Institut für Numerische Simulation
Rheinische Friedrich-Wilhelms-Universität Bonn

Wegelerstraße 6. 53115 Bonn. Germany
phone +49 228 73-3427. fax +49 228 73-7527
www.ins.uni-bonn.de

Maharavo Randrianarivony

Cavity generation of Connolly surfaces for subsequent patchwise chemical Polarizable Continuum Model simulation

INS Preprint No. 1001

March 2010

CAVITY GENERATION OF CONNOLLY SURFACES FOR SUBSEQUENT PATCHWISE CHEMICAL POLARIZABLE CONTINUUM MODEL SIMULATION

MAHARAVO RANDRIANARIVONY

Abstract. We focus on molecular cavities which are represented in the form of Solvent Excluded Surfaces. They are featured by having smooth toroidal and spherical surface blendings between the constituting atoms. The only inputs are the nuclear coordinates, the van der Waals radii and the probe radius. From those information, our purpose is to generate a patch representation of the resulting Connolly surface. We aim at a tessellation which is conforming. The patches are represented as NURBS which are globally continuous. The resulting patches are nicely shaped and neighboring patches have comparable surface areas. An automatic way to make all normal vectors outwardly oriented is provided. We present several practical results obtained from real PDB files. We examine also the influence of different parameters to the number of patches.

1. Introduction

Generating molecular surface cavities is not only an interesting geometric task but it has a real important application in quantum chemistry, molecular docking and synthetic drug design. In this document, we address the problem of modeling cavities for later use in PCM simulation. Hierarchical methods such as multigrid, multilevel, multiscale and wavelets are theoretically known to perform very efficiently. Still, their practical implementation is not yet mature especially in the domain of hierarchical BEM for chemical computation. That is mainly caused by the geometric difficulty of generating hierarchical spaces. To solve such a difficulty, we want to advance the availability of geometric methods in the perspective of chemical simulation. That is, we intend to generate a patch representation of chemical Connolly surfaces [4, 10]. Implementing a program for generating an SES surface beginning from a set of nuclear coordinates is not straightforward because a lot of geometric tasks come into play. It is a long process to start from those coordinates till obtaining the control points of the splines. In this paper, we will not describe all the necessary steps in full detail. Although the geometric tasks which are presented here are mainly for the purpose of subsequent Wavelet simulations, we believe that it can be very helpful for the implementations of other simulation techniques.

Key words and phrases. Connolly, cavity generation, patches, PCM.

Before going into details, a short survey of related past works is in order. Brunnett and Randrianarivony have proposed [13] a splitting method for CAD surfaces. They have also invested a lot to implement their methods by using the IGES format [19]. Additionally, they have proved methods for checking regularity of Coons maps. But they did not treat the global continuity of the resulting patches. The main task in [15] is the correlation between the Coons interpolation which resides in an individual patch and the global continuity. While *approximations* are required to obtain global continuity in [15] for CAD objects, it can be achieved *exactly* for molecular surfaces in [8, 14]. That is due to the fact that both circular arcs and spherical patches can be exactly represented as rational Bézier entities. Harbrecht and Randrianarivony [7, 8] have successfully applied Wavelet methods on CAD as well as molecular models in form of the van der Waals setting. As inputs, they accept a CAD file in an IGES format or a molecular model in a PDB format. Furthermore, a real chemical simulation by using wavelet BEM is described by Weijo *et al* in [20] for the PCM computation.

This paper is structured as follows. In the next section, we begin by introducing the notion of Connolly surface of a given molecule in a solute-solvent interaction. We recall there also the notion of NURBS entities and their treatment by means of the homogeneous coordinates. In Section 3, we show the parametrization of the trimmed surfaces which constitute a Connolly surface. The decomposition of a Connolly surface into four-sided domains will be the purpose of Section 4. Afterwards, we describe in Section 5 the representation of the patches in terms of NURBS functions which are globally continuous. In addition, we want that all normal vectors of the patches are pointing consistently outward. Finally, we present some practical results of our implementation where the molecules come from real PDB files.

2. Definitions and problem formulation

2.1. **Connolly surface.** We address the problem of Connolly representation of a cavity [1] which is the surface separating the solvent from the solute. The solvent is represented by a constant dielectric medium while the solute is located inside the cavity. The cavity comes from the boundary of a molecule where each constituting atom is represented as an imaginary sphere whose center $\mathbf{m}_k$ corresponds to the nuclear coordinates and whose radius r_k to the van der Waals radius of the atom [21] or to a multiple of it. That is, by denoting the sphere of center $\mathbf{m}$ and radius r by $B(\mathbf{m}, r) := \left\{\mathbf{x} \in \mathbb{R}^3 : \|\mathbf{m} - \mathbf{x}\| \leq r\right\}$ where $\|\cdot\|$ denotes the Euclidean norm, the molecule is represented as the union of N spheres

$$\Omega = \bigcup_{k=1}^{N} B(\mathbf{m}_k, r_k).$$

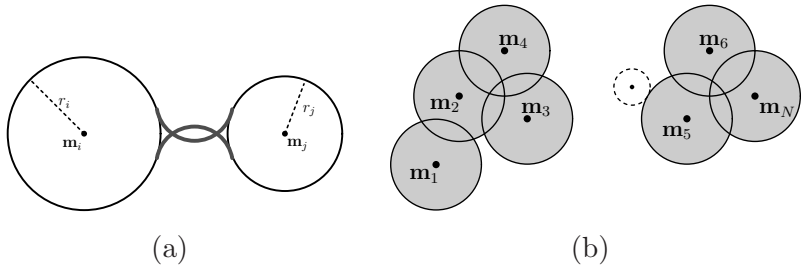


FIGURE 1. (a)Inadmissible self-intersection: the two arcs of radii ρ which are tangent to the two circles have intersections, (b)Inadmissible disconnected adjacency graph.

Apart from the initial molecule, we need also a probe atom which is required for the generation of the Connolly surface. Before formulating the problem setting, let us specify the assumptions concerning the positions of the nuclei $\mathbf{m}_i$ and the properties of the radii r_i with respect to the probe radius ρ. For every two arbitrary atoms $B(\mathbf{m}_i, r_i)$ and $B(\mathbf{m}_j, r_j)$, we assume that one of the following two conditions holds.

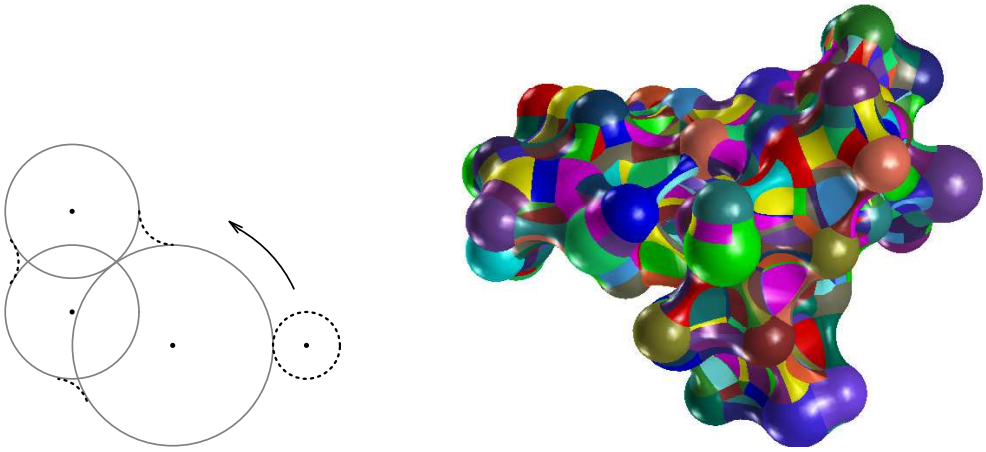
(C1) Either the two enlarged spheres $B(\mathbf{m}_i, r_i + \rho)$ and $B(\mathbf{m}_j, r_j + \rho)$ by the probe radius ρ are completely disjoint such as $\|\mathbf{m}_i - \mathbf{m}_j\| > r_i + r_j + 2\rho$,
(C2) or we have $D_{ij} := \|\mathbf{m}_i - \mathbf{m}_j\| \leq r_i + r_j + 2\rho$ and additionally

$$2D_{ij}r_i^2 + 4D_{ij}\rho - D_{ij}^2 - (r_i + \rho)^2 + (r_j + \rho)^2 > 0.$$

As a consequence, those two assumptions exclude the situation in Fig. 1(a) where the blending torus which is tangent to both $B(\mathbf{m}_i, r_i)$ and $B(\mathbf{m}_j, r_j)$ admits a self-intersection. In addition, we define the *adjacency graph* as follows. The nodes of the graph correspond to the nuclear coordinates. An edge is defined between two graph nodes if the distance between the corresponding two nuclei $\mathbf{m}_i$ and $\mathbf{m}_j$ is smaller than $r_i + r_j + 2\rho$. Our next assumption is that the adjacency graph has only one connected component. As a consequence, the situation illustrated in Fig. 1(b) is excluded.

If assumptions (C1) and (C2) are not fulfilled but one still wants to treat the molecule, one can insert some dummy atoms [12] between every two atoms $\mathbf{m}_i$ and $\mathbf{m}_j$ for which there is a toroidal self-intersection. Note that such an insertion might lead to a further conflict with a third atom $\mathbf{m}_k$. Some careful strategy is essential for the determination of the size and the positions of the dummy atoms whose number should be minimized while keeping the shape of the initial molecule.

The SES model (Surface Excluded Surface), which is also known as Connolly surface, is the surface Γ traced by the probe atom when it is rolled over (see Fig. 2(a)) the



(a) (b)

FIGURE 2. (a)Rolling a probe atom on the molecular surface (b)Connolly surface: composition of trimmed toroidal and spherical surfaces

whole surface

$$\mathcal{S} := \partial\Omega = \partial\Big[\bigcup_{k=1}^{N} B(\mathbf{m}_k, r_k)\Big]. \tag{2.1}$$

The Connolly surface Γ is partly from $\mathcal{S}$ and partly from the blending surfaces traced by the probe atom. The blending surfaces are composed of surfaces of two types. The first type of blending surfaces are toroidal surfaces while the second one is trimmed spherical surfaces. In fact, depending on the position of the atoms, several cases may happen. First, the probe atom is incident upon two atoms between which it can roll. Note that the probe atom can connect two atoms which are completely disjoint. Second, the probe atom is adjacent to more than two atoms where it is fixed. In the first cases, toroidal surfaces are traced by the probe atom. For the second case, spherical surfaces are traced. An illustration of a Connolly surface is depicted in Fig. 2(b). Note that the size of the probe atom can really affect the topology of the whole molecular surface. In chemical applications, the radius of the probe atom is usually chosen between $1.0\mathring{A}$ and $3.0\mathring{A}$ but for our method it can be any positive number. When the probe radius becomes very large, some of the initial atoms might be completely buried inside the whole surface as in Fig. 7(d). When the probe radius approaches zero, then the SES surface practically coincides with the van der Waals surface. As a consequence, the presented method here can be used as an alternative to treat van der Waals surfaces as in Fig.7(a).

2.2. **NURBS, Bézier and homogeneous coordinates.** Since we intend to use NURBS as representation of the Connolly surface, let us introduce the B-spline basis of piecewise polynomials. Consider any constant integer $k \geq 2$ which specifies the smoothness of the spline and a knot sequence $\zeta_0 \leq \zeta_1 \leq \cdots \leq \zeta_{n+k}$ such that $\zeta_{i+k} \neq \zeta_i$. The B-spline basis [9] functions with respect to the knot sequence $(\zeta_i)_i$ verify the following recurrence relation

$$N_i^1(t) := \begin{cases} 1 & \text{if} \quad t \in [\zeta_i, \zeta_{i+1}), \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

$$N_i^k(t) := \left(\frac{t-\zeta_i}{\zeta_{i+k-1}-\zeta_i}\right) N_i^{k-1}(t) + \left(\frac{\zeta_{i+k}-t}{\zeta_{i+k}-\zeta_{i+1}}\right) N_{i+1}^{k-1}(t). \tag{2.3}$$

To ensure that the B-spline functions are open, we assume that the knot sequence is *clampled*. That is, the initial and the final entries of the knot sequence are provided as follows:

$$\zeta_0 = \cdots = \zeta_{k-1}, \tag{2.4}$$

$$\zeta_{n+1} = \cdots = \zeta_{n+k}. \tag{2.5}$$

A NURBS (Non-Uniform Rational B-Spline) curve [9, 11] having control points $d_i \in \mathbb{R}$ and weights $w_i \in \mathbb{R}^+$ with respect to the above knot sequence is a parametric curve of the form

$$\mathbf{X}(t) = \frac{\sum_{i=0}^n w_i \mathbf{d}_i N_i^k(t)}{\sum_{i=0}^n w_i N_i^k(t)} \qquad \forall\, t \in [\zeta_0, \zeta_{n+k}]. \tag{2.6}$$

The above two assumptions (2.4) and (2.5) ensure that the initial and final control points $\mathbf{d}_0$ and $\mathbf{d}_n$ are interpolated. An illustration can be found in Fig. 3(a) where the curve diverges away from a control point $\mathbf{d}_i$ as the corresponding weight w_i becomes smaller.

Similarly, for a NURBS surface we need two clamped knot sequences ζ_i^1 and ζ_i^2. A NURBS surface is defined as

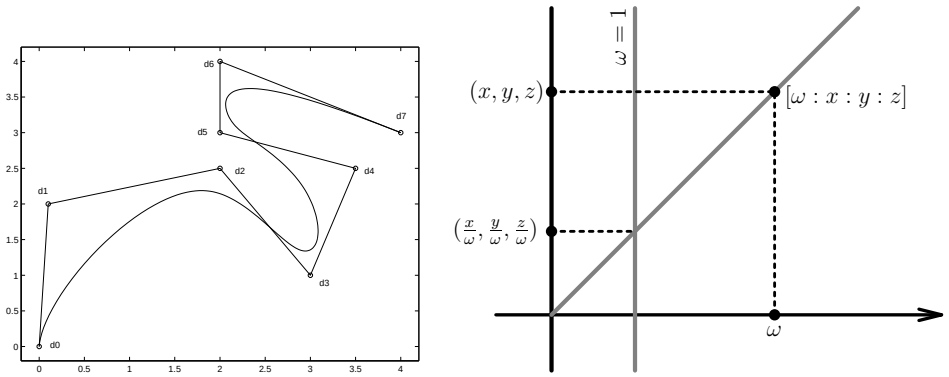
$$\mathbf{X}(u,v) = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} \mathbf{d}_{ij} N_i^{k_1}(u) N_j^{k_2}(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} N_i^{k_1}(u) N_j^{k_2}(v)} \qquad \forall (u,v) \in [\zeta_0^1, \zeta_{n+k_1}^1] \times [\zeta_0^2, \zeta_{m+k_2}^2]. \tag{2.7}$$

Rational Bézier curves and surfaces are defined similarly with the only exception that one uses the Bernstein basis $B_i^n(t) := \binom{n}{i} t^i (1-t)^{n-i}$ instead of piecewise polynomials. That is, a rational Bézier curve with weights w_i and control points $\mathbf{d}_i = (x_i, y_i, z_i)$ is

$$\mathbf{X}(t) = \frac{\sum_{i=0}^n w_i \mathbf{d}_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)} \qquad \forall\, t \in [0,1]. \tag{2.8}$$

A rational Bézier surface is

$$\mathbf{X}(u,v) = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} \mathbf{d}_{ij} B_i^n(u) B_j^m(v)}{\sum_{i=0}^n \sum_{j=0}^m w_{ij} B_i^n(u) B_j^m(v)} \qquad \forall\, (u,v) \in [0,1]^2. \tag{2.9}$$



(a) (b)

FIGURE 3. (a) NURBS curve with $n = 7$, $k = 4$, knot sequence $\boldsymbol{\zeta} = (0, 0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1, 1)$ and weights $\mathbf{W} = (1.4, 0.5, 1.6, 1.8, 0.7, 1.9, 1.5, 0.9)$, (b) The homogeneous coordinates $[\omega : x : y : z]$ represent the Cartesian point $(x/\omega, y/\omega, z/\omega)$

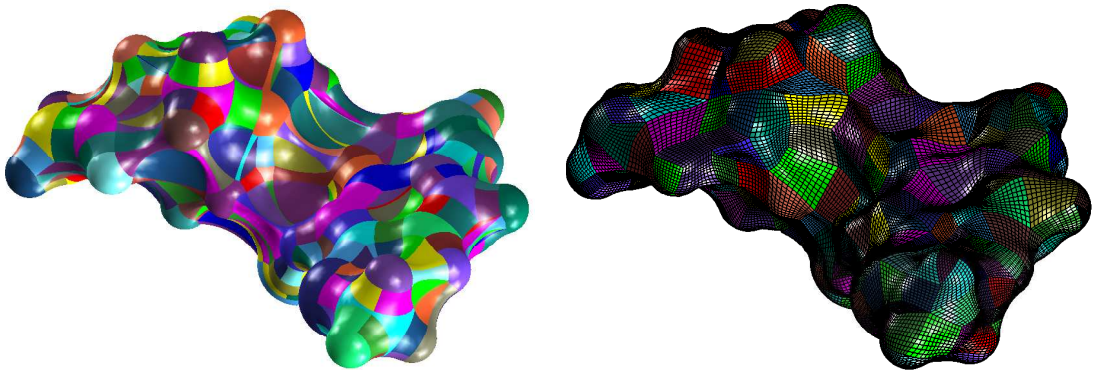
Using homogeneous coordinates considerably simplifies theoretical formulations because rational quantities become polynomial ones which make problems linear instead of nonlinear. An element of the projective space $\mathbb{E}^3$ will be denoted as a column vector with four coordinates or as row vector whose elements are separated by colons. A point with homogeneous coordinates $[\omega : x : y : z]$ corresponds to the Cartesian coordinates $(x/\omega, y/\omega, z/\omega)$. Notice that the homogeneous coordinates $[\omega : x : y : z]$ and $[\lambda\omega : \lambda x : \lambda y : \lambda z]$ represent for any $\lambda \neq 0$ the same point in Cartesian coordinates. The above definitions and properties are illustrated in Fig. 3(b).

A NURBS curve corresponds to a B-spline curve having the homogeneous coordinates $[\omega_i : \omega_i x_i : \omega_i y_i : \omega_i z_i]$ in the projective space. Thus, in homogeneous coordinates the above Bézier curve is represented as

$$\mathbf{X}(t) = \sum_{i=0}^{n} [\omega_i : \omega_i x_i : \omega_i y_i : \omega_i z_i] N_i^k(t)$$

which is a linear combination of the B-spline basis functions $N_i^k(t)$. Vice versa, the homogeneous B-spline curve $\mathbf{X}(t) = \sum_{i=0}^{n} [\alpha_i : \beta_i : \gamma_i : \delta_i] N_i^k(t)$ corresponds uniquely (except for scaling of enumerator and denominator) to a NURBS curve (2.6) with $\omega_i := \alpha_i$ and $\mathbf{d}_i := (\beta_i/\alpha_i, \gamma_i/\alpha_i, \delta_i/\alpha_i)$. Likewise, a NURBS surface $\mathbf{X}(\cdot, \cdot)$ which has the control points $\mathbf{d}_{ij} = (x_{ij}, y_{ij}, z_{ij})$ and the weights ω_{ij} (for $i = 0, 1, \ldots, n$ and $j = 0, 1, \ldots, m$) can be represented in homogeneous coordinates as

$$\mathbf{X}(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} [\omega_{ij} : \omega_{ij} x_{ij} : \omega_{ij} y_{ij} : \omega_{ij} z_{ij}] N_i^{k_1}(u) N_j^{k_2}(v).$$



(a) (b)

FIGURE 4. Connolly surface of a quinine molecule with probe radius $\rho = 1.2$: (a)Trimmed surfaces from a cloud of nuclear coordinates (b)Decomposition and globally continuous NURBS parametrization.

2.3. **Problem setting.** The main objective of the current cavity generation is to find a set of globally continuous NURBS surfaces that approximate the SES surface. For each NURBS patch as in (2.7), we will assume that $[\zeta_0^1, \zeta_{n+k}^1] \times [\zeta_0^2, \zeta_{n+k}^2]$ is the unit square $\square := [0,1]^2$. That is, the Connolly surface $\Gamma \subset \mathbb{R}^3$ will be decomposed into a finite number of patches

$$\Gamma = \bigcup_{i=1}^{M} \Gamma_i, \qquad \Gamma_i = \boldsymbol{\gamma}_i(\square), \qquad i = 1, 2, \ldots, M, \tag{2.10}$$

such that each $\boldsymbol{\gamma}_i : \square \to \Gamma_i$ is described by a bivariate NURBS function. The intersection of two different patches Γ_{i_1} and Γ_{i_2} is supposed to be either $\emptyset$, or a common edge or vertex.

A mesh of level j on Γ is induced by dyadic subdivisions of depth j of the unit square into 4^j cubes $C_{j,\mathbf{k}} \subseteq \square$, where $\mathbf{k} = (k_1, k_2)$ with $0 \le k_1, k_2 < 2^j$. This generates $4^j M$ *elements* (or elementary domains) $\Gamma_{i,j,\mathbf{k}} := \boldsymbol{\gamma}_i(C_{j,\mathbf{k}}) \subseteq \Gamma_i$, $i = 1, \ldots, M$.

In order to ensure that the collection of elements $\{\Gamma_{i,j,\mathbf{k}}\}$ on the level j forms a regular mesh on Γ, the parametric representation is subjected to the following *matching condition*: a bijective, affine mapping $\boldsymbol{\Xi} : \square \to \square$ exists such that for all $\mathbf{x} = \boldsymbol{\gamma}_i(\mathbf{s})$ on a common edge of Γ_{i_1} and Γ_{i_2} it holds that

$$\boldsymbol{\gamma}_{i_1}(\mathbf{s}) = (\boldsymbol{\gamma}_{i_2} \circ \boldsymbol{\Xi})(\mathbf{s}). \tag{2.11}$$

In other words, the NURBS functions $\boldsymbol{\gamma}_{i_1}$ and $\boldsymbol{\gamma}_{i_2}$ coincide pointwise at common edges up to some reorientation.

An instance of such a geometric processing for the quinine molecule is shown in Fig. 4 where the curves are the images of u-isolines and v-isolines from the unit square $\square$. Observe that some patches are allowed to be supported by several atoms. That is, one can merge some parts from different trimmed surfaces in order to form one patch. As a consequence, the size and the shape of the patches are in general very nice. That is a geometric advantage of the Connolly surfaces over the usual van der Waals surfaces where each patch must belong to one atom. That is caused by the fact the transition between two atoms is not smooth in the case of van der Waals surfaces [8].

3. Trimmed surfaces from nuclear coordinates

In this section, we consider the generation of the different constituents of the B-Rep of the SES surfaces. For that, our starting position is a cloud of points $\{\mathbf{m}_i\}$ together with the radii r_i. Note that the whole B-rep model is composed of two types of surfaces: spherical and toroidal trimmed surfaces while there is only one type of curves: circular arcs.

3.1. **Weighted space tessellation.**

First, let us introduce some nomenclatures. For two spheres $\mathcal{B}_1 := B(\mathbf{m}_1, r_1)$ and $\mathcal{B}_2 := B(\mathbf{m}_2, r_2)$, we define their power distance as

$$D(\mathcal{B}_1, \mathcal{B}_2) := \|\mathbf{m}_1 - \mathbf{m}_2\|^2 - r_1^2 - r_2^2. \tag{3.12}$$

That distance coincides to the usual Euclidean distance if points are supposed to be spheres of zero radii. Two spheres are called *orthogonal* if we have

$$D(\mathcal{B}_1, \mathcal{B}_2) = 0. \tag{3.13}$$

Consider any simplex $\Delta \subset \mathbb{R}^3$ which is a segment (resp. triangle, tetrahedron) if it is of dimension $d = 1$ (resp. $d = 2$, $d = 3$). Suppose that the vertices of Δ are $\mathcal{B}_i = B(\mathbf{m}_i, r_i)$ where $i = 0, ..., d$. The orthosphere of Δ is defined to be the smallest sphere b such that one has the orthogonality relations

$$D(\mathcal{B}_i, b) = 0, \qquad i = 0, \cdots, d. \tag{3.14}$$

The center and the radius of an orthosphere are termed *orthocenter* and *orthoradius*. Consider a set of spheres $\mathcal{B}_i$ where $i = 1, ..., N$. The i-th Laguerre cell is composed of points which are closer to $\mathcal{B}_i$ than to any other $\mathcal{B}_j$ $(j \neq i)$ with respect to the power distance:

$$C_i := \big\{\mathbf{x} \in \mathbb{R}^3 :\ D\big(B(\mathbf{x}, 0), \mathcal{B}_i\big) \leq D\big(B(\mathbf{x}, 0), \mathcal{B}_j\big) \qquad \forall\, j \neq i\big\}. \tag{3.15}$$

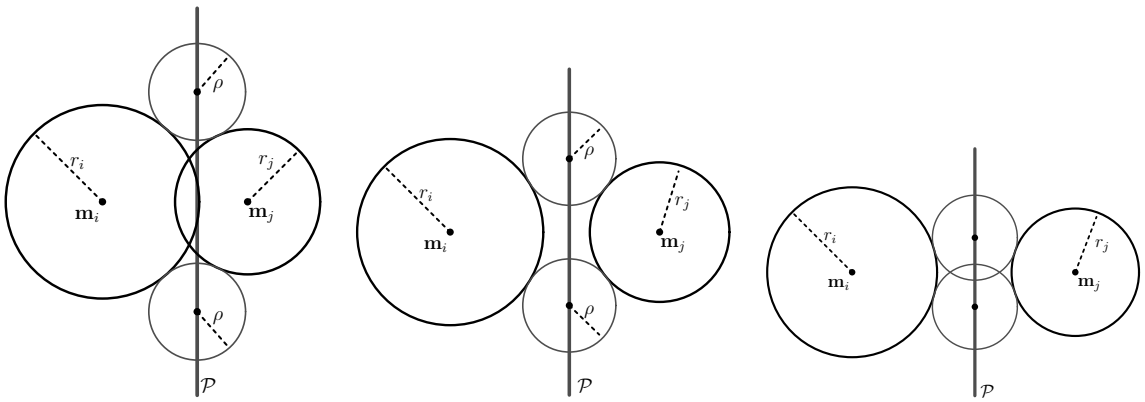


FIGURE 5. The plane $\mathcal{P}$ is a face from the Laguerre decomposition: (a) the spheres $\mathcal{B}_i$ and $\mathcal{B}_j$ are overlapping, (b) they are disjoint but the blending torus has no self-intersection (c) inadmissible toroidal self-intersection.

A Laguerre decomposition of $\mathbb{R}^3$ with respect to the spheres $\mathcal{B}_i$ is

$$\mathbb{R}^3 = \bigcup_{i=1}^{N} C_i. \tag{3.16}$$

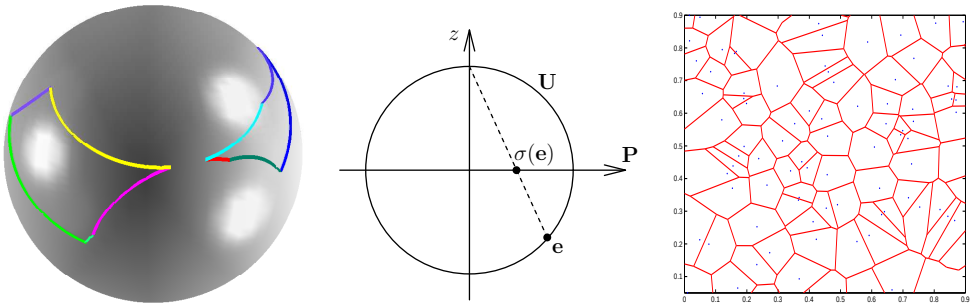
If all radii are equal, then the Laguerre decomposition coincides with the usual Voronoi decomposition as illustrated in Fig.6(c). For two spheres $\mathcal{B}_i$ and $\mathcal{B}_j$, the *radical axis* is the set of points which are equidistant to $\mathcal{B}_i$ and $\mathcal{B}_j$ with respect to the power distance. Such a set is a plane given by

$$R(\mathcal{B}_i, \mathcal{B}_j) = \Big\{ \mathbf{x} \in \mathbb{R}^3 : 2\langle \mathbf{x}, \mathbf{m}_i - \mathbf{m}_j \rangle = \|\mathbf{m}_i\|^2 - \|\mathbf{m}_j\|^2 + r_j^2 - r_i^2 \Big\}. \tag{3.17}$$

A Laguerre cell is a convex polyhedron which has faces from the radical axes and which could be bounded or unbounded. It is impractical to generate the radical axes and compute their intersections to find the Laguerre decomposition because that leads to plane-plane intersections which give an unstable algorithm because not all $R(\mathcal{B}_i, \mathcal{B}_j)$ for $i \neq j$ are present in the decomposition (3.16). Instead, to obtain the Laguerre decomposition, one considers the uplifting function for $\mathbf{m}_i = (x_i, y_i, z_i)$

$$\widetilde{\mathbf{m}}_i := \big(x_i, y_i, z_i, x_i^2 + y_i^2 + z_i^2 + r_i^2\big) \in \mathbb{R}^4. \tag{3.18}$$

One generates the convex hull $\mathcal{H}$ of the set of four dimensional points $\{\widetilde{\mathbf{m}}_i\}$. The projection of the lower face of $\mathcal{H}$ on the space $\mathbb{R}^3$ generates a weighted Delaunay tetrahedral decomposition [6] having apices $\mathbf{m}_i$. Each apex of the Laguerre cell is the orthocenter of such a tetrahedron of the weighted Delaunay. The Laguerre decomposition is obtained as the dual of the weighted Delaunay.



(a) (b) (c)

FIGURE 6. (a)Circular arcs traced on atoms to form closed curves (b)Stereographic projection σ from $\mathbb{S}^2$ to $\mathbb{P}$, (c)Space decomposition for the 2D situation in the case where all radii are equal.

With the Laguerre decomposition at our disposal, let us summarize the way of generating the parts of the Connolly surface which are obtained from the surface $\mathcal{S}$ seen in relation (2.1). That is, for each atom $\mathcal{B}_k$, we describe the way of obtaining its spherical trimmed surfaces by considering its cell C_k defined in (3.15) and (3.16). For each neighboring cell C_j, consider the face $\mathcal{P}_{kj}$ which separates the spheres $B(\mathbf{m}_k, r_k)$ and $B(\mathbf{m}_j, r_j)$ as illustrated in Fig. 5. One generates two offset planes $\mathbf{p}_k$ and $\mathbf{p}_j$ by orthogonally shifting $\mathcal{P}_{kj}$ by $d_k := D_{kj}\rho/(r_k + \rho)$ and $d_j := D_{kj}\rho/(r_j + \rho)$ toward those two spheres respectively. Two circles $\mathbf{c}_k$ and $\mathbf{c}_j$ are traced on those spheres by those two planes. On the sphere $\mathcal{B}_k$, we collect all such circles $\mathbf{c}_k^1$,...,$\mathbf{c}_k^M$. The mutual intersections of those circles $\mathbf{c}_k^q$ generate some points which decompose those circles into circular arcs. We discard the circular arcs on $\mathcal{B}_k$ which are either included inside an atom other than $\mathcal{B}_k$ or beyond a plane $\mathbf{p}_i$. By organizing the remaining circular arcs as illustrated in Fig. 6(a), we obtain several closed curves $\mathcal{K}_1$,...,$\mathcal{K}_R$ on the sphere $B(\mathbf{m}_k, r_k)$. We need then to trim off the spherical parts which are not relevant: the parts of $B(\mathbf{m}_k, r_k)$ which are split by $\mathcal{K}_1$,...,$\mathcal{K}_R$ and which are away from $\mathbf{m}_k$ as illustrated in Fig. 8(b). Afterwards, we obtain on the sphere $B(\mathbf{m}_k, r_k)$ one or several spherical trimmed surfaces. Each one of them could be simply or multiply connected.

Now, let us summarize the process of obtaining the blending surfaces which come in two types. The first case happens when the probe atom exactly touches two atoms. Each face $\mathcal{P}_{kj}$ of the Laguerre decomposition gives rise to one torus T_{kj} which could never be a horn torus or a spindle one because of assumption (C1) and(C2). It is a ring torus of small radius ρ. The torus T_{kj} is tangent upon $\mathcal{B}_k$ and $\mathcal{B}_j$ where the touch-curves are the circles $\mathbf{c}_k$ and $\mathbf{c}_j$ as defined above. Slicing the torus T_{kj} along the curves $\mathbf{c}_k$ and $\mathbf{c}_j$ gives rise to two toroidal components. To obtain the toroidal surface $\mathcal{T}$, trim

off toroidal component which is away from the main axis of T_{kj}. As for the second case, the probe atom touches at least three atoms. It generally touches exactly three atoms but it can be arbitrarily many. The center of the blending sphere of radius ρ which is adjacent to at least three different spheres $B(\mathbf{m}_i, r_i)$, $B(\mathbf{m}_j, r_j)$ and $B(\mathbf{m}_k, r_k)$ is uniquely determined. Thus, the boundary of the each spherical blend is composed of the circular arcs which are intersections of the toroidal surfaces $\mathcal{T}$ and those blending spheres. Those circular arcs need to be organized to form complete closed curves on the blending spheres.

The main difficulty in the implementation of the above method consists of the stability. Depending on the position and the distribution of the atoms, it is possible that some circular arcs as illustrated in Fig.6(a) are so tiny that forming closed curves from them could lead to instability. That could lead also to very long and tight toroidal blend surfaces. A wrong orientation of the direction of the circular arcs could lead to a surface folding. It is impossible to expect that there is no gap at all between the circular arcs caused by all the geometric intersections. Controlling the size of the gaps and the size of the tiny arcs could lead to non-robust algorithm because carelessly discarding arcs which are small enough could lead to topological errors such as non-closed manifold.

3.2. **Stereographic projection.** As seen in the previous section, two instances can generate trimmed spherical patches. First, the patches which have the atoms as base surface. Second, the ones having the probe atom as base surface. In both cases, the boundaries of those spherical patches are circular arcs. We need to obtain a parametrization of the resulting spherical patches and that is the purpose of this section. We shall next be concerned with the stereographic projection σ which [6] is illustrated in Fig. 6(b). To this end, we introduce the plane $z = 0$

$$\mathbb{P} := \{[\omega : x : y : 0] \in \mathbb{E}^3 : \omega \neq 0\} = \{(x, y, z) \in \mathbb{R}^3 : z = 0\} \tag{3.19}$$

and the unit sphere

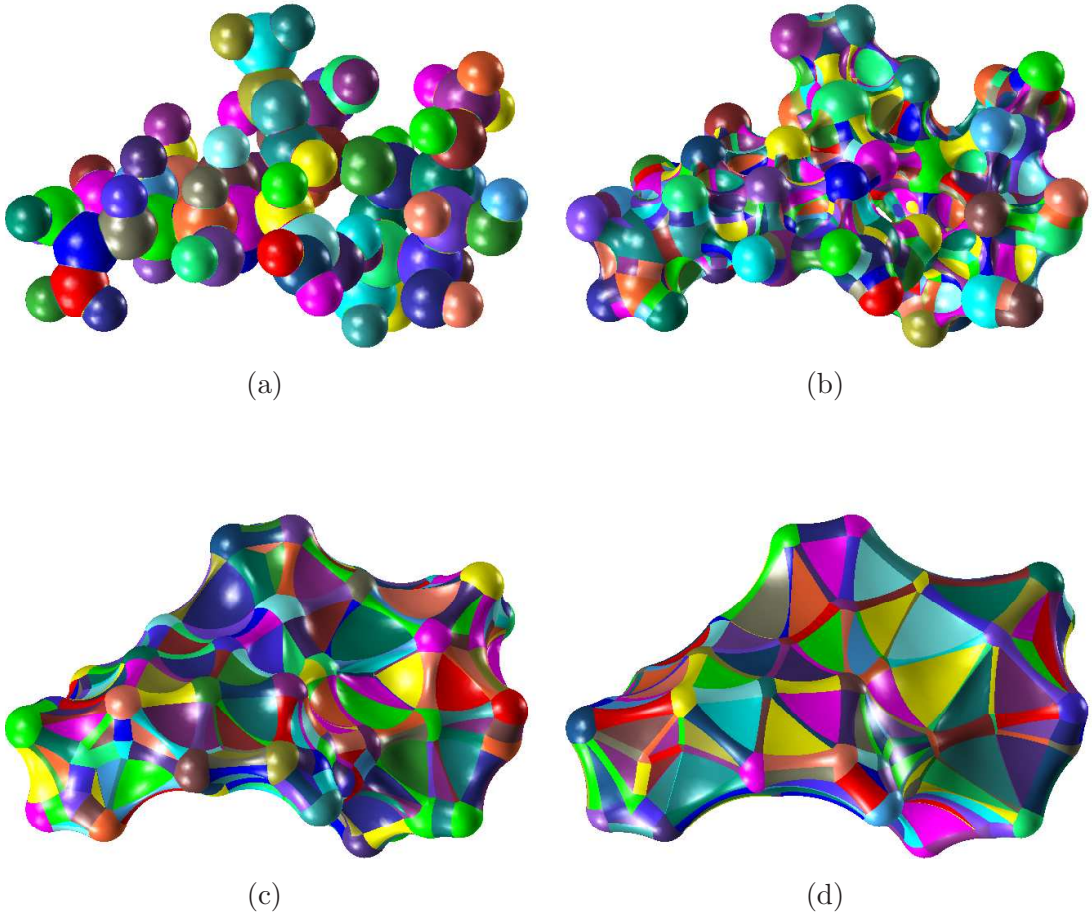
$$\mathbb{S}^2 := \big\{[\omega : x : y : z] \in \mathbb{E}^3 : \omega \neq 0,\ x^2 + y^2 + z^2 = \omega^2\big\} = \big\{\mathbf{x} \in \mathbb{R}^3 : \|\mathbf{x}\| = 1\big\}.$$

Then, the stereographic projection $\sigma : \mathbb{S}^2 \to \mathbb{P}$ maps, in Cartesian coordinates, $(x, y, z) \in \mathbb{S}^2$ to

$$\sigma(x, y, z) = \left(\frac{x}{1 - z}, \frac{y}{1 - z}, 0\right) \in \mathbb{P},$$

see [11]. Expressed in homogeneous coordinates, $\mathbf{e} = [e_0 : e_1 : e_2 : e_3] \in \mathbb{S}^2$ is mapped to

$$\sigma(\mathbf{e}) = [e_0 - e_3 : e_1 : e_2 : 0] \in \mathbb{P}.$$



(a) (b)

(c) (d)

FIGURE 7. Instance of streptomycin. Effect of enlarging the probe atom: (a) 0.025 Angstrom (b) 0.65 Angstrom (c) 1.5 Angstrom (d) 3.0 Angstrom.

The inverse $\tau := \sigma^{-1} : \mathbb{P} \to \mathbb{S}^2$ is given by

$$\tau(x, y, z) = \left(\frac{2x}{x^2 + y^2 + 1}, \frac{2y}{x^2 + y^2 + 1}, \frac{x^2 + y^2 - 1}{x^2 + y^2 + 1} \right) \in \mathbb{S}^2$$

which is in homogeneous coordinates equivalent to

$$\tau(\mathbf{e}) = \left[e_0^2 + e_1^2 + e_2^2 : 2e_0e_1 : 2e_0e_2 : e_1^2 + e_2^2 - e_0^2 \right] \in \mathbb{S}^2.$$

Suppose that the Bézier curve $\mathbf{X} \subset \mathbb{S}^2$ is given in homogeneous coordinates by

$$\mathbf{X}(t) = \sum_{i=0}^{n} [\omega_i : \omega_i x_i : \omega_i y_i : \omega_i z_i] B_i^n(t)$$

with $z_i \neq 1$ for all i. Then, the stereographic projection maps $\mathbf{X}(t)$ onto

$$\mathbf{Y}(t) = \sigma\big(\mathbf{X}(t)\big) = \begin{bmatrix} \sum_{i=0}^{n}(\omega_i - \omega_i z_i)B_i^n(t) \\ \sum_{i=0}^{n}\omega_i x_i B_i^n(t) \\ \sum_{i=0}^{n}\omega_i y_i B_i^n(t) \\ 0 \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n}\widetilde{\omega}_i B_i^n(t) \\ \sum_{i=0}^{n}\widetilde{\omega}_i \widetilde{x}_i B_i^n(t) \\ \sum_{i=0}^{n}\widetilde{\omega}_i \widetilde{y}_i B_i^n(t) \\ 0 \end{bmatrix} \in \mathbb{P}$$

where $\widetilde{\omega}_i := \omega_i(1-z_i)$ and $\widetilde{\mathbf{b}}_i := (\widetilde{x}_i, \widetilde{y}_i, 0) := \big(x_i/(1-z_i), y_i/(1-z_i), 0\big)$. In other words, the preimage with respect to τ of the curve of $\mathbf{X} \subset \mathbb{S}^2$ is the rational Bézier curve

$$\mathbf{Y}(t) = \frac{\sum_{i=0}^{n}\widetilde{\omega}_i \widetilde{\mathbf{b}}_i B_i^n(t)}{\sum_{i=0}^{n}\widetilde{\omega}_i B_i^n(t)} \in \mathbb{P}. \tag{3.20}$$

3.3. **Parametrizing the trimmed surfaces.** As described in Section 3.1, incident circular arcs split the sphere $\mathcal{B}_k$ into p_k subsurfaces $\mathbf{S}_i^{(k)} \subset \mathcal{B}_k$ where $i = 1, \ldots, p_k$. Each subsurface $\mathbf{S}_i = \mathbf{S}_i^{(k)}$ is bounded by some circular arcs $\mathcal{C}_j$. That is, there exists an index set $\mathcal{I}_i$ such that

$$\partial\mathbf{S}_i = \bigcup_{j\in\mathcal{I}_i} \mathcal{C}_j \qquad \text{for all} \qquad i = 1, \ldots, p_k.$$

We shall exploit the stereographic projection σ to represent a surface $\mathbf{S}_i$ as a parametric trimmed surface defined on some planar domain. To this end, let us denote by σ_k the analogue of the stereographic projection σ with respect to an arbitrary sphere $\mathcal{B}_k$. According to Subsection 3.2, setting $\tau_k := \sigma_k^{-1}$, we can compute the curves $\mathcal{E}_j = \tau_k(\mathcal{C}_j) \subset \mathbb{P}$ for each circular arc $\mathcal{C}_j \subset \mathcal{B}_k$ with $j \in \mathcal{I}_i$. Denoting by $\mathbf{D}_i^{(k)}$ the planar trimmed domain bounded by $\partial\mathbf{D}_i^{(k)} = \bigcup_{j\in\mathcal{I}_k} \mathcal{E}_j$ we thus have constructed the trimmed surface [3, 9]

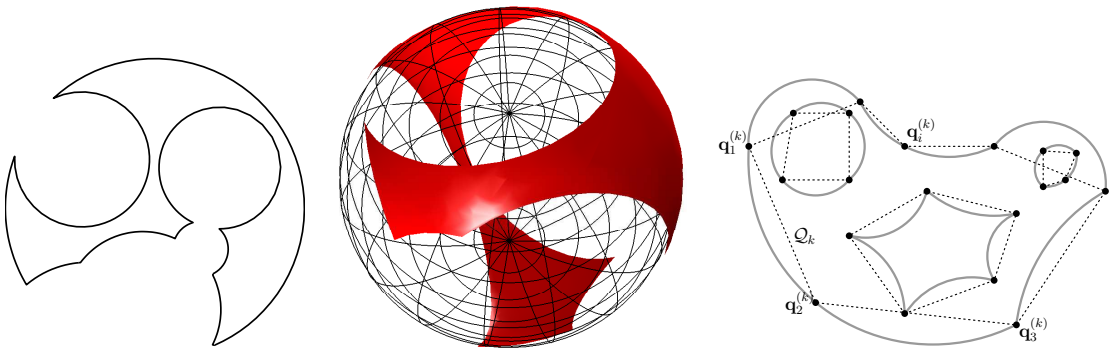
$$\tau_k : \mathbf{D}_i^{(k)} \to \mathcal{B}_k \quad \text{with} \quad \mathbf{S}_i^{(k)} = \tau_k(\mathbf{D}_i^{(k)}) \quad \text{for all} \quad i = 1, \ldots, p_k. \tag{3.21}$$

Note that the preimage by τ_k of a 3D circular arc is a 2D circular arc as illustrated in Fig. 8(a) and Fig. 8(b).

The parametrization of the toroidal blending surfaces consists of a surface of revolution where the directrix is a circular arc. In particular, that can be represented as a rational Bézier surface.

4. Decomposition into four-sided domains

In this section, we describe the decomposition of a Connolly surface into large four-sided subsurfaces. It is very difficult (if possible at all) to directly decompose the Connolly surface from its raw representation. We need to generate a discretization of the whole surface first. The easiest way of representing a discretization while keeping the topology is to use a triangular mesh as an intermediate step.



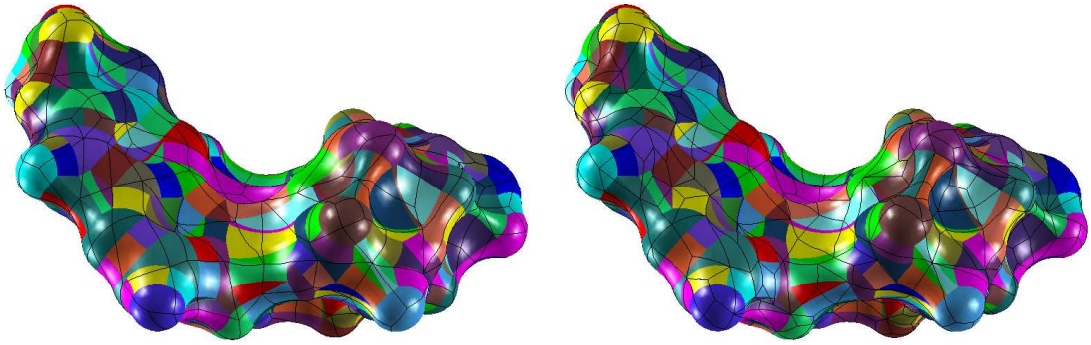
(a) (b) (c)

FIGURE 8. (a) Spherical patch with composite circular arc as boundary, (b) Spherical trimmed surface, (c) Imperfections in polygonal approximation.

We summarize the meshing of a single [17] parametric function $\mathbf{S}_k$ specified by (3.21). To simplify the notation, we will drop the index k in the sequel. The approach in triangulating $\mathbf{S}$ is processed in two steps. First, a 2D mesh on the parameter domain $\mathbf{D}$ is generated according to the first fundamental form. Afterwards, the resulting 2D mesh is lifted to the parametric surface $\mathbf{S}$ by computing its image by τ_k from (3.21). For that purpose, one starts from a coarse 2D mesh of $\mathbf{D}$ and a generalized two dimensional Delaunay refinement is used as summarized below. We will call an edge of a mesh in the parameter domain a *2D edge* and an edge in the lifted mesh a *3D edge*. Similarly to the planar case, we introduce an edge size function ρ which is defined now on the parametric surface $\rho : \mathbf{S} \longrightarrow \mathbf{R}^+$. By composing ρ with the parameterization τ_k of $\mathbf{S}$, we have another function $\widetilde{\rho} := \rho \circ \tau_k$ which we will call henceforth "parameter edge size function" because it is defined for all $\mathbf{u} = (u, v)$ in the parameter domain. Let us consider a 2D edge $[\mathbf{a}, \mathbf{b}] \subset \mathbf{D}$ and let us denote the first fundamental forms at $\mathbf{a}$ and $\mathbf{b}$ by $I_{\mathbf{a}}$ and $I_{\mathbf{b}}$ respectively. Further, we introduce the following average distance between $\mathbf{a}$ and $\mathbf{b}$

$$d_{\text{Riem}}(\mathbf{a}, \mathbf{b}) := \sqrt{\overrightarrow{\mathbf{ab}}^T T \overrightarrow{\mathbf{ab}}} \qquad T := 0.5(I_{\mathbf{a}} + I_{\mathbf{b}}). \tag{4.22}$$

The 2D edge $[\mathbf{a}, \mathbf{b}]$ is split if this average distance exceeds the value of the parameter edge size function $\tilde{\rho}$ at the midnode of $[\mathbf{a}, \mathbf{b}]$. Note that no new boundary nodes are introduced during that refinement because only internal edges are allowed to be split. Consider now a 2D edge $[\mathbf{a}, \mathbf{c}]$ is shared by two triangles which form a convex quadrilateral $[a, b, c, d]$. Denote by T the average values of the first fundamental forms $I_{\mathbf{a}}$, $I_{\mathbf{b}}$, $I_{\mathbf{c}}$ and $I_{\mathbf{d}}$ at those nodes. The edge $[\mathbf{a}, \mathbf{c}]$ is flipped into $[\mathbf{b}, \mathbf{d}]$ if the next generalized



(a) (b)

FIGURE 9. Quadrangulation simplification with NND nodes, NEL quadrilaterals and NED edges: (a)NND=484, NEL=482, NND=964 (b)NND=251, NEL=249, NND=498.

Delaunay angle criterion is met

$$\|\overrightarrow{\mathbf{bc}} \times \overrightarrow{\mathbf{ba}}\|(\overrightarrow{\mathbf{da}}^T T \overrightarrow{\mathbf{dc}}) < \|\overrightarrow{\mathbf{da}} \times \overrightarrow{\mathbf{dc}}\|(\overrightarrow{\mathbf{cb}}^T T \overrightarrow{\mathbf{ba}}). \tag{4.23}$$

As in the planar case, ones starts from a very coarse triangulation and one recursively refines or flips edges according to the ideal mesh size function ρ.

We would like now to describe the procedure of splitting $\mathbf{P}$ into a coarse triangulation. Suppose we have a 2D domain $\mathbf{P}$ which may contain some holes and which has polygonal boundaries. First, the initial polygon is split into a few simply connected polygons $\mathbf{P} = \bigcup_{i=1}^{N} \mathbf{P}^{(i)}$. Afterwards, we do the following for every simply connected polygon $\mathbf{P}^{(i)}$. One initializes its set of triangles as empty set $\mathcal{T}_h^{(i)} = \emptyset$. Then, one finds a triangle T which can be chopped off from $\mathbf{P}^{(i)}$. We can repeat that by updating $P^{(i)} := P^{(i)} \setminus T$ and $\mathcal{T}_h^{(i)} = \mathcal{T}_h^{(i)} \cup T$. Finally, the triangulation of $\mathbf{P}$ is the union of all triangulations: $\mathcal{T}_h := \cup_i \mathcal{T}_h^{(i)}$.

Let us consider a parametric surface $\mathbf{S}$ and a differentiable function $F : \mathbf{S} \longrightarrow \mathbf{R}$. The Laplace-Beltrami operator is defined by

$$\Delta_{\mathbf{S}} F = -\frac{1}{\sqrt{g}} \frac{\partial}{\partial u_j} \left(\sqrt{g} g_{ij} \frac{\partial F}{\partial u_i} \right) \tag{4.24}$$

in which we use Einstein notation in indexing and g is the determinant of I which is the first fundamental form. The function F is said to be harmonic if $\Delta_{\mathbf{S}} F = 0$. The edge size function ρ should be harmonic. In the next description, the mesh of the entire surface Γ from (3.21) will be denoted by $\mathcal{M}$.

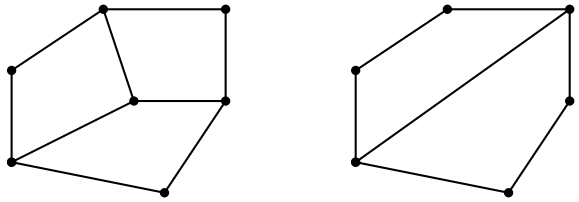
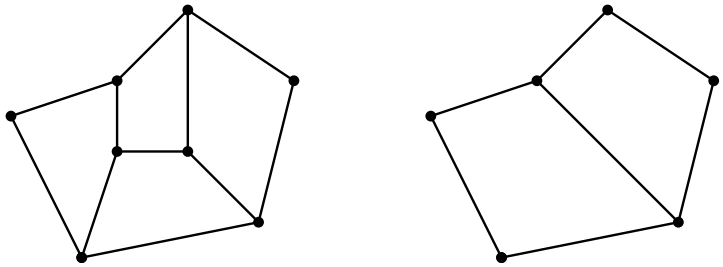


FIGURE 10. Pattern 1 of quadrangulation simplification.

The process of generating the patches starts from finding the underlying structure of a coarse quadrangulation $\mathcal{Q}_{\text{coarse}}$ as illustrated in Fig. 9. That is, we need to know the vertices $\mathbf{P}_i$ of the coarse quadrangulation $\mathcal{Q}_{\text{coarse}}$. We need to know every two vertices $\mathbf{P}_i$ and $\mathbf{P}_j$ that need to be connected to form an edge. Finally, we need to find the four vertices to form the apices of each patch. To find such a structure, we start from a fine quadrangulation $\mathcal{Q}_{\text{fine}}$ which is obtained by subdividing each triangular element of $\mathcal{M}$ into three quadrilaterals. More precisely, a new node is inserted at the center of gravity of each triangle and three new nodes at the middle of its three edges. The resulting quadrangulation $\mathcal{Q}_{\text{fine}}$ is not directly useful for patch representation because it is too fine. As a consequence, we need to coarsen that fine quadrangulation repeatedly in which we initialize $\mathcal{Q}_0 := \mathcal{Q}_{\text{fine}}$. Each coarsening step from $\mathcal{Q}_k$ to $\mathcal{Q}_{k+1}$ consists in amalgamating a few neighboring quadrilaterals in the quadrangulation $\mathcal{Q}_k$ to form a coarser local quadrangulation in $\mathcal{Q}_{k+1}$. Each quadrilateral amalgamation is most easily described by graphically using 2D patterns. For instance, in Fig. 10, Fig. 11 and Fig. 12, we see such patterns. On the left sides, one sees the finer quadrangulation $\mathcal{Q}_k$ and on the right sides the simplified one in $\mathcal{Q}_{k+1}$. Note that in each pattern, the initial local quadrangulation and the simplified one admit exactly the same boundary. In those figures, we present only a few simplification patterns. It is beyond the scope of this paper to describe all possible simplification patterns. Implementing this algorithm is a developmental process and most of the patterns are obtained from practical observations. That method could become stuck if one implements only very few patterns because it is possible that none of the available patterns can be detected in the old quadrangulations $\mathcal{Q}_k$. As more patterns are implemented, the simplification algorithm becomes also more robust. More than fifty patterns seem to be enough in the implementations. Note that those depicted patterns are drawn for quadrangulations on the plane to simplify the description. But, many complications are encountered in practice because we need to apply those patterns to quadrangulations which are embedded in the space. Some additional geometric conflict might occur in the practical application of those patterns. For instance, a manifold folding might result in the application of those operations in 3D. In addition, the notion of *convexity* is lost for a quadrilateral in the space. In the



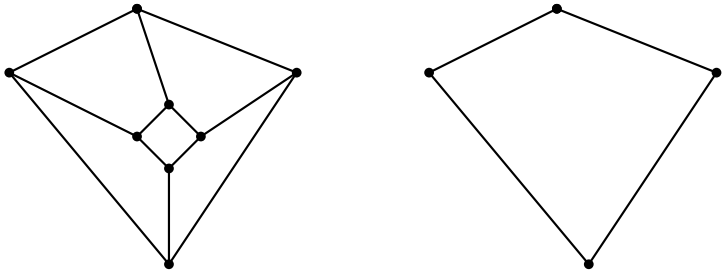
(a) (b)

FIGURE 11. Pattern 2 of quadrangulation simplification.

2D patterns, to verify if a quadrilateral is nicely shaped, you need only to verify its convexity. For quadrilaterals in the space, the simplest method of checking good shape property is to consider the Coons map of the four straight boundary curves. Thus, prior to the application of those geometric operations, tests must be done in order to avoid foldings or irregular mappings. Many tests about edge degeneration, angular quality are needed to be applied in practice. A lot of programming tasks are required to efficiently recognize the used patterns and to detect if a coarsening can be applied.

Since the initial finest quadrangulation $\mathcal{Q}_{\text{fine}}$ is inappropriate to be used as $\mathcal{Q}_{\text{coarse}}$, we impose a maximal allowed number of quadrilaterals (say NQUAD($\mathcal{Q}_{\text{coarse}}$)$\leq$ 40 % NQUAD($\mathcal{Q}_{\text{fine}}$)). Likewise, we impose also the minimal number of quadrilaterals of $\mathcal{Q}_{\text{coarse}}$. For the recursive quadrangulation simplification, we use a coarsening parameter $\alpha \in [0, 1]$ which gauges the density of $\mathcal{Q}_{\text{coarse}}$. A unit value of α corresponds to the finest allowed quadrilateral decomposition while a value of α approaching zero corresponds to a very coarse quadrilateral decomposition. As an illustration, for respective values of $\alpha = 0.5$ and $\alpha = 0.1$ of the coarsening parameter, we see in Fig. 9(a) and in Fig. 9(b) the corresponding quadrangulations $\mathcal{Q}_{\text{coarse}}$ for the same Connolly surface of a Quinine molecule.

When a straight quadrangulation $\mathcal{Q}_{\text{coarse}}$ has been assembled, we need to connect every two nodes $\mathbf{P}_i$ and $\mathbf{P}_j$ on the endpoints of every edge by a curve on the manifold $\mathcal{M}$. That curve is described by means of a geodesic which is the shortest curve on $\mathcal{M}$ joining $\mathbf{P}_i$ to $\mathbf{P}_j$. The search for the geodesic is done in two steps. First, we need a curve which traverses only the nodes and the edges of $\mathcal{M}$. Afterwards, we improve that curve by allowing it to traverse the internal parts of the triangles of $\mathcal{M}$. For the first step, finding geodesics might at first sight be easy because of the existence of algorithms like Dijkstra [2]. But assembling the complete node-edge graph of the whole mesh $\mathcal{M}$ is very memory consuming. Searching for shortest paths becomes very cumbersome for such a large discrete manifold. Even for small molecules having a dozen atoms, some efficient data structure to accelerate the search is necessary in the implementation. As



(a) (b)

FIGURE 12. Pattern 3 of quadrangulation simplification.

for the second step, we improve the discrete geodesic which traverses only the edges of the manifold $\mathcal{M}$ because it is usually of very bad quality unless the manifold $\mathcal{M}$ is extremely fine. To that end, when that initial geodesic is found, one searches for the triangles in its neighborhood and generate the geodesic which traverses the interior of those triangles. That is done by using local refinements and by applying local Dijkstra algorithms.

In many decomposition techniques, *cleanup* is the process of generating a tessellation by improving an available one according to some quality criteria. We are going to keep the numbers of nodes and edges unchanged but we change the position of nodes and edges in order to enhance the quadrangulation quality. Before describing the cleanup operations, let us review the way qualities of a quadrilateral, a node or an edge can be evaluated. We consider a technique of assessing the quality of a quadrilateral $[A, B, C, D]$. It requires the introduction of the following distortion coefficient of any triangle $[a, b, c]$:

$$\alpha := 2\sqrt{3}\frac{\|\overrightarrow{ca} \times \overrightarrow{cb}\|}{\|\overrightarrow{ca}\|^2 + \|\overrightarrow{ab}\|^2 + \|\overrightarrow{bc}\|^2} \in [0, 1]. \tag{4.25}$$

We can easily see that the triangular distortion α is unity if the triangle $[a, b, c]$ is equilateral. From a convex quadrilateral $[A, B, C, D]$, we may derive four triangles $[A, B, C]$, $[A, C, D]$, $[A, B, D]$ and $[D, B, C]$. Let us denote by α_i the triangular distortions of those four triangles such that $\alpha_1 \geq \alpha_2 \geq \alpha_3 \geq \alpha_4$. We define the first quality measurement of the quadrilateral $[A, B, C, D]$ to be

$$\beta := (\alpha_3\alpha_4)/(\alpha_1\alpha_2) \in [0, 1]. \tag{4.26}$$

Effectively, the value of β is unity for rectangles and it approaches zero as a quadrilateral becomes triangular shaped. In the following discussion, we will denote by $\mu(q)$ a quality measure of a quadrilateral q.

We would like to consider quality measurements of a node and an edge inside a quadrangulation. To that end, let us consider an internal node ω which is shared by the

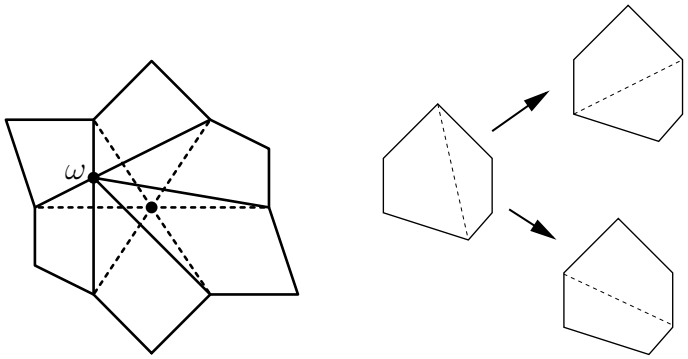


FIGURE 13. Cleanup: (a) Shifting a node ω(b) Flipping an edge

quadrilaterals q_i $i \in \mathcal{J}$. We may now evaluate the quality of the node ω by

$$\mu(\omega) := \frac{1}{\text{card}(\mathcal{J})} \sum_{i \in \mathcal{J}} \mu(q_i). \tag{4.27}$$

Similarly, for an internal edge e upon which two quadrilaterals q_1 and q_2 are incident, its quality can be measured by

$$\mu(e) := \frac{1}{2}[\mu(q_1) + \mu(q_2)]. \tag{4.28}$$

We will treat two types of cleanup operations: *node repositioning* and *edge flipping*. The first one consists in shifting an internal node to another position in order to improve the quality of the neighboring quadrilaterals. In the course of node shifting, we have to make sure that all incident quadrilaterals remain convex. The second operation modifies the endpoints of an internal edge.

Let us first show how to find the region inside which a node ω can be shifted. The node repositioning consists in moving ω inside the interior of $\mathcal{R}$ in order to minimize $\mu(\omega)$. Let us denote by E_ω the set of edges which emanate from the node ω. Then we take the shortest edge $\tilde{e}$ from among E_ω and consider a circle centered at the node ω and having radius $\rho := \lambda \cdot \text{length}(\tilde{e})$ where λ is a user defined parameter (say λ=0.25) from $]0, 1[$. The new position of ω is then sought inside this circle. The practical realization of such a shifting is to pick p (say $p = 5$) positions q_i inside the circle. For every q_i, we test if by replacing ω by q_i, we would still have incident nicely shaped quadrilaterals. We replace then ω by q_i which gives new incident quadrilaterals and which minimizes $\mu(q_i)$. If none of the q_i fulfils those desirable properties, then we keep ω in its current position.

The second operation consists in flipping an edge in order to improve the qualities of the neighboring quadrilaterals. In the best case, there are two possibilities for flipping an

edge by considering the union of the incident quadrilaterals as explained in Fig. 13(b). We flip an internal edge e to a position which keeps the two incident quadrilaterals nicely shaped and which improves the value of $\mu(e)$.

There is no exact rule about where to start but the method that many people use in such a retouching technique is to start from the worst entities. That means, one measures the qualities $\mu(\omega_i)$ of all internal nodes. One searches for the m (say $m = 5$ or m is the number of all internal nodes) nodes having the largest μ values. Then one applies the cleanup operations to those nodes. One can repeat that operation a few number of times. One can do also the same thing for edges. It is also possible to alternate the edge and node quality improvements.

5. Globally continuous spline representation

In this section, we would like to determine the splines with the main objective of obtaining global continuity. In addition, we would like to provide a method to specify the orientations of all the normal vectors. We are given four boundary NURBS curves

$$K_1^h(t) = \frac{\sum_{i=0}^{n_u} \mathbf{a}_i^h w_{a,i}^h N_i^k(t)}{\sum_{i=0}^{n_u} w_{a,i}^h N_i^k(t)}, \qquad K_2^h(t) = \frac{\sum_{i=0}^{n_u} \mathbf{b}_i^h w_{b,i}^h N_i^k(t)}{\sum_{i=0}^{n_u} w_{b,i}^h N_i^k(t)}, \tag{5.29}$$

$$K_1^v(t) = \frac{\sum_{i=0}^{n_v} \mathbf{a}_i^v w_{a,i}^v N_i^k(t)}{\sum_{i=0}^{n_v} w_{a,i}^v N_i^k(t)}, \qquad K_2^v(t) = \frac{\sum_{i=0}^{n_v} \mathbf{b}_i^v w_{b,i}^v N_i^k(t)}{\sum_{i=0}^{n_v} w_{b,i}^v N_i^k(t)}. \tag{5.30}$$

We assume that all four curves are defined on the interval $[0, 1]$ and that the knot sequences of the opposite curves K_1^p and K_2^p for $p = h, v$ are the same. Otherwise, we need to apply a preprocessing step of knot insertions by using discrete B-splines. Additionally, we have compatibility conditions (see Fig. 14(a)) related to the control points and weights at the corners:

$$\begin{array}{llll} \mathbf{a}_{n_v}^h = \mathbf{b}_0^h, & \mathbf{a}_0^v = \mathbf{a}_0^h, & \mathbf{a}_{n_u}^h = \mathbf{b}_0^v, & \mathbf{b}_{n_u}^h = \mathbf{b}_{n_v}^v, \\ \omega_{a,n_v}^h = \omega_{b,0}^h, & \omega_{a,0}^v = \omega_{a,0}^h, & \omega_{a,n_u}^h = \omega_{b,0}^v, & \omega_{b,n_u}^h = \omega_{b,n_v}^v. \end{array} \tag{5.31}$$

Suppose we are given a set of samples $(u_i, v_i) \in [0, 1]^2$ and $\mathbf{P}_i \in \mathbb{R}^3$ for $i = 0, ..., M$. To approximate those samples, we seek a NURBS surface of the form

$$\boldsymbol{\gamma} = \left(\frac{x_1(u,v)}{\omega(u,v)}, \frac{x_2(u,v)}{\omega(u,v)}, \frac{x_3(u,v)}{\omega(u,v)} \right) \quad \text{where} \tag{5.32}$$

$$x_q(u,v) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} w_{i,j} d_{i,j}^q N_i^{k_u}(u) N_j^{k_v}(v), \tag{5.33}$$

$$\omega(u,v) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} w_{i,j} N_i^{k_u}(u) N_j^{k_v}(v) \tag{5.34}$$

in which $q = 1, 2, 3$ and $\mathbf{d}_{i,j} := (d^1_{i,j}, d^2_{i,j}, d^3_{i,j}) \in \mathbb{R}^3$. That surface is equivalent to the following in homogeneous case

$$\overline{\boldsymbol{\gamma}}(u,v) = \Big[\omega(u,v) : x_1(u,v) : x_2(u,v) : x_3(u,v)\Big]. \tag{5.35}$$

A point $\mathbf{P} = (p_1, p_2, p_3)$ will be identified to the following in the projective space

$$\overline{\mathbf{P}} = \big[\beta : \beta p_1 : \beta p_2 : \beta p_3\big] \qquad \text{where} \qquad \beta(\mathbf{P}) := 1/\sqrt{1+\mathbf{P}} = 1/(1+p_1^2+p_2^2+p_3^2). \tag{5.36}$$

The following lexicographic ordering of the surface information for $i = 0, .., n_u$, $j = 0, ..., n_v$ are used for the NURBS patch

$$\begin{aligned} \widetilde{w}_{i(n_v+1)+j} &:= w_{i,j}, \qquad \widetilde{\mathbf{d}}_{i(n_v+1)+j} := \mathbf{d}_{i,j}, \\ \widetilde{N}_{i(n_v+1)+j}(u,v) &:= N_i^{k_u}(u) N_j^{k_v}(v) \qquad \forall\, (u,v) \in [0,1]^2. \end{aligned}$$

Note that the new expressions $\widetilde{w}_s$, $\widetilde{\mathbf{d}}_s$, $\widetilde{N}_s(u,v)$ have only one index $s = 0, ..., n$ where $n := n_u(n_v+1)+n_v$ whereas the old ones $w_{i,j}$, $\mathbf{d}_{i,j}$ and $N_i^{k_u}(u)N_j^{k_v}(v)$ have two indices $i = 0, ..., n_u$ and $j = 0, ..., n_v$. With the help of the new notations, the definitions in (5.33) and (5.34) are equivalent to

$$x_q(u,v) = \sum_{s=0}^{n} \widetilde{w}_s \widetilde{d}^q_s \widetilde{N}_s(u,v), \qquad \omega(u,v) = \sum_{s=0}^{n} \widetilde{w}_s \widetilde{N}_s(u,v). \tag{5.37}$$

Introduce the distance functional [5] of the above surface to the point $\overline{\mathbf{P}}$

$$Q\Big[\overline{\boldsymbol{\gamma}}(u,v), \overline{\mathbf{P}}\Big] := \overline{\boldsymbol{\gamma}}(u,v)^2 - \big(\overline{\boldsymbol{\gamma}}(u,v) \cdot \overline{\mathbf{P}}\big). \tag{5.38}$$

By applying that distance functional to $\overline{\boldsymbol{\gamma}}(u,v)$ and $\overline{\mathbf{P}}$, we obtain

$$Q\Big[\overline{\boldsymbol{\gamma}}(u,v), \overline{\mathbf{P}}\Big] = \omega(u,v)^2 + \sum_{q=1}^{3} x_q(u,v)^2 - \beta\Big[\omega(u,v) + \sum_{q=1}^{3} p_q x_q(u,v)\Big]^2. \tag{5.39}$$

The problem is then reduced to solving the minimization $F := \sum_{i=0}^{M} Q[\overline{\boldsymbol{\gamma}}(u_i, v_i), \overline{P}_i]$ where the unknowns are the control points $\mathbf{d}_{i,j}$ and the weights $w_{i,j}$. Since that formulation could generate a NURBS function having negative or zero weights, it is corrected by a regularizing term

$$\sum_{i=0}^{M} Q\Big[\overline{\boldsymbol{\gamma}}(u_i, v_i), \overline{P}_i\Big] + \lambda R\Big[\overline{\boldsymbol{\gamma}}(u_i, v_i)\Big] \tag{5.40}$$

where $R[\overline{\boldsymbol{\gamma}}(u_i, v_i)] = [w(u_i, v_i) - 1]^2$. As the value of the parameter λ becomes large, the weights are likely to be positive. Thus, that value is a trade-off between the approximation and the regularity. In practice, the value of λ starts from a small one and it is incremented gradually.

For the purpose of global continuity, the control points and weights on the boundary are interpolated while the internal ones need to be determined. Thus, the boundary entities verify

$$\begin{aligned}\mathbf{d}_{i,0} &:= \mathbf{a}_i^h, \quad \omega_{i,0} := \omega_{a,i}^h, \quad \mathbf{d}_{i,n_v} := \mathbf{b}_i^h, \quad \omega_{i,n_v} := w_{b,i}^h, \quad \text{for} \quad i = 0, ..., n_u,\\ \mathbf{d}_{0,j} &:= \mathbf{a}_j^v, \quad \omega_{0,j} := w_{a,j}^v, \quad \mathbf{d}_{n_u,j} := \mathbf{b}_j^v, \quad \omega_{n_u,j} := w_{b,j}^v, \quad \text{for} \quad j = 0, ..., n_v.\end{aligned}$$

Let us denote by $\mathcal{J}$ the set of indices $s = 0, 1, ..., n$ such that the corresponding control points $\mathbf{d}_{i,j} = \widetilde{\mathbf{d}}_s$ are internal ones. Likewise, the set of indices for the boundary control points are denoted by $\mathcal{B}$. As for the coordinates, we define $Z(\mathcal{J}) := \{1, 2, 3\} \times \mathcal{J}$ while the set $Z(\mathcal{B})$ is defined analogously. The coordinates of the internal control points are $\widetilde{d}_s^q$ for $(q, s) \in Z(\mathcal{J})$. For the mixed values, we denote $\alpha_i(s_1, s_2) := \widetilde{N}_{s_1}(u_i, v_i)\widetilde{N}_{s_2}(u_i, v_i)$ for all $s_1, s_2 = 0, 1, ..., n$ and $i = 0, 1, ..., M$. For any $q_0 = 1, 2, 3$ and $s_0 = 0, 1, ..., n$ such that $(q_0, s_0) \in Z(\mathcal{J})$, the partial derivative of F with respect to $\widetilde{d}_{s_0}^{q_0}$ is where $\beta_i := \beta(\mathbf{P}_i)$

$$\sum_{i=0}^{M}\Big[\sum_{s=0}^{n}\widetilde{d}_s^{q_0}\alpha_i(s, s_0) - \beta_i\Big(\sum_{s=0}^{n}\widetilde{\omega}_s p_{q_0,i}\alpha_i(s, s_0) + \sum_{q=1}^{3}\sum_{s=0}^{n}p_{q,i}p_{q_0,i}\widetilde{d}_s^{q_0}\alpha_i(s, s_0)\Big)\Big] \tag{5.41}$$

and the contribution of the regularizing term with respect to $\widetilde{d}_{s_0}^{q_0}$ is zero. The one for F with respect to $\widetilde{\omega}_{s_0}$ is

$$\sum_{i=0}^{M}\Big[-\beta_i\sum_{q=1}^{3}\sum_{s=0}^{n}p_{q,i}\widetilde{d}_s^q\alpha_i(s, s_0) + \sum_{s=0}^{n}(1-\beta_i)\widetilde{\omega}_s\alpha_i(s, s_0)\Big]. \tag{5.42}$$

The one for regularizing term with respect to $\widetilde{\omega}_{s_0}$ is

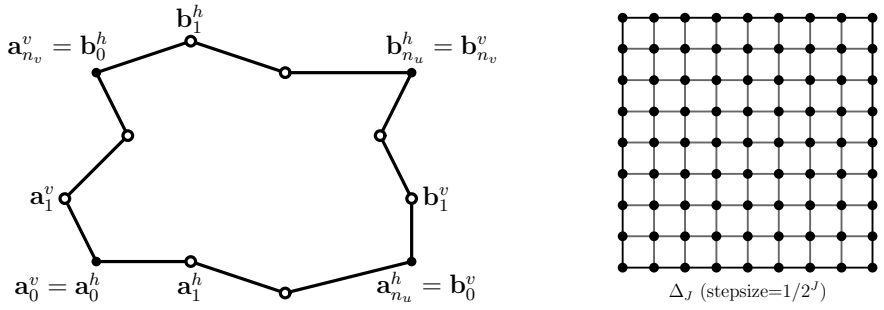
$$\sum_{s=0}^{n}\Big[\sum_{i=0}^{M}\alpha_i(s, s_0)\Big]\widetilde{\omega}_s - \sum_{i=0}^{M}\widetilde{N}_{s_0}(u_i, v_i). \tag{5.43}$$

After collecting the terms in (5.41), one obtains

$$\sum_{q=1}^{3}\sum_{s=0}^{n}\Big[\sum_{i=0}^{M}\Big(\delta_{q,q_0} - \beta_i p_{q,i}p_{q_0,i}\Big)\alpha_i(s, s_0)\Big]\widetilde{d}_s^q + \sum_{s=0}^{n}\Big(\sum_{i=0}^{M} -\beta_i p_{q_0,i}\alpha_i(s, s_0)\Big)\widetilde{\omega}_s = 0. \tag{5.44}$$

Similarly, a combination of the expressions in (5.42) and (5.43) yields

$$\sum_{q=1}^{3}\sum_{s=0}^{n}\Big[\sum_{i=0}^{M} -\beta_i p_{q,i}\alpha_i(s, s_0)\Big]\widetilde{d}_s^q + \sum_{s=0}^{n}\Big[\sum_{i=0}^{M}(1+\lambda-\beta_i)\alpha_i(s, s_0)\Big]\widetilde{\omega}_s = \lambda\sum_{i=0}^{M}\widetilde{N}_{s_0}(u_i, v_i). \tag{5.45}$$



(a) (b)

FIGURE 14. (a)Compatibility condition of the control points of the bounding curves (b)Grid-points for dyadic stepsize at level J.

By considering the boundary condition in virtue of the global continuity, we obtain

$$\sum_{(q,s)\in Z(\mathcal{J})} \Big[\sum_{i=0}^{M} \Big(\delta_{q,q_0} - \beta_i p_{q,i} p_{q_0,i}\Big)\alpha_i(s,s_0)\Big]\widetilde{d}^q_s - \sum_{s\in\mathcal{J}} \Big[\sum_{i=0}^{M} \beta_i p_{q_0,i}\alpha_i(s,s_0)\Big]\widetilde{\omega}_s = G \tag{5.46}$$

$$\sum_{(q,s)\in Z(\mathcal{J})} \Big[\sum_{i=0}^{M} -\beta_i p_{q,i}\alpha_i(s,s_0)\Big]\widetilde{d}^q_s + \sum_{s\in\mathcal{J}} \Big[\sum_{i=0}^{M} (1+\lambda-\beta_i)\alpha_i(s,s_0)\Big]\widetilde{\omega}_s = H \tag{5.47}$$

where

$$G := \sum_{(q,s)\in Z(\mathcal{B})} \Big[\sum_{i=0}^{M} \Big(\beta_i p_{q,i} p_{q_0,i} - \delta_{q,q_0}\Big)\alpha_i(s,s_0)\Big]\widetilde{d}^q_s + \sum_{s\in\mathcal{B}} \Big(\sum_{i=0}^{M} \beta_i p_{q_0,i}\alpha_i(s,s_0)\Big)\widetilde{\omega}_s.$$

$$H := \sum_{(q,s)\in Z(\mathcal{B})} \Big[\sum_{i=0}^{M} \beta_i p_{q,i}\alpha_i(s,s_0)\Big]\widetilde{d}^q_s + \sum_{s\in\mathcal{B}} \Big[\sum_{i=0}^{M} (\beta_i - 1 - \lambda)\alpha_i(s,s_0)\Big]\widetilde{\omega}_s + \\ + \lambda \sum_{i=0}^{M} \widetilde{N}_{s_0}(u_i, v_i).$$

The equations in (5.46) are of number $\operatorname{card}(Z(\mathcal{J})) = 3\operatorname{card}(\mathcal{J})$ and those in (5.47) of number $\operatorname{card}(\mathcal{J})$. A combination of them yields a linear system of $4\operatorname{card}(\mathcal{J})$ equations with $4\operatorname{card}(\mathcal{J})$ unknowns. By solving that system, the values of $\widetilde{d}^q_s$ and $\widetilde{\omega}^q_s$ for $(q,s) \in Z(\mathcal{J})$ are found.

The above process can of course be applied to the determination of a NURBS curve from a set of points. The determination of all NURBS patches is then done as follows. First, find all separating curves $\mathcal{C}_i$ such that for each curve the two boundary nodes are exactly interpolated. For each patch, select from among $\mathcal{C}_i$ the four NURBS curves K^h_1, K^h_2, K^v_1, K^v_2 on its sides. Those four curves should fulfil the compatibility conditions in (5.31). Afterward, determine the NURBS surface interpolating those four curves as described in (5.46) and (5.47).

Before showing some practical results, let us briefly consider the orientations of the normal vectors. It is usually necessary for the subsequent numerical solvers such as the wavelet BEM that the directions of the normal vectors are consistent and outwardly oriented. The automatic fast check if the normal vector is pointing outward needs only to be done on one patch Γ_{seed} which we call *seed* patch. Consider an infinite line $\mathcal{L}$ having any unitary directional vector $\mathbf{v}$ which can be supposed to be $(1, 0, 0)$. The seed patch will be the patch having the largest projection value of $\boldsymbol{\gamma}_i(0.5, 0.5)$ on the line $\mathcal{L}$. We consider now the plane $\mathcal{P}$ tangent to the normal $\vec{n}_{\text{seed}}$ at $\boldsymbol{\gamma}_{\text{seed}}(0.5, 0.5)$. The plane $\mathcal{P}$ splits the space into two half-spaces. If all other patches Γ_i are on the same half-space which is not pointed by $\vec{n}_{\text{seed}}$ then the normal orientation is outward. Otherwise, we need to flip the direction of $\vec{n}_{\text{seed}}$. Flipping the normal orientation of a NURBS patch consists in applying $\mathbf{d}_{i,j} := \mathbf{d}_{j,i}$ to its control points and $w_{i,j} := w_{j,i}$ to its weights. Two adjacent patches Γ_p and Γ_q admit the same normal orientation if their common edge is traversed from both sides in opposite directions. As a consequence, we initialize a set of patches as $\mathcal{Z} := \{\Gamma_{\text{seed}}\}$ and we consider $\mathcal{R}$ the set of patches adjacent upon $\mathcal{Z}$. If a patch Γ_q of $\mathcal{R}$ has an opposite normal as an element of $\mathcal{Z}$, we flip the normal $\vec{n}_q$. Afterwards, we update $\mathcal{Z} := \mathcal{Z} \cup \mathcal{R}$ and we repeat the same process. Provided that the molecular surface is an *orientable* manifold (i.e. not like Möbius strip), that process should terminate after each patch has been visited once.

6. Practical results

In this section, we would like to present several practical results of the method described formerly. First, we show some graphical outcomes of our method when applied to real molecules. We present two results in Fig. 15 and Fig. 16 where we use molecules of a DNA and a water cluster respectively. Now, we would like to provide some numerical results along which we display equally the runtimes of the cavity generation. Note that these are not the final version of the execution time since our software is constantly under development and improvement. We give only these numbers so that the readers have a certain idea about the expected time. The running time depends of course on several factors. First, it depends on the number of atoms in the molecule. Second, it depends on the atom distributions. Further, it depends on how coarse the patch decomposition should be. Finally, it depends on the level used in the dyadic points generations. As a first numerical test, we want to investigate the number of patches in accordance to the size of the molecule. The results of such a test are gathered in Tab. 6.1. According to our experience, the interesting practical values of α range between 0.2 and 0.4. A smaller value of α indicates that one has a coarser decomposition. It amounts also to fewer number of fitting tasks. On the other hand, a single large NURBS surface area needs many sampling points. It means also that it takes more time to complete

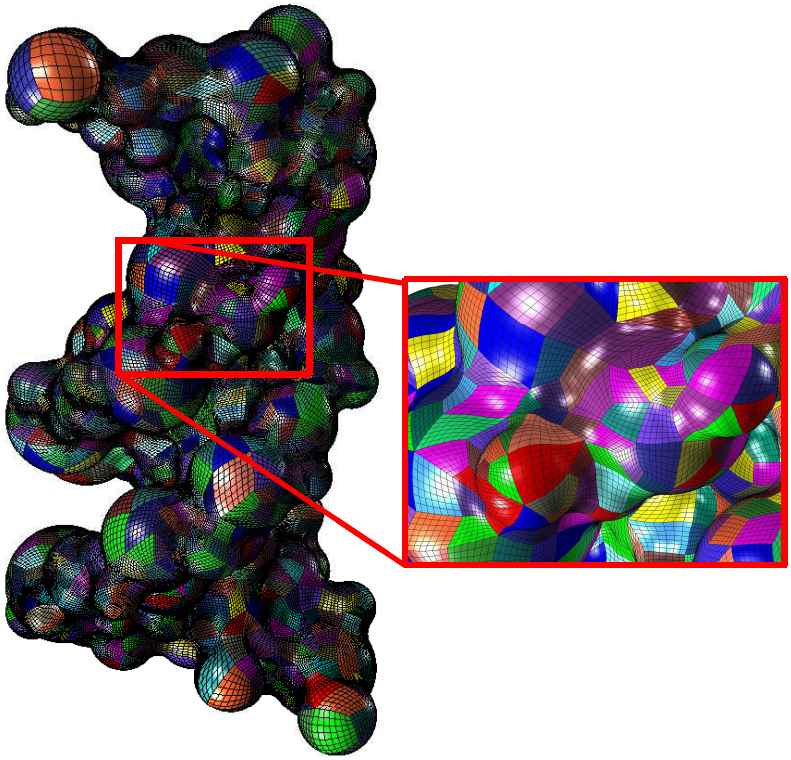


FIGURE 15. Patch representation of a DNA with 1905 NURBS

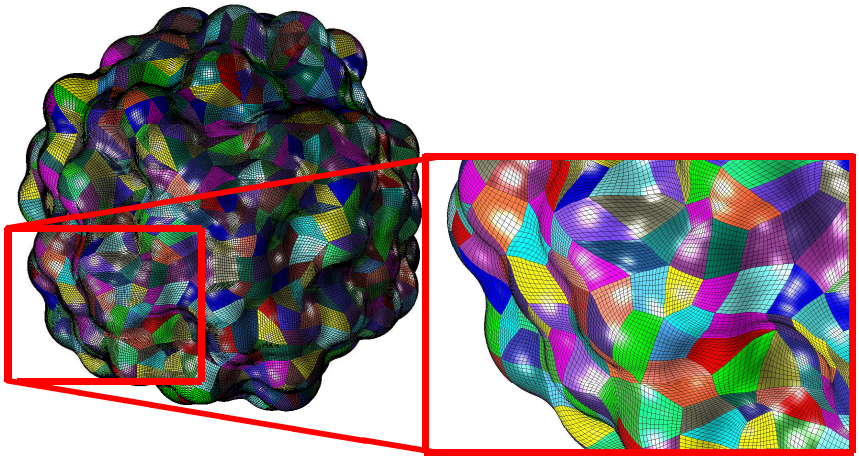


FIGURE 16. Patch representation of a water cluster with 1089 NURBS

the NURBS determination. Hence, the running time depends not only on the initial molecular size but also on the surface area of the cavity. For example, lecithin has more atoms than DNA but the latter takes almost four times longer than the former. The surface areas can be reflected from the number of patches.

Molecules	Nb. atoms	Number of patches (Runtime in second)		
		$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$
Benzene	12	84 (13.86 s)	97 (14.06 s)	119 (14.60 s)
Cyclohexane	18	99 (15.27 s)	108 (16.27 s)	127 (16.13 s)
LDS	49	286 (50.36 s)	346 (53.73 s)	409 (52.76 s)
Streptomycin	81	399 (68.69 s)	468 (67.78 s)	564 (74.02 s)
Lecithin	128	687 (118.50 s)	821 (120.68 s)	932 (145.11 s)
PDMPG	225	1102 (219.23 s)	1345 (219.12 s)	1556 (226.34 s)
DNA	116	1610 (461.67 s)	1899 (475.33 s)	2232 (487.38 s)

TABLE 6.1. Number of patches with respect to the number of atoms and the coarseness factor α.

The purpose of the second test is the investigation of the size $\mu(\Gamma_i)$ of each patch Γ_i. We would like to compare the areas of the patches in two perspectives: in the vicinity of each patch and global comparison. For the vicinity test, let $\mathcal{N}_i$ denote the set of neighboring patches which share at least one corner node with a patch Γ_i. We have computed

$$M(i) := \mu(\Gamma_i) \Big/ \frac{1}{m_i} \sum_{j \in \mathcal{N}_i} \mu(\Gamma_j) \qquad \text{where} \qquad m_i = \text{Card}(\mathcal{N}_i). \tag{6.48}$$

The results for the average values of $M(i)$ are collected in the third column of Tab. 6.2 where we observe that the sizes of the patches vary slowly because $M(i)$ approximates averagely the unity. That fact illustrates clearly the advantage of the Connolly surfaces over the van der Waals surface [8]. Since patches are allowed here to span several atoms. The size and the shape of the patches are in general very nice. As for the global comparison, the ideal patch area μ_{ideal} of a patch is the area of the whole molecular surface divided by the number of patches. We compute for each patch Γ_i the ratio

$$R(i) := \min \Big\{ \mu(\Gamma_i)/\mu_{\text{ideal}},\, \mu_{\text{ideal}}/\mu(\Gamma_i) \Big\} \in [0, 1]. \tag{6.49}$$

The resulting average values of $R(i)$ are seen in the last column of Tab. 6.2. We observe that the sizes of the patches are not too different from the ideal size. In fact, we have that in general $\mu = \rho\mu_{\text{ideal}}$ (or vice versa $\mu_{\text{ideal}} = \rho\mu$) with a factor $\rho \in [0.65, 0.75]$.

Our third test consists in investigating the quality of the patches. Consider a patch Γ_k by using the level L as illustrated in Fig. 14(b). For each $i = 0, ..., 2^L - 1$ and $j = 0, ..., 2^L - 1$, we consider the quadrilateral $Q_k(i, j)$ embedded in the space such that the vertices of $Q_k(i, j)$ are $a_k(i, j) := \boldsymbol{\gamma}_k(i/2^L, j/2^L)$, $a_k(i+1, j) := \boldsymbol{\gamma}_k((i+1)/2^L, j/2^L)$, $a_k(i + 1, j + 1) := \boldsymbol{\gamma}_k((i + 1)/2^L, (j + 1)/2^L)$, $a_k(i, j + 1) := \boldsymbol{\gamma}_k(i/2^L, (j + 1)/2^L)$. By inserting a diagonal $[a_k(i, j + 1), a_k(i + 1, j)]$ in $Q_k(i, j)$, one obtains two triangles T_1

Molecules	Number of patches	area/(neighb area)	area /ideal
Benzene	12	0.960095	0.706800
Cyclohexane	97	0.959639	0.695809
Tamoxifen	410	1.012924	0.665099
Streptomycin	462	0.959130	0.662762
Lecithin	821	0.971143	0.676250
PDMPG	2175	1.015307	0.688102
DNA	3144	0.950066	0.651584

TABLE 6.2. Investigation of patches area.

and T_2. Two other triangles T_3 and T_4 by inserting the diagonal $[a_k(i,j), a_k(i+1,j+1)]$. Let us denote by $\theta(T_p)$ the smallest angle in the triangle T_p. In order to quantify the quality of the quadrilateral $Q_k(i,j)$, we introduce the quality metric

$$\Psi\big(Q_k(i,j)\big) := 0.5\Big[\min\Big(\theta(T_1),\theta(T_2)\Big) + \min\Big(\theta(T_3),\theta(T_4)\Big)\Big]. \tag{6.50}$$

The ideal quadrilateral which is a perfect square corresponds to $\Psi(Q_k(i,j)) = 0.25\pi$. In general, it is impossible to attain the ideal value $\Psi_{\text{ideal}} = 0.25\pi \simeq 0.785398$ because all patches could not be a scaling of the unit square. A value tending to zero indicates a bad quad quality. In Tab. 6.3, we gather the results of our tests which consists in computing the average values of Ψ over the whole patches. We find there also the ratio between the Ψ-value and the ideal value Ψ_{ideal}

Molecules	Number of patches	Ψ-value	
		Average	Ratio with Ψ_{ideal}
Benzene	133	0.595737	0.758516
Quinine	358	0.554006	0.705383
Borane	812	0.531560	0.676803
Lecithin	821	0.555027	0.706682
Water cluster	1567	0.571700	0.727911
DNA	3348	0.576848	0.734465

TABLE 6.3. Quality of the resulting patches.

In virtue of the wavelet Galerkin BEM, we need only the value at the dyadic stepsize as illustrated in Fig.14(b). That is, at level L for each NURBS patch Γ_k, we need for each one the values

$$a_k(i,j) = \boldsymbol{\gamma}_k(i/2^L, j/2^L) \qquad \forall\, i,j = 0,1,...,2^L. \tag{6.51}$$

Our next test is the runtime for the punctual evaluations. The results are seen in Tab. 6.4 where we use two levels $L = 3$ and $L = 5$ for seven different molecules.

Molecules	Nb. patches	Levels for multiscale	
		L=3	L=5
Benzene	97	7857 points (0.28 s)	105.64 K-points (2.81 s)
Cyclohexane	108	8748 points (0.30 s)	117.62 K-points (2.54 s)
Streptomycin	399	32.32 K-points (0.97 s)	434.52 K-points (10.12 s)
LDS	438	35.56 K-points (1.00 s)	478.07 K-points (10.36 s)
Lecithin	821	66.51 K-points (2.04 s)	894.07 K-points (25.80 s)
PDMPG	1344	108.87 K-points (3.37 s)	1463.62 K-points (47.81 s)
DNA	1899	153.82 K-points (4.82 s)	2068.02 K-points (71.37 s)

TABLE 6.4. Generation of the dyadic points on the Connolly surfaces (K=kilo).

REFERENCES

[1] C. BAJAJ, G. XU, Q. ZHANG, *A fast variational method for the construction of adaptive resolution C^2 smooth molecular surfaces*, Comput. Methods Appl. Mech. Eng. **198**, 1684-1690 (2009).

[2] H. BAST, K. MEHLHORN, G. SCHÄFER, H. TAMAKI, *A heuristic for Dijkstra's algorithm with many targets and its use in weighted matching algorithms*, Algorithmica **36**, No. 1, 75-88 (2003).

[3] G. BRUNNETT, *Geometric design with trimmed surfaces*, Comput. Suppl. **10**, 101–115 (1995).

[4] M. CONNOLLY, *Analytical molecular surface calculation*, J. Appl. Cryst. **16**, 548–558 (1983).

[5] B. ELSÄSSER, *Approximation mit rationalen B-Spline Kurven und Flächen*, *Ph.D. thesis*, Darmstadt (1998).

[6] J. GALLIER, *Geometric methods and applications for computer science and engineering*, Springer Verlag, TAM Vol. 38 (2000).

[7] H. HARBRECHT, M. RANDRIANARIVONY, *From Computer Aided Design to wavelet BEM*, Comput. Vis. Sci. **13**, No. 2, 69–82 (2010).

[8] H. HARBRECHT, M. RANDRIANARIVONY, *Wavelet BEM on molecular surfaces: parametrization and implementation*, Computing **86**, 1–22 (2009).

[9] J. HOSCHEK, D. LASSER, *Fundamentals of computer aided geometric design*, A.K. Peters, Wellesley (1993).

[10] B. LEE, F. RICHARDS, *The interpretation of protein structures estimation of stratic accessibility*, J. Mol. Biol. **55**, 379–400 (1971).

[11] L. PIEGL, W. TILLER, *The NURBS book*, Springer, Berlin (1995).
[12] C. POMELLI, *A tessellationless integration grid for the polarizable continuum model reaction field*, J. Comp. Chem. **25**, 1532–1541 (2004).
[13] M. RANDRIANARIVONY, *Geometric processing of CAD data and meshes as input of integral equation solvers*, Ph.D. thesis, Technische Universität Chemnitz, Germany (2006).
[14] M. RANDRIANARIVONY, G. BRUNNETT, *Molecular surface decomposition using geometric techniques*, In: Proc. Conf. Bildverarbeitung für die Medizine, Berlin, 197–201 (2008).
[15] M. RANDRIANARIVONY, *On global continuity of Coons mappings in patching CAD surfaces*, Comput.-Aided Design **41**, No. 11, 782–791 (2009).
[16] M. RANDRIANARIVONY, G. BRUNNETT, *Preparation of CAD and molecular surfaces for meshfree solvers*, Lect. Notes Comput. Sci. Eng. **65**, 231–245 (2008).
[17] M. RANDRIANARIVONY, *Harmonic variation of edge size in meshing CAD geometries from IGES format*, Lect. Notes Comput. Sci. **5102**, pp. 56-65 (2008).
[18] J. TOMASI, B. MENNUCCI, R. CAMMI, *Quantum Mechanical Continuum Solvation Models*, Chem. Rev. **105**, 2999–3094 (2005).
[19] U. S. PRODUCT DATA ASSOCIATION, *Initial Graphics Exchange Specification. IGES 5.3*. Trident Research Center, SC (1996).
[20] V. WEIJO, M. RANDRIANARIVONY, H. HARBRECHT, L. FREDIANI, *Wavelet formulation of the Polarizable Continuum Model*, J. Comput. Chem. **31**, No. 7, pp. 1469-1477 (2010).
[21] D. WHITLEY, *Van der Waals graphs and molecular shape*, J. Math. Chem. **23**, 377–397 (1998).

Maharavo Randrianarivony, Institut für Numerische Simulation, Universität Bonn, Wegelerstr. 6, 53115 Bonn, Germany.

E-mail address: randrian@ins.uni-bonn.de