

Article

Level Sets of Weak-Morse Functions for Triangular Mesh Slicing

Daniel Mejia-Parra ^{1,2} , Oscar Ruiz-Salguero ² , Carlos Cadavid ³, Aitor Moreno ^{1,*} 
and Jorge Posada ¹ 

¹ Vicomtech Foundation, Basque Research and Technology Alliance (BRTA), Mikeletegi 57, 20009 San Sebastián, Spain; dmejia@vicomtech.org (D.M.-P.); jposada@vicomtech.org (J.P.)

² Laboratory of CAD CAM CAE, Universidad EAFIT, Cra 49 no 7-sur-50, Medellín 050022, Colombia; oruiz@eafit.edu.co

³ Matemáticas y Aplicaciones, Departamento de Ciencias Matemáticas, Universidad EAFIT, Medellín 050022, Colombia; ccadavid@eafit.edu.co

* Correspondence: amoreno@vicomtech.org

Received: 31 July 2020; Accepted: 14 September 2020; Published: 19 September 2020



Abstract: In the context of CAD CAM CAE (Computer-Aided Design, Manufacturing and Engineering) and Additive Manufacturing, the computation of level sets of closed 2-manifold triangular meshes (mesh slicing) is relevant for the generation of 3D printing patterns. Current slicing methods rely on the assumption that the function used to compute the level sets satisfies strong Morse conditions, rendering incorrect results when such a function is not a Morse one. To overcome this limitation, this manuscript presents an algorithm for the computation of mesh level sets under the presence of non-Morse degeneracies. To accomplish this, our method defines *weak-Morse* conditions, and presents a characterization of the possible types of degeneracies. This classification relies on the position of vertices, edges and faces in the neighborhood outside of the slicing plane. Finally, our algorithm produces oriented 1-manifold contours. Each contour orientation defines whether it belongs to a hole or to an external border. This definition is central for Additive Manufacturing purposes. We set up tests encompassing all known non-Morse degeneracies. Our algorithm successfully processes every generated case. Ongoing work addresses (a) a theoretical proof of completeness for our algorithm, (b) implementation of interval trees to improve the algorithm efficiency and, (c) integration into an Additive Manufacturing framework for industry applications.

Keywords: level sets; Morse theory; mesh slicing; additive manufacturing

1. Introduction

In the context of CAD CAM CAE (Computer-Aided Design, Manufacturing and Engineering) and Additive Manufacturing, the computation of closed 2-manifold level sets (mesh slices) is crucial for the generation of contours required in pattern filling. The filling of these patterns require correct classification and orientation of closed 1D contours, allowing the application of filling algorithms that rely on line/polygon boolean operations. The resulting fill patterns are used to generate the G-Code or CLI (Common Layer Interface) code required to 3D print an input model layer by layer, where each layer corresponds to a filled level set [1,2]. Other applications for mesh slicing include visual dimensional inspection, where computed level sets are used to assess 2D geometric features [3,4].

The problem of slicing closed 2-manifold meshes addresses the computation of level sets of a height function $f : \mathcal{M} \rightarrow \mathbb{R}$. The inverse f^{-1} of such a function at a given point c is known as a level set of f . The properties of the level set $f^{-1}(c)$ (i.e., 0-, 1- or 2-manifoldness, or non-manifoldness at all) heavily depend on the critical points of f (in $f^{-1}(c)$).

A critical point of f is a point $x \in \mathcal{M}$ such that its tangent gradient $\nabla_{\mathcal{M}}f(x)$ is 0. In addition, a critical point $x \in \mathcal{M}$ is degenerate, if its tangent Hessian matrix $\mathbf{H}_{\mathcal{M}}f(x)$ is degenerate (i.e., if its matrix determinant is 0).

A function $f : \mathcal{M} \rightarrow \mathbb{R}$ is a Morse (*strong-Morse*) function if all of its critical points are non-degenerate. The assumption of Morseness on f guarantees that the resulting level set f^{-1} is either (see Figure 1): (a) a single (0-manifold) point, (b) a 1-manifold closed contour, (c) a self-intersecting 1D closed contour or, (d) the disconnected union of any of the previously mentioned cases. As a consequence, the Morseness of f facilitates the computation of $f^{-1}(c)$ as only closed oriented contours are generated (aside from the trivial 0-manifold points).

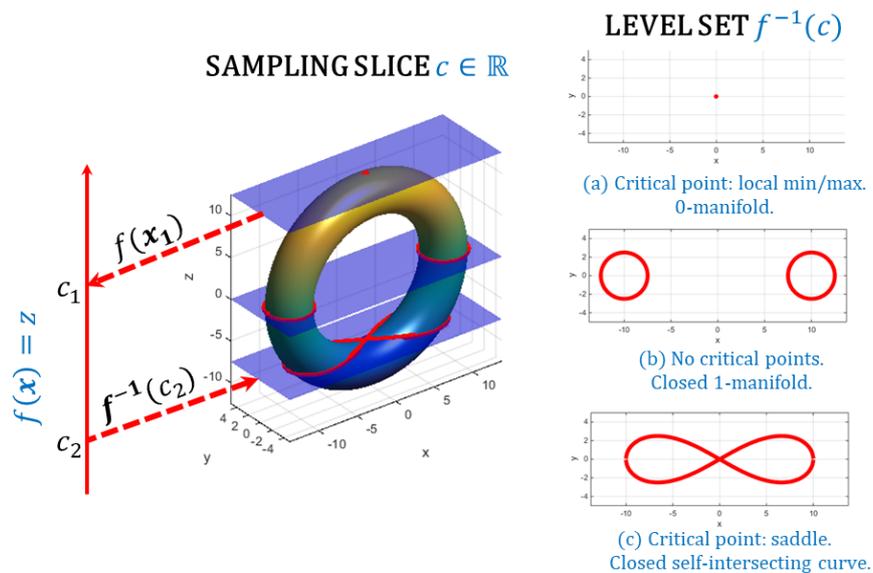


Figure 1. Type of level sets f^{-1} for a Morse function f . f is a slicing function defined on a closed 2-manifold $\mathcal{M} \subset \mathbb{R}^3$.

However, in most application cases f is a non-Morse function, with several degeneracies along the domain \mathcal{M} . Therefore, $f^{-1}(c)$ can produce other types of level sets, such as: non-oriented compact contours, 2-manifold regions, self-intersecting contours (with or without border) or a connected union between Morse and non-Morse cases. It is well-known that a slight rotation of the manifold may remove non-Morse conditions of the height (slicing) function. However, in CAD CAM CAE (and particularly in Additive Manufacturing), these non-Morse degeneracies are of special interest as any tilt to the original model changes the manufacturing feasibility of the workpiece.

In the current literature, most mesh slicing algorithms assume *strong-Morseness* on the slicing function f , or they do not correctly classify the different types of non-Morse cases that arise in real application cases. To organize the solution of this problem, this manuscript defines *weak-Morse* functions as functions with a finite set of points $C \subset \mathbb{R}$, such that f is only degenerate (non-Morse) at such points $f^{-1}(C) \subset \mathcal{M}$, satisfying Morseness elsewhere. Such a definition allows us to define a classification of different Morse and non-Morse cases in order to develop an algorithm that correctly reflects this classification, producing the correct level sets for each case. Such classification requires our algorithm the evaluation of each vertex, edge and triangle neighborhood, outside of the slicing plane c . Finally, closed contours are correctly oriented to differentiate body holes from external contours. This final step is crucial in Additive Manufacturing operations where pattern filling algorithms require this differentiation for correct line-polygon intersection operations.

The remainder of this manuscript is organized as follows: Section 2 discusses the relevant literature. Section 3 presents the proposed *weak-Morse* classification and the mesh slicing algorithm. Section 4 presents and discusses some test results. Finally, Section 5 presents the conclusions and introduces what remains for future work.

2. Literature Review

Mesh slicing and pattern filling are crucial steps in the Additive Manufacturing workflow. This section discusses some of the most relevant literature concerning the computation of 2-manifold level sets. For a more detailed review, the reader may refer to the surveys in References [1,2].

2.1. Parallel-Planes Mesh Slicing

Height functions $f(x) = x_z$ are the most common functions used for mesh slicing. These functions naturally produce parallel planar slices of M in the z -direction, and they can be generalized to any direction $\vec{v} \in \mathbb{R}^3$ by applying a 3D rotation to the input mesh. Standard triangle/plane intersection algorithms are used to compute the level sets of f [5–12], which in most cases are assumed to produce closed (and oriented) 2-manifold contours. Adaptive slicing algorithms introduce techniques to compute non-uniformly spaced slicing planes [13–15]. Adaptive techniques are relevant for avoiding surface aliasing, without having to rely on oversampling.

To reduce the complexity of slicing algorithms, Reference [16] introduces interval trees, which reduces the complexity of mesh slicing algorithms by testing subsets of mesh triangles in given z -intervals. For regularly spaced slicing planes, the complexity can be further reduced by sorting the mesh triangles in ascending z -order [17].

Other methods for surface slicing include point clouds [18], Bezier level sets [19], NURBS [13], implicit surfaces [20,21], Catmull-Clark solids [22] and slicing of graph-based lattice structures [23].

All of the aforementioned slicing algorithms limit their level set retrieval using only triangle/plane intersection information. This relies on the assumption of the slicing function f to be a Morse function. However, this assumption does not always hold true, specially for CAD models where sharp edges, planar faces and saddle points intersect the slicing plane. The non-Morseness of f requires additional evaluation of local vertex, edge and triangle neighborhoods, which do not necessarily lie on the slicing plane but are crucial for determining the type of slice degeneracy. Reference [24] addresses the problem of some non-Morse slicing functions by considering degenerate 2-manifold faces lying on the slicing plane. However, degenerate 1-manifold regions and saddle points are still not considered.

2.2. Non-Parallel-Planes Mesh Slicing

As discussed before, most slicing algorithms consider height maps as slicing functions. However, some approaches have been recently proposed to include other kind of slicing functions [1,2]. These functions vary the slicing direction through the surface, producing non-parallel slicing planes.

References [25,26] apply an error minimization algorithm, which segments the mesh model and produces a different slicing direction for each computed sub-mesh. Reference [27] maps surface normals to the unit sphere and solves an optimization problem inside this sphere. The optimization procedure results in a slicing function that produces non-parallel slicing planes.

Reference [28] defines a slicing function that travels along a parametric curve (which the authors name *streamline*). References [29,30] fit B-Spline surfaces to the mesh. Many non-planar slices of M are produced afterwards by offsetting the fitted B-Splines in the surface normal direction.

In terms of Morseness of the slicing function f , non-parallel-planes methods also arise the same problems of their parallel-plane methods counterparts. Computation of level sets still happens at a local triangle/plane intersection level, neglecting the information of local vertex, edge and triangle neighborhoods.

2.3. Conclusions of the Literature Review

Current mesh slicing algorithms rely on the assumption of Morseness of the slicing function f . However, non-Morse slicing functions are frequent in Computational Geometry applications and (particularly) in CAD applications, where input models present saddle points, sharp edges and planar faces that intersect the slicing plane. These non-Morse cases, produce incorrect slicing results

when not correctly addressed. Furthermore, literature on discrete Morse theory has been already discussed by different authors [31–33]. Nevertheless, to the best of our knowledge, there is not a Morse-based mathematical description of the mesh slicing problem in the context of Engineering and Manufacturing applications.

To overcome these problems, this manuscript discusses and classifies the different types of plane/surface intersections that arise when computing level sets. To do so, we define *weak-Morse* functions, in which non-Morse degeneracies are allowed in a finite set of planar slices $C \subset \mathbb{R}$. Additional to the classic plane/triangle intersection algorithm, this classification requires our algorithm to evaluate vertex, edge, and triangle neighborhoods outside of the slicing plane. The resulting slices may produce not only closed 1-manifold contours, but also non-manifold (with or without boundary) self-intersecting contours, 0-manifold points and 2-manifold surfaces. In order to differentiate holes from external contours, the resulting closed contours are correctly oriented. The orientation of the closed contours is crucial in Additive Manufacturing operations where pattern filling algorithms require this specification for correct line/polygon boolean operations. Future work is intended to produce a theoretical proof of the completeness of the algorithm presented in this manuscript.

3. Methodology

The idea of mesh slicing considers a triangular discretization $M = (X, T)$ of a continuous, closed 2-manifold \mathcal{M} that is embedded in \mathbb{R}^3 . M is a triangular mesh defined by a set of vertices $X = \{x_0, x_1, \dots, x_n\}$ and a set of triangles $T = \{t_0, t_1, \dots, t_m\}$. The slicing of M requires to compute the level sets f^{-1} of a *slicing function* $f : \mathcal{M} \rightarrow \mathbb{R}$.

For a given slicing value $c \in \mathbb{R}$, the level set $f^{-1}(c) \subset \mathcal{M}$ produces an oriented 1-manifold contour if such a level set does not contain any critical points of f . However, when $f^{-1}(c)$ contains one (or more) critical points of M , the resulting level set cannot be guaranteed to be 1-manifold closed contour. In fact, several cases arise depending on the nature of the critical point, and the resulting contour can either be: (a) a 0-manifold point, (b) a self-intersecting (non-manifold), closed, 1D oriented contour, (c) a non-manifold, 1D non-oriented contour (possibly with borders), and (d) a closed 2-manifold region. The nature of these critical points can be classified in terms of Morse theory, as illustrated in Figure 2. The mathematical and algorithmic details for the computation of the level sets of f are discussed in the following subsections.

3.1. Mathematical Description of the Problem

3.1.1. Strong-Morse Function

Let \mathcal{M} be a closed and oriented 2-manifold embedded in \mathbb{R}^3 , and consider a twice differentiable function $f : \mathcal{M} \rightarrow \mathbb{R}$. The function f , is a *strong-Morse* function if all critical points of f are non-degenerate. That is to say, f is a Morse function if:

$$\forall x \in \mathcal{M}, \quad \nabla_{\mathcal{M}} f(x) \neq 0 \vee \det(\mathbf{H}_{\mathcal{M}} f(x)) \neq 0, \tag{1}$$

where $\nabla_{\mathcal{M}}$ and $\mathbf{H}_{\mathcal{M}}$ are the tangent gradient operator and tangent 2×2 Hessian matrix, respectively. These tangent operators are defined in the tangent planes $T_x \mathcal{M}$ of \mathcal{M} . The function f is known as the *slicing function*, and the inverse $f^{-1} : \mathbb{R} \rightarrow \mathcal{P}(\mathcal{M})$ defines the level sets of f .

3.1.2. Weak-Morse Function

We define a *weak-Morse* function, as a function f that satisfies the Morse criterion everywhere, except in a finite set of slice values $C \subset \mathbb{R}$, that is:

$$\forall c \in C, \exists x \in f^{-1}(c), \quad \nabla_{\mathcal{M}} f(x) = 0 \wedge \det(\mathbf{H}_{\mathcal{M}} f(x)) = 0. \tag{2}$$

Since the set C is allowed to be the empty set, any *strong-Morse* function is also a *weak-Morse* one.

It is worth noting that in the discrete domain (e.g., in the case of triangle meshes), *weak-Morse* and non-Morse are equivalent definitions. However, in the continuous domain these two definitions are not the same. As an example, consider the surface $S \subset \mathcal{M}$, defined as $x, y \in (-a, a)$ (with $a \in \mathbb{R}$), and the non-Morse function $z = f(x, y)$ defined as follows (see Figure 3):

$$f(x, y) = \begin{cases} x^5 \cos\left(\frac{1}{x}\right) & x \neq 0 \\ 0 & x = 0. \end{cases} \tag{3}$$

	Continuous case	Discrete case	3D Slicing	2D Level Set
Morse level sets: $\nabla_{\mathcal{M}} f(x) \neq 0 \vee \det(\mathbf{H}_{\mathcal{M}} f(x)) \neq 0$	Non-critical point $x \in \mathcal{M}$: $f^{-1}(c)$: • 1-manifold • Oriented	Face/plane intersection: $f^{-1}(c), t_k \in T$: • p_0 : edge intersection $e_i \in t_k$ • p_f : edge intersection $e_j \in t_k$		
	Local min (or max) $x \in \mathcal{M}$: $f^{-1}(c)$: • 0-manifold	Point/plane intersection: $f^{-1}(c), x_i \in X, x_j \in N_X^T(x_i)$: • $x_i^z = c$ • $x_j^z > c$ (< for local max)		
	Non-degenerate saddle $x \in \mathcal{M}$: $f^{-1}(c)$: • 1D non-manifold • Oriented, self-intersecting	Point/plane, face/plane intersection: $f^{-1}(c), x_i \in X, N_X^T(x_i)$: • $x_i^z = c$ • Adjacent faces $N_X^T(x_i)$: $num_intersections = 4$		
Non-Morse level sets: $\nabla_{\mathcal{M}} f(x) = 0 \wedge \det(\mathbf{H}_{\mathcal{M}} f(x)) = 0$	Degenerate saddle $x \in \mathcal{M}$: $f^{-1}(c)$: • 1D non-manifold • Oriented, self-intersecting	Point/plane, face/plane intersection: $f^{-1}(c), x_i \in X, N_X^T(x_i)$: • $x_i^z = c$ • Adjacent faces $N_X^T(x_i)$: $num_intersections = 2m, m > 2$		
	Critical 1D set $E \subset \mathcal{M}$: $f^{-1}(c)$: • 1D non-manifold • Non-Oriented, self-intersecting	Edge/plane intersection: $f^{-1}(c), (x_i, x_j) \in E$: • $x_i^z = c, x_j^z = c$ • $(x_{opp}^z(x_i, x_j) - c) \cdot (x_{opp}^z(x_j, x_i) - c) > 0$		
	Critical 2D set $F \subset \mathcal{M}$: $f^{-1}(c)$: • 2-manifold	Face/plane intersection: $f^{-1}(c), (x_i, x_j, x_k) \in T$: • $x_i^z = c, x_j^z = c, x_k^z = c$		

Figure 2. Type of level sets f^{-1} for a *weak-Morse* function f . Level sets computed on a discrete 2-manifold $M = (X, T)$.

The function $f \in \mathcal{C}^2(S)$ has infinite critical values and, since for all $x, y \in S, \frac{\partial f^2}{\partial xy} = \frac{\partial f^2}{\partial y^2} = 0$, the Hessian matrix $\mathbf{H}_{\mathcal{M}} f(x, y, z)$ is singular everywhere in S . As a consequence, the non-Morse function f does not satisfy *weak-Morse* conditions on $\mathcal{M} \supset S$.

In particular, this manuscript considers the function f as the height function $f(x) = x_z$, which maps each point $x \in \mathcal{M}$ to its corresponding z -value (see Figure 1). In this case, the level sets of f are parallel planar slices of \mathcal{M} .

3.1.3. Level Set Types

Let $\mathcal{M} \subset \mathbb{R}^3$ be a closed, oriented and differentiable 2-manifold, and the slicing function $f : \mathcal{M} \rightarrow \mathbb{R}, f \in \mathcal{C}^2(\mathcal{M})$. For a given $c \in \mathbb{R}$, this subsection defines a classification of the level set $f^{-1}(c)$ based on the characteristics of the critical points of f (in $f^{-1}(c)$).

Without loss of generality, the following classification assumes that $f^{-1}(c)$ is connected. In the case that $f^{-1}(c)$ is disconnected, each connected component of such a level set can be treated separately.

Furthermore, such a classification is applied locally (i.e., in a small neighborhood $N_\epsilon(x) \subset \mathcal{M}$ around a point $x \in \mathcal{M}$), allowing more than one type instances at the same level set $f^{-1}(c)$.

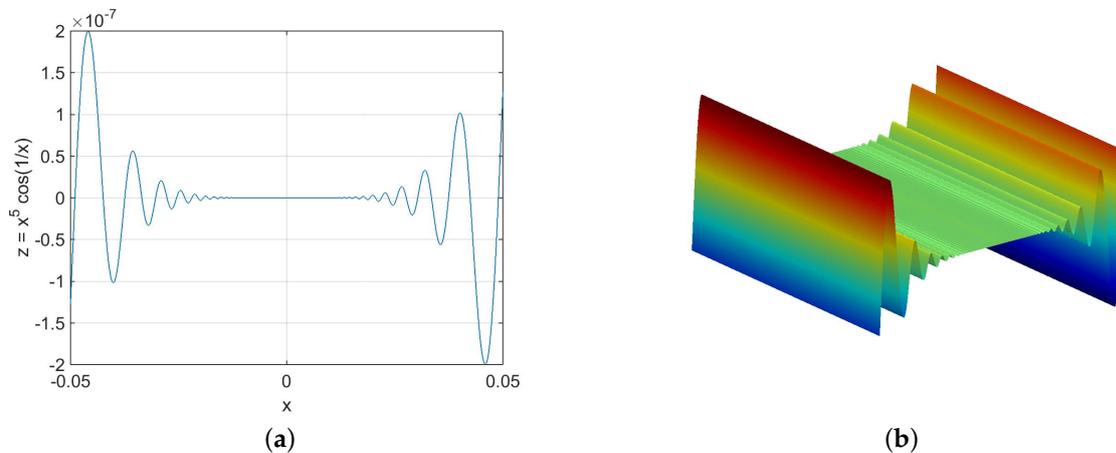


Figure 3. The surface $z(x, y) = x^5 \cos(\frac{1}{x})$. An example of a non-Morse function that is not a *weak-Morse* one in a bounded region $S \subset \mathbb{R}^3$. **(a)** The function $z(x) = x^5 \cos(\frac{1}{x})$ has infinite critical points in the interval $(-a, a)$; **(b)** Surface $z(x, y) = x^5 \cos(\frac{1}{x})$. The Hessian of $z(x, y)$ is singular everywhere in S .

Without further ado, we identify the following level set types (see Figure 2):

1. **Closed 1-manifold contour:** $f^{-1}(c)$ contains no critical points:

$$0 \notin \nabla_{\mathcal{M}} f(f^{-1}(c))$$

In this case $f^{-1}(c)$ defines a simple, oriented, closed curve (1-manifold homeomorphic to the unit circle).

2. **0-manifold point:** $f^{-1}(c)$ contains a strict local minimum (or maximum) $f(x) = c$:

$$\exists \epsilon > 0, \forall y \in N_\epsilon(x) \setminus x, \quad f(x) < f(y)$$

where, $N_\epsilon(x) \subset \mathcal{M}$ is a small neighborhood around the point x . Notice that for a local minimum (or maximum), $f^{-1}(c) \cap N_\epsilon(x) = \{x\}$. The level set $f^{-1}(c)$ corresponds to a 0-manifold isolated point $x \in \mathcal{M}$.

3. **Non-degenerate saddle:** $f^{-1}(c)$ contains a non-degenerate saddle point $f(x) = c$:

$$\nabla_{\mathcal{M}} f(x) = 0 \wedge \det(\mathbf{H}_{\mathcal{M}} f(x)) \neq 0,$$

in which, for a small neighborhood, $N_\epsilon(x) \subset \mathcal{M}$:

$$f^{-1}(c) \cap N_\epsilon(x) \setminus \{x\} \text{ is homeomorphic to } \bigcup_{i=0}^3 (i, i + 1).$$

This contour locally produces a self intersecting (non-manifold), 1D curve with the shape of an "X" (see Figure 2).

4. **Degenerate saddle:** $f^{-1}(c)$ contains degenerate saddle point $f(x) = c$:

$$\nabla_{\mathcal{M}} f(x) = 0 \wedge \det(\mathbf{H}_{\mathcal{M}} f(x)) = 0$$

then for a small neighborhood $N_\epsilon(x) \subset \mathcal{M}$:

$$f^{-1}(c) \cap N_\varepsilon(x) \setminus \{x\} \text{ is homeomorphic to } \bigcup_{i=0}^{2k-1} (i, i+1), \quad k > 2.$$

Similar to the non-degenerate saddle, the degenerate saddle produces a self-intersecting curve with $2k$ ($k > 2$) 1-manifold branches (see Figure 2).

- Degenerate 1D region:** $f^{-1}(c)$ contains a local minimum (or maximum) along a 1D path:

$$\exists x \in f^{-1}(c), \exists \varepsilon > 0, \quad c = \min(f(N_\varepsilon(x)))$$

and

$$f^{-1}(c) \cap N_\varepsilon(x) \text{ is homeomorphic to } \mathbb{R}.$$

This region defines a non-manifold 1D compact region, possibly with non-empty boundaries.

- Degenerate 2-Manifold region:** $f^{-1}(c)$ contains a critical 2-manifold region:

$$\exists x \in f^{-1}(c), \exists \varepsilon > 0, \forall y \in N_\varepsilon(x) \quad f(y) = c.$$

The region $N_\varepsilon(x) \subset f^{-1}(c)$ is locally homeomorphic to \mathbb{R}^2 . The definition of locally homeomorphic permits the existence of holes in the region $f^{-1}(c)$.

3.1.4. Contours Orientation

Since the slicing function is the height function $f(x) = x_z$, each contour $f^{-1}(c)$ naturally lies in a 2D linear space (i.e., planar slice). The orientation of each contour is defined as follows (see Figure 2):

- Closed 1-manifold:** The orientation of $f^{-1}(c)$ on the slicing plane is defined by the projection of the surface normal vector, that is:

$$\vec{n}_{XY}(x) = \text{Proj}_{XY}(\vec{n}(x)),$$

where $\vec{n}(x) \in \mathbb{R}^3$ is the vector normal to \mathcal{M} at x , and $\vec{n}_{XY}(x) \in \mathbb{R}^2$ is the corresponding projection on the slicing plane.

- 0-manifold point:** Not oriented.
- Non-degenerate saddle:** The orientation of the non-degenerate saddle point $f(x) = c$ is equally defined by the projection to the surface normal vector:

$$\vec{n}_{XY}(y) = \text{Proj}_{XY}(\vec{n}(y))$$

for all $y \in N_\varepsilon(x) \setminus \{x\}$.

- Degenerate saddle:** Exactly the same as the non-degenerate saddle case.
- Degenerate 1D region:** Not oriented.
- Degenerate 2-manifold region:** Not oriented in the slicing plane. However, the boundary $\partial f^{-1}(c)$ is an oriented, closed 1-manifold embedded in \mathbb{R}^2 .

3.2. Slicing Algorithm

Let $M = (X, T)$ be a triangular 2-manifold mesh with set of vertices $X = \{x_0, x_1, \dots, x_n\}$, and set of triangles $T = \{t_1, t_2, \dots, t_m\}$. M is a discrete piecewise linear representation of \mathcal{M} . The slicing function $f(x) = z(x)$ is therefore defined on M as $\mathbf{f} = \{z(x_0), z(x_1), \dots, z(x_n)\}$.

For a given $c \in \mathbb{R}$, a level set (planar slice) $f^{-1}(c) = (S_O, P, S_{NO}, F)$ of M is composed by the following discrete sets:

- $S_O = \{s_0, s_1, \dots\} \subset \mathbb{R}^2$ is the set of all closed and oriented polylines on the slicing plane. The set S_O includes closed 1-manifolds and saddle points with its adjacent neighborhoods

2. $P = \{p_0, p_1, \dots\} \subset X$ is the 0-manifold set of all local minima/maxima vertices in M that lie on the slicing plane.
3. $S_{NO} = \{e_0, e_1, \dots\} \subset E$ is the set of all non-oriented (and possibly with border) contours on the slicing. S_{NO} includes all degenerate 1D regions of M .
4. $F = \{f_0, f_1, \dots\} \subset T$ is the set of all 2-manifold regions (triangles) in M that lie on the slicing plane.

The different types of level sets $f^{-1}(c)$ on M are defined as follows (see Figure 2):

1. **Degenerate 2-manifold region:** Let $t = (x_i, x_j, x_k) \in T$. $t \in F$ if:

$$z(x_i) = c, \quad z(x_j) = c, \quad z(x_k) = c.$$

2. **Degenerate 1D region:** Let $(x_i, x_j) \in E$, such that $z(x_i) = z(x_j) = c$. $(x_i, x_j) \in S_{NO}$ is a degenerate 1D region if:

$$(x_{opp}^z(x_i, x_j) - c)(x_{opp}^z(x_j, x_i) - c) > 0$$

where $x_{opp}^z(x_i, x_j)$ is the z-coordinate of the vertex that is opposite to the oriented edge (x_i, x_j) .

3. **Degenerate and non-degenerate saddle:** Let $x_i \in X$, $z(x_i) = c$. $x_i \in S_O$ is a saddle point if:

$$|A(x_i)| = 2m, m > 1 \quad \text{with} \quad A(x_i) = \{t_j \in N_X^T(x_i) \mid t_j \cap f^{-1}(c) \neq \emptyset\},$$

where $N_X^T : X \rightarrow \mathcal{P}(T)$ maps each vertex x_i to the set of all triangles t_j that are incident to it. Therefore, $A(x_i)$ is the set of all triangles that intersect the slicing plane and $|A(x_i)|$ is the number of segments in the triangle/plane intersections in such a set.

4. **0-manifold point:** Let $x_i \in X$, $z(x_i) = c$. $x_i \in P$ is a local minimum (or maximum) if:

$$(\forall x_j \in N_X^X(x_i), f(x_j) - c < 0) \quad \vee \quad (\forall x_j \in N_X^X(x_i), f(x_j) - c > 0),$$

where $N_X^X : X \rightarrow \mathcal{P}(X)$ maps each vertex $x_i \in X$ to the set of all vertices $x_j \in X$ that are adjacent to it.

5. **Closed 1-manifold:** Let $t \in T$, $t \cap f^{-1}(c) \neq \emptyset$. The edge $(p_1, p_2) \in S_O$ if:

$$p_i = \alpha x_j + (1 - \alpha)x_k, \quad (x_j, x_k) \cap f^{-1}(c) \neq \emptyset$$

with $0 \leq \alpha \leq 1$ and $(x_j, x_k) \in E$ being one of the triangle t edges that intersect the plane $z(x) = c$. It is worth noting that $p_i \in \mathbb{R}^3$ not necessarily belongs to the vertex set X .

For the discrete case, the list is defined in reverse order, considering the most degenerate (non-Morse) cases first and the non-degenerate (Morse) cases at the end. The given order defines the priority of the slicing Algorithm when finding *weak-Morse* degeneracies.

Algorithm 1 presents the slicing algorithm according to the presented classification, as follows: (1) The computation of the degenerate faces is performed in lines 1–6. (2) The computation of degenerate 1D edges is performed in lines 7–15, where $x_{opp}(x_i, x_j)$ is the vertex that is opposite to the oriented edge (x_i, x_j) . (3) The computation of critical 0-manifold points is performed in lines 16–30, where each vertex is tested as a local minimum (or maximum). (4) Finally, lines 31–40 perform the computation of closed and oriented contours, including degenerate and non-degenerate saddle points. Line 33 checks for triangle/plane intersection, and line 34 computes explicitly such an intersection. Finally, line 35 compares the orientation between the computed edge (p_0, p_f) and the current triangle $t \in T$, swapping the edge orientation if required at line 36.

The orientation of the closed 1D contours (in \mathbb{R}^2) is defined such that a counter-clockwise polyline represents an external contour, while a clockwise polyline represents an internal hole.

Algorithm 1 Slice triangular mesh.

Require: $M = (X, T), c \in \mathbb{R}$
Ensure: $S = (S_O, P, S_{NO}, F)$

- 1: **for** $t \in T$ **do**
- 2: $(x_i, x_j, x_k) \leftarrow t$
- 3: **if** $z(x_i) = c \wedge z(x_j) = c \wedge z(x_k) = c$ **then**
- 4: $\text{push}(F, t)$
- 5: **end if**
- 6: **end for**
- 7: **for** $(x_i, x_j) \in E$ **do**
- 8: **if** $z(x_i) = c \wedge z(x_j) = c$ **then**
- 9: $a \leftarrow x_{opp}(x_i, x_j)$
- 10: $b \leftarrow x_{opp}(x_j, x_i)$
- 11: **if** $(z(a) - c)(z(b) - c) > 0$ **then**
- 12: $\text{push}(S_{NO}, (x_i, x_j))$
- 13: **end if**
- 14: **end if**
- 15: **end for**
- 16: **for** $x_i \in X$ **do**
- 17: $\text{is_min} \leftarrow \text{TRUE}$
- 18: $\text{is_max} \leftarrow \text{TRUE}$
- 19: **for** $x_j \in N_X^X(x_i)$ **do**
- 20: **if** $z(x_j) - c \geq 0$ **then**
- 21: $\text{is_max} \leftarrow \text{FALSE}$
- 22: **end if**
- 23: **if** $z(x_j) - c \leq 0$ **then**
- 24: $\text{is_min} \leftarrow \text{FALSE}$
- 25: **end if**
- 26: **end for**
- 27: **if** $\text{is_min} \vee \text{is_max}$ **then**
- 28: $\text{push}(P, x_i)$
- 29: **end if**
- 30: **end for**
- 31: **for** $t \in T$ **do**
- 32: $(x_i, x_j, x_k) \leftarrow t$
- 33: **if** $\max(z(x_i), z(x_j), z(x_k)) \geq c \wedge \min(z(x_i), z(x_j), z(x_k)) \leq c$ **then**
- 34: $(p_0, p_f) \leftarrow \text{intersection}(t, \text{plane}_{z(x)=c})$
- 35: **if** $(\vec{n}_{XY}(p_0, p_f)) \cdot \text{Proj}_{XY}(\vec{n}(t)) < 0$ **then**
- 36: $\text{swap}(p_0, p_f)$
- 37: **end if**
- 38: $\text{push}(S_O, (p_0, p_f))$
- 39: **end if**
- 40: **end for**
- 41: **return** $S = (S_O, P, S_{NO}, F)$

4. Results

This section presents graphical results for the resulting slices of different closed 2-manifold meshes. These test datasets have been generated explicitly to show different *weak-Morse* degeneracies that usually arise in CAD CAM CAE applications. On the one hand, Section 4.1 presents the slicing results on different *weak-Morse* datasets. On the other hand, Section 4.2 presents an application case in the context of Additive Manufacturing.

4.1. Weak-Morse Slicing

This section presents the slicing results for different *weak-Morse datasets*. As already stated above, these simple datasets have been manually designed and generated in order to provoke and illustrate all the different *weak-Morse* cases that have been already discussed in previous sections. In addition, benchmark datasets have been used to illustrate the application of our implementation. Figure 4 presents the slicing results for the *Upright Torus* dataset already introduced in Figure 1. In this case, the function f is a *strong-Morse* one, containing a non-degenerate saddle point on the slicing plane. The resulting slices are the oriented contours S_O . Note that despite the self-intersection in Figure 4b, our algorithm correctly computes the orientation of the contour.

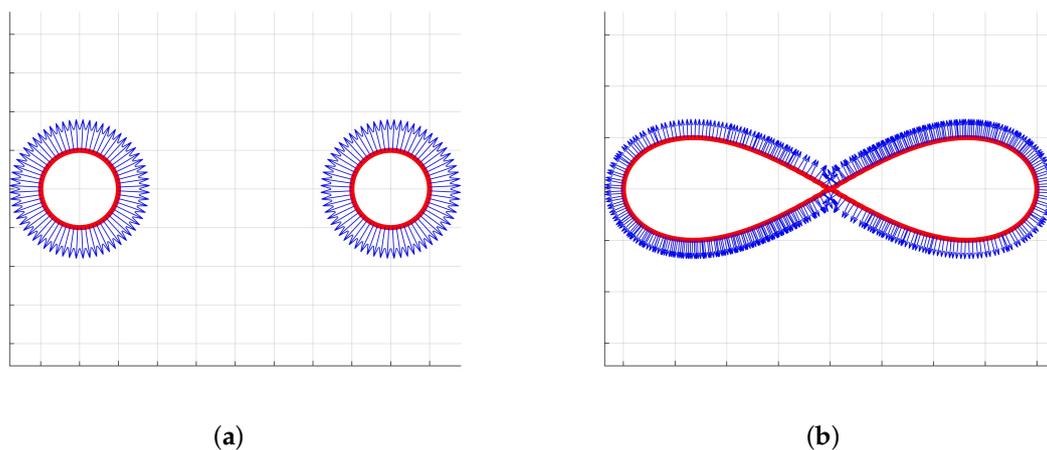


Figure 4. *Upright Torus* dataset (Figure 1). Morse level sets S_O computed with our slicing algorithm. (a) Oriented contours with no critical points; (b) Oriented contour with non-degenerate saddle point.

Figure 5 shows the *Double Torus* dataset. This dataset presents a degenerate, non-Morse saddle point ($m = 4$, see Figure 2). Similar to the *Upright Torus* dataset, the resulting level set is an oriented polyline that self-intersects $2m - 1$ times at such saddle point. The orientation of the polyline is always preserved according to the input mesh M orientation.

Figure 6 illustrates some cases where degenerate 1D regions arise. On the one hand, Figure 6a plots a *Lying Torus* and the planar slice at a non-Morse (degenerate) region. Figure 6b plots the corresponding S_{NO} 1D region. Despite of being closed and non-self intersecting, S_{NO} has no orientation. This fact is consistent with the mesh M , as S_{NO} has no internal or external region. As a contrast, Figure 6c plots the resulting level set computed using a naive *strong-Morse* approach. The resulting level set is filled to illustrate the fact that a naive approach assumes oriented contours and does not discriminate the possible degeneracies. This filling is relevant since in Additive Manufacturing, the fabricated workpiece would end (incorrectly) with such a cover at the top.

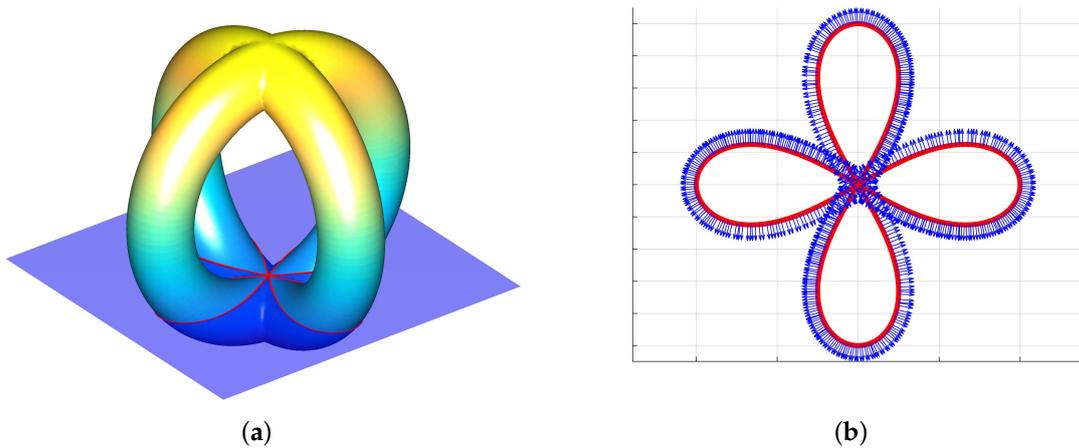


Figure 5. Double Torus dataset. Computation of non-Morse 1D level set S_O . (a) Double Torus dataset; (b) Oriented contour with degenerate saddle point.

On the other hand, Figure 6d plots a *Grid* dataset. The resulting level set $f^{-1}(c)$ at the top of the *Grid* is a non-Manifold 1D region S_{NO} (see Figure 6e). In this case, S_{NO} is not closed (i.e., it has borders), and since it is non-manifold, our slicing algorithm S_{NO} into 1-manifold subset. Again, this 1D contour is non-oriented as there is no external or internal 2D region associated to the contour. In addition, Figure 6f plots the resulting level set computed by the naive *strong-Morse* approach. Again, the result is filled illustrating the orientated polyline produced by the algorithm and 3D printing of such a workpiece would end with a top cover.

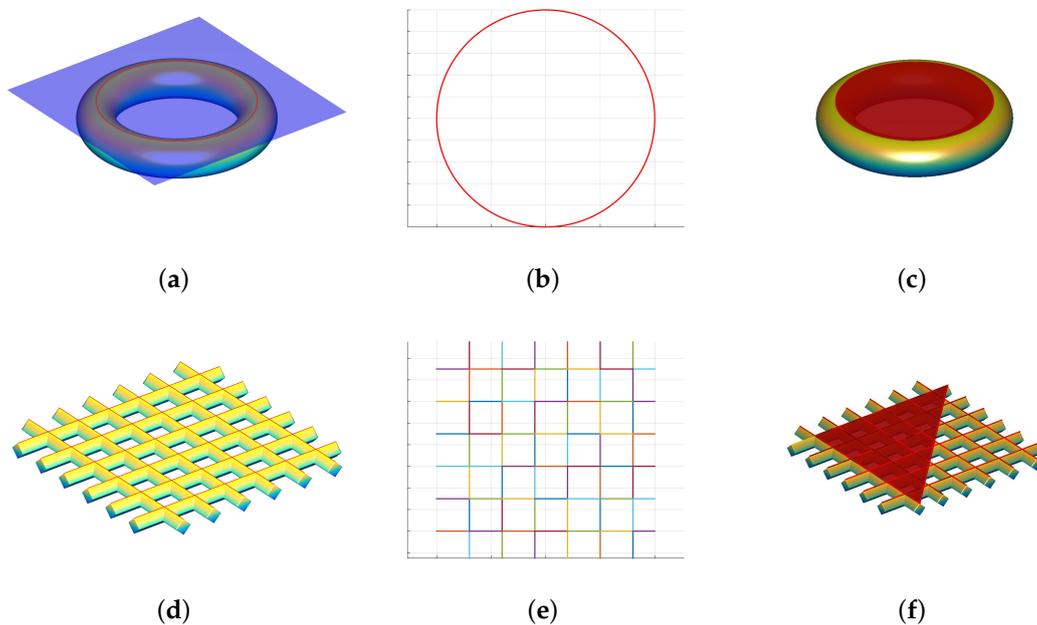


Figure 6. First column: Degenerate datasets *Lyjng Torus* and *Grid* used as input. Second column: correct level sets computed by our algorithm. Third column: Incorrect results produced by naive Morse slicer. (a) *Lyjng Torus* dataset; (b) Correct uppermost level set of *Lyjng Torus*, being a 0-width annulus S_{NO} ; (c) Incorrect uppermost level set of *Lyjng Torus*, being a full disk; (d) Correct uppermost level set of *Grid*, being a set S_{NO} of non-oriented polylines with borders; (e) *Grid* dataset; (f) Incorrect uppermost level set of *Grid*.

Figure 7 plots a *Cylinder* dataset with a slicing occurring at the *Cylinder* top. The resulting level set is a 2-manifold region F composed by the set of all mesh triangles that intersect the slicing plane. A contour can be extracted from F by computing all of its boundary edges and its orientation can be computed from the information of the triangles incident to such boundary (as illustrated in Figure 7b).

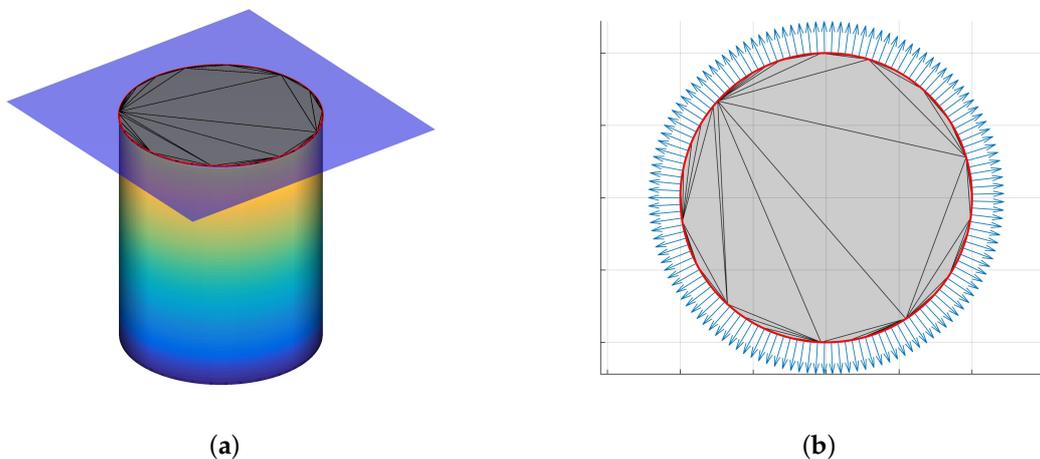


Figure 7. *Cylinder* dataset. Computation of a 2-manifold level set F and its oriented boundary contour $\partial f^{-1}(c)$. (a) *Cylinder* dataset; (b) 2-manifold region F and its oriented contour $\partial f^{-1}(c)$.

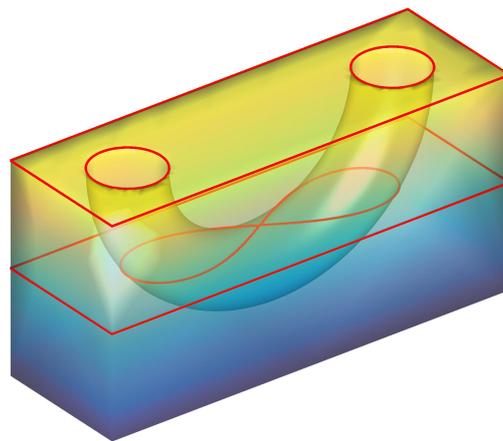
Figure 8 presents the *Internal Torus* dataset. This dataset is generated by a box with a torus hole in the inside. The torus hole naturally produces a saddle point of an internal (oriented) contour hole S_O , which is correctly computed by our slicing algorithm (see Figure 8a). Furthermore, our algorithm is capable of differentiating internal holes from external contours in 2-manifold regions F , as illustrated in Figure 8b.

As already mentioned in the previous section, the current *weak-Morse* classifications works locally. As a consequence, it is not uncommon (specially in CAD datasets) to find multiple *weak-Morse* degeneracies of different types on the same level set $f^{-1}(c)$. Furthermore, these degeneracies can be connected. As an example, the *Peg Cube* dataset (Figure 9a) presents an oriented contour S_O extracted from a region with no critical points, connected to a non-oriented contour S_{NO} extracted from a degenerate 1D critical region, both from the same connected level set (see Figure 9b).

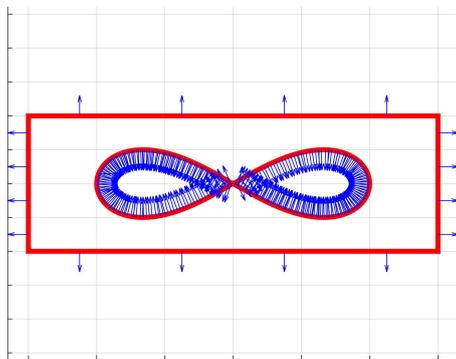
Figure 9c plots the *Chamfered Cylinder* dataset. In this case, the level set contains an oriented contour S_O from a non-degenerate region and a face F extracted from a degenerate 2-manifold region (see Figure 9d. It is worth nothing that both contours S_O and ∂F share a polyline section, each with its own orientation (which is consistent with the orientation of its owner polyline). This guarantees that both contours are closed and oriented consistently, according to the orientation of the input mesh M .

Finally, Figure 9e presents the *Platform Grid* dataset. The level set $f^{-1}(c)$ at the top of this dataset contains both a non-oriented polyline S_{NO} from a degenerate 1D region, connected to a 2-manifold region F (with its respective oriented boundary contour ∂F). The level set results are presented in Figure 9f.

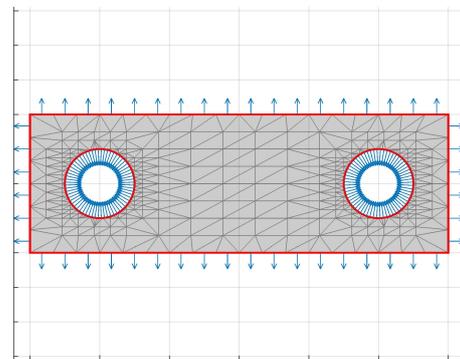
All of these cases are easily managed by our algorithm due to the data structure used $f^{-1}(c) = (S_O, P, S_{NO}, F)$, which handles each type of Morse/non-Morse case separately.



(a)

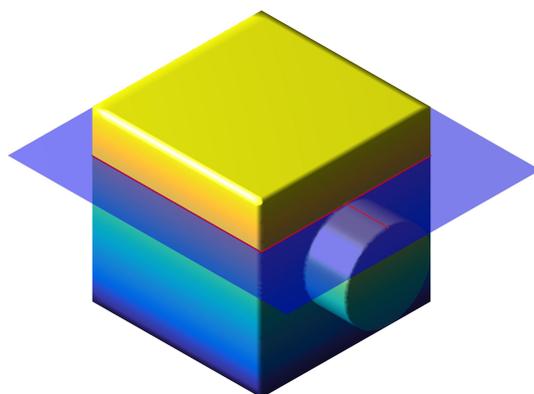


(b)

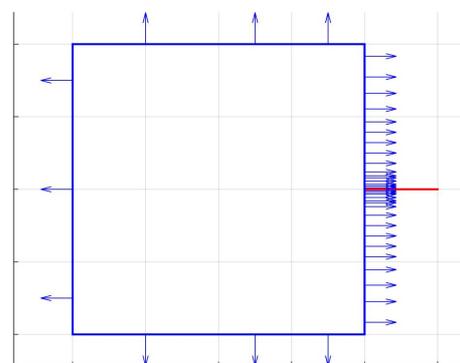


(c)

Figure 8. *Internal Torus* dataset. Our algorithm correctly handles contours, faces with holes, and internal saddle points. (a) *Internal Torus* dataset; (b) Oriented contour S_O for an internal hole with a saddle point; (c) 2-manifold level set F with holes.

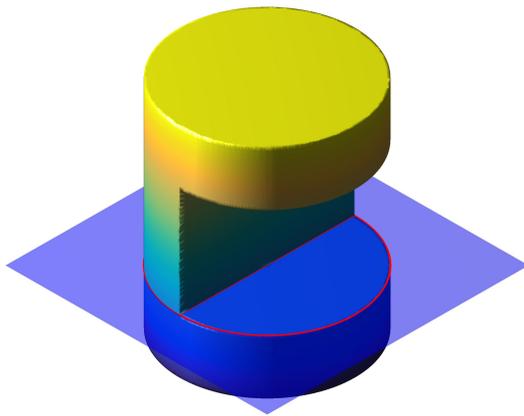


(a) *Peg Cube* dataset

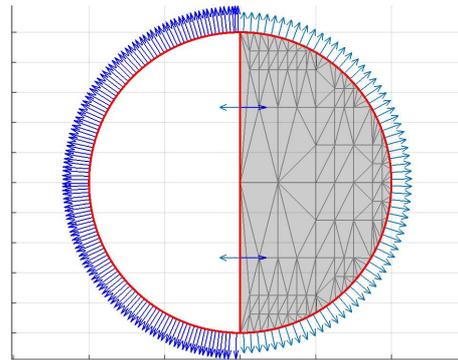


(b) Level set with connected S_O and S_{NO} regions

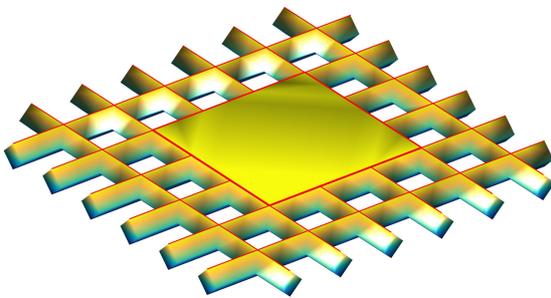
Figure 9. *Cont.*



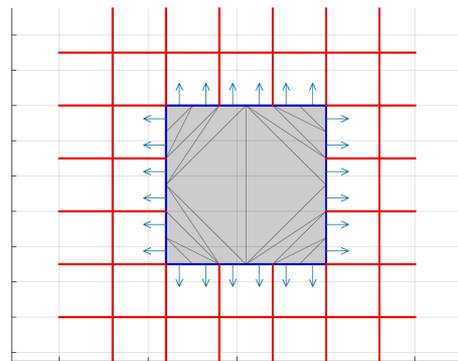
(c) Chamfered Cylinder dataset



(d) Level set with connected S_O and F regions



(e) Grid Platform dataset



(f) Level set with connected S_{NO} and F regions

Figure 9. Level set f^{-1} results for: (a,b), an oriented contour S_O (blue) connected to a non-oriented contour S_{NO} (red), (c,d) an oriented contour S_O connected to a 2-manifold region F and, (e,f) a non-oriented S_{NO} contour connected to a 2-manifold region F .

4.2. Applications in Additive Manufacturing

As already illustrated in the previous subsection, our level set computation algorithm allows to easily back track the normals of \mathcal{M} from each level set $f^{-1}(c)$. Using this information, it is possible to produce a filling pattern for $f^{-1}(M)$. This subsection applies the presented mesh slicing algorithm to real-life CAD datasets. Afterwards, a fill pattern is computed for each level set $f^{-1}(c)$ using a line-polygon intersection algorithm [34].

Figure 10 presents the resulting mesh slices and their corresponding fill patterns. CAD workpiece. The datasets have been extracted from the National Design Repository [35]. The resulting filling patterns can be used to produce 3D printing routines that accurately fabricate the original CAD models, even in the presence of non-Morse level sets. Future work aims to explore this topic in detail.

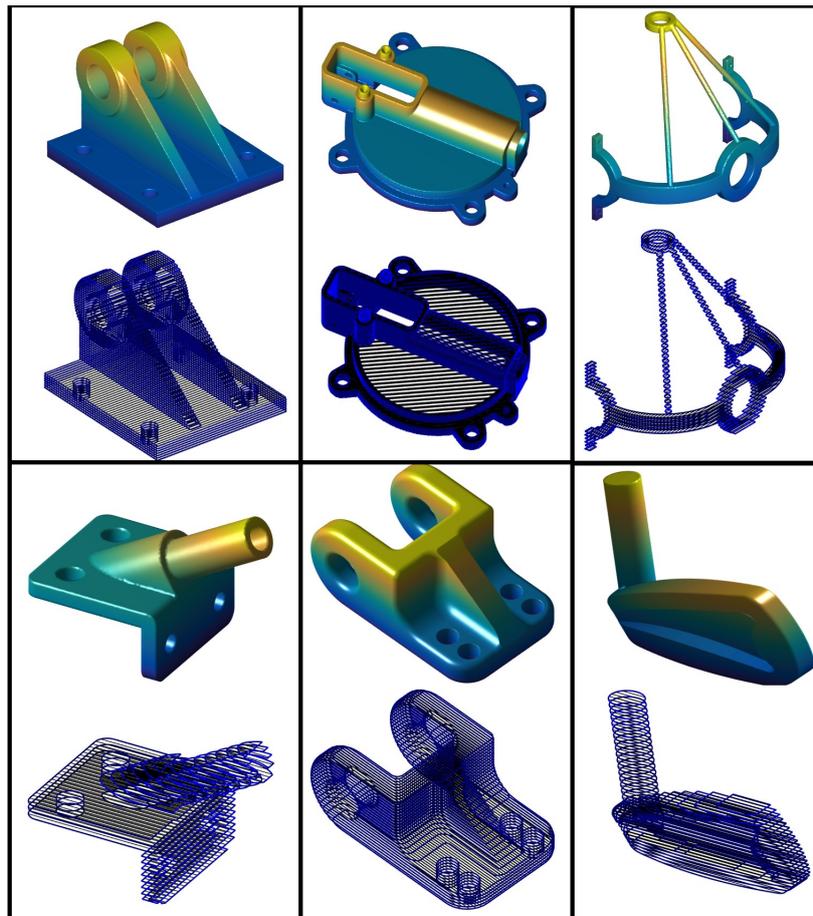


Figure 10. Mesh slicing of benchmark *weak-Morse* CAD datasets. Our algorithm computes the level sets (slices). As a test for our level sets, they are then fed to another algorithm (not part of this manuscript) which correctly generates the tool paths for Additive Manufacturing. Datasets from the National Design Repository [35].

5. Conclusions

This manuscript develops a mathematical and algorithmic classification of the level sets of closed 2-manifold triangular meshes. These level sets are classified in terms of *weak-Morse* functions, where Morse and non-Morse degeneracies arise. This classification relies on the position of vertices, edges and triangles in the neighborhood outside of the slicing plane. The test results show that the presence of such degeneracies on the slicing function f results on level sets that are not only oriented, 1-manifold closed contours, but also self-intersecting (non-manifold) contours, 0-manifold points, non-oriented contours, and even 2-manifold level sets. Our algorithm correctly computes and classifies each resulting contour, and produces its orientation (if it exists) in concordance with the input mesh M . Furthermore, our algorithm correctly handles different degeneracies that arise simultaneously at the same level set. Finally, the presented slicing algorithm enables consistent line-polygon operations (even in *weak-Morse* cases), allowing the computation of filling patterns that are relevant in the context of Additive Manufacturing.

Future work addresses—(a) A theoretical proof of the completeness of the proposed *weak-Morse* classification and the slicing algorithm. Such a proof would guarantee the correctness of every (Morse and non-Morse) level set for any given 2-manifold model. (b) The implementation of the interval tree data structure [16,17] to speed-up the computation times. Such a speed-up could prove valuable for large CAD datasets (i.e., with high triangle count). (c) The integration of the proposed slicing algorithm into an Additive Manufacturing framework, enabling the fabrication of complex models with problematic non-Morse degeneracies.

Author Contributions: Conceptualization, D.M.-P., O.R.-S. and C.C.; Investigation, D.M.-P. and O.R.-S.; Methodology, D.M.-P., O.R.-S., C.C., A.M. and J.P.; Software, D.M.-P.; Supervision, O.R.-S., C.C., A.M. and J.P.; Visualization, D.M.-P., O.R.-S., A.M. and J.P.; Writing—original draft, D.M.-P., O.R.-S., C.C., A.M. and J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

\mathcal{M}	A closed 2-manifold embedded in \mathbb{R}^3 .
f	A scalar function $f : \mathcal{M} \rightarrow \mathbb{R}$. In this manuscript, $f(x) = z(x)$ is the height function, named also as the <i>slicing</i> function.
f^{-1}	Level sets of the function f . $f^{-1} : \mathbb{R} \rightarrow \mathcal{P}(\mathcal{M})$.
$c \in \mathbb{R}$	Scalar that defines the <i>slicing</i> plane. $f^{-1}(c)$ is the slicing plane at $z(x) = c$.
$T_x\mathcal{M}$	For any $x \in \mathcal{M}$, $T_x\mathcal{M}$ is the tangent plane (\vec{e}_1, \vec{e}_2) of \mathcal{M} at x .
$\nabla_{\mathcal{M}}$	Gradient operator defined on the surface (tangent planes) of \mathcal{M} . $\nabla_{\mathcal{M}} = \nabla_{T_x\mathcal{M}}$.
$\mathbf{H}_{\mathcal{M}}$	2×2 Hessian matrix defined on the surface (tangent planes) of \mathcal{M} . $\mathbf{H}_{\mathcal{M}} = \mathbf{H}_{T_x\mathcal{M}}$.
<i>strong-Morse</i>	A <i>strong-Morse</i> (or simply Morse) function f , is a function in which all of its critical points are non-degenerate. A critical point $x \in \mathcal{M}$ is degenerate if $\det(\mathbf{H}_{\mathcal{M}}f(x)) = 0$.
<i>weak-Morse</i>	Denomination used exclusively for the purpose of this manuscript. f is <i>weak-Morse</i> if there exists a finite set of values $C \subset \mathbb{R}$ (each corresponding to a cross cut of \mathcal{M}) where $f(f^{-1}(C))$ is degenerate (i.e., non-Morse). f satisfies Morse criteria elsewhere.
$\vec{n}(x)$	Vector normal to the tangent plane $T_x\mathcal{M}$. $\vec{n} \in \mathbb{R}^3$.
$\vec{n}_{XY}(x)$	Projection of the vector $\vec{n}(x)$ onto the slicing plane. $\vec{n}_{XY}(x) \in \mathbb{R}^2$.
$N_\varepsilon(x)$	Small neighborhood in \mathcal{M} around the point $x \in \mathcal{M}$. $N_\varepsilon(x) = \{y \in \mathcal{M} \mid \ x - y\ < \varepsilon\}$
M	2-manifold triangular mesh $M = (X, T)$, which is a piecewise linear discretization of \mathcal{M} . $X = \{x_0, x_2, \dots, x_n\} \subset \mathcal{M}$ is the set of mesh vertices (geometry), and $T = t_0, t_1, \dots, t_m$ is the set of mesh triangles (topology).
E	Set of bidirectional edges that belong to the graph of the mesh M . $(x_i, x_j) \in E$ if for any k , $(x_i, x_j, x_k) \in T$.
$N_X^X(x_i)$	Set of all vertices $x_j \in X$ that are adjacent to the vertex $x_i \in X$. $N_X^X : X \rightarrow \mathcal{P}(X)$.
$N_X^T(x_i)$	Set of all triangles $t_j \in T$ that are incident to the vertex $x_i \in X$. $N_X^T : X \rightarrow \mathcal{P}(T)$.
\mathbf{f}	Slicing function defined on the triangular mesh M . $\mathbf{f} = \{z(x_0), z(x_1), \dots, z(x_n)\}$.
S_O	Set of all oriented and closed polylines in $f^{-1}(c)$ for a given slice c of M . $S_O = \{s_0, s_1, \dots\} \subset \mathbb{R}^2$.
P	0-manifold set of all local minima/maxima vertices of M that lie on the slicing plane. $P = \{p_0, p_1, \dots\} \subset X$.
S_{NO}	Set of all non-oriented compact (i.e., possibly with borders) polylines in $f^{-1}(c)$ for a given slice c of M . $S_O = \{e_0, e_1, \dots\} \subset E$.
F	Set of all 2-manifold regions (triangles) in M that lie on the slicing plane. $F = \{f_0, f_1, \dots\} \subset T$.

References

- Ding, D.; Pan, Z.; Cuiuri, D.; Li, H.; van Duin, S. Advanced Design for Additive Manufacturing: 3D Slicing and 2D Path Planning. In *New Trends in 3D Printing*; Shishkovsky, I.V., Ed.; IntechOpen: Rijeka, Croatia, 2016; Chapter 1. [\[CrossRef\]](#)
- Zhao, D.; Guo, W. Shape and Performance Controlled Advanced Design for Additive Manufacturing: A Review of Slicing and Path Planning. *J. Manuf. Sci. Eng.* **2019**, *142*. 010801, [\[CrossRef\]](#)
- Mejia-Parra, D.; Sánchez, J.R.; Ruiz-Salguero, O.; Alonso, M.; Izaguirre, A.; Gil, E.; Palomar, J.; Posada, J. In-Line Dimensional Inspection of Warm-Die Forged Revolution Workpieces Using 3D Mesh Reconstruction. *Appl. Sci.* **2019**, *9*. [\[CrossRef\]](#)

4. Sánchez, J.R.; Segura, Á.; Barandiaran, I. Fast and accurate mesh registration applied to in-line dimensional inspection processes. *Int. J. Interact. Des. Manuf.* **2018**, *12*, 877–887. [[CrossRef](#)]
5. Steuben, J.C.; Iliopoulos, A.P.; Michopoulos, J.G. Implicit slicing for functionally tailored additive manufacturing. *Comput. Aided Des.* **2016**, *77*, 107–119. [[CrossRef](#)]
6. Jin, Y.; He, Y.; Fu, G.; Zhang, A.; Du, J. A non-retraction path planning approach for extrusion-based additive manufacturing. *Robot. Comput. Integr. Manuf.* **2017**, *48*, 132–144. [[CrossRef](#)]
7. Jin, Y.; Du, J.; He, Y. Optimization of process planning for reducing material consumption in additive manufacturing. *J. Manuf. Syst.* **2017**, *44*, 65–78. [[CrossRef](#)]
8. Nilsiam, Y.; Sanders, P.; Pearce, J.M. Slicer and process improvements for open-source GMAW-based metal 3-D printing. *Addit. Manuf.* **2017**, *18*, 110–120. [[CrossRef](#)]
9. Budinoff, H.; McMains, S. Prediction and visualization of achievable orientation tolerances for additive manufacturing. In Proceedings of the 15th CIRP Conference on Computer Aided Tolerancing, CIRP CAT, Milan, Italy, 11–13 June 2018; [[CrossRef](#)]
10. Campocasso, S.; Chalvin, M.; Reichler, A.K.; Gerbers, R.; Droder, K.; Hugel, V.; Dietrich, F. A framework for future CAM software dedicated to additive manufacturing by multi-axis deposition. In Proceedings of the 6th CIRP Global Web Conference—Envisaging the Future Manufacturing, Design, Technologies and Systems in Innovation Era (CIRPe2018), 23–25 October 2018. Available online: <http://www.cirpe2018.org/> (accessed on 17 September 2020) [[CrossRef](#)]
11. Michel, F.; Lockett, H.; Ding, J.; Martina, F.; Marinelli, G.; Williams, S. A modular path planning solution for Wire + Arc Additive Manufacturing. *Robot. Comput. Integr. Manuf.* **2019**, *60*, 1–11. [[CrossRef](#)]
12. Roschli, A.; Gaul, K.T.; Boulger, A.M.; Post, B.K.; Chesser, P.C.; Love, L.J.; Blue, F.; Borish, M. Designing for Big Area Additive Manufacturing. *Addit. Manuf.* **2019**, *25*, 275–285. [[CrossRef](#)]
13. Ma, W.; But, W.C.; He, P. NURBS-based adaptive slicing for efficient rapid prototyping. *Comput. Aided Des.* **2004**, *36*, 1309–1325. [[CrossRef](#)]
14. Alexa, M.; Hildebrand, K.; Lefebvre, S. Optimal Discrete Slicing. *ACM Trans. Graph.* **2017**, *36*. [[CrossRef](#)]
15. Mao, H.; Kwok, T.H.; Chen, Y.; Wang, C.C. Adaptive slicing based on efficient profile analysis. *Comput. Aided Des.* **2019**, *107*, 89–101. [[CrossRef](#)]
16. Gregori, R.M.M.H.; Volpato, N.; Minetto, R.; Silva, M.V.G.D. Slicing Triangle Meshes: An Asymptotically Optimal Algorithm. In Proceedings of the 14th International Conference on Computational Science and Its Applications, Guimaraes, Portugal, 30 June–3 July 2014; pp. 252–255.
17. Minetto, R.; Volpato, N.; Stolfi, J.; Gregori, R.M.; da Silva, M.V. An optimal algorithm for 3D triangle mesh slicing. *Comput. Aided Des.* **2017**, *92*, 1–10. [[CrossRef](#)]
18. Xu, J.; Hou, W.; Sun, Y.; Lee, Y.S. PLSP based layered contour generation from point cloud for additive manufacturing. *Robot. Comput. Integr. Manuf.* **2018**, *49*, 1–12. doi:10.1016/j.rcim.2017.05.006. [[CrossRef](#)]
19. Gohari, H.; Barari, A.; Kishawy, H. Using Multistep Methods in Slicing 2 1/2 Dimensional Parametric Surfaces for Additive Manufacturing Applications. In Proceedings of the 2th IFAC Workshop on Intelligent Manufacturing Systems IMS 2016, Austin, TX, USA, 5–7 December 2016. [[CrossRef](#)]
20. Song, Y.; Yang, Z.; Liu, Y.; Deng, J. Function representation based slicer for 3D printing. *Comput. Aided Geom. Des.* **2018**, *62*, 276–293. [[CrossRef](#)]
21. Feng, J.; Fu, J.; Lin, Z.; Shang, C.; Niu, X. Layered infill area generation from triply periodic minimal surfaces for additive manufacturing. *Comput. Aided Des.* **2019**, *107*, 50–63. [[CrossRef](#)]
22. Luu, T.H.; Altenhofen, C.; Ewald, T.; Stork, A.; Fellner, D. Efficient slicing of Catmull-Clark solids for 3D printed objects with functionally graded material. *Comput. Graph.* **2019**, *82*, 295–303. [[CrossRef](#)]
23. Messner, M.C. A fast, efficient direct slicing method for slender member structures. *Addit. Manuf.* **2017**, *18*, 213–220. [[CrossRef](#)]
24. Hu, J. Study on STL-Based Slicing Process for 3D Printing. In Proceedings of the 28th Annual International Solid Freeform Fabrication Symposium—An Additive Manufacturing Conference, Austin, TX, USA, 7–9 August 2017; pp. 885–895.
25. Hildebrand, K.; Bickel, B.; Alexa, M. Orthogonal slicing for additive manufacturing. *Comput. Graph.* **2013**, *37*, 669–675. [[CrossRef](#)]

26. Murtezaoglu, Y.; Plakhotnik, D.; Stautner, M.; Vaneker, T.; van Houten, F.J. Geometry-Based Process Planning for Multi-Axis Support-Free Additive Manufacturing. In Proceedings of the 6th CIRP Global Web Conference—Envisaging the Future Manufacturing, Design, Technologies and Systems in Innovation Era (CIRPe 2018), 23–25 October 2018. Available online: <http://www.cirpe2018.org/> (accessed on 17 September 2020). [[CrossRef](#)]
27. Ding, D.; Pan, Z.; Cuiuri, D.; Li, H.; Larkin, N.; van Duin, S. Automatic multi-direction slicing algorithms for wire based additive manufacturing. *Robot. Comput. Integr. Manuf.* **2016**, *37*, 139–150. [[CrossRef](#)]
28. Ezair, B.; Fuhrmann, S.; Elber, G. Volumetric covering print-paths for additive manufacturing of 3D models. *Comput. Aided Des.* **2018**, *100*, 1–13. [[CrossRef](#)]
29. Jin, Y.; Du, J.; He, Y.; Fu, G. Modeling and process planning for curved layer fused deposition. *Int. J. Adv. Manuf. Technol.* **2017**, *91*, 273–285. [[CrossRef](#)]
30. Zhao, D.; Guo, W. Mixed-layer adaptive slicing for robotic Additive Manufacturing (AM) based on decomposing and regrouping. *J. Intell. Manuf.* **2019**, *31*, 985–1002. [[CrossRef](#)]
31. Banchoff, T.F. Critical Points and Curvature for Embedded Polyhedral Surfaces. *Am. Math. Mon.* **1970**, *77*, 475–485. [[CrossRef](#)]
32. Forman, R. Morse Theory for Cell Complexes. *Adv. Math.* **1998**, *134*, 90–145. [[CrossRef](#)]
33. Fugacci, U.; Landi, C.; Varli, H. Critical sets of PL and discrete Morse theory: A correspondence. *Comput. Graph.* **2020**, *90*, 43–50. [[CrossRef](#)]
34. Vatti, B.R. A Generic Solution to Polygon Clipping. *Commun. ACM* **1992**, *35*, 56–63. [[CrossRef](#)]
35. Regli, W.C.; Foster, C.; Hayes, E.; Ip, C.Y.; McWherter, D.; Peabody, M.; Shapirsteyn, Y.; Zaychik, V. National Design Repository: A Status Report. In Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI) and AAAI/SIGMAN Workshop on AI in Manufacturing Systems, Seattle, WA, USA, 4–10 August 2001.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).