Ambient Isotopic Meshing of Implicit Algebraic Surface with Singularities

Jin-San Cheng[1,2]*, Xiao-Shan Gao[1]†, Jia Li[1]‡
[1]Key Lab of Mathematics Mechanization
Institute of Systems Science, AMSS, Academia Sinica
[2]Loria, INRIA Nancy

Abstract

A complete method is proposed to compute a certified, or ambient isotopic, meshing for an implicit algebraic surface with singularities. By certified, we mean a meshing with correct topology and any given geometric precision. We propose a symbolic-numeric method to compute a certified meshing for the surface inside a box containing singularities and use a modified Plantinga-Vegter marching cube method to compute a certified meshing for the surface inside a box without singularities. Nontrivial examples are given to show the effectiveness of the algorithm (see Fig. 1). To our knowledge, this is the first method to compute a certified meshing for surfaces with singularities.

Keywords. Surface, curve, topology, ambient isotopic meshing, marching cube, symbolic computation, interval arithmetic.

1 Introduction

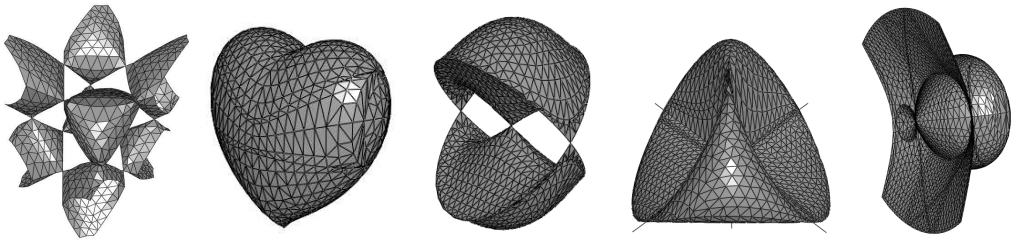

Figure 1: Isotopic meshing for surfaces with singular points and singular curves

To determine the topology of a given algebraic surface and to use triangular meshes to approximately represent the surface are fundamental operations in computer graphics and geometric model-

*e-mail: jcheng@amss.ac.cn
†e-mail:xgao@mmrc.iss.ac.cn
‡lijia@mmrc.iss.ac.cn

ing. Meshing of surfaces could be used to display the surface correctly and to perform engineering applications on the surface, such as the finite element analysis. A survey on this topic can be found in [5].

We consider an implicit surface defined by $f(x, y, z) = 0$ where $f(x, y, z)$ is a square free polynomial with rational numbers as coefficients. There exists a large amount of work on meshing implicit surfaces. Please see the work [1, 4, 26] and the literatures cited in them. Recent work focuses on isotopic meshing [5]. Simply speaking, a meshing is called **isotopic** if it has the same topology and the same geometry as the surface (for definition see Section 2). A meshing is called **ambient isotopic** or **certified** if it is isotopic and approximates the surface to any give precision. There exist four main approaches to compute isotopic meshings for surfaces: the marching cube method, the Morse theory method, the Delaunay refinement method, and the CAD (Cylindrical Algebraic Decomposition) based method.

The famous marching cube method repeatedly subdivides the space into smaller cubes until the structure of the surface inside each cube is known [19]. For implicit surfaces, Snyder proposed the globally parameterizable criterion for that purpose [28]. Plantinga and Vegter proposed the small normal variation condition which leads to a better meshing algorithm [24, 25].

Hart et al proposed a method based on Morse theory [18, 23, 29]. The idea is to check when the topology of $f(x, y, z) = a$ will change for a parameter a. When a changes from some initial value where $f(x, y, z) = a$ has no solution to $a = 0$, the topology of the surface is found. Fortuna et al presented improved algorithms for surfaces in the projective space [15, 14].

For a set of points on the surface, one can form the restricted Delaunay triangles and the corresponding Delaunay triangulation can be used to approximate the surface. Boissonnat and Oudot proved that when the sample point set satisfies certain conditions, the Delaunay triangulation has the same topology as the surface [7, 6]. Cheng et al established similar results using different strategies [12].

The CAD method proposed by Collins can be used to divide the Euclidean space into cylindrical cells such that the given surface has the same sign on each of the cells. Then to determine the topology of the surface, we need only give the adjacency information between the cells [3, 20]. Alone this line, new ideas are introduced to compute the topology of surfaces [10, 22].

All the above methods except the one based on CAD work for surfaces without singularities only. In this paper, we give a method to compute a certified meshing for implicit algebraic surfaces with singularities. The method is a hybrid one based on the CAD approach and the marching cube approach. We propose a CAD based method to compute a certified meshing for the surface inside a box containing singularities and use a modified Plantinga-Vegter method to compute a certified meshing for the surface inside a box without singularities. Our main contribution is how to treat the singularities.

This paper consists of three parts. The algorithms for surfaces are the main contributions. In Section 3, a new method is proposed to compute a certified meshing for a plane algebraic curve. This section also provides preparations for algorithms about surfaces. There exist many methods to compute the topology of plane curves, e.g., [2, 9, 13, 17]. Our contribution is to give an interval based method to compute the adjacency information and to give an ambient isotopic meshing for a curve. The method in [9] can also compute an ambient isotopic meshing based on root bounds of equation systems. Our method is based on symbolic-numerical computation, which is practically more effective.

In Section 4, a new method is proposed to compute an isotopic meshing for a surface. The method has two advantages. First, we use symbolic computation methods to guarantee the completeness and whenever possible use interval arithmetics to increase the efficiency. Actually, computations of algebraic numbers are totally avoided. The work [2, 20] uses algebraic numbers. Second, our algorithm does not change the surface to generic positions as done in [22], which is generally expensive. Our method need only to project the surface once, while the algorithm proposed in [22] need to do projections twice.

In Section 5, a method is proposed to compute a certified meshing for a surface. A well-known technical to treat a singular point P is to find a **segregating box** which contains P but does not intersect the surface at its bottom and top faces. We f extend this concept to singular curve segments and give an interval based method to compute such boxes and meshes in the boxes. Another key ingredient is a careful analysis of the extremal points of surfaces and spatial curves. It is pointed out in [5], that the method in [22] "makes no guarantees about the geometric accuracy of the mesh, and it cannot be extended in a straightforward way to provide a more accurate mesh." To our knowledge, the method proposed in this paper is the first one to compute a certified meshing for surfaces with singularities.

Algorithms in Sections 3 and 4 are implemented in Maple and nontrivial examples are used to show that the algorithm is quite effective for surfaces with singular points and curves.

2 Preparations

In this section, we give several known results and algorithms needed in this paper. Following [5], we will compute a meshing with correct topology for a curve or a surface in the following sense.

An **isotopic meshing** for a variety $\mathcal{S} \subset \mathbb{R}^n$ $(n = 2, 3)$ consists of a graph/polyhedron $\mathscr{G}$ (for $n = 2, 3$) and a continuous mapping $\gamma : \mathbb{R}^n \times [0, 1] \to \mathbb{R}^n$ which, for any fixed $t \in [0, 1]$, is a homeomorphism $\gamma(\cdot, t)$ from R^n to itself, and which continuously deforms $\mathscr{G}$ into S: $\gamma(\cdot, 0) = id$, $\gamma(\mathscr{G}, 1) = \mathcal{S}$.

For a number $\epsilon > 0$, an **ϵ-meshing** for $\mathcal{S}$ is an isotopic meshing $\mathscr{G}$ for $\mathcal{S}$, which gives an ϵ-approximation for S in the following sense $\| P - \gamma(P, 1) \| \leq \epsilon$ for all $P \in \mathscr{G}$. Please note that isotopy is stronger than homeomorphism [5].

2.1 Real root isolation of triangular system

A basic step of our algorithm is to isolate the real roots of a **triangular system** which consists of equations like

$$\Sigma_n = \{f_1(x_1), f_2(x_1, x_2), \ldots, f_n(x_1, x_2, \ldots, x_n)\} \tag{1}$$

where $f_i \in \mathbb{Q}[x_1, \ldots, x_i]$ involves x_i effectively.

We use intervals to isolate real numbers: let $\square\mathbb{Q}$ denote the set of intervals of the form $[a, b]$ where $a < b \in \mathbb{Q}$. The **length** of an interval box $\mathbf{B}_n = [a_1, b_1] \times \cdots \times [a_n, b_n] \in \square\mathbb{Q}^n$ is defined to be $|\mathbf{B}_n| = \max_i(b_i - a_i)$.

In this paper, when we say a **point**, we mean a point $P = (\xi_1, \ldots, \xi_n)$ with real algebraic numbers as coordinates, which is represented by a triangular system Σ_n like (1) with P as a solution and an isolation box $\mathbf{B}_n$ for ξ. For instance $\sqrt{2}$ is represented by $x_1^2 - 2 = 0$ and $(1, 2)$.

Now, we give a formal description of the root isolation algorithm.

Algorithm 2.1 **RootIsol**$(\Sigma_n, \mathbf{B}_n, \epsilon)$. *The input consists of a triangular system Σ_n of form (1), a box $\mathbf{B}_n \in \square\mathbb{Q}^n$, and a positive number ϵ. The output is a set of isolation boxes for all the real roots of $\Sigma_n = 0$ in $\mathbf{B}_n$ such that the length of the isolation boxes is smaller than ϵ and any two of the isolation boxes are disjoints.*

A modified version of the root isolation algorithms in [11, 27] is used in our implementation.

Let $f(x_1, \ldots, x_n) \in \mathbb{Q}[x_1, \ldots, x_n]$ and $\mathbf{B}_n = [a_1, b_1] \times \cdots \times [a_n, b_n] \in \square\mathbb{Q}^n$. The **box operation** $\square f(\mathbf{B}_n)$ returns an interval containing all the points $\{f(x_1, \ldots, x_n) \,|\, a_i \leq x_i \leq b_i, i = 1, \ldots, n\}$. Furthermore, when $|\mathbf{B}_n|$ approaches to zero, the length of interval $\square f(\mathbf{B}_n)$ also approaches to zero. If $a_i > 0$ and $b_i > 0$, we can construct $\square f(\mathbf{B}_n)$ as follows

$$\square f(\mathbf{B}_n) = f^+(b_1, \ldots, b_n) - f^-(a_1, \ldots, a_n)$$

where $f = f^+ - f^-$ such that $f^+, f^- \in \mathbb{Q}[x_1 \ldots, x_n]$ each has only positive coefficients and minimal number of monomials. For the general case, please consult [11]. It is clear that such an operation satisfies the following property.

Lemma 2.2 *If $\xi = (\xi_1, \ldots, \xi_n)$ is not a zero of $f(x_1, \ldots, x_n) = 0$ and $\mathbf{B}_n$ an isolation box for ξ. Then if the length of $\mathbf{B}_n$ is small enough, the interval $\square f(\mathbf{B}_n)$ will not contain $(0, \ldots, 0)$, which means that $\mathbf{B}_n$ has no intersections with $f = 0$. We denote this as $\square f(\mathbf{B}_n) \neq 0$.*

2.2 Delineable polynomials

Delineable polynomials are important in determining the topology of algebraic surfaces. Let $f(x_1, \ldots, x_{r-1}, x_r) \in \mathbb{R}[x_1, \ldots, x_r]$ and $P = (p_1, \ldots, p_r)$ a point of $\mathbb{R}^r$. We say that f has **order** k at point P, if $k \geq 0$ is the least non-negative integer such that some partial derivative of total order k does not vanish at P. And f is said to be **order-invariant** in a subset R of $\mathbb{R}^r$ provided that the order of f is the same at every point of R.

For simplification, we denote the $(r-1)$-tuple $(x_1, \ldots, x_{r-1})$ as $\bar{x}$. An r-variate polynomial $f(\bar{x}, x_r)$ over the reals is said to be **delineable** on a submanifold R of $\mathbb{R}^{r-1}$ if it holds that:

(1) the portion of the real variety of f that lies in the cylinder $R \times \mathbb{R}$ over R consists of the union of the function graphs of some $k > 0$ analytic functions $\theta_1 < \ldots < \theta_k$ from R into $\mathbb{R}$; and

(2) there exist positive integers m_i such that for every $\alpha \in R$, the multiplicity of the root of $f(\alpha, x_r)$ corresponding to θ_i is m_i.

Polynomial f is said to **vanish identically** on R if $f(P, x_r) = 0$ for every point $P \in R$. In addition, f is said to be **degree-invariant** on R if the degree of $f(P, x_r)$ as a polynomial in x_r is the same for every point $P \in R$. In this situation, the following theorem holds (see [20], pp. 246).

Theorem 2.3 *(McCallum and Collins) Let $f(\bar{x}, x_r)$ be a polynomial in $\mathbb{R}[\bar{x}, x_r]$ of positive degree in x_r. Let $D(\bar{x})$ be the discriminant of f as a univariate polynomial in x_r and suppose that $D(\bar{x})$ is a nonzero polynomial. Let R be a connected submanifold of $\mathbb{R}^{r-1}$ on which f is degree-invariant and does not vanish identically, and over which D is order-invariant. Then, f is delineable on R.*

The following theorem improves the above result.

Theorem 2.4 *([8]) Let $f \in \mathbb{R}(\bar{x}, x_r)$ ($r \geq 2$) be an r-variate polynomial of positive degree in x_r with discriminant $D(\bar{x}) \neq 0$. Let R be a connected submanifold of $\mathbb{R}^{r-1}$ in which D is order-invariant, the leading coefficient of f w.r.t. x_r is sign-invariant, and such that f vanishes identically at no point in R. Then, f is degree-invariant on R.*

3 Ambient isotopic meshing of plane curve

In this section, we give an algorithm to compute an isotopic meshing for an algebraic curve. The main purpose of this section is to provide preliminary algorithms for later sections. We also give a new and fast method to compute the adjacency information based on interval arithmetics.

3.1 Determine the topology of plane algebraic curve

We use a graph to represent the topology of a plane curve. A **topology graph** is a graph $\mathscr{G} = \{\mathcal{P}, \mathcal{E}\}$ where

- $\mathcal{P}$ is a set of plane points defined by triangular systems Σ_i and isolation boxes $\mathbf{B}_{i,j}$:

$$\begin{aligned} \mathcal{P} &= \{P_{i,j} = (\alpha_i, \beta_{i,j}), 0 \leq i \leq s, 0 \leq j \leq s_i\} \\ \Sigma_i &= \{h_i(x), g_i(x,y)\}, \mathbf{B}_{i,j} = [a_i, b_i] \times [c_{i,j}, d_{i,j}] \end{aligned} \tag{2}$$

  where $\alpha_0 < \alpha_1 < \cdots < \alpha_s$ and $\beta_{i,0} < \beta_{i,1} < \cdots < \beta_{i,s_i}$. When drawing the graph, we use $M_{i,j} = ((a_i + b_i)/2, (c_{i,j} + d_{i,j})/2)$ to represent $P_{i,j}$.

- $\mathcal{E} = \{(P_1, P_2) | P_1, P_2 \in \mathcal{P}$, such that either $P_1 = P_{i,p}, P_2 = P_{i+1,q}$ or $P_1 = P_{i,p}, P_2 = P_{i,p+1}\}$. In the first case, the edge is called **non-vertical**. In the second case, the edge is called x**-vertical**. We further assume that any two edges do not intersect except at the end points.

Consider a plane algebraic curve $\mathcal{C} : g(x, y) = 0$ where $g(x, y) \in \mathbb{Q}[x, y]$ is a square free polynomial. A point P_0 is an x**-critical point** of $\mathcal{C}$ if $g(P_0) = g_y(P_0) = 0$.

We will consider the part of $\mathcal{C}$ in a bounding box

$$\mathbf{B}_2 = [\mathcal{X}_1, \mathcal{X}_2] \times [\mathcal{Y}_1, \mathcal{Y}_2] \in \square\mathbb{Q}^2 \tag{3}$$

which is denoted as $\mathcal{C}_{\mathbf{B}_2} = \mathcal{C} \cap \mathbf{B}_2$. In the rest part of this paper, $\mathbf{B}_2$ is always assumed to be of this form.

Let P be a point on curve $\mathcal{C}$, the **left (right) branch number of** P, also denoted as $L\#(P)$ ($R\#(P)$), is the number of curve segments of $\mathcal{C}$ which pass through P and are on the left (right) side of P in a small neighborhood of P.

We introduce the key concept of segregating box. A box $\mathbf{B} = [a, b] \times [c, d] \in \square\mathbb{Q}^2$ is called **segregating** w.r.t. $\mathcal{C}$ if

$$\mathcal{C} \cap [a, b] \times [c, c] = \mathcal{C} \cap [a, b] \times [d, d] = \emptyset.$$

A curve behaves nicely in a segregating box, as illustrated by the following lemma. See Fig. 2(a) for an illustration.

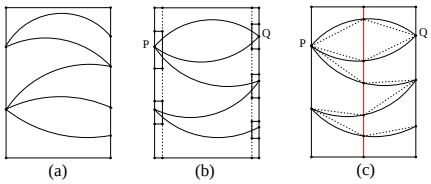

Figure 2: Curve segments inside a segregating box.

Lemma 3.1 *Let $\mathbf{B} = [a, b] \times [c, d] \in \square\mathbb{Q}^2$ be a box segregating w.r.t. $\mathcal{C}$ and the interior of $\mathbf{B}$ contains no $x-$critical points of $\mathcal{C}$. Let $\mathcal{C}$ intersect the left and right boundaries of $\mathbf{B}$ at points $L_i, i = 1, \ldots, l$ and $R_j, j = 1, \ldots, r$ respectively. Then $\mathcal{C}$ is delineable over $R = (a, b)$ and the number of curve segments of $\mathcal{C}$ inside $\mathbf{B}$ equals $\sum_{i=1}^{l} R\#(L_i) = \sum_{j=1}^{r} R\#(R_i)$. (See Figure 2(a) for an illustration)*

Proof. Note that the leading coefficient $C(x)$ of $g(x, y)$ w.r.t. to y is a factor of the discriminant $D(x)$ of $g(x, y)$ as a univariate polynomial in y. Since there exist no x-critical points of $\mathcal{C}$ inside $\mathbf{B}$, $C(x)$ is not zero. Hence $g(x, y)$ is degree invariant over R. Also $D(x) = 0$ has no roots over R. Then, by Theorem 2.3, $\mathcal{C}$ is delineable over R and $\mathcal{C}_{\mathbf{B}}$ consists of curve segments starting from certain L_i and ending at certain R_j. Furthermore, these curve segments do not intersect. So $\sum_{i=1}^{l} R\#(L_i) = \sum_{j=1}^{r} R\#(R_i)$. This proves the lemma. ∎

A box $\mathbf{B}$ is called a **segregating box** for a point P on $\mathcal{C}$ if P is inside $\mathbf{B}$, $\mathbf{B}$ is segregating w.r.t. $\mathcal{C}$, and $\mathcal{C}_{\mathbf{B}} \setminus \{P\}$ contains no x-critical points of $\mathcal{C}$. See Fig. 3(a) for an illustration. It is known that (Theorem 5 in [2]):

Lemma 3.2 *If $\mathbf{B} = [a, b] \times [c, d] \in \square\mathbb{Q}^2$ is a segregating box of P on $\mathcal{C}$, then $R\#(P)$ and $L\#(P)$ are the numbers of real roots of $g(b, y) = 0$ and $g(a, y) = 0$ in (c, d) respectively. See Fig. 2(b).*

The following algorithm computes the branch numbers.

Algorithm 3.3 **NumCur**$(\mathcal{P})$. *$\mathcal{P}$ is a set of points defined by (2). Output $R\#(P_{i,j})$ for $0 \le i \le s-1$ and $L\#(P_{i,j})$ for $1 \le i \le s$.*

1. For $0 \le i \le s$, if $g_i(x, y)$ has a factor of the form $V(x) \in \mathbb{Q}[x]$, let $g_i = g_i/V(x)$.
2. While $0 \in \square g_i([a_i, b_i], c_{i,j})$ or $0 \in \square g_i([a_i, b_i], d_{i,j})$, repeat $[a_i, b_i] = \mathbf{RootIsol}(h_i(x), [a_i, b_i], (b_i - a_i)/2)$.
3. Let $R = \mathbf{RootIsol}(g(b_i, y), [c_{i,j}, d_{i,j}], 1)$ and $L = \mathbf{RootIsol}(g(a_i, y), [c_{i,j}, d_{i,j}], 1)$. By Lemma 3.2, $R\#(P_{i,j}) = |R|$ and $L\#(P_{i,j}) = |L|$. (See Fig. 2(b))

Proof of correctness. Since $\mathbf{B}_{i,j}$ is an isolation box for $P_{i,j}$, then $g_i(\alpha_i, c_{i,j})g_i(\alpha_i, d_{i,j}) \neq 0$. By Lemma 2.2, the procedure in Step 2 will terminate. At the end of Step 2, $g_i(x, c_{i,j})g_i(x, d_{i,j}) = 0$ has no real roots in $[a_i, b_i]$, that is, $\mathbf{B}_{i,j}$ is a segregating box for $P_{i,j}$. The third step is clearly true. ∎

Remark 3.4 *In Step 2 of Algorithm 3.3, the boundary points need special consideration. If the boundary points are on the curve, that is, if $g(\alpha_i, \mathcal{Y}_1) = 0$ or $g(\alpha_i, \mathcal{Y}_2) = 0$, we will make sure that the following condition holds: α_i is the only real root of $g_i(x, \mathcal{Y}_1) = 0$ or $g_i(x, \mathcal{Y}_2) = 0$ in $[a_i, b_i]$. Then, the algorithm also works.*

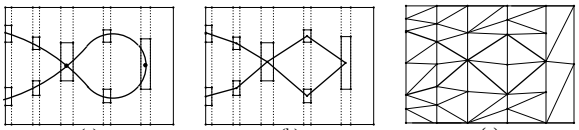

Figure 3: Compute topology graph of a curve

The following algorithm to compute a topology graph follows the basic idea in [2]. Our main contribution is to use interval arithmetics instead of algebraic numbers. Also, we do not need changing the curve to generic positions as done in [13, 17]

Algorithm 3.5 **TopCur**$(g(x, y), \mathbf{B}_2, \epsilon)$. *$\mathcal{C} : g(x, y) = 0$ is the curve, $\mathbf{B}_2$ is defined in (3), and $\epsilon > 0$ is a number. Output a topology graph $\mathcal{G} = (\mathcal{P}, \mathcal{E})$ which is an* **isotopic meshing** *for $\mathcal{C}_{\mathbf{B}_2}$. Further, each isolation box $\mathbf{B}$ of a point in $\mathcal{P}$ satisfies $|\mathbf{B}| \leq \epsilon$.*

1. Let $\mathcal{E} = \emptyset$ and $g(x, y) = V(x)g_v(x, y)$, where $V(x)$ is the factor of $g(x, y)$ in x only.
2. Let $D(x) = \text{Res}(g_v, \frac{\partial g_v}{\partial y}, y)$ be the resultant of g_v and $\frac{\partial g_v}{\partial y}$.
3. Let $\mathcal{P} =$**RootIsol**$(\Sigma_{21}, \mathbf{B}, \epsilon) \cup$**RootIsol**$(\Sigma_{22}, \mathbf{B}, \epsilon)$, where

$$\begin{aligned} H(x) &= (x - \mathcal{X}_1) \cdot (x - \mathcal{X}_2) \cdot g_v(x, \mathcal{Y}_1) \cdot g_v(x, \mathcal{Y}_2) \cdot D(x) \\ H_v(x) &= H(x)/\gcd(H(x), V(x)) \\ \Sigma_{21} &= \{H_v(x), g_v(x, y)\} \\ \Sigma_{22} &= \{V(x), g_v(x, y)(y - \mathcal{Y}_2)(y - \mathcal{Y}_1)\}. \end{aligned} \tag{4}$$

   Assume that $\mathcal{P}$ is of form (2). See Fig. 3(a) for an illustration.

4. Execute Algorithm 3.3 to compute $L\#(P_{i,j})$ $(1 \leq i \leq s)$ and $R\#(P_{i,j})$ $(0 \leq i \leq s - 1)$.
5. Add an auxiliary line at $x = (b_i + a_{i+1})/2$ and construct the non-vertical edges. See Fig 2(c) for an illustration. For $i = 0, \ldots, s - 1$, execute the following steps
   (a) Let $\mathcal{Q}_i =$**RootIsol**$(\{x - \frac{b_i + a_{i+1}}{2}, g_v(x, y)\}, \mathbf{B}, \epsilon)$, where a_i, b_i are from (2). Arrange the points in $\mathcal{Q}_i$ bottom up, we have $\mathcal{Q}_i = \{Q_{i,1}, \ldots, Q_{i,u_i}\}$. Set $R\#(Q) = L\#(Q) = 1$.
   (b) Let $\mathcal{R}_i$ be the list of points $P_{i,k}$ arranged bottom up and point $P_{i,k}$ will be repeated $R\#(P_{i,k})$ times in R_i. Similarly, $\mathcal{L}_i$ is the list of points $P_{i+1,t}$. By Lemma 3.1, $\mathcal{L}_i, \mathcal{R}_i$, and $\mathcal{Q}_i$ contain the same number of points. Let $\mathcal{L}_i = (L_1, \ldots, L_{u_i}), \mathcal{R}_i = (R_1, \ldots, R_{u_i})$.
   (c) For $j = 1, \ldots, u_i$, add $(L_j, Q_{i,j}), (Q_{i,j}, R_j)$ to $\mathcal{E}$.
   (d) Let $\mathcal{P} = \mathcal{P} \cup \mathcal{Q}_i$. Still assume that $\mathcal{P}$ is of form (2).
6. Add the x-vertical edges. If α_i is a root of $V(x) = 0$, add $(P_{i,k}, P_{i,k+1}), k = 0, \ldots, s_i$ to $\mathcal{E}$.

7. Output the topology graph $\mathscr{G} = \{\mathcal{P}, \mathcal{E}\}$. The isotopy map can be constructed in the usual way [25].

Theorem 3.6 *Algorithm 3.5 computes an isotopic meshing for* $\mathcal{C}_{\mathbf{B_2}}$.

Proof. First, we prove that each edge $e \in \mathcal{E}$ represent exact one curve segment of $\mathcal{C}$, and for each degree invariant segment of $\mathcal{C}$, there exist exact one $e \in \mathcal{E}$ presenting it. Hence $\mathcal{E}$ and some y-vertical line decompose $\mathbf{B}_2$ into cylindric regions .

It is clear that the curve $\mathcal{C}$ consists of two parts $\mathcal{C}_v : g_v(x, y) = 0$ and $V(x) = 0$. The part $V(x) = 0$ consists of straight lines $x - \gamma_i = 0, i = 1, \ldots, t$, where γ_i are the real roots of $V(x) = 0$. To determine the topology of $\mathcal{C}$, we need only to find the topology graph $\mathscr{G}_v$ of $\mathcal{C}_v$ and then to add the lines $x - \gamma_i = 0$ to $\mathscr{G}_v$. So we may consider $\mathcal{C}_v$ only.

From Steps 2-4, we know that $\mathcal{P}$ contains all the x-critical points of the curve $\mathcal{C}_v$ and the *boundary points* which are the intersection points of $\mathcal{C}$ and the boundaries of $\mathbf{B}_2$. In Steps 6 and 7, we add auxiliary points $\mathcal{Q}_i$ to $\mathcal{P}$. Since points in $\mathcal{Q}_i$ are not critical points of $\mathcal{C}$, we have $R\#(Q) = L\#(Q) = 1$ for $Q \in \mathcal{Q}_i$. This makes sure that all the edges $(L_j, Q_{i,j})$ and $(Q_{i,j}, R_j)$ are distinct.

Let $B_i = (\alpha_i, \alpha_{i+1}) \times [\mathcal{Y}_1, \mathcal{Y}_2], i = 0, \ldots, s-1$. We need only to show that $\mathcal{C}_v$ and $\mathscr{G}_v$ have the same topology in B_i. Let S_i be the interval (α_i, α_{i+1}). Then $D(x)$ does not vanish on any point of S_i. As a consequence, $g_v(x, y)$ must be degree invariant on S_i. By Theorem 2.3, $g_v(x, y)$ is delineable over S_i and $\mathscr{G}_v$ is obtained by replacing a curve segment of $\mathcal{C}_v$ in B_i by a line segment with same end points. It is clear that $\mathcal{C}_v$ and $\mathscr{G}_v$ have the same topology. We are going to make explicit the isotopy from $\mathcal{E}$ to $\mathcal{C}_{\mathbf{B_2}}$. Let $\mathscr{G} = (\mathcal{P}, \mathcal{E})$ be a topology graph for curve $\mathcal{C}_{\mathbf{B_2}}$, and $\mathcal{P}$ of form (2). Let $P_{i,j} = (\alpha_i, \beta_{i,j})$ be of form (2). Let $Q_{i,j} = (\tau_i, \rho_{i,j})$ where $\tau_i = \frac{a_i + b_i}{2}, \rho_{i,j} = \frac{c_{i,j} + d_{i,j}}{2}$. Then $\mathscr{G}$ decomposes $\mathbf{B}_2$ into cylindrical regions $\cup_{i,j} R_{i,k}$, where $R_{i,j}$ is bounded by $[\tau_i, \tau_{i+1}]$ in the x-direction and by $f_1 = (Q_{i,u}, Q_{i+1,v})$ and $f_2 = (Q_{i,s}, Q_{i+1,t})$ for ceratin u, v, s, t. Note that $R_{i,k}$ could be a triangle or a quadrilateral.

First, we consider one cylindrical region $R_{i,k}$ defined as above. Let $e_1 = (P_{i,u}, P_{i+1,v}))$ and $e_2 = (P_{i,s}, P_{i+1,t})$. Without loss of generality, assume $\alpha_1 < \alpha_2, \beta_{1,1} < \beta_{1,2}$. According to the correctness prove of Algorithm 3.5, $g_v(x, y)$ is delineable over $[\alpha_1, \alpha_2]$, we can find two root functions $\theta_i(x)$ of g_v on $[\alpha_1, \alpha_2]$ corresponding to the two curve segments $C(e_1)$ and $C(e_2)$. Denote $y = \delta_i(x), x \in [\alpha_1, \alpha_2]$ to be the definition functions of line segments e_i and $y = \varphi_i(x), x \in [\tau_1, \tau_2]$ to be the definition functions of line segments f_i. Consider the maps:

$$F_1 : ([\alpha_1, \alpha_2] \times \mathbb{R}) \times [0, 1] \to \mathbb{R}^2$$

defined by

$$\begin{array}{rl} & (x, \lambda\delta_1(x) + (1 - \lambda\delta_2(x)), t) \\ \to & (x, \lambda(t\theta_1(x) + (1-t)\delta_1(x)) + (1-\lambda)(t\theta_2(x) + (1-t)\delta_2(x))) \end{array}$$

and

$$F_2 : ([\tau_1, \tau_2] \times \mathbb{R}) \times [0, 1] \to [\alpha_1, \alpha_2] \times \mathbb{R}$$

defined by

$$\begin{array}{rl} & (x, \lambda\varphi_1(x) + (1 - \lambda\varphi_2(x)), t) \\ \to & (x', \lambda(t\delta_1(x') + (1-t)\varphi_1(x)) + (1-\lambda)(t\delta_2(x') + (1-t)\varphi_2(x))), \end{array}$$

where $x' = \alpha_1 + \frac{x-\tau_1}{\tau_2-\tau_1}(\alpha_2 - \alpha_1)$.

The map F_2 is a homeomorphism from $[\tau_1, \tau_2] \times \mathbb{R}$ to $[\alpha_1, \alpha_2] \times \mathbb{R}$ and F_1 is a homeomorphism from $[\alpha_1, \alpha_2] \times \mathbb{R}$ to itself. So the composed map $F_{i,k} := F_1 \circ F_2$ is a homeomorphism from $[\tau_1, \tau_2] \times \mathbb{R}$ to $[\alpha_1, \alpha_2] \times \mathbb{R}$ and it deforms f_i to $C(e_i)$ continuously. Extend this map to $\mathbb{R}^2 \times [0, 1]$ by setting it to be the identity map outside $R_{i,k}$, we obtain an isotopy from line segments $f_1 \cup f_2$ to the curve segments $C(e_1) \cup C(e_2)$.

Now we consider the whole topology graph $\mathscr{G}$. For each cylindrical region $R_{i,k}$, we can construct an isotopy $F_{i,k}$ as above. Consider the following map:

$$F : \mathbb{R}^2 \times [0, 1] \to \mathbb{R}^2$$

denoted by

$$F(P, t) = \begin{cases} F_{i,j}(P, t), & P \in R_{i,j}, \\ id, & P \in \mathbb{R}^2 \setminus \mathbf{B}_2 \end{cases} \tag{5}$$

Note that $F_{i,j}|_{R_{i,j} \cap R_{u,v}} = F_{u,v}|_{R_{i,j} \cap R_{u,v}}$, and $F_{i,j}|_{R_{i,j} \cap (\mathbb{R}^2 \setminus \mathbf{B}_2)} = id$ for all i, j, u, v. $(\mathscr{G}, F)$ is an isotopy for $\mathcal{C}_{\mathbf{B_2}}$. ∎

As a consequence of the above proof, we have

Corollary 3.7 *Let $G = (\mathcal{P}, \mathcal{E})$ be a topology graph of the curve $\mathcal{C}_{\mathbf{B}_2}$ obtained by Algorithm 3.5. Then all the singular points of $\mathcal{C}_{\mathbf{B}_2}$ are in $\mathcal{P}$ and $g(x, y)$ is y-degree invariant over the intervals $(\alpha_i, \alpha_{i+1}), i = 0, \ldots, s-1$.*

When computing the topology of a surface, we need to introduce the concept of extended topology graph. An **extended topology graph** associate with a box $\mathbf{B}_2$ is a triplet $\mathcal{EG} = \{\mathcal{EP}, \mathcal{EE}, \mathcal{EC}\}$ where $\{\mathcal{EP}, \mathcal{EE}\}$ is a topology graph and $\mathcal{EC} = \{(P_1, P_2, P_3) \,|\, P_i \in \mathcal{EP}, (P_1, P_2), (P_2, P_3), (P_3, P_1) \in \mathcal{EE}\}$ is a set triangular cells in $\mathbf{B}_2$. We further assume that the cells in $\mathcal{EC}$ are disjoint except on their edges and provide a cover for $\mathbf{B}_2$.

We can obtain an extended topology graph of a curve from a topology graph by adding more auxiliary points and edges.

Algorithm 3.8 **ETopCur**$(g(x, y), \mathbf{B}_2, \rho)$ *The input is the same as Algorithm 3.5. The output is an extended topology graph of $\mathcal{C}_{\mathbf{B}_2}$. (See Fig. 3(c) for an illustration)*

1. Let $\mathscr{G} = \{\mathcal{P}, \mathcal{E}\} = \mathbf{TopCur}(g(x, y), \mathbf{B}_2, \rho)$.
2. Let $\mathcal{EP} = \mathcal{P}$. For $i = 0, \ldots, s$, add $(\alpha_i, \mathcal{Y}_1)$ and $P_{i,s_i} = (\alpha_i, \mathcal{Y}_2)$ to $\mathcal{EP}$ if they are not in it.
3. For points $P_{i,j}, j = 0, \ldots, s_i$, let $[a_i, b_i] \times [c_{i,j}, d_{i,j}]$ be the isolation box for $P_{i,j}$. Add $N_{i,j} = (\alpha_i, (d_{i,j} + c_{i,j+1})/2), j = 0, \ldots, s_i - 1$ to $\mathcal{EP}$. We still assume that $\mathcal{EP}$ is of form (2).
4. Let $\mathcal{EE} = \mathcal{E}$. For $j = 0, \ldots, s_0 - 1$, add the edges $(P_{0,j}, N_{0,j}), (N_{0,j}, P_{0,j+1})$ to $\mathcal{EE}$.
5. For each $0 \le i \le s - 1$, add the edges $(P_{i,0}, P_{i+1,0})$ and $(P_{i,s_i}, P_{i+1,s_{i+1}})$ to $\mathcal{EE}$. Then the edges in $\mathcal{EE}$ divide the rectangular region $B = [\alpha_i, \alpha_{i+1}] \times [\mathcal{Y}_1, \mathcal{Y}_2]$ into triangular and quadrilateral regions. We will subdivide these regions into triangular regions such that each point in $\mathcal{EP}$ is the vertex of at least one triangles.

6. Let $\mathcal{EC} = \emptyset$. For any two adjacent edges $e_1 = (P_{1,1}, P_{2,1}), e_2 = (P_{1,2}, P_{2,2})$ inside B, execute Steps 7 and 8.

7. If $P_{1,1} \neq P_{1,2}$, there exists one point N_1 added before in Step 3 between $P_{1,1}$ and $P_{1,2}$. Furthermore,

   - If $P_{2,1} \neq P_{2,2}$, there exists one point N_2 between $P_{2,1}$ and $P_{2,2}$. $P_{1,1}, P_{1,2}, P_{2,2}, P_{2,1}$ form a quadrilateral region. We can divide the quadrilateral region into four triangles. Add the edges $(P_{2,1}, N_2), (N_2, P_{2,2}), (P_{1,2}, N_2), (N_1, N_2), (N_1, P_{2,1})$ to $\mathcal{EE}$. Add the triangles $(P_{1,1}, N_1, P_{2,1})$, $(N_1, P_{2,1}, N_2), (N_1, P_{1,2}, N_2)$, $(P_{1,2}, N_2, P_{2,2})$ to $\mathcal{EC}$.
   - If $P_{2,1} = P_{2,2} = P$, $P_{1,1}, P_{1,2}, P$ form a triangular region. We can divide the triangular region into two triangles. Add the edges $(P_{1,1}, N_1), (N_1, P_{1,2}), (N_1, P)$ to $\mathcal{EE}$. Add the triangles $(P_{1,1}, N_1, P), (N_1, P_{1,2}, P)$ to $\mathcal{EC}$.

8. If $P_{1,1} = P_{1,2} = P$, then there must exist a point N_2 added before in Step 3 between $P_{2,1}$ and $P_{2,2}$. $P, P_{2,2}, P_{2,1}$ form a triangular region. We can divide the quadrilateral region into two triangles. Add the edges $(P_{2,1}, N_2), (N_2, P_{2,2}), (P, N_2)$ to $\mathcal{EE}$. Add the triangles $(P, P_{2,1}, N_2), (P, P_{2,2}, N_2)$ to $\mathcal{EC}$.

9. Output $\mathcal{EG} = \{\mathcal{EP}, \mathcal{EE}, \mathcal{EC}\}$.

Remark. The purpose to add points $N_{i,j}$ in Step 3 is to make sure that topology representation for surfaces possible. These points has similar function as the the auxiliary points added in Step 6 of Algorithm 3.5. Hence, they are also called **auxiliary points**. Figure 3 is an extend topology graph of the curve $G(x, y) = x \cdot y \cdot (16x^2 + 16y^2 - 49) = 0$.

Let $\mathcal{EG} = \{\mathcal{EP}, \mathcal{EE}, \mathcal{EC}\}$ be an extend topology graph of $\mathcal{C}_{\mathbf{B}_2}$ and $e = (P_1, P_2) \in \mathcal{EE}$. If e corresponds to a curve segment of $\mathcal{C}_{\mathbf{B}_2}$, we use $C(e)$ to represent the corresponding curve segment; otherwise, we use $C(e)$ to represent the line segment P_1P_2. Let $I(e) = C(e) \setminus \{P_1, P_2\}$. For a $c \in \mathcal{EC}$, we use $R(c)$ and $I(c)$ to denote the cell and interior of the cell represented by c respectively.

3.2 Compute ϵ-meshing for plane curve

The meshing given in Section 3.1 has no guarantee of precision. In this section, we will show how to compute a meshing for a curve to any given precision.

Let $\mathcal{G} = (\mathcal{P}, \mathcal{E})$ be a topology graph for a curve $\mathcal{C}$ inside a box $\mathbf{B}_2$ defined in (3). Assume that $\mathcal{P}$ is of form (2). Consider the two disjoint regions $\mathbf{S}_2$ and $\mathbf{N}_2$ of $\mathbf{B}_2$:

$$\begin{aligned} \mathbf{S}_2 &= \cup_i \mathbf{S}_2^i, \mathbf{N}_2 = \cup_j \mathbf{N}_2^j, \text{ where} \\ \mathbf{S}_2^i &= (a_i, b_i) \times [\mathcal{Y}_1, \mathcal{Y}_2], i = 0, \ldots, s \\ \mathbf{N}_2^j &= [b_j, a_{j+1}] \times [\mathcal{Y}_1, \mathcal{Y}_2], j = 0, \ldots, s-1. \end{aligned} \tag{6}$$

Then, $\mathcal{C}_{\mathbf{B}_2} \subset \mathbf{S}_2 \cup \mathbf{N}_2$ and is smooth in $\mathbf{N}_2$.

The idea of our algorithm is to determine the topology of the curve in the region $\mathbf{S}_2$ with Algorithm 3.5, to determine the topology of the curve in the region $\mathbf{N}_2$ with a modified marching cube method of

Pantinga-Vegter [24], and to compute the adjacency information on the border lines $x = a_i, x = b_i$. We could use the marching cube method in $\mathbf{N}_2$ because $\mathcal{C}$ has no singular point in it.

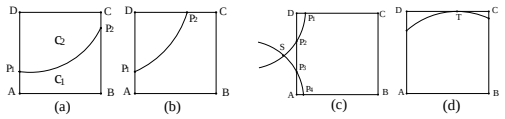

Figure 4: Nice boxes: (a), (b). Boxes in (c), (d) not nice.

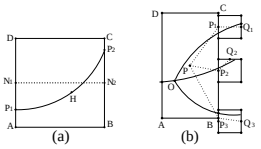

Figure 5: Meshing curve segments

In order for the above idea to work, we need to modify the Pantinga-Vegter method such that each output box contains only one curve segment of $\mathcal{C}$, as shown in Fig. 4(a) and (b). Such boxes are called **nice boxes**.

The original Pantinga-Vegter method could output a box containing two curve segments and this will cause problems when the box is near a singular point, as shown in Fig. 4(c). A point is called a y**-extremal point** of curve $\mathcal{C}$ if $\mathcal{C}$ achieves a local extremum value at this point in the y-direction. Pantinga-Vegter's method could output a box shown in Fig. 4(d).

To make the process precise, we introduce the following definition. An ϵ**-meshing graph** of a curve $\mathcal{C}$ is a triplet $\mathcal{M} = \{\mathcal{P}, \mathcal{E}, \mathcal{B}\}$ where $(\mathcal{P}, \mathcal{E})$ is a graph whose vertices are with rational numbers as coordinates and whose **edges are the meshes** for $\mathcal{C}$; $\mathcal{B}$ is a set of nice boxes and segregating boxes of singular points of $\mathcal{C}$ such that for each $e \in \mathcal{E}$, there exists a $\mathbf{B}_e \in \mathcal{B}$ with the property: $|\mathbf{B}_e| < \epsilon$ and $\mathcal{C} \cap \mathbf{B}_e$ is a connected curve segment of $\mathcal{C}$ (See Fig. 5). In Fig. 5(a), $\mathcal{P} = \{N_1, N_2\}, e = (N_1, N_2), \mathbf{B}_e = ABCD$ forms a meshing graph for curve segment $C(e) = P_1HP_2$.

It is easy to show that an ϵ-meshing graph for a curve $\mathcal{C}$ provides an ϵ**-meshing** for $\mathcal{C}$ according to the definition given in Section 2.

Algorithm 3.9 MPV2$(g(x, y), \mathbf{B}_2, \epsilon)$. *Input:* $\mathcal{C} : g(x, y) = 0$ *is a curve with no* x*-critical points and no* y*-extremal points in box* $\mathbf{B}_2$. *Output an* ϵ*-meshing graph* $\mathscr{G} = \{\mathcal{P}, \mathcal{E}, \mathcal{B}\}$ *for* $\mathcal{C}_{\mathbf{B}_2}$.

We need only add some extra criterions for the boxes: (1) For each edge (A, B) of $\mathbf{B}$, if $0 \in$ $g((A, B))$ and $g(A)g(B) > 0$, we continue to subdivide $\mathbf{B}$. (2) For each box $\mathbf{B}$, if $|\mathbf{B}| > \epsilon$, we continue to subdivide $\mathbf{B}$.

Since $\mathcal{C}$ has no x-critical points and no y-extremal points in box $\mathbf{B}_2$, a box like the one in Fig. 4(d) does not exist and the algorithm will terminate.

Now, we can give the meshing algorithm for curves.

Algorithm 3.10 ATopCur$(g(x, y), \mathbf{B}_2, \epsilon)$. *The input is the same as that of Algorithm 3.5. Output an* ϵ*-meshing graph for* $g(x, y) = 0$.

1. Execute the first four steps of Algorithm 3.5 with input $(g(x, y), \mathbf{B}_2, \epsilon)$. We need to modify Algorithm 3.5 as follows: Let $g_u(x, y) = g_v(x, y)/U(y)$ where $U(y)$ is the gcd of the coefficients of $g_v(x, y)$ as a univariate polynomial in x; and use $H(x) \cdot \text{Res}(g_u, \frac{\partial g_u}{\partial x}, y)$ as the new $H(x)$ in (4).

   We need $V(x), g_v(x, y)$, and $\mathscr{G}_1 = \{\mathcal{P}_1, \mathcal{B}_1\}$ from Algorithm 3.5, where $\mathcal{B}_1$ is the segregating boxes for the points in $\mathcal{P}_1$.

2. Compute $\mathscr{G}_2 = \{\mathcal{P}_2, \mathcal{E}_2, \mathcal{B}_2\}$=**MPV2**($g(x,y), \mathbf{N}_2, \epsilon$). The modification in Step 1 makes sure that $\mathcal{C}$ has no x-critical points and y-extremal points in $\mathbf{N}_2$.

3. Compute the connection between the boxes computed by Step S1 and Step S2. (Fig. 5(b) shows how to mesh $\mathcal{C}$ near a singular point O with $\mathbf{B} = ABCD$ as its segregating box. Fig. 6(b) provides a global picture for meshing a curve.)For $i = 0, \ldots, s$, consider the adjacency information on the border lines $x = a_i, b_i$. We only consider $x = b_i$. For each $P \in \mathcal{P}$ and its segregating box $\mathbf{B} = [a,b] \times [c,d] \in \mathcal{B}$, do the following

   (a) Let $\mathbf{E}_k = [b, c_k] \times [e_k, f_k] \in \mathcal{B}_2$ be the boxes satisfying $\mathbf{B} \cap \mathbf{E}_k \neq \emptyset$ and $g(b, \hat{e}_k) g(b, \hat{f}_k) < 0$, where $\hat{e}_k = \min\{e_k, c\}$ and $\hat{f}_k = \max\{f_k, d\}$. As a consequence, $\mathcal{C}$ passes through these $\mathbf{E}_k$ through the interval $[b_i, b_i] \times [e_k, f_k]$.

   (b) Let $Q = ((a+b)/2, (c+d)/2)$, $m_k = (\hat{e}_k + \hat{f}_k)/2$.

   (c) Add the edge $e = (Q, (b, m_k))$ to $\mathcal{M}$. Add $\mathbf{B}_e = \mathbf{B}$ to $\mathcal{M}$.

4. Add the meshes for the straight lines defined by $V(x) = 0$.

5. Output the meshing graph $\mathcal{M}$.

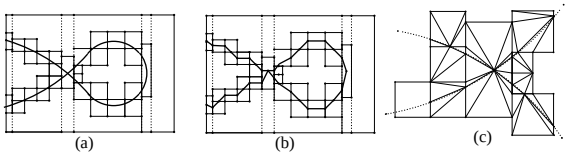

Figure 6: ϵ-meshing for a curve

Theorem 3.11 *Algorithm 3.10 terminates and computes an ϵ-ambient meshing for $\mathcal{C}_{\mathbf{B_2}}$.*

Proof. By Lemma 3.6, $\mathcal{M} \cap \mathbf{S}_2$ is the isotopic meshing of $\mathcal{C} \cap \mathbf{S}_2$. Marching cube method compute the isotopic meshing of $\mathcal{C} \cap \mathbf{N}_2$. Hence, $\mathcal{M}$ is a isotopic meshing for $\mathcal{C}_{\mathbf{B}_2}$. Furthermore, for each $e \in \mathcal{E} \cap \mathbf{N}_2$, $C(e) \subseteq \mathbf{B}_e, |\mathbf{B}_e| < \epsilon$ and each part of $\mathcal{C}_{\mathbf{B}_2}$ around the singular point P is contained in the segregating boxes $\mathbf{B}_P, |\mathbf{B}_P| < \epsilon$ of P. Therefore, for any point $P \in \mathcal{M}$, $F(P,1)$ and P are in the same box $\mathbf{B}_e$ with $|\mathbf{B}_e| < \epsilon$, so $\| F(P,1) - P \| < \epsilon$. This gives a proof of Theorem 3.11. ∎

In order to compute the ϵ-meshing for surfaces, we need to add more information to the ϵ-meshing graph. Let $\mathcal{M} = \{\mathcal{P}, \mathcal{E}, \mathcal{B}\}$ be a meshing graph for a curve $\mathcal{C}$. Then an **extended meshing graph** $\mathcal{EM} = \{\mathcal{EP}, \mathcal{EE}, \mathcal{EC}\}$ for $\mathcal{C}$ in $\mathcal{B}$ can be defined similarly as the extended topology graph. The difference is that $\mathcal{EC}$ provides a triangular decomposition for $\mathcal{B}$.

The following algorithm computes an extended meshing graph.

Algorithm 3.12 **METopCur**($g(x,y), \mathscr{G}_1, \mathscr{G}_2$). *$\mathscr{G}_1 = \{\mathcal{P}_1, \mathcal{B}_1\}$ where $\mathcal{P}_1$ is the set of points on $\mathcal{C} : g(x,y) = 0$ of form (2) and $\mathcal{B}_1$ is the set of their segregating boxes. $\mathscr{G}_2 = \{\mathcal{P}_2, \mathcal{E}_2, \mathcal{B}_2\}$ is an ϵ-meshing of the curve $\mathcal{C}$ in $\mathbf{N}_2$ defined in (6). Output an extended meshing graph for $\mathcal{C}$ in $\mathcal{B}_1 \cup \mathcal{B}_2$. (Fig. 6(c) is the extended meshing graph for the box in the center of Fig. 6(a) and its surrounding boxes.)*

S1 Let $\mathcal{EP} = \emptyset, \mathcal{EE} = \emptyset, \mathcal{EC} = \emptyset$.

S2 For any $\mathbf{B} = [a, b] \times [c, d] \in \mathcal{B}_1$, it is the segregating box of one point $P \in \mathcal{P}_1$. Compute the extended topology in $\mathcal{B}_1$.

1. Compute

$$\begin{aligned} \{Q_1, \ldots, Q_l\} &= \textbf{RootIsol}(\{x - a, g(a, y)\}, [c, d], 1) \\ \{T_1, \ldots, T_r\} &= \textbf{RootIsol}(\{x - b, g(b, y)\}, [c, d], 1). \end{aligned}$$

   Denote $A_i, i = 1, \ldots, 4$ to be the four vertices of $\mathbf{B}$. Denote $B_j, j = 1, \ldots, p$ to be the points on the edge of $\mathbf{B}$ which are the vertices of boxes adjacent to $\mathbf{B}$.(Note that if $l > 1$($r > 1$), there exists some point B_k between Q_i and Q_{i+1}(T_i and T_{i+1})).

2. $\mathcal{TP} = \emptyset$. Add points Q_i, T_j, A_k, B_p into $\mathcal{TP}$.

3. Denote the sets of points $L = \{L_1, \ldots, s\}$ and $R = \{R_1, \ldots, R_t\}$ where L is the points in $\mathcal{TP}$ which are on the left edge of $\mathbf{B}$ sorted from bottom to up and R the points in $\mathcal{TP}$ which are on on the right edge of $\mathbf{B}$ sorted from bottom to up.

4. Add point P and all points in $\mathcal{TP}$ into $\mathcal{EP}$. Add edges $(P, L_i), (P, R_j)$ into $\mathcal{EE}$. Add triangular cell $(P, L_i, L_{i+1}), (P, R_i, R_{i+1})$ and $(P, L_1, R_1), (P, L_s, R_t)$ and (L_i, L_{i+1}), (R_j, R_{j+1}) to $\mathcal{EC}$.

S3 For any boxes $\mathbf{B} \in \mathcal{B}_2$. Compute the extended topology in $\mathbf{B}$. For any line segment $e \in \mathcal{E}_2$ with $\mathbf{B} = [a, b] \times [c, d] \in \mathcal{B}_2$ containing it. Doing the following operations(There are six conditions that e divide $\mathbf{B}$ into two parts, see fig 4. We can distinguish them according to $\mathcal{P}_2$ and $\mathcal{B}_2$. Here we consider the condition (a), the other conditions are dealt with in the similar way).

1. Compute $Q =$ **RootIsol**$(\{x-a, g(x, y)\}, \mathbf{B}, \epsilon/4)$ and $T =$ **RootIsol**$(\{x-b, g(x, y)\}, \mathbf{B}$, $\epsilon/4)$. Obviously, Q and T both contain only one point. We still call them Q and T. Denote $A_i, i = 1, \ldots, 4$ to be the four vertices of $\mathbf{B}$. Denote $B_j, j = 1, \ldots, p$ to be points on one edge of $\mathbf{B}$ which are the vertices of boxes adjacent to $\mathbf{B}$.

2. $\mathcal{TP} = \emptyset$. Add points Q, T, A_i, B_j into $\mathcal{TP}$.

3. Add all points in $\mathcal{TP}$ into $\mathcal{EP}$. Similar to the forth step in Step **S2**, we can decompose $\mathbf{B}$ into triangular cells and insert these cells into $\mathcal{EC}$, and insert corresponding edges into $\mathcal{EE}$ such that each point in $\mathcal{TP}$ connects to at least another point in this set.

S4 Output $\mathcal{EM} = \{\mathcal{EP}, \mathcal{EE}, \mathcal{EC}\}$.

4 Topology of surface

In this section, an algorithm will be given to compute a polyhedron with triangular faces, which is isotopic to a given surface.

4.1 Outline of the algorithm

We use a polyhedron with triangular faces to represent the topology of a surface. A **topology polyhedron** is a triplet $\mathscr{P} = \{\mathcal{SP}, \mathcal{SE}, \mathcal{SF}\}$ where $\mathcal{SP}$, $\mathcal{SE}$, and $\mathcal{SF}$ are defined below.

- $\mathcal{SP}$ is a set of 3D points determined by a triangular system Σ_i and an isolation boxes $\mathbf{B}_{i,j,k}$:

$$\begin{aligned} \mathcal{SP} &= \{P_{i,j,k}, 0 \le i \le s, 0 \le j \le s_i, 0 \le k \le t_{i,j}\} \\ \Sigma_i &= \{h_i(x), g_i(x,y), f_i(x,y,z)\} \\ \mathbf{B}_{i,j,k} &= [a_i, b_i] \times [c_{i,j}, d_{i,j}] \times [e_{i,j,k}, f_{i,j,k}] \in \square\mathbb{Q}^3. \end{aligned} \tag{7}$$

  where $P_{i,j,k} = (\alpha_i, \beta_{i,j}, \gamma_{i,j,k})$ satisfy $\alpha_0 < \cdots < \alpha_s$, $\beta_{i,0} < \cdots < \beta_{i,s_i}$, and $\gamma_{i,j,0} < \cdots < \gamma_{i,j,t_{i,j}}$. Point $P_{i,j,k}$ is said to be **lifted** from the plane point $P_{i,j} = (\alpha_i, \beta_{i,j})$. $P_{i,j}$ is said to be the **projection** of $P_{i,j,k}$.

- $\mathcal{SE} = \{(P_1, P_2) | P_1, P_2 \in \mathcal{SP}$, such that either $P_1 = P_{i,u,v}, P_2 = P_{i+1,p,q}$ or $P_1 = P_{i,u,v},, P_2 = P_{i,u+1,t}\}$. We further assume that any two edges do not intersect except at the end points.

- $\mathcal{SF} = \{(P_1, P_2, P_3) | P_1, P_2, P_3 \in \mathcal{SP}\}$ such that its three edges are in $\mathcal{SE}$. We further assume that any two faces do not intersect except at the edges.

Let $\mathcal{S} : f(x,y,z) = 0$ be an algebraic surface, where $f(x,y,z) \in \mathbb{Q}[x,y,z]$ is square free. A point P_0 is a **critical point** of $\mathcal{S}$ if $f(P_0) = f_z(P_0) = 0$. Write f as a univariate polynomial in z: $f(x,y,z) = f_d(x,y)z^d + \cdots + f_0(x,y)$. $f_d(x,y)$ is called the **leading coefficient** of $f(x,y,z)$. We further assume that

$$f_d(x,y) = \cdots = f_0(x,y) = 0 \text{ have no common zeros.} \tag{8}$$

Geometrically, this means that $\mathcal{S}$ does not contain a line parallel to the z-axis. We will consider surfaces that do not satisfy this condition in Section 4.7.

Similar to the case of algebraic curves, we will consider the topology of $\mathcal{S}$ in a bounding box

$$\mathbf{B}_3 = [\mathcal{X}_1, \mathcal{X}_2] \times [\mathcal{Y}_1, \mathcal{Y}_2] \times [\mathcal{Z}_1, \mathcal{Z}_2] \in \square\mathbb{Q}^3. \tag{9}$$

Let

$$D(x,y) = \text{Res}(f, \frac{\partial f}{\partial z}, z) \tag{10}$$

$$G(x,y) = \text{sqrfree}(D(x,y) f(x,y,\mathcal{Z}_1) f(x,y,\mathcal{Z}_2)) \tag{11}$$

where sqrfree$(P(x,y))$ is the square free part of $P(x,y)$. The plane curve $G(x,y) = 0$ is called the **projection curve** of $\mathcal{S}$.

To determine the topology of a surface is to find a topology polyhedron with the same topology as the surface. We first give an outline of the algorithm, which consists of four main steps.

S1 Compute an extended topology graph $\mathcal{EG} = \{\mathcal{EP}, \mathcal{EE}, \mathcal{EC}\}$ of the projection curve of $\mathcal{S}$ in $\mathbf{B}_2$ defined in (3).

S2 Determine $\mathcal{SP}$. For any $P \in \mathcal{EP}$, determine the intersection points of $\mathcal{S}$ and the line segment $P \times [\mathcal{Z}_1, \mathcal{Z}_2]$.

S3 Determine $\mathcal{SE}$. For each edge $e \in \mathcal{EE}$, compute the intersection of $\mathcal{S}$ and the cylindrical surface patch $I(e) \times [\mathcal{Z}_1, \mathcal{Z}_2]$, which are delineable curve segments of $\mathcal{S}$ whose end points are in $\mathcal{SP}$. We will use line segments in $\mathcal{SE}$ to represent these curve segments. See Fig. 7.

S4 Determine $\mathcal{SF}$. For each $c \in \mathcal{EC}$, compute the intersection of $\mathcal{S}$ and the prism $I(c) \times [\mathcal{Z}_1, \mathcal{Z}_2]$, which are delineable surface patches of $\mathcal{S}$ whose edges are in $\mathcal{SE}$. We will use triangular faces in $\mathcal{SF}$ to represent these surface patches. See Fig. 8.

4.2 Theoretical preparations for the algorithm

In the outline of the algorithm given in the preceding section, Step **S**1 has been solved in Section 3.1. Step **S**2 can be solved with Algorithm **RootIsol**. We will explain Steps **S**3 and **S**4 below.

Roughly speaking, Step **S3** is to determine the topology of the spatial curve defined by $f(x, y, z) = G(x, y) = 0$. The following result, which is a consequence of Theorem 2.3, allows us to determine the singularities of this curve easily.

Lemma 4.1 *Use the notations introduced above. For each edge $e = (P_1, P_2) \in \mathcal{EE}$, $f(x, y, z) = 0$ is delineable over $I(e) = C(e) \setminus \{P_1, P_2\}$.*

Proof. Let $\mathcal{C} : G(x, y) = 0$ be the projection curve of $\mathcal{S}$ and $D(x, y)$ the discriminant of f w.r.t. z. Since $I(e)$ is a continous curve segment of $\mathcal{C}$, G is order-invariant on $I(e)$. From (10), $D(x, y)$ is order-invariant on $I(e)$. Since condition (8) holds, f does not vanish identically on any point of xy-plane. So, f does not vanish identically on $I(e)$. Now, we will prove that f is degree-invariant on $I(e)$. It is clear that all the singular points of $\mathcal{C}$ are in $\mathcal{EP}$. Then $f_d(x, y)$ is either identically zero on $I(e)$ or does not vanish on any point on $I(e)$. So we can conclude that $f_d(x, y)$ is sign-invariant on $I(e)$. By Theorem 2.4, f is degree-invariant on $I(e)$. By Theorem 2.3, f is delineable on $I(e)$. ∎

As a corollary, we have

Corollary 4.2 *For $e = (P_1, P_2) \in \mathcal{EE}$, the intersection of $\mathcal{S}$ and $I(e) \times [\mathcal{Z}_1, \mathcal{Z}_2]$ consists of disjoint curve segments of $\mathcal{S}$ whose end points are in $\mathcal{SP}$.*

These curve segments together with their endpoints are called the **spatial cylindrical curve segments (SCCS)** of $\mathcal{S}$ **lifted from** e.

To determine the edges of the topology polyhedron, an SCCS with end points P_1 and P_2 is represented by the line segment $e = (P_1, P_2)$. $\mathcal{SE}$ is the set of these line segments. For an edge $E \in \mathcal{SE}$, we use $S(E)$ to denote the corresponding SCCS of $\mathcal{S}$.

Let $P_{i,j,k} \in \mathcal{SP}$ and $e = (P_{i,j}, P_{u,v}) \in \mathcal{EE}$. We use $\#(P_{i,j,k}, e)$ to represent the **number of SCCSes** which have $P_{i,j,k}$ as an end point and are lifted from $C(e)$. We use $\#(e)$ to denote the **number of SCCSes lifted from** e. Define $\#(P_{u,v,w}, e)$ similarly. As a direct consequence of Lemma 4.1, the following equation

$$\#(e) = \sum_k \#(P_{i,j,k}, e) = \sum_w \#(P_{u,v,w}, e) \tag{12}$$

holds for each $e = (P_{i,j}, P_{u,v}) \in \mathcal{EE}$. (See Fig. 7)

In Step **S4**, we find the surface patches lifted from a triangular cell $c \in \mathcal{EC}$ by identifying their boundaries which are SCCSes of $\mathcal{S}$. As a consequence of Theorem 2.3, we have

Lemma 4.3 *Let $c \in \mathcal{EC}$. Then $f(x, y, z) = 0$ is delineable over $S = I(c)$.*

Proof. For any $P = (\alpha, \beta) \in S$, f is degree-invariant and does not vanish. The discriminant $D(x, y)$ of f does not vanish on P. So D is order-invariant over S. By Theorem 2.3, the lemma holds. ∎

Lemma 4.4 *$\mathcal{S} \cap (I(c) \times [\mathcal{Z}_1, \mathcal{Z}_2])$ consists of disjoint surface patches whose edges are SCCSes and whose vertices are points in $\mathcal{SP}$. These surface patches with their edges and vertices are called* **triangular surface patches** *(TSP) lifted from c.*

Proof. By Lemma 4.3, the intersection of $\mathcal{S}$ and $I(c) \times [\mathcal{Z}_1, \mathcal{Z}_2]$ consists of disjoint surface patches. The edges of a surface patch s are the intersection of $\mathcal{S}$ and $I(e_i) \times [\mathcal{Z}_1, \mathcal{Z}_2], i = 1, 2, 3$, where e_i are the three sides of c. As a consequence, the edges of these surface patches are SCCSes. If $c = (P_1, P_2, P_3)$, the vertices of an intersection surface patch are the intersection points of $\mathcal{S}$ and $P_i \times [\mathcal{Z}_1, \mathcal{Z}_2], i = 1, 2, 3$. As a consequence, the vertices Q_1, Q_2, Q_3 of a triangular surface patch are points in $\mathcal{SP}$ lifted from P_1, P_2, P_3 respectively. ∎

It is clear that the TSPs are the intersection of $C(e) \times [\mathcal{Z}_1, \mathcal{Z}_2]$ and $\mathcal{S}$.

For a cell $c = (P_{i,j}, P_{u,v}, P_{s,t}) \in \mathcal{EC}$ and an edge $E = (P_{i,j,k}, P_{u,v,w}) \in \mathcal{SE}$ lifted from the side $e = (P_{i,j}, P_{u,v})$ of c, we use $\#(c)$ to denote the **branch number** of TSPs lifted from $R(c)$ and $\#(E, c)$ to denote the **number of TSPs** which pass through $S(E)$ and lifted from $R(c)$. Notations $\#((P_{u,v,w}, P_{s,t,l}), c)$ and $\#((P_{i,j,k}, P_{s,t,l}), c)$ can be similarly defined. As a consequence of Lemma 4.4, for $c = (P_{i,j}, P_{u,v}, P_{s,t}) \in \mathcal{EC}$, we have

$$\#(c) = \sum_{E_1} \#(E_1, c) = \sum_{E_2} \#(E_2, c) = \sum_{E_3} \#(E_3, c), \tag{13}$$

where $E_1 = (P_{i,j,k_1}, P_{u,v,k_2})$, $E_2 = (P_{u,v,k_2}, P_{s,t,k_3})$, $E_3 = (P_{i,j,k_1}, P_{s,t,k_3})$ for all possible k_1, k_2, k_3. (See Fig. 8)

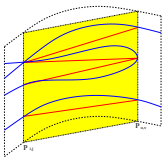

Figure 7: Mesh SCCSes

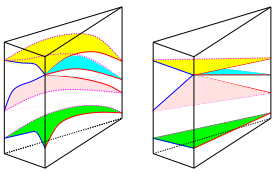

Figure 8: Mesh TSPs

4.3 The algorithm

Following the analysis in the preceding section, we now give the algorithm to construct a topology polyhedron for a given surface.

Algorithm 4.5 **TopSur**$(f(x,y,z), \mathbf{B}_3)$. *$\mathcal{S} : f(x,y,z) = 0$ is the surface satisfying condition (8) and f is square free. $\mathbf{B}_3$ is defined in (9). Output* **an isotopic topology polyhedron** *$\mathscr{P}$ for $\mathcal{S}_{\mathbf{B}_3}$.*

1. **Compute the projection curve** $\mathcal{C} : G(x,y) = 0$ as in (11).
2. **Compute the extended topology graph**: $\mathcal{EG} = \{\mathcal{EP}, \mathcal{EE}, \mathcal{EC}\}$ of $\mathcal{C}_{\mathbf{B}_2}$ with Algorithm 3.8, where $\mathbf{B}_2$ is defined in (3).
3. **Compute $\mathcal{SP}$.** For any $P_{i,j} \in \mathcal{EP}$, use Algorithm 4.7 with input $(f, \mathbf{B}_3, P_{i,j}, 1)$ to compute $P_{i,j,k}$.
4. **Compute $\mathcal{SE}$.** Let $\mathcal{SE} = \emptyset$.
   (a) For each $P_{s,t} \in \mathcal{EP}$ and $e \in \mathcal{EE}$ with $P_{s,t}$ as an endpoint, use Algorithm 4.9 to compute $\#(P_{s,t,k}, e)$.
   (b) For any $e = (P_{i,j}, P_{u,v}) \in \mathcal{EE}$, let $L_1 = (P_{i,j,0}, \ldots, P_{i,j,s_{i,j}})$ such that point $P_{i,j,k}$ repeats $\#(P_{i,j,k}, e)$ times. Similarly, define $L_2 = (P_{u,v,0}, \ldots, P_{u,v,s_{u,v}})$.
   (c) By (12), $|L_1| = |L_2| = m$. Let $L_1 = (P_1, \ldots, P_m)$ and $L_2 = (Q_1, \ldots, Q_m)$. Add (P_i, Q_i) to $\mathcal{SE}$. See Fig. 7 for an illustration.
5. **Compute $\mathcal{SF}$.** Let $\mathcal{SF} = \emptyset$.
   (a) For each cell $c \in \mathcal{EC}$ and $E \in \mathcal{SE}$ lifted from a side of c, compute $\#(E, c)$ with Algorithm 4.11.
   (b) Let $e_1, e_2, e_3 \in \mathcal{EE}$ be the three sides of c. Let S_i be the sequence of edges in $\mathcal{SE}$ lifted from e_i ordered bottom up and an E is repeated $\#(E, c)$ times in the sequence.
   (c) By (13), $|S_1| = |S_2| = |S_3| = t$. Let $S_i = \{E_{i,k}, k = 1, \ldots, t\}$. Then the three line segments $E_{1,k}, E_{2,k}, E_{3,k}$ should form a triangle $f = (P_{1,k}, P_{2,k}, P_{3,k})$. Add f to $\mathcal{SF}$. See Fig. 8 for an illustration.
6. Output $\mathscr{P} = \{\mathcal{SP}, \mathcal{SE}, \mathcal{SF}\}$. The isotopic map can be computed as usual [25].

Theorem 4.6 *Algorithm 4.5 computes an isotopic meshing for $\mathcal{S}_{\mathbf{B_3}}$.*

Proof. First, we prove the algorithm compute the correct topology of given surface. Note that with the auxiliary points added in Step 6(a) of Algorithm 3.5 and Step 3 of Algorithm 3.8, the edges in Step 4(c) and the faces in Step 5(d) are mutually different. Thus, we have a well-defined polyhedron.

The extended topology graph EG divides the rectangle $\mathbf{B}_2$ into triangular cells. We need only to show that for each edge $e \in \mathcal{EE}$ and each cell $c \in \mathcal{EC}$, $\mathscr{P}$ and $\mathcal{S}$ have the same topology on $C(e) \times [\mathcal{Z}_1, \mathcal{Z}_2]$ and $C(c) \times [\mathcal{Z}_1, \mathcal{Z}_2]$ respectively.

For $e \in \mathcal{EE}$, from Step 4 the SCCSes of $\mathcal{S}$ on the cylindrical surface $S_1 = C(e) \times [\mathcal{Z}_1, \mathcal{Z}_2]$ do not intersect except at the end points. By Corollaries 4.2 and (12), the edges of $\mathcal{SE}$ are the line segments with the same end points as those SCCSes. Then, the plane graph $\mathscr{P}$ on $e \times [\mathcal{Z}_1, \mathcal{Z}_2]$ and $\mathcal{S}$ on S_1 have the same topology. See Figure 7 for an illustration. The spatial curve segments are presented by line segments. With similar arguments, we could show that the part of $\mathscr{P}$ on $c \times [\mathcal{Z}_1, \mathcal{Z}_2]$ and $\mathcal{S}$ on

$C(c) \times [\mathcal{Z}_1, \mathcal{Z}_2]$ have the same topology. See Figure 8 for an illustration. This proves the topology correctness of the algorithm.

Then we prove the topology polyhedron is a isotopic meshing of the given surface.

The extended topology graph $\mathcal{EG} = \{\mathcal{EP}, \mathcal{EE}, \mathcal{EC}\}$ for the curve $\mathcal{C}_{\mathbf{B}_2}$ decompose $\mathbf{B}_2$ into triangular cells. According to Theorem 3.6, $\mathcal{EG}$ and $\mathcal{C}$ are isotopic and we can construct a homeomorphism F from $\mathbb{R}^2$ to itself that deforms $\mathcal{EE}$ to $\mathcal{C}$ continuously:

$$F : \mathbb{R}^2 \times [0, 1] \to \mathbb{R}^2.$$

Let $\mathscr{P} = (\mathcal{SP}, \mathcal{SE}, \mathcal{SF})$ be a topology polyhedron for a surface $\mathcal{S}_{\mathbf{B_3}}$ which decomposes $\mathbf{B}_3$ into cylindrical regions in a similar way as described in the proof of Theorem 3.6. Extend F to $\mathbb{R}^3 \times [0, 1]$:

$$T_1 = (F(x, y), z) : \mathbb{R}^3 \times [0, 1] \to \mathbb{R}^3.$$

The inverse transformation T_1^{-1} of T_1 deforms all SCCSs of $\mathcal{S}$ into planes $\{C(e) \times \mathbb{R}, e \in \mathcal{EE}\}$ which are perpendicular to the xy-pane. Denote $\mathcal{S}_1$ to be the surface $T_1^{-1}(\mathcal{S})$. We need only to prove that $\mathcal{S}_1$ and $\mathscr{P}$ are isotopic.

We can construct a homeomorphism T_2 from $\mathbb{R}^3$ to itself similar to that give in the proof of Theorem 3.6 to deform the z direction such that $T_2(\mathcal{SF}, 0) = \mathcal{SF}$ and $T_2(\mathcal{SF}, 1) = \mathcal{S}_1$.

The transformation $T = T_1 \circ T_2$ is a homeomorphism from $\mathbb{R}^3$ to itself which deforms $\mathcal{SF}$ to $\mathcal{S}$ continuously. ∎

We implemented Algorithm 4.5 in Maple. Two groups of experiments are done for the following five surfaces with singularities.

$\mathcal{S}_1 : f_1 = x^4 + y^4 + z^4 - x^2 - y^2 - z^2 - x^2y^2 - x^2z^2 - y^2z^2 + 1 = 0$, $\mathbf{B}_3 = [[-1.5, 1.5], [-1.25, 1.25], [-2, 2]]$.

$\mathcal{S}_2 : f_2 = -1 + (27/2)z^2y^2x^2 - (27/2)x^2y^2 - 6x^2z^2 - (27/2)y^2z^2 + 3x^2 + 3z^2 + (27/4)y^2 - 3x^4 - (243/16)y^4 - 3z^4 + x^6 + (729/64)y^6 + z^6 + (27/4)x^4y^2 + 3x^4z^2 + (243/16)x^2y^4 + 3x^2z^4 + (243/16)z^2y^4 + (27/4)z^4y^2 - x^2z^3 - (9/80)y^2z^3 = 0$, $\mathbf{B}_3 = [[-2, 2], [-2, 2], [-4, 4]]$.

$\mathcal{S}_3 : f_3 = -2y^4 + 2y^2z^2 + y^2 + z^4 - 2z^2 + x^6 + 3x^4y^2 - 3x^4 + 3x^2y^4 - 6x^2y^2 + 3x^2 + y^6 = 0$, $\mathbf{B}_3 = [[-2, 2], [-2, 2], [-2, 2]]$.

$\mathcal{S}_4 : f_4 = x^2y^2 + y^2z^2 + z^2x^2 - 7xyz/2 = 0$, $\mathbf{B}_3 = [[-2, 2], [-2, 2], [-2, 2]]$.

$\mathcal{S}_5 : f_5 = 16 - 2x^2z^2 - 8z^2 + 4x^3 - x^5 + (1/4)x^6 + x^4 + y^4 + y^2x^3 + z^4 + z^2x^3 - 2x^2y^2 + 2y^2z^2 - 8x^2 - 8y^2 = 0$, $\mathbf{B}_3 = [[-2, 2], [-3, 3], [-6, 6]]$.

The first experiment is to compute an isotopic polyhedron for the surfaces without considering precision. The timings are given in the second row of Table 1. Two of the polyhedrons are shown in Fig. 9. In the second experiment, we continue to subdivide the intervals between $[\mathcal{X}_1, \mathcal{X}_2]$ to compute a more accurate meshing. The results are given in Fig. 1. The timings are given in the third row of Table 1. #Mesh in the fourth row gives the number of meshes in these meshings. Considering that implementations in Maple are generally slow due to overhead costs, our algorithm is quite effective.

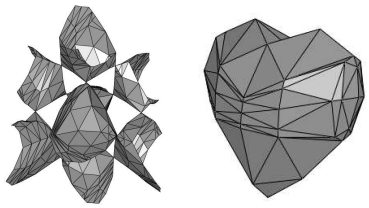

Figure 9: Topology polyhedrons for surfaces $\mathcal{S}_1$ and $\mathcal{S}_2$.

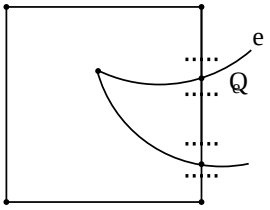

Figure 10: Isolation intervals

TYPE	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$	$\mathcal{S}_5$
Topology	0.544	0.816	0.760	0.684	1.280
Meshing	11.7	11.8	22.0	51.1	92.0
#Mesh	1472	1612	3032	3658	5456

Table 1: Timings on a PC with Linux OS, 3.00G Core 2Duo CPU, and 2G RAM.

4.4 Segregating box for a point on $\mathcal{S}$

Assume that $\mathcal{SP}$ is of form (7). Then $\mathbf{B}_{i,j}$ in (2) is an isolation box for $P_{i,j}$ and $\mathbf{B}_{i,j,k}$ is an isolation box for $P_{i,j,k}$. It is clear that

$$f(\alpha_i, \beta_{i,j}, e_{i,j,k}) f(\alpha_i, \beta_{i,j}, d_{i,j,k}) \neq 0 \tag{14}$$

The isolating box $\mathbf{B}_{i,j,k}$ of $P_{i,j,k}$ is called a **segregating box** if $f(x, y, z)$ does not intersect with the top and bottom faces of $\mathbf{B}_{i,j,k}$. Due to (14), when sufficiently subdividing $\mathbf{B}_{i,j}$, $\mathbf{B}_{i,j,k}$ will become a segregating box. This leads to the following algorithm.

Algorithm 4.7 **SegBoxP3**$(f(x, y, z), \mathbf{B}_3, P, \epsilon)$ *where* $\mathcal{S}$*:* $f(x, y, z) = 0$ *is the surface,* $\mathbf{B}_3$ *defined in (9),* P *a plane point defined by* $\Sigma_2 = \{h(x), g(x, y)\}$ *and an isolation box* $\mathbf{B}$*, and* $\epsilon > 0$*. Output the set of points* $\{P_i\}$ *on* $\mathcal{S}$ *lifted from* P*, segregating boxes for* P_i*, and a new segregating box* $\mathbf{B}$ *of* P*.*

1. Let $\{\mathbf{B}_1, \ldots, \mathbf{B}_s\} = \mathbf{RootIsol}(\Sigma_3, \mathbf{B} \times [\mathcal{Z}_1, \mathcal{Z}_2], \epsilon)$, where $\Sigma_3 = \{h(x), g(x, y), f(x, y, z)\}$.
2. Let $\mathbf{B}_i = \mathbf{B} \times [e_i, f_i]$ be the isolation box for P_i on $\mathcal{S}$.
3. Let $\eta = \epsilon$. While $0 \in \square f(\mathbf{B} \times [e_i, e_i])$ or $0 \in \square f(\mathbf{B} \times [f_i, f_i])$ for some $k \in \{1, \ldots, s\}$, repeat

   $\eta = \eta/2$ and $\mathbf{B} := \mathbf{RootIsol}(\Sigma_2, \mathbf{B}, \eta)$.
4. Output the points P_i defined by Σ_3 and $\mathbf{B}_i$, and the new $\mathbf{B}$.

In Step 3, if $f(\alpha_i, \beta_{i,j}, \mathcal{Y}_1) = 0$ or $f(\alpha_i, \beta_{i,j}, \mathcal{Y}_2) = 0$, then we need to use the minimal circle method introduced in [10] to find a segregating box in order for Lemma 4.8 to be true at this boundary point.

4.5 Compute number of SCCSes adjacent to a point

Let $P_{i,j,k}$ be a point lifted from point $P_{i,j}$ and $e = (P_{i,j}, P_{u,v}) \in \mathcal{EE}$. We will show how to compute $\#(P_{i,j,k}, e)$.

For any point P on the projection curve $\mathcal{C} : G(x, y) = 0$ and a segregating box $\mathbf{B} = [a, b] \times [c, d]$ of P, $\mathcal{C}$ intersects only with the vertical boundaries of $\mathbf{B}$.

For an edge $e \in \mathcal{E}$, consider the right boundaries of $\mathbf{B}$. We denote the intersection point of $C(e)$ with line $x = b$ as Q_e and

$$[b, b] \times [u_e, v_e] \tag{15}$$

is an isolation interval for Q_e on line $x = b$, which is called the **isolation interval** of $C(e)$. See Fig. 10.

Lemma 4.8 *Use the above notations. If $\mathbf{B}_{i,j,k}$ is a segregating box for $P_{i,j,k}$ and $\mathcal{S}$ is delineable over $I(e)$, then $\#(P_{i,j,k}, e)$ equals to the number of solutions of the triangular system $\Sigma_R = \{G(b_i, y), f(b_i, y, z)\}$ in the interval box $[u_e, v_e] \times [e_{i,j,k}, f_{i,j,k}]$. Geometrically, this is the number of intersection points of the line segment $\{x = b_i, y = \gamma_i, e_{i,j,k} \le z \le f_{i,j,k}\}$ and the surface $\mathcal{S}$ where (b_i, γ_i) is a point on $G(b_i, y) = 0$. See Fig. 11 for an illustration.*

Proof. From Algorithm 3.3, each SCCS passing through $P_{i,j,k}$ and projecting to $C(e)$ must pass through the rectangle $[b_i, b_i] \times [u_e, v_e] \times [\mathcal{Z}_1, \mathcal{Z}_2]$. Since $\mathbf{B}_{i,j,k}$ is a segregating box, these SCCSes must intersect with the the rectangle $\mathcal{R} = [b_i, b_i] \times [u_e, v_e] \times [e_{i,j,k}, f_{i,j,k}]$. Further, each SCCS can intersect with the rectangle only once since these SCCSes are delineable by Lemma 4.1. Note that the number of solutions of the triangular system Σ_R is the number of intersections of the SCCSes and the rectangle $\mathcal{R}$. ∎

Remark. Similarly, we can compute the number of the SCCSes on the left hand side of the point $P_{i,j,k}$ by computing the number of solutions for $\{G(a_i, y) = 0, f(a_i, y, z) = 0\}$. When $G(x, y) = 0$ contains vertical lines, we can compute the number of SCCSes passing through $P_{i,j,k}$ and projecting to these lines by solving $\{G(x, w), f(x, w, z)\}$ for $w = c_{i,j}$ and $w = d_{i,j}$ respectively.

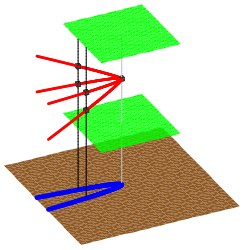

Figure 11: Compute $\#(P_{i,j,k}, e)$

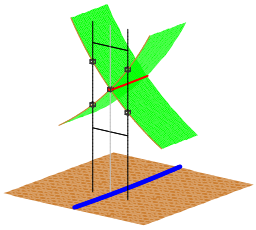

Figure 12: Compute $\#(S, c)$

We now give the following algorithm to compute the number of curve branches.

The following algorithm is based on Lemma 4.8.

Algorithm 4.9 **NumSCCS**$(f(x, y, z), P_{i,j,k}, e)$ *$\mathcal{S} : f(x, y, z) = 0$ is a surface delineable over $I(e)$, $P_{i,j,k} \in \mathcal{SP}$ is of form (7), and $e \in \mathcal{EE}$ is an edge with $P_{i,j}$ as an end point, where $P_{i,j}$ is the projection point of $P_{i,j,k}$. The output is $\#(P_{i,j,k}, e)$.*

1. If e is an x-vertical line segment above $P_{i,j}$ in the y-direction, then form the triangular system $\Sigma_{22} = \{g_i(x, d_{i,j}), f(x, d_{i,j}, z)\}$ and let $\mathcal{Q} = \mathbf{RootIsol}(\Sigma_{22}, [a_i, b_i] \times [e_{i,j,k}, f_{i,j,k}], 1)$. Output $\#(P_{i,j,k}, e) = |\mathcal{Q}|$.

2. If e is not an x-vertical line segment, we need to compute the isolation intervals defined in (15). We only consider the right branches. Let $\mathcal{R} = \mathbf{RootIsol}(g_i(b_i, y), [c_{i,j}, d_{i,j}], 1)$ where $r = R\#(P_{i,j}) = |\mathcal{R}|$. By Lemma 3.2, one of intervals in $\mathcal{R}$ is the isolation interval $[u_e, v_e]$ for e.

3. Let $\Sigma_{21} = \{g_i(b_i, y), f(b_i, y, z)\}$ be a triangular system in y and z and $\mathcal{Q} = \mathbf{RootIsol}(\Sigma_{21}, [u_e, v_e] \times [e_{i,j,k}, f_{i,j,k}], 1)$. Output $\#(P_{i,j,k}, e) = |\mathcal{Q}|$. See Fig. 11 for an illustration.

If there exist no SCCSes originating from a point, it is an *isolated singularity*.

4.6 Compute number of TSPs adjacent to an SCCS

We compute the number of TSPs originating from an $E \in \mathcal{SE}$. That is, for an $E = (P_{i,j,k}, P_{u,v,w}) \in \mathcal{SE}$ and a $c \in \mathcal{EC}$ with $e = (P_{i,j}, P_{u,v})$ as an edge, we will compute $\#(E, c)$.

Use the notations in Algorithm 4.9. Denote the SCCSs passing through point $P_{i,j,k}$ and projecting to $C(e)$ as $S(s_i), i = 1, \ldots, m$. Assume that $\mathcal{Q}$ (Step 3 of Algorithm 4.9) is the set of isolation boxes of m points $Q_1, \ldots, Q_s$ with Q_i on $S(s_i)$. Then in the plane $x = b_i$ (or $x = a_i$), the surface becomes a plane curve $f(b_i, y, z) = 0$ and each surface patch passing through $S(s_i)$ becomes a curve segment of the curve $f(b_i, y, z) = 0$ passing through Q_i. We summarize this as the following lemma.

Lemma 4.10 *Use the above notations. If $\mathcal{S}$ is delineable over $I(e)$ and $I(c)$ respectively, then the number of TSPs passing through $S(s_i)$ and projecting to $R(c)$ is the number of curve branches passing through Q_i and projecting to the region $R(c)$.*

According to the above discussion, we have the following algorithm.

Algorithm 4.11 **NumTSP**$(f(x, y, z), P_{i,j,k}, e, c)$ *$\mathcal{S} : f(x, y, z) = 0$ is the surface delineable over $I(e)$ and $I(c)$ respectively, $P_{i,j,k} \in \mathcal{SP}$, $e = (P_{i,j}, P_{u,v}) \in \mathcal{EE}$, and $c \in \mathcal{EC}$ with e as an edge. The output is $\#(E_i, c)$ where $S(E_i)$ are all the SCCSes passing through $P_{i,j,k}$ and projecting to $C(e)$.*

1. Execute Algorithm **NumSCCS**$(f(x, y, z), P_{i,j,k}, e)$.

2. If e is not an x-vertical edge, execute the following steps

   (a) Let $\mathcal{Q} = \{Q_1, \ldots, Q_m\}$ be the points obtained in Step 3 of Algorithm **NumSCCS**.

   (b) Let $\Sigma_{21} = \{g(b_i, y), f(b_i, y, z)\}$ be the defining triangular system for $\mathcal{Q}$. Execute Algorithm 3.3 with input $\mathcal{Q}$ to compute $L\#(Q_i)$ and $R\#(Q_i)$.

   (c) Let c_1 be the cell under e in the y direction and c_2 the one above e. By Lemma 4.10, $\#(S_l, c_1) = L\#(Q_i)$ and $\#(S_l, c_2) = R\#(Q_i)$. See Fig. 12 for an illustration.

3. If e is an x-vertical edge, execute the following steps

   (a) Let $\mathcal{R} = \{R_1, \ldots, R_s\}$ be the points obtained in Step 1 of Algorithm **NumSCCS**.

   (b) Let $\Sigma_{22} = \{g(x, d_{i,j}), f(x, d_{i,j}, z)\}$ be the defining triangular system for $\mathcal{R}$. Execute Algorithm 3.3 with input $\mathcal{R}$.

(c) Let c_1 be the cell on the left hand side of e and and c_2 the cell on the right hand side of e. By Lemma 4.10, $\#(T_l, c_1) = L\#(R_l)$ and $\#(T_l, c_2) = R\#(R_l)$.

If there exist no TSPs connect to a SCCS, then the SCCS is an *isolated spatial curve segment*.

4.7 The General Case

Until now, we assume that the surface $\mathcal{S}$ does not contain straight lines parallel to the z-axis. In this subsection, we will show how to treat surfaces that contain such lines.

The aim is to get the points on the vertical lines where the topology of the surface changed, and the intersections between some SCCSes and the vertical lines, then the SCCSes originating from these points, and the surface patches originating from the line segments defined by these points.

The following will show how to compute the special case when $f(\alpha, \beta, z) \equiv 0$ for some point $P = (\alpha, \beta)$. It is clear that $g(x, y) = 0$ has a finite number of such points since $f(x, y, z)$ has no factor containing x, y only. We can solve the problem in the following way.

1. Take a coordinate system transformation such that the transformed line L_1 of the vertical line L_0 can be projected as a line L_2 on the new XY-plane.
2. Determine the topological information of L_2: the intersections of L_2 and the new projection curve, the number of curve segments originating from each intersection on its two sides.
3. Determine the topological information of L_1: lifting the intersections of L_2 and the projection curve of the new surface to determine the corresponding points on L_1. Find the points where the topology of surface changed on L_1.
4. We can made the same coordinate system transformation for the intersection of two surfaces $G(x, y) = 0$ and $f(x, y, z) = 0$. Then we can decide the points on the vertical lines which are the intersections of SCCSes and the vertical line.
5. Find the points where the topology of the original surface changed on L_0 from L_1 by coordinate relationship. Determine the topological information of L_0.

Remark: It is convenient to take a transformation such that L_2 is a vertical line of the new projection curve if L_2 can't overlap other line(s) of the projection curve.

In this way, we can solve the special case in Algorithm 4.5. Since we have introduced the operations we need before, we just use an example to show the effectivity.

In this special case, $\mathcal{SE}$ contains the edge with the from $(P_{i,j,k}, P_{i,j,k+1})$. Similarly, $\mathcal{SF}$ contain the face with the form $(P_{i,j,k}, P_{i,j,k+1}, P_{u,v,w})$.

We will continue the same example. Let us consider the following surface inside $\mathbb{B} = [-2, 2] \times [-2, 2] \times [-2, 2]$ as an example.

$$\mathcal{S} : f(x, y, z) = x^2y^2 + x^2z^2 + y^2z^2 - \frac{7}{2}xyz = 0. \tag{16}$$

It is clear that $f(0,0,z) \equiv 0$. So $(0,0)$ is a point in special case. So $L_0 : \{x = 0, y = 0, -2 \leq z \leq 2\}$. The topology polyhedron of the surface is shown in Figure 14.

1. Take the system transformation

$$\{x = X - Z, y = Y - Z, z = Z.\} \tag{17}$$

We get a new surface

$$\begin{aligned} \mathcal{S}' : F(X,Y,Z) &= X^2Y^2 - 2\,X^2YZ + 2\,X^2Z^2 - 2\,XZY^2 + 4\,XZ^2Y - 4\,XZ^3 + 2\,Z^2Y^2 \\ &\quad -4\,Z^3Y + 3\,Z^4 - \tfrac{7}{2}\,ZXY + \tfrac{7}{2}\,XZ^2 + \tfrac{7}{2}\,YZ^2 - \tfrac{7}{2}\,Z^3 \\ &= 0. \end{aligned}$$

Now L_0 corresponds to the line segment $L_1 : \{X = Z, Y = Z, -2 \leq Z \leq 2\}$ on the new surface.

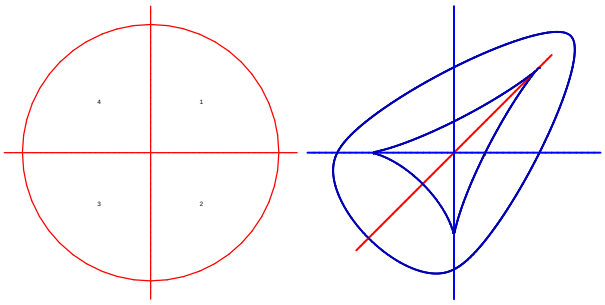

Figure 13: Determining the V_i in special case

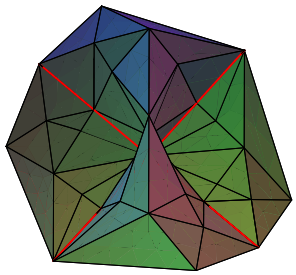

Figure 14: Topology polyhedron of a surface with vertical line

2. The projection curve of $\mathcal{S}'$ is shown in the right part of Figure 13. The red line segment is the $L_2 : \{X - Y = 0, -2 \leq X \leq 2\}$. It corresponds to L_1. The isolation boxes of the singularities of the projection curve of $\mathcal{S}'$ on L_2 are below.

$$[P_1, P_2, P_3, P_4] := [[-\tfrac{7}{4}, -\tfrac{7}{4}] \times [-\tfrac{7}{4}, -\tfrac{7}{4}], [-\tfrac{41}{64}, -\tfrac{655}{1024}] \times [-\tfrac{41}{64}, -\tfrac{655}{1024}], [0,0] \times [0,0], [\tfrac{7}{4}, \tfrac{7}{4}] \times [\tfrac{7}{4}, \tfrac{7}{4}]].$$

3. The corresponding points Q_i on L_1 of these singularities P_1, P_2, P_3, P_4 are

$$[Q_1, Q_2, Q_3, Q_4] := [[t \times t \times t],$$

$$t = [-\frac{7}{4}, -\frac{7}{4}], [-\frac{41}{64}, -\frac{655}{1024}], [0,0], [\frac{7}{4}, \frac{7}{4}]].$$

Assume the endpoints of L_1 are Q_0, Q_5. Computing the SCCSes originating from $Q_i (i = 0, \ldots, 5)$ with Algorithm 4.9, we can find that:

There are no SCCSes originating from $Q_0 (Q_5)$ except for $\overline{Q_0Q_1}$ ($\overline{Q_4Q_5}$) on L_1, we can find there are on surface patches originating from $\overline{Q_0Q_1}$ ($\overline{Q_4Q_5}$) by Algorithm 4.11; Similarly, Q_1 originates one SCCS from L_1's two sides respectively, and the SCCSes originates two surface patches in the two cells besides $\overline{Q_1Q_2}$, so Q_1 is a point we are interested; Q_2 originates two SCCSes from Q_2 besides L_1 respectively, and the four SCCSes all originate one surface patch on the cells besides $\overline{Q_1Q_2}$ and $\overline{Q_2Q_3}$, which means Q_2 is not a point where the topology of the new surface changes on L_1; Q_3 originate two line segments parallelling to XY-plane as SCCSes on L_1's two sides respectively, all

originating 2 surface patches on the cell bodies besides them, which means the SCCSes are singular curve of the surface, so Q_3 is a point we are interested; Q_4 originate one SCCS on L_1's two sides respectively, $\overline{Q_3Q_4}$ originates surface patches but there is no surface patches originating from $\overline{Q_4Q_5}$, so Q_5 is a point we are interested.

So we can conclude that Q_1, Q_3, Q_4 are the points where the topology of the surface changed on L_1.

4. Take the same coordinate system transformation as (17) for $g = xy(16x^2 + 16y^2 - 49)$, We can get a surface:

$$G(X, Y, Z) = (X - Z)(Y - Z)(16(X - Z)^2 + 16(Y - Z)^2 - 49).$$

We just need to decide some points on the line corresponding to the vertical line on the space curve defined by $G(X, Y, Z) = 0$ and $F(X, Y, Z) = 0$. Use the method in [16], we can find that there is only one point on the vertical line which is the intersection of SCCSes and the vertical line. It is $[0, 0] \times [0, 0] \times [0, 0]$.

5. Now we can get the points where we are interested on L_0, we can simply call these points as **vertical points**. By the coordinate relationship of L_1 and L_0, we can get the points we are interested on L_0 which correspond to Q_i. Since the topology of the surface does not change on L_1 at Q_2, we need not to consider the corresponding point on L_0. Let V_0, V_1, V_2, V_3, V_4 be the points on L_0 corresponds to Q_0, Q_1, Q_3, Q_4, Q_5 on L_1. We have the points

$$[V_0, V_1, V_2, V_3, V_4] = [[[0, 0] \times [0, 0] \times t],$$
$$t = [-2, -2], [-\tfrac{7}{4}, -\tfrac{7}{4}], [0, 0], [\tfrac{7}{4}, \tfrac{7}{4}], [2, 2]]$$

and the edges $(V_0, V_1), (V_1, V_2), (V_2, V_3), (V_3, V_4)$.

Now we need to find out the SCCSes of the original surface which originate from these points and edges on vertical line. The basic idea is as below.

At first, find a separate point W_i on each vertical edge, that is, between two adjacent points V_{i-1}, V_i, then construct a plane Θ_{W_i} paralleling to XY-plane passing W_i, search a rectangle R_i containing W_i such that all the curve segments inside R_i originate from W_i, and when projected into XY-plane, all this kind of R_i correspond to a same rectangle R which only contains one critical point P. In order to determine the number of SCCSes originating from each vertical point, we need the following lemma.

Lemma 4.12 *The number of SCCSes originating from the point V_i equals the number of intersections of line $\{x = \alpha, y = \beta\}$ and the surface $\mathcal{S}$ between two planes Θ_{W_i} and $\Theta_{W_{i+1}}$, where (α, β) is a point on $C(e)$ inside R.*

Proof. Since there is only one vertical points between Θ_{W_i} and $\Theta_{W_{i+1}}$, the SCCSes between two planes originate from V_i. There is no part of the surface in R_i or R_{i+1} has intersection with $I(e)$ when projected to R. Otherwise, there exists a critical point on R besides P. By Lemma 4.1, the SCCS originating from V_i only intersects the line $\{x = \alpha, y = \beta\}$ once. So the lemma is true. ∎

So we have the method to decide the number of SCCSes for vertical line case.

Then, we need to decide the number of surface patches originating from each vertical line segment in each plane cell. In fact, this is done! The boundaries of R_i have some intersections with the

surface, the number of the intersections in each cell body is the number of surface patches originating from the corresponding vertical line segment. For our example, since V_0, V_4 are endpoints and $(V_0, V_1), (V_3, V_4)$ do not originate surface patch, we just need to find rectangles for $(V_1, V_2), (V_2, V_3)$. So we can conclude that (V_1, V_2) originate two surface patches in cell bodies "2" and "4" respectively, and (V_2, V_3) originate two surface patches in cell bodies "1" and "3" respectively. When computing the SCCSes originating from V_1, V_2, V_3, we can find that V_1, V_3 do not originate non-vertical SCCSes, V_2 originates four line segments as SCCSes.

In the end, we should form triangles for this case. The curve branches in R_i can intersect the plane triangles when projected to XY-plane. Use these points to subdivide the plane triangles, and then to form triangular patches. Note that when an endpoint of a plane triangle corresponding to a vertical line, some of the surface patches corresponding to the triangle should contain two or three TSPs.

Figure 14 is a triangular polyhedron representation of the surface defined by Equation 16 which has a vertical line $\{x = 0, y = 0\}$.

5 Ambient isotopic meshing of surface

In this section, we will show how to compute an ϵ-meshing of a surface $\mathcal{S}$ for a given $\epsilon > 0$.

Let $\mathcal{M}_1$ be an ϵ-meshing graph of the projection curve of $\mathcal{S}$ computed with Algorithm 3.10. Consider the two disjoint regions of $\mathbf{B}_3$:

$$\mathbf{S}_3 = \cup_{e \in \mathcal{M}_1} \mathbf{B}_e \times [\mathcal{Z}_1, \mathcal{Z}_2] \quad (18)$$

$$\mathbf{N}_3 = \mathbf{B}_3 \setminus \mathbf{S}_3. \quad (19)$$

Surface $\mathcal{S}$ has no singularities in the cylindrical region $\mathbf{N}_3$, so we can use a modified Pantinga-Vegter method [24] to compute its meshing. What we need to do is to compute the correct meshing inside $\mathbf{S}_3$. To present the algorithm, we need preparations given in Sections 5.1 and 5.2.

5.1 Extremal points of surfaces and spatial curves

In order to give an ambient isotopic meshing for a surface, we need to consider z-extremal points of surfaces and spatial curves. A point is called z-**extremal** if the surface achieves a local extremum value at this point in the z-direction. We have

Lemma 5.1 *Let $f(x, y, z) = \prod_i f_i(x, y, z)$ be a square free polynomial and f_i irreducible polynomials. A necessary condition for the surface $f(x, y, z) = 0$ to have a z-extremal point is*

$$G_1(x, y) = \prod_i \text{Res}(f_i, \frac{\partial f_i}{\partial x}, z) \prod_i \text{Res}(f_i, \frac{\partial f_i}{\partial y}, z) = 0 \quad (20)$$

where only the nonzero resultants are included.

The following example shows that we need to consider the irreducible factors. Let $f = (z-y)(z-x)(x^2+y^2+z^2-1)$. Then $\text{Res}(f, f_x, z) = \text{Res}(f, f_y, z) \equiv 0$. But the surface indeed has an z-extremal point at $(0, 0, 1)$.

Lemma 5.2 *Let $f(x, y, z)$ be a square free polynomial, $D(x, y)$ defined in (10), $G_1(x, y)$ defined in (20), and r a fixed number. Then $D(x, y)G_1(x, y) = 0$ is a necessary condition for the curves $f(x, y, r) = 0$, $f(x, r, z) = 0$, and $f(r, y, z) = 0$ to have x-extremal, y-extremal, or z-extremal points.*

We also need to consider the z-extremal points of spatial curves defined by $g(x, y) = f(x, y, z) = 0$, where g and f are polynomials. For this purpose, we need to decompose the curve into irreducible ones. The leading coefficient of g (f) as an univariate polynomial in y (z) is called the **initial** of g (f). Two polynomials of the form $g(x, y), f(x, y, z)$ is called an **irreducible chain** if the following conditions are satisfied [21] (pages, 297-381)

- $g(x, y)$ is an irreducible polynomial.
- $f(x, y, z)$ is an irreducible polynomial of z module $g = 0$, $\deg(f, y) < \deg(g, y)$, and the initial of f is a polynomial in x.

For instance, $g = y^2 - x$, $f = z^2 - x$ is not irreducible, since $f = (z-y)(z+y)+g = (z-y)(z+y) \mod (g)$.

For an irreducible chain $g(x, y), f(x, y, z)$, we define its **saturation ideal** to be

$$\text{Sat}(g(x, y), f(x, y, z)) = \{P \mid I_1^s I_2^k P \in (f, g)\}$$

where I_1 and I_2 are the initials of g and f respectively. It is known that the saturation ideal of an irreducible chain is a prime ideal, and thus defines an irreducible spatial curve [21] (pages, 297-381).

Any spatial curve $f(x, y, z) = g(x, y) = 0$ can be decomposed into the union of irreducible curves algorithmically:

$$V(g(x, y), f(x, y, z)) = \cup_i V(\text{Sat}(g_i(x, y), f_i(x, y, z))) \tag{21}$$

where $g_i(x, y), f_i(x, y, z)$ are irreducible chains. We can prove the following result:

Theorem 5.3 *Let $g(x, y), f(x, y, z)$ be an irreducible chain and*

$$\begin{aligned} I(x) &= \textit{product of the initials of } f, g. \\ T(x) &= \text{Res}(\text{Res}(h, f, z), g, y) \textit{ where } h(x, y, z) = f_x g_y - f_y g_x. \end{aligned} \tag{22}$$

Let E be the set of z-extremal points of the curve $\mathcal{C} : f = g = 0$. Then

$$Proj_x(E) \subset V(T(x)) \cup V(I(x)). \tag{23}$$

Furthermore, if $T(x) \equiv 0$, then the curve $V(\text{Sat}(\{f, g\}))$ is contained in several planes perpendicular to the z-axis.

Proof. For any point $P = (\alpha, \beta, \gamma)$ on $\mathcal{C}$, the necessary condition of P being a z-extremal point of $\mathcal{C}$ is the tangent line of $\mathcal{C}$ at P is perpendicular to z-axes. If P is neither the singular point of $f = 0$ nor $g = 0$, the tangent planes of $f = 0$ and $g = 0$ at P are both well defined. The tangent line of $\mathcal{C}$

at P is the intersection of the tangent planes of $f = 0$ and $g = 0$ at P. The tangent direction $\mathbf{n}$ of $\mathcal{C}$ at P is

$$\begin{aligned}\mathbf{n} &= < f_x, f_y, f_z > |_P \times < g_x, g_y, 0 > |_P \\ &= < -g_y f_z, f_z g_x, f_x g_y - f_y g_x > |_P.\end{aligned}$$

Since the tangent planes of $f = 0$ and $g = 0$ at P are:

$$\begin{cases} (x-\alpha) f_x(\alpha,\beta,\gamma) + (y-\beta) f_y(\alpha,\beta,\gamma) + (z-\gamma) f_z(\alpha,\beta,\gamma) = 0, \\ (x-\alpha) g_x(\alpha,\beta) + (y-\beta) g_y(\alpha,\beta) = 0 \end{cases}$$

respectively. $\mathbf{n}$ is perpendicular to the z-axes, that is:

$$\begin{aligned}\mathbf{n} \cdot < 0, 0, 1 > &= f_x(\alpha,\beta,\gamma) g_y(\alpha,\beta) - f_y(\alpha,\beta,\gamma) g_x(\alpha,\beta) \\ &= h(\alpha,\beta,\gamma) = 0.\end{aligned}$$

Therefore, $E \subset V(h(x, y, z))$. (23) is true.

If $T(x) \equiv 0$, we prove that $V(\{f, g\}/I)$ is contained in several planes perpendicular to the z-axis.

First we claim that $\mathbf{n}$ is well defined on $\mathcal{C}$ except finite number of points, that is only finite number of points on $\mathcal{C}$ are the singular points of $f = 0$ or $g = 0$. If it is not true, at least one of the following conditions occurs:

C1. $V(f, g, f_x, f_y, f_z)$ has 1-dimensional component.

C2. $V(f, g, g_x, g_y)$ has 1-dimensional component.

If C1. occurs, it means that $f_z \in \text{Sat}(f, g)$. It is impossible. Condition C2. could not take place for the same reason. Note that $h(x_0, y_0, z_0) = 0$ for any point (x_0, y_0, z_0) on $\mathcal{C}$. The tangent direction of $\mathcal{C}$ at almost all points is the form $(A, B, 0)$.

Then we prove this component of $\mathcal{C}$ lies in some planes $z = z_0$. This component of $\mathcal{C}$ can be parametrization in some segments. Assume the parametric equation is $\mathbf{r}(t)$. We have

$$\mathbf{r}(t) = \mathbf{r}(t_0) + \int_0^t \mathbf{r}'(t) = (x(t), y(t), z_0),$$

where $\mathbf{r}(t_0) = (x_0, y_0, z_0)$. It implies that this segment of $\mathcal{C}$ lies in the plane $z = z_0$. Therefore, the irreducible component of $\mathcal{C}$ which contains this segment lies in the plane $z = z_0$. We prove this theorem. ∎

The following example shows that we need to decompose the curve into irreducible ones. Let $f = z(x^2 + z^2 - 1), g = y$. Then $\text{Res}(f_x g_y - f_y g_x, f, z) \equiv 0$. But the curve indeed has a z-extremal point at $(0, 0, 1)$.

5.2 Compute segregating box for an SCCS

In Section 4.5, we showed how to compute the segregating box for a singular point. In this section, we introduce the concept of segregating boxes for singular curve segments.

In Algorithm 3.10, a curve segment $C(e)$ of the projection curve $\mathcal{C}$ is represented by a segment e contained in a box $\mathbf{B}_e$, as shown in Fig. 5. When lifting $\mathcal{C}$ to the space, we obtain a set of SCCSes $S_i, i = 1, \ldots, d$ of $\mathcal{S}$ represented by edges $E_i \in \mathcal{SE}$ (see Section 4.2). A box $\mathbf{B}_{S_i} = \mathbf{B}_e \times [e_i, f_i]$ is called a **segregating box** for S_i if $\mathbf{B}_{S_j} \cap \mathbf{B}_{S_i} = \emptyset$ for $i \neq j$ and $\mathcal{S}$ does not intersect with the top and bottom faces of $\mathbf{B}_{S_i}$. In Fig. 15, we give a segregating box for the surface patches $A_1B_1C_1D_1$ and $A_2B_2C_2D_2$ intersecting at curve segment $P_{11}P_{21}$ which is lifted from curve segment P_1P_2.

Assume that all S_i are monotonous in the direction of z, the following algorithm shows how to compute segregating boxes for the SCCSes: $S_i, i = 1, \ldots, d$.

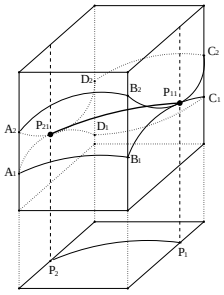

Figure 15: Segregating boxes

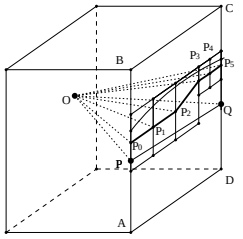

Figure 16: Merge meshes

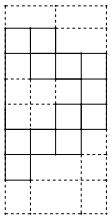

Figure 17: Divide $\mathbf{N}_2$ into boxes

Algorithm 5.4 SegBoxC$(f(x, y, z), g(x, y), \mathbf{B}_3, \mathbf{B}_e, \epsilon)$**.** *Let $\mathcal{S} : f(x, y, z) = 0$ be the surface, $\mathbf{B}_3$ the bounding box, $\mathbf{B}_e$ a nice box (see Fig. 4) containing a curve segment $C(e)$ of the projection curve $\mathcal{C} : g(x, y) = 0$ of $\mathcal{S}$, $\epsilon > 0$.*

The output is a pair $(\mathbf{P}, \mathbf{S})$. $\mathbf{P}$ is a set of interior-disjoint boxes contained in $\mathbf{B}_e$, the union of which contains $C(e)$. For each $\mathbf{P}_i \in \mathbf{P}$, there exist 3D boxes $\mathbf{S}_{i,j} \in \mathbf{S}$ which are the segregating boxes for the SCCSes lifted from $C(e) \cap \mathbf{P}_i$.

1. We consider case (a) in Fig. 4. Other cases can be treated similarly. $C(e)$ divides $\mathbf{B}_e$ into two cells c_1 and c_2.
2. Let $\mathbf{P}_1 = \{\mathbf{B}_e\}$, $\mathbf{P} = \emptyset$, $\mathbf{S} = \emptyset$. Repeat the following steps until $\mathbf{P}_1 = \emptyset$.
   (a) Let $\mathbf{B} = [a, b] \times [c, d] \in \mathbf{P}_1$ and remove $\mathbf{B}$ from $\mathbf{P}_1$.
   (b) Execute **RootIsol**$(\{g(a, y), f(a, y, z)\}, [c, d] \times [\mathcal{Z}_1, \mathcal{Z}_2], \epsilon)$ to compute the points $P_{1,i}, i = 1, \ldots, N_1$ lifted from P_1. See Fig. 15 for an illustration. Let the isolation box for $P_{1,i}$ be $\mathbf{S}_{1,i} \times [e_{1,i}, f_{1,i}]$.
   (c) Similarly, let $P_{2,i}, e_{2,i}, f_{2,i}, i = 1, \ldots, N_2$ be the points lifted from P_2. By Lemma 4.1, $N_1 = N_2$.
   (d) Let $\mathbf{B}_i = \mathbf{B}_e \times [\min\{e_{1,i}, e_{2,i}\}, \max\{f_{1,i}, f_{2,i}\}], i = 1, \ldots, N_1$.
   (e) If $|\mathbf{B}_i| < \epsilon$ for all i, add $\mathbf{B}_e$ to $\mathbf{P}$ and add $\mathbf{B}_i$ to $\mathbf{S}$.
   (f) Otherwise, subdivide $\mathbf{B}_e$ into four equal boxes and add the boxes intersecting with $\mathcal{C}$ into $\mathbf{P}_1$.
3. Repeat the following steps until all boxes in $\mathbf{S}$ are segregating.

(a) Let $\bar{\mathbf{B}} = \mathbf{B} \times [e, f] \in \mathbf{S}$.

(b) If $0 \notin \square f(\mathbf{B} \times [e, e])$ and $0 \notin \square f(\mathbf{B} \times [f, f])$, then $\bar{\mathbf{B}}$ is segregating, we do nothing for $\mathbf{B}$.

(c) Otherwise, remove $\mathbf{B}$ from $\mathbf{P}$ and $\bar{\mathbf{B}}$ from $\mathbf{S}$. Subdivide $\mathbf{B}$ into four equal boxes $\mathbf{C}_1, \mathbf{C}_2$, $\mathbf{C}_3, \mathbf{C}_4$, add each $\mathbf{C}_i$ intersecting with $\mathcal{C}$ into $\mathbf{P}$, and add $\mathbf{C}_i \times [e, f]$ into $\mathbf{S}$.

4. Return $\mathbf{P}$ and $\mathbf{S}$.

Proof of correctness. We need only prove the termination of the algorithm. According to the assumption, all S_i are monotonous in the z direction. So Step 2 terminates in a finite number of steps.

At the beginning of Step **3**, for any $C(e) \subset \mathbf{B} = [a, b] \times [c, d] \in \mathbf{P}$ where $e = (P_1, P_2),\ P_1 = (a, \alpha), P_2 = (b, \beta)$, let $C(s_j)$ be a curve segment lifted from e and $\mathbf{B}_i = [a, b] \times [c, d] \times [e_i, f_i]$ be the corresponding box where $s_i = (P_{1,i}, P_{2,i}),\ P_{1,i} = (a, \alpha, \gamma_i), P_{2,i} = (b, \beta, \tau_i)$, we have $|\mathbf{B}_i| < \epsilon$ and

$$f(a, \alpha, e_i) f(a, \alpha, f_i) f(b, \beta, e_i) f(b, \beta, f_i) \neq 0.$$

Furthermore, s_i does not intersect with the top nor bottom faces of $\mathbf{B}_i$ since s_i is monotonous in the direction of z. That is $0 \notin \square f(C(e) \times [e_i, e_i]) f(C(e) \times [f_i, f_i])$. So there exists a positive number δ such that

$$0 \notin \square f(d(C(e), \delta) \times [e_i, e_i]) f(d(C(e), \delta) \times [f_i, f_i])$$

where $d(C(e), \delta)$ is a zonal region in $\mathbb{R}^2$ containing the points Q such that the distance between Q and $C(e)$ is less than δ. We can get the set of sub-boxes of $\mathbf{B}$ in a finite steps such that all boxes in it are contained in the region $d(C(e), \delta)$. Then the algorithm clearly terminates. ∎

5.3 Compute ϵ-meshing of surface

Similar to the case of curves, we need to modify the Pantinga-Vegter method. A box $\mathbf{B}$ is called a **nice box** if each face of $\mathbf{B}$ is a nice 2D box. For an illustration, see the 2D case in Fig. 4. To make the process precise, we introduce the following definition.

A **meshing polyhedron** of a surface $\mathcal{S}$ is a four-tuple $\mathcal{M} = \{\mathcal{P}, \mathcal{E}, \mathcal{F}, \mathcal{B}\}$ where $(\mathcal{P}, \mathcal{E}, \mathcal{F})$ is a polyhedron whose vertices are with rational numbers as coordinates and whose **faces are the meshes** for $\mathcal{S}$; $\mathcal{B}$ is a set of nice boxes and segregating boxes of singular points of $\mathcal{S}$ s.t. for each $F \in \mathcal{F}$, there exists a $\mathbf{B}_F \in \mathcal{B}$ with the property that the surface patch $\mathcal{S} \cap \mathbf{B}_F$ is connected.

A meshing polyhedron $\mathcal{M}$ is called an **ϵ-meshing polyhedron** if each box $\mathbf{B}$ in $\mathcal{B}$ satisfies $|\mathbf{B}| < \epsilon$. It is easy to show that an ϵ-meshing polyhedron for a surface $\mathcal{S}$ provides an **ϵ-meshing** for $\mathcal{S}$ according to the definition given in Section 2.

Algorithm 5.5 MPV3$(f(x, y, z), \mathbf{N}_3, \epsilon)$. *$\mathcal{S} : f(x, y, z) = 0$ is the surface. $\mathbf{N}_3$ is a box contains no zero of $D(x, y)G_1(x, y) = 0$ where $D(x, y)$ is defined in (10) and $G_1(x, y)$ is defined in (20). Output an ϵ-meshing polyhedron for $\mathcal{S}_{\mathbf{N}_3}$.*

1. Subdivide $\mathbf{N}_3$ into boxes $\mathbf{B}_i$ at the corner lines (Fig. 17 shows how to subdivide the region $\mathbf{N}_2^1$ in Fig. 6(a), where the dotted lines are newly added.) and execute the Pantinga-Vegter algorithm with initial values $\{\mathbf{B}_i\}$. Let $\mathbf{S}$ be the output.

2. For each cube $\mathbf{B} \in \mathbf{S}$, repeat subdividing $\mathbf{B}$ until all of the following statements are false.
   (a) There exists an edge (A, B) of $\mathbf{B}$ s.t. $0 \in \square f((A, B))$ and $f(A)f(B) > 0$.
   (b) There exists a face $ABCD$ of $\mathbf{B}$ s.t. $f(A)f(B) < 0 \wedge f(B)f(C) < 0 \wedge f(C)f(D) < 0 \wedge f(D)f(A) < 0$.
   (c) $|\mathbf{B}| > \epsilon$.

Termination of the algorithm is guaranteed by Lemma 5.2.

Now we can compute the ϵ-meshing for $\mathcal{S}_{\mathbf{B_3}}$.

Algorithm 5.6 ATopSur$(f(x, y, z), \mathbf{B}_3, \epsilon)$. *The input is the same as Algorithm 4.5. The output is an* ϵ**-meshing polyhedron** *for* $\mathcal{S}_{\mathbf{B_3}}$.

S1 Compute the critical points of the projection curve and their segregating boxes.

1. Let
$$G(x, y) = \text{sqrfree}(G(x, y)G_1(x, y)), \tag{24}$$
where G is defined in (11) and G_1 is defined in (20).
2. Execute the first four steps of Algorithm 3.5 with input $(G(x, y), \mathbf{B}_2, \epsilon)$ to compute a set of points $\mathcal{P}_1$ and the segregating box for each point in $\mathcal{P}_1$. We need to modify the algorithm as follows. In Step 3 of Algorithm 3.5, we use the new projection polynomial:
$$H(x) := H(x) \prod_i I_i(x) \prod_i T_i(x), \tag{25}$$
where $H(x)$ is defined in (4), $I_i(x)$ and $T_i(x)$ are defined in (22) with decomposition (21). Only the nonzero T_i are considered.

S2 Compute $\mathcal{SP}_0$ and the set $\mathcal{SB}_0$ of segregating boxes for points in $\mathcal{SP}_0$. For any $P_{i,j} \in \mathcal{P}_1$, use Algorithm 4.7 with input $(f, \mathbf{B}_3, P_{i,j}, \epsilon)$ to compute the points lifted from $P_{i,j}$ and their segregating boxes. Let $\mathcal{B}_1$ be the set of all updated segregating boxes $\mathbf{B}_{i,j}$ of $P_{i,j}$. Let $\mathcal{M}_1 = \{\mathcal{P}_1, \mathcal{B}_1\}$.

S3 Compute an ϵ-meshing graph for the non-singular part of $\mathcal{C}_{B_2}$ in $\mathbf{N}_2$ defined in (6). Let $\mathcal{M}_0 = \textbf{MPV2}(G(x, y), \mathbf{N}_2, \epsilon)$, where $\mathbf{N}_2$ is defined in (6).

S4 Compute segregating boxes for SCCSs:

1. Assume $\mathcal{M}_0 = \{\mathcal{P}_0, \mathcal{E}_0, \mathcal{B}_0\}$. Let $\mathcal{SB}_1 = \mathcal{P}_2 = \mathcal{E}_2 = \mathcal{B}_2 = \emptyset$.
2. For each $\mathbf{B} \in \mathcal{B}_0$, execute the following steps:
   (a) Compute $\{\mathbf{P}, \mathbf{S}\}$=**SegBoxC**$(f(x, y, z), G(x, y), \mathbf{B}_3, \mathbf{B}, \epsilon)$.[1]
   (b) $\mathcal{SB}_1 = \mathcal{SB}_1 \cup \mathbf{S}$ and update $\mathcal{P}_2, \mathcal{E}_2, \mathcal{B}_2$ according to $\mathbf{P}$ which subdivides $\mathbf{B}$.
3. Let $\mathcal{M}_2 = \{\mathcal{P}_2, \mathcal{E}_2, \mathcal{B}_2\}$.

S5 Compute the extended meshing graph $\mathcal{EG}_S$ of $\mathcal{C}$ with Algorithm **??** with input $\mathcal{M}_1$ and $\mathcal{M}_2$.

[1] Step S1 ensures all z-extremal points of the curve $\mathcal{C} : f = 0, G = 0$ are in $\mathbf{S}_3$. Hence the SCCS in $\mathbf{B}$ is monotonous in the direction of z.

S6 Meshing the singular part of $\mathcal{S}$ in $\mathbf{S}_3$.

1. Let $\{\mathcal{SP}_1, \mathcal{SE}_1, \mathcal{SF}_1\}$=**TopSur**$(f(x,y,z), \mathbf{B}_3)$. Modify Algorithm **TopSur** as follows: use $G(x,y)$ defined in (24) in Step 1, use $\mathcal{EG}_S$ in the Step 2, and use $\mathcal{SP}_0$ in Step 3. We actually only run Steps 4 and 5 of Algorithm **TopSur**.
2. Let $\mathcal{M}_1 = \{\mathcal{SP}_1, \mathcal{SE}_1, \mathcal{SF}_1, \mathcal{SB}_0 \cup \mathcal{SB}_1\}$ where $\mathcal{SB}_0$ and $\mathcal{SB}_1$ are from Steps S2 and S4 respectively. $\mathcal{M}_1$ is an ϵ-meshing polyhedron for $\mathcal{S}_{\mathbf{S}_3}$.

S7 Meshing the non-singular part of $\mathcal{S}$ in $\mathbf{N}_3$. Let $\mathcal{M}_2 = \{\mathcal{MP}_2, \mathcal{ME}_2, \mathcal{MF}_2, \mathcal{MB}_2\} =$ **MPV3**$(f(x,y,z), \mathbf{N}_3, \epsilon)$, where $\mathbf{N}_3$ is defined in (19).

S8 Merge $\mathcal{M}_1$ and $\mathcal{M}_2$ to obtain an ϵ-meshing polyhedron for $\mathcal{S}$. Output **Merge**$(\mathcal{M}_1, \mathcal{M}_2)$ (with Algorithm 5.8).

Theorem 5.7 *Algorithm 5.6 computes an ϵ-AIMESH for $\mathcal{S}_{\mathbf{B_3}}$.*

Proof. The prove is similar to the proof of Theorem 3.11. ∎

In principle, there exist no difficulties to implement the algorithm. But, it will take a lots of time, since we need to incorporate algorithms from symbolic computation, interval arithmetics, and marching cube into one program. This will be our further work,

In the final step of Algorithm 5.6, we need to merge two meshing polyhedrons, which will be done by the following algorithm.

Algorithm 5.8 Merge$(\mathcal{M}_1, \mathcal{M}_2)$. *$\mathcal{M}_1 = \{\mathcal{MP}_1, \mathcal{ME}_1, \mathcal{MF}_1, \mathcal{MB}_1\}$ and $\mathcal{M}_2 = \{\mathcal{MP}_2, \mathcal{ME}_2, \mathcal{MF}_2, \mathcal{MB}_2\}$ are the ϵ-meshing polyhedrons of $\mathcal{S}$ in $\mathbf{S}_3$ and $\mathbf{N}_3$ respectively. The algorithm merges $\mathcal{M}_1$ and $\mathcal{M}_2$ and outputs an ϵ-meshing polyhedron $\mathcal{M} = \{\mathcal{MP}, \mathcal{ME}, \mathcal{MF}\}$ for the surface.*

S1 Let $\mathcal{MB}_t = \mathcal{MB}_1$.

S2 While $\mathcal{MB}_t \neq \emptyset$, repeat

1. Remove $\mathbf{B} = [a,b] \times [c,d] \times [e,f]$ from $\mathcal{MB}_t$. Insert box $\mathbf{B}_i = [b,b_i] \times [c_i,d_i] \times [e_i,f_i] \in \mathcal{MB}_2$ which is connected with $\mathbf{B}$ according to $\mathcal{S}$ and adjacent to the face $\mathbf{F} = [b,b] \times [c,d] \times [e,f]$ into $\mathbf{B}_a$ and insert corresponding $V(\mathbf{B}_i)$ into $\mathcal{P}_a$. Pick out boxes $\mathbf{B}_i$ satisfying $d_i = \min_{\mathbf{B}_j \in \mathbf{B}_a}\{d_j\}$. Rename them to be $\mathbf{B}_1, \ldots, \mathbf{B}_m$. Sort the residual boxes in $\mathbf{B}_a$ as $\{\mathbf{B}_{m+1}, \ldots, \mathbf{B}_r\}$ such that $c_{m+1} \leq c_{m+2} \leq \ldots \leq c_r$ and for each $\mathbf{B}_k, k > m$, $\mathbf{B}_k$ is connected with some $\mathbf{B}_j,\ j < m$ according to $\mathcal{S}$(Note that the result is not unique, and any $B_k, 0 < k < m$ only overlaps with $\mathbf{B}$ on the vertical edge $[b,b] \times [d_i,d_i] \times [e_i,f_i]$).
2. For i from 1 to r do
   (a) Remove the points P from $V(\mathbf{B}_i)$ and insert points $Q \in \mathcal{SP} \cap (\mathbf{B} \cap \mathbf{B}_i)$ into $V(\mathbf{B}_i)$ if $P \in \mathbf{L} \cup \mathbf{R}$ where $\mathbf{L} = [b,b] \times [c,c] \times [e,f],\ \mathbf{R} = [b,b] \times [d,d] \times [e,f]$. Remove edge (P, P_i) from $\mathcal{ME}_2$ which are the edges with P as an ending point and insert (Q, P_i) into $\mathcal{MF}_2$. Remove triangular faces (P, P_i, P_j) from $\mathcal{MF}_2$ which are the faces with (P, P_i) as an edge and insert (Q, P_i, P_j) into $\mathcal{MF}_2$.

(b) If there exist $P \in \mathbf{R_i}$ where $\mathbf{R_i} = [b, b] \times [d_i, d_i] \times [e_i, e_i]$ and $\mathbf{R_i} \cap \mathbf{L} \cap \mathbf{R} = \emptyset$, add P into $\mathcal{MP}_1$. goto (e).

(c) If there exist$P \in \mathbf{D_i} \subset \mathbf{F}$ where $\mathbf{D_i} = [b, b] \times [c_i, d_i] \times [e_i, e_i]$. Add P in $\mathcal{SP}$. goto (e).

(d) If there exist$P \in \mathbf{U_i} \subset \mathbf{F}$ where $\mathbf{U_i} = [b, b] \times [c_i, d_i] \times [f_i, f_i]$. Add P in $\mathcal{SP}$. goto (e).

(e) Assume the other point contained in the face $[b, b] \times [c_i, d_i] \times [e_i, f_i]$ of $\mathbf{B}_i$ is Q and $(Q, S, T) \in \mathcal{MF}_1$ is the triangular face with Q as a vertex where $S \in \mathbf{R}$. Remove (Q, S) from $\mathcal{ME}_1$ and insert $(Q, P), (P, S)$ into $\mathcal{ME}_1$. Remove triangular faces (Q, S, T) from $\mathcal{MF}_1$ and insert $(Q, P, S),\ (P, S, T)$ into $\mathcal{MF}_1$(The four edges of this face of $\mathbf{B}_i$ contains two points. We can always assume that we have dealt with the other one point, since $\mathbf{B}_i, i > m$ is connected with some $\mathbf{B}_j$ we have dealt with).

(f) Update $\mathcal{MB}_2$ according to the new $V(\mathbf{B}_i)$.

3. Determine the connection information of the other three faces of $\mathbf{B}$ in the similar way.

S3 Out $\mathcal{M} = \{\mathcal{MP}, \mathcal{ME}, \mathcal{MF}\}$ where $\mathcal{MP} = \mathcal{MP}_1 \cup \mathcal{MP}_2, \mathcal{ME} = \mathcal{ME}_1 \cup \mathcal{ME}_2$, and $\mathcal{MF} = \mathcal{MF}_1 \cup \mathcal{MF}_2$.

A box $\mathbf{B}_1$ is said to be **adjacent to a box** $\mathbf{B}_2$ w.r.t. the surface $\mathcal{S}$ if $\mathbf{B}_1$ and $\mathbf{B}_2$ are interiorly disjoint and $\mathcal{S}$ intersects $\mathbf{B}_1 \cap \mathbf{B}_2$. We need only consider how to merge the meshes in two adjacent boxes.

We use the example in Fig. 16 to explain the algorithm. The large box $\mathbf{B}$ is in $\mathbf{S}_3$ and contains singularities. We consider the right face $\mathbf{F}$ of $\mathbf{B}$. Let $\mathbf{B}_i, i = 1, \ldots, 5$ be the boxes in $\mathbf{N}_3$ adjacent to $\mathbf{B}$ at face $\mathbf{F}$. By Step 1 of Algorithm 5.5, $\mathbf{B}_i$ must be completely between lines AB and CD. We will adjust the meshes in $\mathbf{B}$ and leave the meshes in $\mathbf{B}_i$ unchanged. Since all the meshes are triangular, let OPQ be the mesh of $\mathcal{S}$ in $\mathbf{B}$, and $N_iP_{i-1}P_i$ the mesh of $\mathcal{S}$ in $\mathbf{B}_i$. We will replace the mesh OPQ with the meshes $M_i = OP_{i-1}P_i, i = 0, \ldots, 4$. If P_i is above BC, P_i is taken to be the intersection of BC and the line passing through P_i and parallel to AB. Other cases can be treated similarly.

6 Conclusion

This paper proposes complete methods to compute isotopic and ambient isotopic meshings for implicit algebraic curves and surfaces. We use symbolic computation to achieve completeness and whenever possible use interval arithmetics to achieve practical effectiveness. Note that an isotopic meshing without precision and an ϵ-meshing are quite different and can be used for different purposes.

References

[1] E. L. Allgower and S. Gnutzmann. An algorithm for piecewise linear approximation of implicitly defined two-dimensional surfaces. *SIAM J. Numer. Anal.*, 24:452–469, 1987.

[2] D. S. Arnon, G. Collins, and S. McCallum. Cylindrical algebraic decomposition, ii: an adjacency algorithm for plane. *SIAM J. on Comput.*, 13(4):878–889, 1984.

[3] D. S. Arnon, G. Collins, and S. McCallum. Cylindrical algebraic decomposition, iii: an adjacency algorithm for three-dimensional space. *J. Symbolic Comput.*, 5(1,2):163–187, 1988.

[4] C. Bajaj and G. L. Xu. Spline approximations for real algebraic surfaces. *J. Symbolic Comput.*, 5:285–307, 1997.

[5] J. D. Boissonnat, D. Cohen-Steiner, B. Mourrain, G. Rote, and G. Vegter. Meshing of surfaces. In J. D. Boissonnat and M. Teillaud, editors, *Effective Computational Geometry for Curves and Surfaces*, pages 181–230. Springer-Verlag, Berlin, Chapter 5, 2006.

[6] J. D. Boissonnat, D. Cohen-Steiner, and G. Vegter. Isotopic implicit surface meshing. *Discrete and Computational Geometry*, 39:138–157, 2008.

[7] J. D. Boissonnat and S. Oudot. Provably good surface sampling and approximation. In *Proc. Eurographics 2003*, pages 9–18. Eurographics Association, 2003.

[8] C. W. Brown. Improved projection for cylindrical algebraic decomposition. *J. Symbolic Comput.*, 5:447–465, 2001.

[9] M. Burr, S. Choi, B. Galehouse, and C. Yap. Complete subdivision algorithms, ii: isotopic meshing of algebraic curves. In *Proc. ACM ISSAC 2008*. ACM Press, 2008.

[10] J. Cheng, X. S. Gao, and M. Li. Determine the topology of real algebraic surfaces. In *Mathematics of Surfaces*, pages 121–146. Springer-Verlag, 2005.

[11] J. Cheng, X. S. Gao, and C. Yap. Complete numerical isolation of real roots in zero-dimensional triangular systems. In *Proc. ACM ISSAC 2007*, pages 92–99. ACM Press, 2007.

[12] S. W. Cheng, T. K. Dey, A. Ramos, and T. Ray. Sampling and meshing a surface with guaranteed topology and geometry. In *Proc. 20th Symp. on Computational Geometry*, pages 280–289. ACM Press, New York, 2004.

[13] A. Eigenwillig, M. Kerber, and N. Wolpert. Fast and exact geometric analysis of real algebraic plane curves. In *Proc. ACM ISSAC 2007*, pages 151–158. ACM Press, 2007.

[14] E. Fortuna, P. Gianni, and D. Luminati. Algorithmical determination of the topology of a real algebraic surface. *J. Symbolic Comput.*, 38:1551–1567, 2004.

[15] E. Fortuna, P. Gianni, P. Parenti, and C. Traverso. Algorithms to compute the topology of orientable real algebraic surfaces. *J. Symbolic Comput.*, 36:343–364, 2003.

[16] L. A. M. B. Gatellier, G. and J. P. T*é*court. Computing the topology of 3-dimensional algebraic curves. In *In Computational Methods for Algebraic Spline Surfaces*, pages 27–44. Springer-verlag, 2005.

[17] L. Gonzalez-Vega and I. Necula. Efficient topology determination of implicitly defined algebraic plane curves. *Computer Aided Geometric Design*, 19:719–743, 2002.

[18] J. C. Hart. Morse theory for implicit surface modeling. In H. Hege and K. Polthier, editors, *Mathematical Visualization*, pages 257–268. Springer-verlag, Oct. 1998.

[19] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3d surface construction algorithm. In *Proc. SIGGRAPH 1987*. ACM Press, 1987.

[20] S. Mccallum and G. E. Collins. Local box adjacency a lgorithms for cylindrical algebraic decompositions. *J. Symbolic Comput.*, 33:321–342, 2002.

[21] B. Mishra. *Algorithmic algebra*. Springer, Berlin, 1993.

[22] B. Mourrain and J. P. T*é*court. Computing the topology of real algebraic surfaces. In *MEGA electronic proceedings*, 2005.

[23] X. L. Ni, M. Garland, and J. C. Hart. Fair morse functions for extracting topological structure of a surface mesh. *ACM Trans. on Graphics*, 23(2):613–622, 2004.

[24] S. Plantinga and G. Vegter. Isotopic implicit surface meshing. In *Proc. Symp. on Geometry Processing*, pages 251–260, Nice, France, 2004.

[25] S. Plantinga and G. Vegter. Isotopic meshing of implicit surfaces. *The Visual Computer*, 23:45–58, 2007.

[26] J. Ploomenthal. Polygonization of implicit surfaces. *Computer Aided Geometric Design*, 5:341–355, 1998.

[27] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *AAECC*, 9:433–461, 1999.

[28] J. M. Snyder. Interval analysis for computer graphics. In *Proc. 19th Annual Conf. on Computers*, pages 121–130, 1992.

[29] B. T. Stander and J. C. Hart. Guaranteeing the topology of an implicit surface polygonization for interactive modeling. In *Proc. SIGGRAPH 97*, pages 279–286. ACM Press, August 1997.