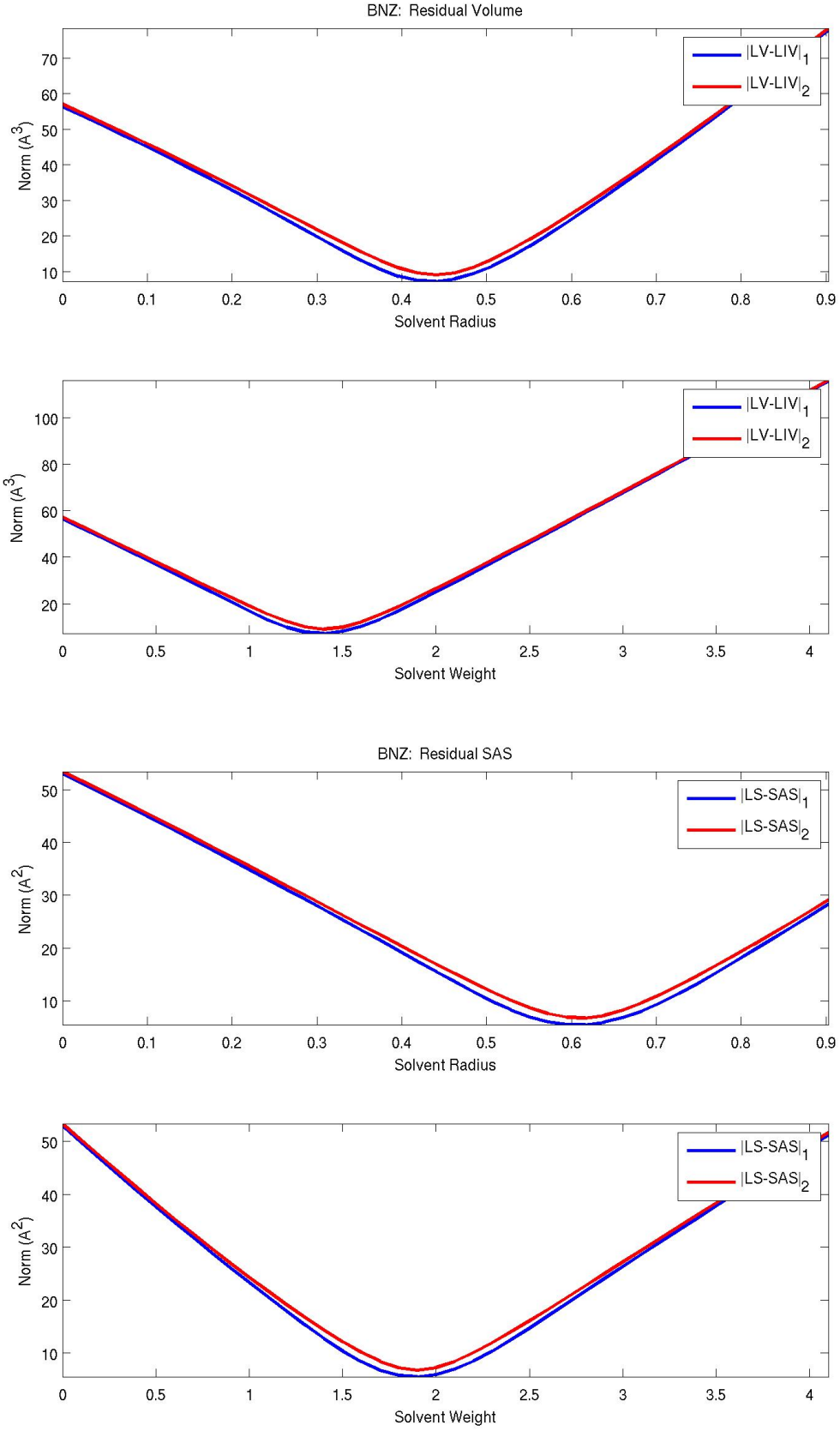


Figure D.1: Top two plots show error curves for the optimization of the solvent parameter for benzene with respect to residual volume. Lower two plots show the optimization of solvent parameter with respect to surface area.

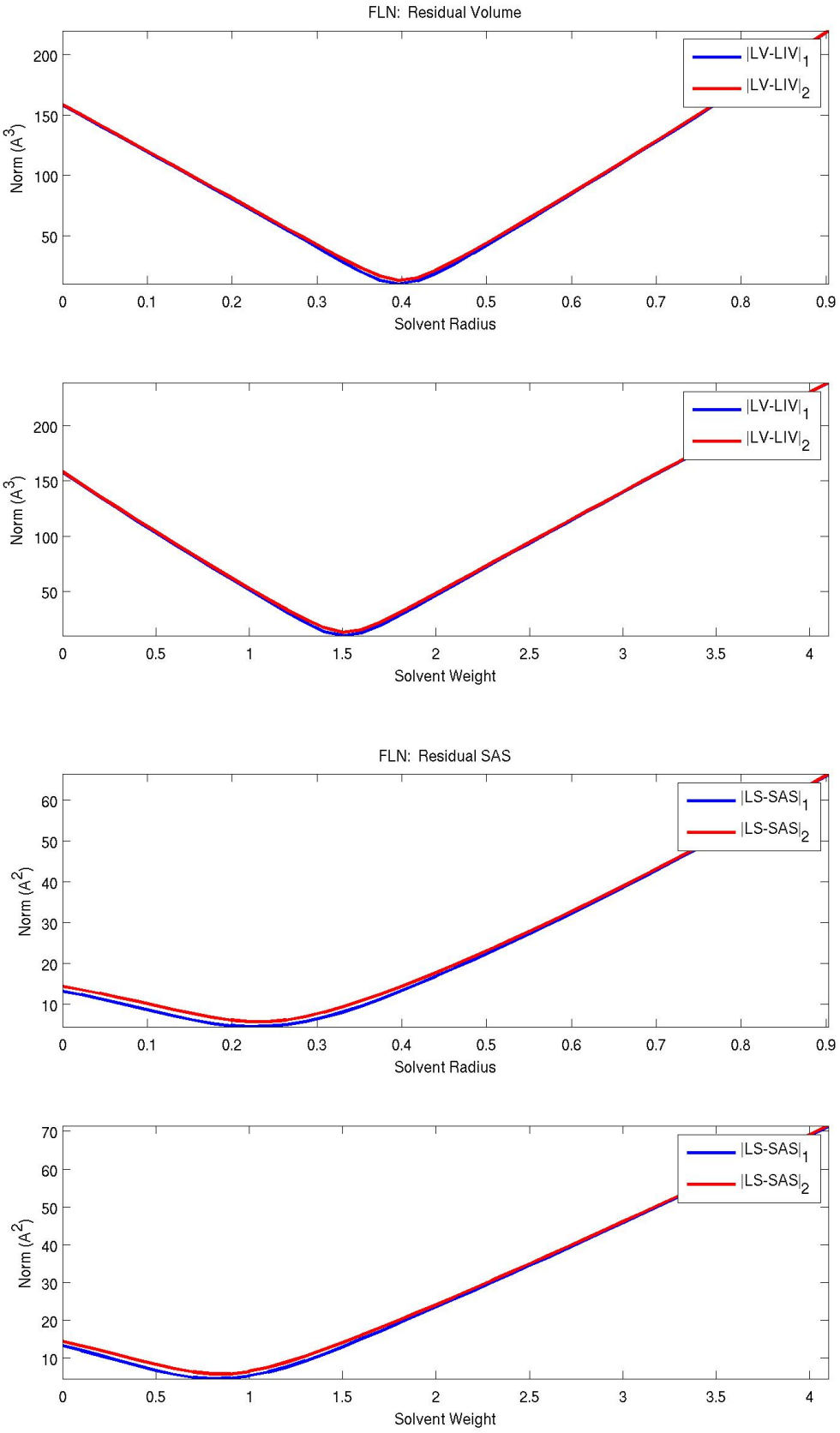


Figure D.2: Top two plots show error curves for the optimization of the solvent parameter for fullerene with respect to residual volume. Lower two plots show the optimization of solvent parameter with respect to surface area.

Appendix E

Glossary

Π

Power product between two weighted points; $\Pi(p, q) = |p' - q'|^2 - p'' - q''$.

π

Power distance between weighted point and unweighted point; $\pi(p, x') = |p' - x'|^2 - p''$.

ρ_T

Size of the simplex σ_T. Equal to the weight of the characteristic point x_T.

σ_T

A simplex which is the convex hull of point centers in set T.

$\phi(p)$

A map from $\mathbb{R}^3 \times \mathbb{R}$ to $\mathbb{R}^4$; $\phi(p) = \overline{p} = (p', |p'|^2 - p'')$.

ϕ^{-1}

Inverse function of ϕ.

A

$\mathcal{A}$

Set of weighted data points (spheres) which represents atoms in a molecule.

$\mathcal{A}'$

Set of data point centers; $\mathcal{A}' = \{p' \text{ such that } p \in \mathcal{A}\}$.

$\mathcal{A}_\alpha$

Set of expanded points, $\mathcal{A}_\alpha = \{p_\alpha \text{ such that } p \in \mathcal{A}\}$.

$\overline{\mathcal{A}}$

The set $\mathcal{A}$ under the map ϕ.

A_i

Individual atomic surface area of atom i. May also be written as $\mathcal{S}^i$.

B

$\mathcal{B}_\alpha$
: Space filling model of $\mathcal{A}_\alpha$. Equal to $\bigcup_{p \in \mathcal{A}} B_{p_\alpha}$.

B_p
: The weighted point p can be thought of as a closed ball B_p with center at p' and weight p''.

$\bigcap B_T$
: Intersection of balls represented by points in T.

$\bigcap B_{T^\alpha}$
: Intersection of balls in T that whose weights are expanded by α.

$\bigcup B_{T^\alpha}$
: Union of balls in T that whose weights are expanded by α.

C

$\mathcal{C}_\alpha(\mathcal{A})$
: Alpha complex of $\mathcal{A}$ for α.

$\partial\mathcal{C}$
: Boundary of the alpha complex $\mathcal{C}$.

D

D
: Displacement gradient.

E

e_{ij}
: Edge connecting the centers of p_i and p_j.

F

F_T
: Surface area of the intersection of the Laguerre facet corresponding to simplex σ_T with the interior of the alpha complex.

H

$H(\overline{p})$
: Polar hyperplane of $\overline{p}$. Equal to $\overline{p}^*$.

H^+
: Space above the hyperplane H.

H^-
: Space below the hyperplane H.

H^*
: Pole of hyperplane H.

I

inc_T
: The set of higher dimensional simplices that are incident to σ_T.

L

$\mathcal{L}(\mathcal{A})$
: Laguerre diagram of the data set $\mathcal{A}$.

$l(t)$
: Set of all weighted points that are mutually orthogonal to all $p \in t$ where t is a triangle. Equivalent to X_t.

$l'(t)$
: Centers of points in the set $l(t)$.

L_i
: Laguerre cell of atom i. Equivalent to L_{p_i}.

$L_i(w)$
: Laguerre cell of $p_i(w)$ in $\mathcal{A}(w)$.

L_{ij}
: Laguerre facet corresponding to edge e_{ij}.

L_p
: The Laguerre cell of p.

$LI_i(w)$
: Laguerre-Intersection cell of atom i or residue i (depending on the context) with weight w.

$LIS_i(w)$
: Surface area of $LI_i(w)$.

$LIS_T^{(i)}$
: Contribution of the simplex σ_T to LIS_i.

$LIV_i(w)$
: Volume of $LI_i(w)$.

$LIV_T^{(i)}$
: Contribution of the simplex σ_T to LIV_i.

LS_i
: Surface area of the Laguerre cell of atom i or the Laguerre surface area of residue i depending on the context.

LS_{ij}
: Surface area of facet L_{ij}.

LV_i
: Volume of the Laguerre cell of atom i or Laguerre volume of residue i depending on the context.

LV_{ij}
: Volume contribution of the edge e_{ij} to the Laguerre cell of atom i.

N

$nnbr\_wdd^+(t)$
: Nearest weighted Delaunay distance neighbor to triangle t in the halfspace P^+.

P

P
: Plane that contains triangle t. P^+ and P^- are the upper and lower halfspaces, respectively.

$p = (p', p'')$
: Weighted point (sphere) in $\mathbb{R}^3 \times \mathbb{R}$ with center p' and weight (radius squared) p''.

p'
: Unweighted point or location in $\mathbb{R}^3$.

p''
: Weight or radius squared of point p. This can also be written as w.

$\overline{p}$
: The point p under the map ϕ.

$\overline{p}^*$
: Polar hyperplane of $\overline{p}$. Equal to $H(\overline{p})$.

p_α
: Expanded point $p_\alpha = (p', p'' + \alpha)$.

$p_i(w)$
: Expanded point that is equivalent to $(p_i)_w$.

$P_T^{(i)}$
: Pyramidal volume contribution of σ_T to LIV_i.

Q

$\overline{q}$
: Equal to the point $r(\overline{p})$.

R

$r(\overline{p})$
: Vertical reflection of $\overline{p}$ over the parabola X^2.

S

$\vec{\mathcal{S}}$

Vector in which the ith element is $\mathcal{S}^i$.

$\mathcal{S}(\mathcal{A})$

Surface area of the molecule represented by $\mathcal{A}$.

$\mathcal{S}^i$

Accessible surface area of atom corresponding to point p_i or equivalently the contributions of atom i to total surface area $\mathcal{S}$.

S_p

Boundary of the ball B_p.

S_T

Surface area of the intersection of balls represented by points in T. Equal to $surf(\bigcap S_T)$.

$S_T^{(i)}$

The contribution of S_T to $\mathcal{S}^i$.

$\bigcap S_{T^\alpha}$

Intersection of spheres corresponding to points in T whose weights are expanded by α.

$\bigcap Surf_T$

Intersection of the surfaces of balls represented by points in T.

T

$\mathcal{T}(\mathcal{A})$

Weighted Delaunay (Regular) tetrahedrization of the data set $\mathcal{A}$.

T

In Chapters 1-9, subset of $\mathcal{A}$. We usually consider $|T| \leq 4$. In Chapters 10-14, a tetrahedron.

$|T|$

Number of elements in the set T.

T_a

Tetrahedron consisting of points in a triangle t and the point p_a; $T_a = \{t, p_a\}$.

t

A triangle.

V

$\mathcal{V}(\mathcal{A})$

Volume of the molecule represented by $\mathcal{A}$.

V_T

Volume of the intersection of balls represented by points in T. Equal to $vol(\bigcap B_T)$. In Chapter 14, the volume of tetrahedron T.

W

wdd
: Weighted Delaunay distance.

X

X^2
: Parabola $x_4 = x_1^2 + x_2^2 + x_3^2$.

X_T
: The set of points which are equipowerdistant to all points in T.

x_T
: Characteristic point of the simplex σ_T. $x_T = (x'_T, x''_T)$ where x'_T is the center of the characteristic point and the weight x''_T is the size of the simplex.

x_{ij}
: Characteristic point of the edge e_{ij}.

Bibliography

[1] Thanks to Carlos Simmerling for sharing unpublished explicit water molecular dynamics trajectory data. Simulations were carried out using the AMBER molecular modeling program.

[2] Javier Bernal, *Regtet: A program for computing regular tetrahedralizations*, National Institute of Standards and Technology Internal Report **6786** (2001).

[3] J-D. Boissonnat and M. Yvinec, *Algorithmic geometry*, Cambridge University Press, Cambridge, UK, 1998.

[4] Robert Bryant, Herbert Edelsbrunner, Patrice Koehl, and Michael Levitt, *The area derivative of a space-filling diagram*, Discrete and Computational Geometry (2004).

[5] P. Cignoni, C. Montani, and R. Scopigno, *Dewall: A fast divide and conquer Delaunay triangulation algorithm in ed*, Computer-Aided Design **30** (1998), 5: 333–341.

[6] H. Edelsbrunner and N.F. Shah, *Incremental topological flipping works for regular triangulations*, Proceedings of the eight annual symposium on Computation geometry, Berlin, Germany, June 10-12, 1998.

[7] Herbert Edelsbrunner, *The union of balls and its dual shape*, Proceeding of the ninth annual symposium on Computational Geometry, San Diego, CA, 1993.

[8] Herbert Edelsbrunner, Michael Facello, and Jie Liang, *On the definition and the construction of pockets in macromolecules*, Discrete Applied Mathematics **88** (1998), 83–102.

[9] Herbert Edelsbrunner and Ping Fu, *Measuring space filling diagrams and voids*, Proc. 28th Ann. HICSS (1995).

[10] Herbert Edelsbrunner and Patrice Koehl, *The geometry of biomolecular solvation*, Discrete and Computational Geometry: MSRI publications **52**.

[11] ———, *The weighted-volume derivative of a space-filling diagram*, PNAS **100** (2003), 2203–2208.

[12] Herbert Edeslbrunner, *Weighted alpha shapes*, Technical Report (1992).

[13] David Eisenberg and Andrew D. McLachlan, *Solvation energy in protein folding and binding*, Nature **319** (1986), 199–203.

[14] Jeremy Esque, Christophe Oguey, and Alexandre G. de Brevern, *A novel evaluation of residue and protein volumes by means of Laguerre tessellation*, J. Chem. Inf. Model **50** (2010), 947–960.

[15] _____, *Comparative analysis of threshold and tessellation methods for determining protein contacts*, Journal of Chemical Information and Modeling **51** (2011), 493–507.

[16] Tsung-Pao Fang and Les A. Piegl, *Delaunay triangulation in three dimensions*, Computer Graphics and Applications, IEEE **15** (1995), 5:62–69.

[17] Christopher J. Fennell, Charles W. Kehoe, and Ken A. Dill, *Oil/water transfer is partly driven by molecular shape, not just size*, J Am Chem Soc. **132** (2010), 234–240.

[18] _____, *Modeling aqueous solvation with semi-explicit assembly*, Proceedings of the National Academy of Sciences of the United States of America (2011), www.pnas.org/cgi/doi/10.1073/pnas.1017130108.

[19] Mark Gerstein, Jerry Tsai, and Michael Levitt, *The volume of atoms on the protein surface: Calculated from simulation, using Voronoi polyhedra*, J. Mol. Biol. **249** (1995), 955–966.

[20] K.D. Gibson and Harold A. Scheraga, *Volume of the intersection of three spheres of unequal size: a simplified formula*, Journal of Physical Chemistry **91**, 4121–4122.

[21] Jyrki Katajainen and Karkku Koppinen, *Constructing Delaunay triangulations by merging buckets in quadtree order*, Fundamenta Informaticae **XI** (1988), 275–288.

[22] Libo Li, Christopher J. Fennell, and Ken A. Dill, *Field-sea: A model for computing the solvation free energies of nonpolar, polar, and charged solutes in water*, The Journal of Physical Chemistry **118** (2013), 6431–6437.

[23] Jie Liang, *Computation of protein geometry and its applications: Packing and function prediction*, Cornell University Library, arXiv (2006), q–bio/0601020v1 [q–bio.BM].

[24] Jie Liang and Ken A. Dill, *Are proteins well-packed?*, Biophysical Journal **81** (2001), 751–766.

[25] Jie Liang, Herbert Edelsbrunner, Ping Fu, and Pamidighantam V. Sudhakar, *Analytical shape computation of macromolecules: II. inaccessible cavities in proteins*, PROTIENS: Structure, Function and Genetics **33** (1998), 18–29.

[26] Jie Liang, Herbert Edelsbrunner, Ping Fu, Pamidighantam V. Sudhaker, and Shankar Subramaniam, *Analytical shape computation of macromolecules: I. molecular area and volume through alpha shapes*, PROTEINS: Structure, Function, and Genetics **33** (1998), 1–17.

[27] George Marsaglia, *Choosing a point from the surface of a sphere*, The Annals of Mathematical Statistics **43** (1972), 645–646.

[28] B.J. McConkey, V. Sobolev, and M. Edelman, *Quantification of protein surfaces, volumes and atom-atom contacts using a constrained Voronoi procedure*, Bioinformatics **18** (2002), 1365–1373.

[29] Ray N, Cavier X, Paul JC, and Maigret B, *Intersurf: dynamic interface between proteins*, J Mol Graph Model **23** (2005), 347–354.

[30] Kliment Olechnovic, Eleonora Kulberkyte, and Ceslovas Venclovas, *Cad-score: A new contact area difference-based function for evaluation of protein structural models*, Proteins **81** (2013), 149–162.

[31] S.H. Park, S.S. Lee, and J.H. Kim, *The Delaunay triangulation by grid subdivision*, Lecture Notes in Computer Science **3482** (2005), 385–429.

[32] Nicolas Rayand, Xavier Cavin, Inria Lorraine, and Bernard Maigret, *Interactive poster: visualizing the interaction between two proteins*.

[33] J.F. Sadoc, R. Jullien, and N. Rivier, *The Laguerre polyhedral decomposition: application to protein folds*, The European Physical Journal B **33** (2003), 355–363.

[34] Alain Soyer, Jaques Chomilier, Jean-Paul Mornon, Remi Jullien, and Jean-Francois Sadoc, *Voronoi tessellation reveals the condensed matter character of folded proteins*, Physical Review Letters **85** (2000), 3532–3535.

[35] F. Vanhoutte, *www.wblut.com/2009/04/28/ooh-ooh-ooh-3d-voronoi/*, Accessed: 08-02-2014.

[36] Ryszard J. Wawak, Kenneth D. Gibson, and Harold A. Scheraga, *Gradient discontinuities in calculations involving molecular surface area*, Journal of Mathematical Chemistry **15** (1994), 207–232.

[37] Stephen H. White, *Experimentally determined hydrophobicity scales*, Stephen White Laboratory Homepage (2011), Figure 2.

[38] Wikipedia, *Dodecahedron*, 2014, Image available from: http://en.wikipedia.org/wiki/Dodecahedron.

[39] _____, *Octahedron*, 2014, Image available from: http://en.wikipedia.org/wiki/Octahedron.

[40] William C. Wimley and Stephen H. White, *Experimentally determined hydrophobicity scale for proteins at membrane interfaces*, Nature Structural Biology **3** (1996), 842–848, Figure 2.

[41] Takahisa Yamato, Junichi Higo, Yasunobu Seno, and Nobuhiro Go, *Conformational deformation in deoxymyoglobin by hydrostatic pressure*, PROTEINS: Structure, Function, and Genetics **16** (1993), 327–340.

[42] Afra J. Zomorodian, *Topology for computing*, Cambridge University Press, Cambridge, UK, 2009.