university of groningen / faculty of science and engineering

Design of continuous envelope surfaces with cylindrical tools

Bachelor's Project Mathematics and Computing Science

July 2024

Author: Carolina Aranda Bassegoda
Student Number: S4010779
First supervisor(Mathematics): Marcello Seri
First supervisor(Computer Science): Jiří Kosinka

Abstract

It is increasingly popular to use Computer Numerically Controlled machining to mass-produce objects that require high levels of precision. These machines work through a pre-programmed computer software that dictates the movements of the tools and machinery. Tangent continuity is often needed throughout the surface of the machined object for aerodynamic and aesthetic purposes.

Consequently, in this thesis we derive the necessary and sufficient conditions for tangent continuity between the surfaces generated by two adjacent passes of a tool. We developed an interface that allows the user to visualize the different surfaces that can be obtained when imposing the necessary conditions for tangent continuity. Moreover, it illustrates the geometrical elements that come into play when defining a tool, an envelope and the continuity conditions between two envelopes.

Contents

List of Figures

1 Introduction

Computer Numerically Controlled (CNC) machining is a widely used manufacturing method in the industry. As such, it plays a crucial role in producing precise components across various sectors, including automotive, aerospace, and electronics. This manufacturing method relies on computer software which defines the movements of the machinery.

A big concern in these industries lies on the manufacture of objects with a smooth finish, be it for aerodynamic or aesthetic purposes. To obtain such smoothness, the machined object must be milled with high levels of precision during the finishing stage. Thus it has inspired numerous research in the field of computer aided design with the goal of better approximating free form surfaces fed to the computer software while minimizing computation and manufacturing times. Numerous research has been carried out with the aim of better approximating the surfaces that the industry calls for, see Bartoň et al. (2021); Bo et al. (2016); Bo et al. (, 2017); Rajaina et al. (2023); Skopenkov et al. (2020).

In particular, Bartoň et al. (2021) focuses on tool selection and optimal path planning in approximating free form surfaces. However, not all research follows this school of thought, Bo et al. (2016) deals with non-conventional tool shapes since it also looks to optimize the shape of the milling tool.

Rather than approximating free form surfaces, in this thesis we want to study the continuity conditions that result on such smooth finish and illustrate the elements that come into play, in order to improve our understanding of the freedoms that we have in designing such surfaces. As such, our goal is to define the conditions for tangent continuity between two tool passes and provide the reader with an application that allows them to visualize the concepts discussed. In said application the user should be able to interact with the elements that play a role in defining the continuity constraints that result in a smooth surface.

We will begin by formalizing some fundamental concepts which are essential for the mathematical discussion in this project; see section 2. Once we cover our bases we will go over the mathematical representation of a rotatory tool in section 3 and that of the surface of a milled object in section 4. Using the latter representation, in section 5 we will define and prove the necessary and sufficient conditions for tangent continuity of this kind of surfaces.

We want to make an application to illustrate the concepts discussed in this thesis. As such, one of our concerns when defining the conditions for tangent continuity is that they help us define an algorithm for the tool movement of the second envelope. The design and functionalities of said application will then be discussed in the last section, namely section 6.

2 Preliminaries

We will be working with various geometrical concepts for which we first need to establish rigorous definitions. There exist numerous definitions for these geometrical concepts, some of which are more general or abstract than others. For mathematically rigorous definitions one can refer to DoCarmo (2016); Montiel & Ros (2000); Zhang (2020). Moreover, there are also many books which redefine these concepts by focusing on the characteristics which are useful to real world applications like Computer-Aided Design or various areas of engineering; see Farin (1993); Radzevich (2020).

Since our interest lies in applications to CNC-machining, we will use definitions which help us tackle this problem specifically. For instance, although we can define curves and surfaces in any number of dimensions, we will focus on geometry that lives in three-dimensional space. A more general definition would not add value to this paper since one cannot find such geometries in CNC-machining. First we want to define a curve as a subset of $\mathbb{R}^3$:

Definition 2.1. *A (differentiable) curve is a (differentiable) map, $c: I \to \mathbb{R}^3$, that maps an open interval $I = (a,b) \subseteq \mathbb{R}$ into $\mathbb{R}^3$.*
In other words, c maps $t \in I$ to a point $c(t) = (x(t), y(t), z(t)) \in \mathbb{R}^3$.
The mapping c is called a ***parametrization*** *of the curve. See Figure 1.*

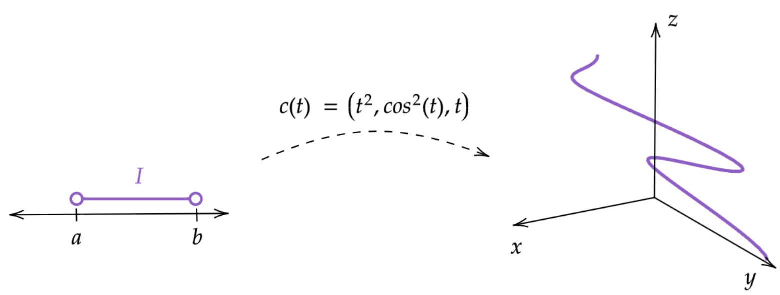

Figure 1: A mapping of an interval (a,b) into $\mathbb{R}^3$

For some examples of a parametrization of a curve, see Figure 2. To better understand the mapping $c(t)$ of a curve as defined above, note that when one of the parameters of the mapping is a constant function, we obtain a planar curve, namely a curve that can be parametrized as a mapping of an interval into $\mathbb{R}^2$.

In Definition 2.1, c being differentiable means that the functions $x, y, z : I \to \mathbb{R}$ are differentiable. Moreover, the vector $c'(t) = (x'(t), y'(t), z'(t))$ is called the tangent vector of the curve at $t \in I$. For the study of curve properties it is essential that such tangent vector exists at every point of the curve, namely, we require $c'(t) \neq 0$. Thus we have the following definition.

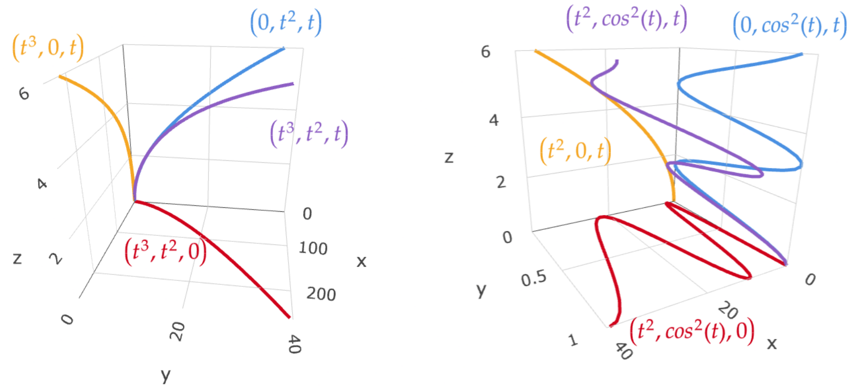

Figure 2: Examples of curve parametrizations

Definition 2.2. *A parametrization of a differentiable curve $c : I \to \mathbb{R}^3$ is called* ***regular*** *if $c'(t) \neq 0$ for all $t \in I$.*

We can extend the definition of the curve to define a surface. A differentiable surface can be defined informally as a two-dimensional subset of $\mathbb{R}^3$ which can be locally approximated by a plane.

Definition 2.3. *A (differentiable) surface is a (differentiable) map, $s : \Omega \to \mathbb{R}^3$, which maps an open set $\Omega \subseteq \mathbb{R}^2$ into $\mathbb{R}^3$.*
In other words, s maps $(a,t) \in \Omega$ to a point $s(a,t) = (x(a,t), y(a,t), z(a,t))$.
The mapping s is called a ***paremetrization****. See Figure 3*

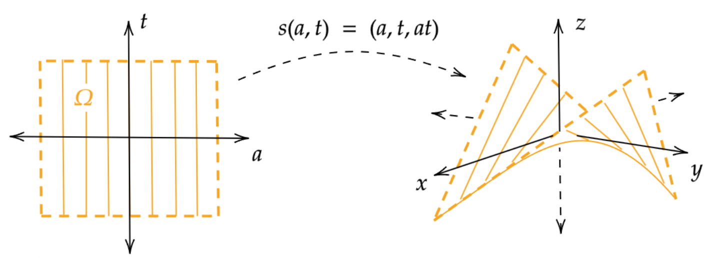

Figure 3: A mapping of an open set Ω into $\mathbb{R}^3$

Similarly, s being differentiable means that the functions $x, y, z : \Omega \to \mathbb{R}$ have continuous partial derivatives and the vectors $\frac{\partial}{\partial a} s(a,t) = s_a(a,t) = (x_a(a,t), y_a(a,t), z_a(a,t))$ and $\frac{\partial}{\partial t} s(a,t) = s_t(a,t) = (x_t(a,t), y_t(a,t), z_t(a,t))$ are tangent to the surface.

Definition 2.4. *A **regular surface** is a surface $s : \Omega \to \mathbb{R}^3$ which satisfies the following conditions:*

1. *s is differentiable*
2. *s has a continuous inverse $s^{-1} : s(\Omega) \to \Omega$*
3. *at all points $(a,t) \in \Omega$, $s_a \times s_t \neq 0$, i.e. there exists a normal*

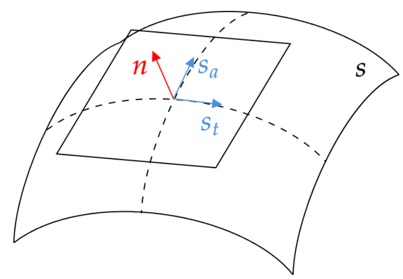

Figure 4: Local frame of a surface

Given a regular surface $s(a,t)$, we then have that the partial derivatives $s_a(a,t)$ and $s_t(a,t)$ span the plane tangent to the surface s at the point (a,t); see Figure 4. Moreover, the normal of the tangent plane coincides with the normal of the surface at (a,t), so the normal of the surface at $(a,t) \in \Omega$ is

$$n(a,t) = \frac{s_a(a,t) \times s_t(a,t)}{|s_a(a,t) \times s_t(a,t)|}$$

We can now define a local frame of a regular surface. Since we are only interested in regular surfaces, we will now be referring to regular surfaces as surfaces for simplicity.

Definition 2.5. *Given a surface $s : \Omega \to \mathbb{R}^3$ where $\Omega \subseteq \mathbb{R}^2$, the **local frame** of s at a point $(a,t) \in \Omega$ consists of the tangent vectors $s_a(a,t)$ and $s_t(a,t)$ and the normal $n(a,t)$, where s_a and s_t span the plane tangent to the surface and n is perpendicular to s_a and s_t.*

Let us consider a surface that changes size, shape or position through time. We can regard each position of said surface as a different surface in a family of surfaces. Such a family of surfaces can be described as a function, $S(m)$, where $m \in M$ and $M \subseteq \mathbb{R}$ is a time interval. We denote $S(m)$ a one-parameter family of surfaces. Similarly, a two-parameter family of surfaces can be regarded as a one-parameter family of surfaces that change through time. We can describe this family as $S(m)$ where $m = (a,t) \in M \subseteq \mathbb{R}^2$. See Figure 5.

Furthermore, a family of surfaces define a volume whose shell or boundary is itself a surface. We refer to this surface as an enveloping surface and we can define it formally as follows:

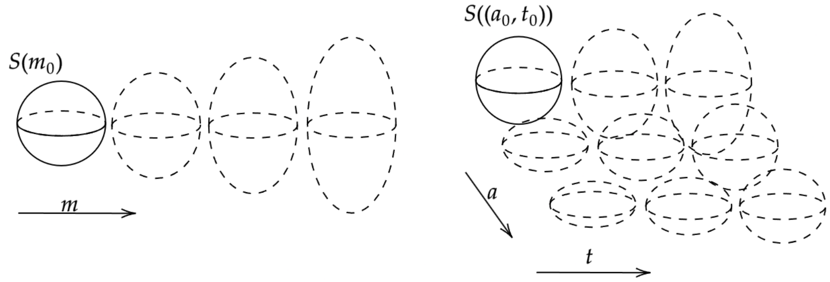

Figure 5: A one-parameter family of surfaces (left) and two-parameter family of surfaces (right)

Definition 2.6. *An **enveloping surface** X is such that at every point $p \in X$, X is tangent to one of the surfaces of the family $S(m)$, $m \in M$ where M is an open set. See Figure 6.*

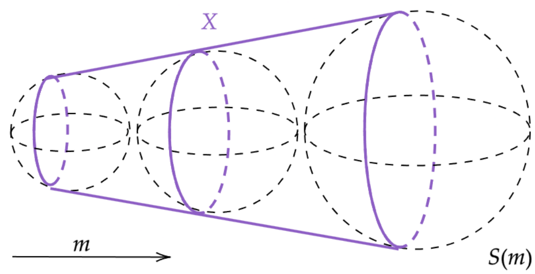

Figure 6: Enveloping surface of a family of spheres

We are interested in enveloping surfaces since they can be used to describe a section of the surface of an object after a passing of a cutting tool of a CNC machine. The surface generated by the cutter is an envelope of the family of surfaces representing each position that the cutters takes.

Furthermore, in CNC machining two passes of a cutting tool generate two unique surfaces. Our main interest lies in defining the conditions for tangent continuity between these surfaces. In order to define necessary and sufficient conditions for two of these surface patches to be continuous, let us begin by defining a composite surface.

Definition 2.7. *A **composite surface** $s = s(a,t)$ is a surface that consists of patches or segments of different surfaces. Here, a patch is a curve bounded surface, $s_p : \Omega \to \mathbb{R}^3$, whose bounding curves result from fixing the value of one parameter of the function of the composite* *surface. See Figure 7. Note that fixing the first parameter of the composite surface $s(a, t)$ results in a curve $c(t) = s(a_0, t)$ with free variable t.*

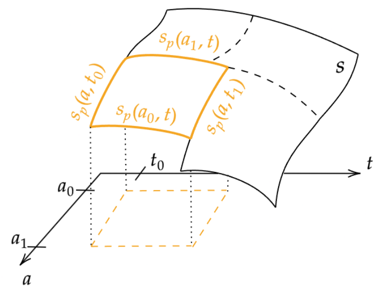

Figure 7: A composite surface with one of its patches highlighted.

Now that we have the tools to describe the surfaces that we encounter in CNC machining, it remains to describe the necessary and sufficient conditions for tangent continuity. Since we can tackle this problem locally, we need to simply define tangent continuity between two patches of a composite surface, or two bounded surfaces, as shown in Figure 8.

Definition 2.8. *Let $s_1(a, t)$ and $s_2(a, t)$ be two patches of a composite surface s. We say that s is* ***tangent plane continuous*** *along the boundary curve of the patches, $c(t) = s_1(a_1, t) = s_2(a_0, t)$ where $t \in [t_0, t_1]$, if the unit normal vectors along the boundary curve of both patches $s_1(a_1, t)$ and $s_2(a_0, t)$ are parallel, namely, $n_1(a_1, t) = \pm n_2(a_0, t)$ for each $t \in [t_0, t_1]$.*

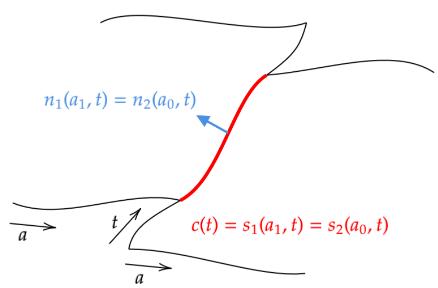

Figure 8: Tangent continuity for composite surfaces

3 Representation of a revolving tool

A cutting tool of a CNC machine sweeps a surface of revolution, which we will denote the enveloping surface of the tool. Following the envelope theory of sphere congruence presented in L. M. Zhu et al. (2009), we can parameterize a rotational tool by regarding it as the envelope of a family of spheres. The enveloping surface of the simplest of milling tools is simply a cone or a cylinder. In this thesis we will focus on these two surfaces but it is also possible to parametrize more complex tools the same way.

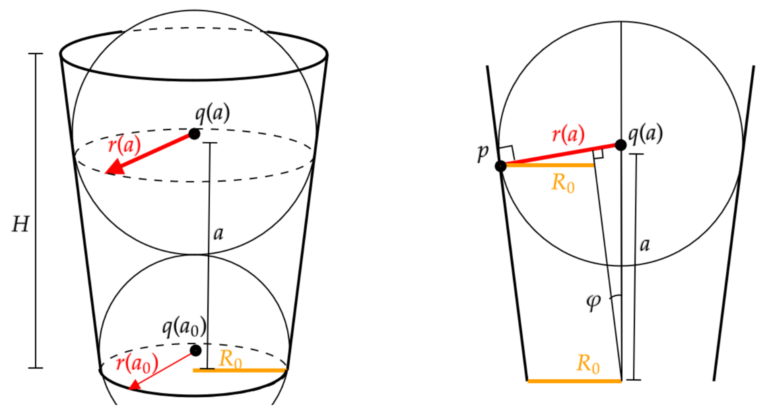

Figure 9: Sphere-swept envelope, with a vertical section on the right

Let us define $\mathbf{q}(a)$ and $r(a)$ as functions of the centers and the radii respectively of the family of spheres representing the rotatory tool. The function $\mathbf{q}(a) = [0,0,a]^T$ denotes the centers of all spheres which are situated along the z-axis, thus for $a \in (a_0, a_1)$ where $a_1 - a_0 = H$, we have that $\mathbf{q}(a)$ gives us the axis of the cylinder. Furthermore, a cylinder has a constant radius and thus the function of the radii would simply be $r = R_0$, where R_0 is the radius of the base of the tool.

A natural extension is to parameterize a conical rotary tool by allowing the radius, r, to change along the axis. Note that in this case, the family of spheres do not graze the enveloping surface at the equator because the tangent to the spheres is not perpendicular to the axis of revolution. For any point p on the surface of revolution, p is a grazing point of a sphere of radius r centered at q where the vector $p - q$ corresponds to the normal at p. Using this property we can parameterize the radii $r(a)$ of spheres with centers on the axis of revolution $\mathbf{q}(a)$.

We can see from Figure 9 that the radius r of the sphere centered at $\mathbf{q}(a)$ can be obtained from the opening angle of the cone φ and the radius of the base of the cone R_0. As a result, we obtain the following parameters.

$$\mathbf{q}(a) = [0,0,a]^T \tag{1}$$

$$r(a) = R_0 \cos\varphi + a \sin\varphi \tag{2}$$

where $a \in [a_0, a_1]$. Equation (2) can easily be derived by looking at the left-hand diagram in Figure 9.

Then the conical tool can be parametrized as follows.

$$s(a,t) = (r(a)\cos\varphi\cos t, r(a)\cos\varphi\sin t, h(a)) \tag{3}$$

where $t \in [-\pi, \pi)$ and $h(a) = a - r(a)\sin\varphi$. $h(a)$ is a function of the tool height with respect to the sphere centers which can be derived from the diagram in Figure 9.

The main focus of this thesis are conical and cylindrical tools, however, it is good to note that this method for representing a revolving tool is highly versatile. This method allows us to parameterize the enveloping surface of many CNC-machining tools by modifying $\mathbf{q}(a)$ and $r(a)$ accordingly. For instance one can easily represent a drum shaped cutter (see Figure 10) or a toroidal cutter in the manner described above.

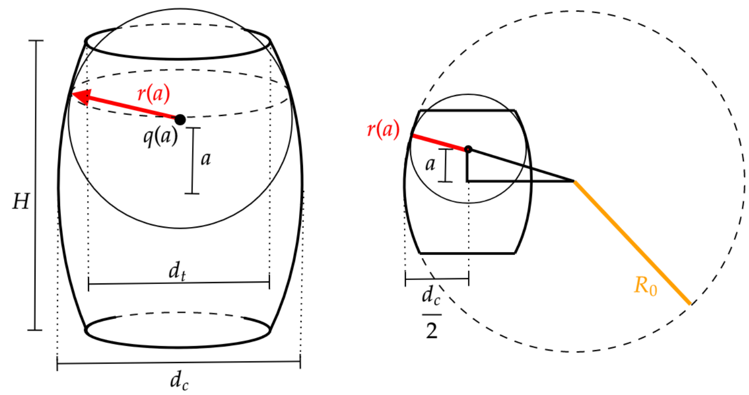

Figure 10: Sphere-swept envelope of a drum shaped tool, with a vertical section on the right

Let us parametrize a drum shaped cutter. As before, let $\mathbf{q}(a)$ and $r(a)$ be functions of the centers and the radii of the family of spheres respectively. H is then the height of the tool and d_c the diameter of the center of the tool. From this information and the diameter d_t of the top of the tool, one can easily derive R_0. R_0 denotes the radius of the circle which defines the curvature of the tool as seen on the right-hand diagram of Figure 10.

Let us define

$$b = \frac{d_c - d_t}{2}.$$

Then using the Pythagorean theorem we can obtain an expression for R_0 in terms of b

$$(R_0 - b)^2 + (\frac{H}{2})^2 = R_0^2,$$

$$\Rightarrow R_0 = \frac{b^2 + (\frac{H}{2})^2}{2b}.$$

Now we can define $\mathbf{q}(a) = [0,0,a]^T$ and $r(a)$ as a function of R_0 and d_c using the Pythagorean theorem (see Figure 10). Thus, obtaining the following expression

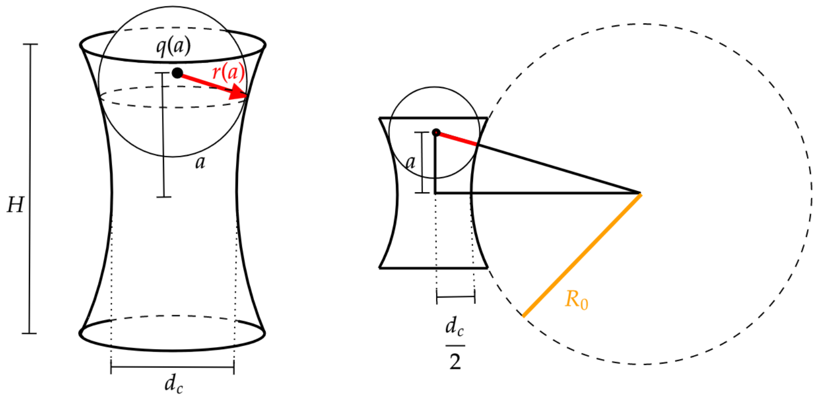

Figure 11: Sphere-swept envelope of a hyperbolic shaped tool, with a vertical section on the right

$$r(a) = R_0 - \sqrt{(R_0 - \frac{d_c}{2})^2 + a^2}.$$

Analogously, we can parametrize a hyperbolic shaped tool by modifying $r(a)$ according to the diagram in Figure 11. Thus obtaining

$$r(a) = \sqrt{(R_0 + \frac{d_c}{2})^2 + a^2} - R_0.$$

The details are left to the reader.

4 Enveloping surface of the tool

We can now parameterize the movement of the tool and thus its enveloping surface which represents the surface of a milled object. Since the tool is parametrized as an envelope of spheres, we may parametrize the tool envelope as an envelope of sphere congruence following the method presented in L. M. Zhu et al. (, 2009).

Here, the movement of the tool depends on the tool position and orientation along time t. Let us describe the path of the tool by defining a guiding curve $P(t)$ which denotes the path of the sphere centered at $q(a_0)$. Note that for a cylinder, this is simply the base of the tool axis but in the case of a conical tool it lies slightly higher (or lower in the case of a negative opening angle) on the tool axis (recall Figure 9). Moreover, let us define $A(t)$ to be the unit vector describing the orientation of the tool axis at each moment in time. Then we can describe the movement of the tool as the surface generated by the tool axis which is defined as

$$S(a,t) = P(t) + (a - a_0)A(t) \tag{4}$$

where $(a,t) \in [a_0, a_1] \times [t_0, t_1]$.

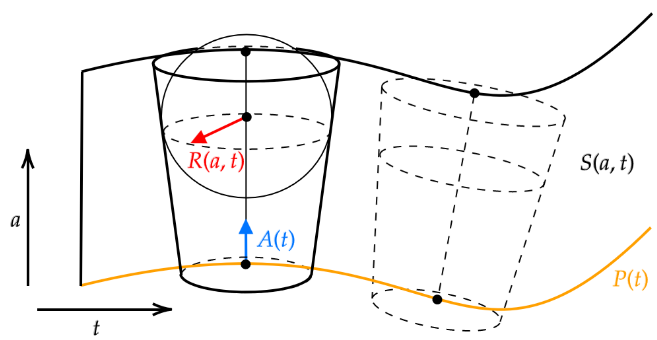

Figure 12: Surface generated by the tool movement

Note that $S(a,t_0)$ in turn describes the position of the center of the spheres that define our cutting tool at time t_0. This means that we can also see $S(a,t)$ as the function defining the position of the center of the family of spheres through time. Therefore, in order to parameterize the envelope of the cutting tool we can treat it as the envelope of the 2-parameter family of spheres with centers at $S(a,t)$. Let us define the radius of the family of spheres throughout time as $R(a,t)$. Since the tool is a rigid body (namely, it should retain its original shape throughout the milling process) we have that

$$R(a,t) = r(a) \tag{5}$$

where $r(a)$ is as in Equation (2).

The cutter swept surface is then given by

$$X(a,t) = S(a,t) + R(a,t)n(a,t) \tag{6}$$

where $(a,t) = [a_0, a_1] \times [t_0, t_1]$ and $n(a,t)$ is the unit normal of the envelope.

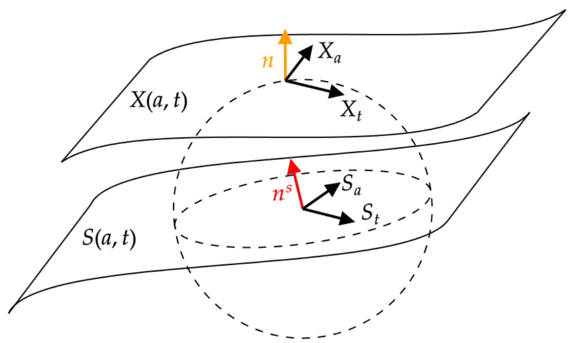

Figure 13: Frame of a sphere of the family of spheres with centers at $S(a,t)$ and the corresponding frame of the enveloping surface at the grazing point.

We now want to obtain an expression for $n(a,t)$. Let us denote the normal of the surface $S(a,t)$ as $n^s(a,t)$. Note that by definition n^s is perpendicular to the partial derivatives of said surface, namely $S_a(a,t)$ and $S_t(a,t)$. In fact, n^s, S_a and S_t generally span $\mathbb{R}^3$ and thus we can write n as a linear combination of these three vectors

$$n(a,t) = \alpha S_a(a,t) + \beta S_t(a,t) + \gamma n^s(a,t). \tag{7}$$

In order to find α, β and γ we begin by computing the partial derivatives of $X(a,t)$

$$\begin{aligned}
\frac{\partial}{\partial a} X(a,t) &= S_a(a,t) + R_a(a,t)n(a,t) + R(a,t)n_a(a,t) \\
&= S_a(a,t) + r_a(a)n(a,t) + r(a)n_a(a,t), \\
\frac{\partial}{\partial t} X(a,t) &= S_t(a,t) + R_t(a,t)n(a,t) + R(a,t)n_t(a,t) \\
&= S_t(a,t) + r(a)n_t(a,t).
\end{aligned}$$

Note that $R_t(a,t) = r_t(a) = 0$. Taking the inner product of the resulting partial derivatives with n, namely computing $X_a \cdot n$ and $X_t \cdot n$, we obtain the following expressions.

$$\begin{aligned}
X_a(a,t) \cdot n(a,t) &= (S_a(a,t) + r_a(a)n(a,t) + r(a)n_a(a,t)) \cdot n(a,t) \\
&= S_a(a,t) \cdot n(a,t) + r_a(a) + r(a)n_a(a,t) \cdot n(a,t), \\
X_t(a,t) \cdot n(a,t) &= (S_t(a,t) + r(a)n_t(a,t)) \cdot n(a,t) \\
&= S_t(a,t) \cdot n(a,t) + r(a)n_t(a,t) \cdot n(a,t).
\end{aligned}$$

Since n is the unit normal of the envelope surface, it is perpendicular to X_a and X_t. Therefore, $X_a \cdot n = X_t \cdot n = 0$. Moreover, n is also perpendicular to its partial derivatives n_a and n_t. We use this to further simplify expressions (8) and (11) which we obtained above.

$$\begin{aligned}
X_a(a,t)\cdot n(a,t) &= S_a(a,t)\cdot n(a,t) + r_a(a) + r(a)n_a(a,t)\cdot n(a,t) && (8)\\
0 &= S_a(a,t)\cdot n(a,t) + r'(a), && (9)\\
S_a(a,t)\cdot n(a,t) &= -r_a(a), && (10)\\
X_t(a,t)\cdot n(a,t) &= S_t(a,t)\cdot n(a,t) + r(a)n_t(a,t)\cdot n(a,t), && (11)\\
S_t(a,t)\cdot n(a,t) &= 0. && (12)
\end{aligned}$$

This results in Equations (10) and (12) which indicate the relationships between the envelope normal and the surface S. We want to find the unknowns of Equation (7), so we take its dot product with S_a and S_t. We will omit the arguments (a,t) of the functions for compactness.

$$\begin{aligned}
S_a \cdot n &= \alpha S_a \cdot S_a + \beta S_a \cdot S_t + \gamma S_a \cdot n^s, && (13)\\
S_t \cdot n &= \alpha S_t \cdot S_a + \beta S_t \cdot S_t + \gamma S_t \cdot n^s. && (14)
\end{aligned}$$

We now substitute Equations (10) and (12) into equations (13) and (14) respectively. Furthermore, we can simplify equations (13) and (14) by observing that S_t and S_a are orthogonal to n^s.

$$\begin{aligned}
-r_a &= \alpha S_a \cdot S_a + \beta S_a \cdot S_t,\\
0 &= \alpha S_t \cdot S_a + \beta S_t \cdot S_t.
\end{aligned}$$

Note that the dot product is commutative, this means that $S_a \cdot S_t = S_t \cdot S_a$. This along with the fact that vectors S_a and S_t cannot be parallel to each other allows us to conclude that the matrix

$$\begin{bmatrix} S_a \cdot S_a & S_a \cdot S_t \\ S_t \cdot S_a & S_t \cdot S_t \end{bmatrix}$$

is invertible. To convince yourself of this, consider its determinant

$$\begin{vmatrix} S_a \cdot S_a & S_a \cdot S_t \\ S_t \cdot S_a & S_t \cdot S_t \end{vmatrix} = (S_a \cdot S_a)(S_t \cdot S_t) - (S_a \cdot S_t)(S_a \cdot S_t).$$

Note that $a \cdot b = |a||b|\cos\theta$, where θ is the angle between the vectors. Hence, the determinant is equal to zero only when $(S_a \cdot S_a)(S_t \cdot S_t) = (S_a \cdot S_t)(S_a \cdot S_t)$ which is only the case when S_a and S_t are parallel to each other.

We then have a system of equations for the unknowns α and β.

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} S_a \cdot S_a & S_a \cdot S_t \\ S_t \cdot S_a & S_t \cdot S_t \end{bmatrix}^{-1} \begin{bmatrix} -r_a \\ 0 \end{bmatrix}. \tag{15}$$

Let

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} S_a \cdot S_a & S_a \cdot S_t \\ S_t \cdot S_a & S_t \cdot S_t \end{bmatrix}^{-1}.$$

Then,

$$\alpha = -A_{11} r_a, \tag{16}$$

$$\beta = -A_{21} r_a. \tag{17}$$

By substituting Equations (16) and (17) in Equation (7) we obtain the following expression:

$$n = -A_{11} r_a S_a - A_{21} r_a S_t + \gamma n^s.$$

Taking the inner product with n, and using (10), (12), we can solve for γ:

$$\begin{aligned} n \cdot n &= -A_{11} r_a S_a \cdot n - A_{21} r_a S_t \cdot n + \gamma n^s \cdot n, \\ 1 &= -A_{11} r_a(-r_a) - A_{21} r_a(0) + \gamma n^s \cdot n, \\ 1 &= A_{11} r_a 2 + \gamma n^s \cdot n. \end{aligned}$$

Recall that S_t and S_a are orthogonal to n^s and compute the product $n^s \cdot n$ using Equation (7).

$$n^s \cdot n = \alpha n^s \cdot S_a + \beta n^s \cdot S_t + \gamma n^s \cdot n^s, \tag{18}$$

$$n^s \cdot n = \gamma. \tag{19}$$

Thus,

$$\gamma = \pm\sqrt{1 - r_a^2 A_{11}}. \tag{20}$$

Finally we can substitute Equations (16), (17) and (20) into (7) to obtain the expression for $n(a, t)$:

$$n = -A_{11} r_a S_a - A_{21} r_a S_t \pm \sqrt{1 - r_a^2 A_{11}}\, n^s. \tag{21}$$

We have defined all of the parameters in our parametrization of a cutter swept surface (see Equation 6). Using this parametrization we can now study the continuity conditions of adjacent envelopes.

5 Conditions for tangent continuity

In this section we define the necessary and sufficient conditions for ensuring tangent continuity between two adjacent enveloping surfaces as done in L. Zhu & Lu (2015). Let us define two distinct envelopes

$$X^1(a,t) = S^1(a,t) + r(a)n^1(a,t), \tag{22}$$

$$X^2(a,t) = S^2(a,t) + r(a)n^2(a,t), \tag{23}$$

where $(a,t) \in [a_0, a_1] \times [t_0, t_1]$.

In order to obtain tangent plane continuity, or G^1-continuity, among the envelopes we need to ensure positional continuity and tangent convergence at the junction. Therefore the following are the necessary conditions for tangent continuity:

$$X^1(a_1,t) = X^2(a_0,t), \tag{24}$$

$$n^1(a_1,t) = n^2(a_0,t), \tag{25}$$

Equation (24) requires the curve at a_1 of the first enveloping surface to coincide with the curve at a_0 of the second surface. Similarly, form Equation (25) we have that along said curve, the normal vectors of both surfaces must be equal to each other. Note that we assume the normal vectors are unit length and are oriented away from the tool centers. Although $n^1(a_1,t) = -n^2(a_0,t)$ would in theory result in a tangentially continuous surface, we disregard this because it would not make sense for us given that our motivation lies in milling objects through CNC-machining. These two conditions ensure tangential convergence of the enveloping surfaces.

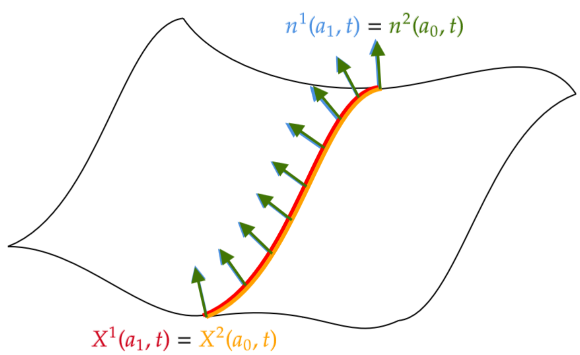

Figure 14: Visualization of G^1-continuity between two surfaces

We now derive the necessary and sufficient conditions for tangent continuity such that they impose the necessary constraints on the second enveloping surface. In order for these conditions to be useful to our implementation, the aim is that they define the necessary constraints on the tool path and orientation that generate the second enveloping surface, X_2, with respect to the parameters of the first surface X_1.

Let us first consider Equation (24). We can use the parametrizations of the surfaces from Equations (22) and (23) to work out the following expression.

$$
\begin{aligned}
&X^1(a_1,t) = X^2(a_0,t),\\
&S^1(a_1,t) + r(a_1)n^1(a_1,t) = S^2(a_0,t) + r(a_0)n^2(a_0,t),\\
&S^2(a_0,t) = S^1(a_1,t) + r(a_1)n^1(a_1,t) - r(a_0)n^2(a_0,t).
\end{aligned}
$$

Using Equation (25) we then have

$$S^2(a_0,t) = S^1(a_1,t) + (r(a_1) - r(a_0))n^1(a_1,t). \tag{26}$$

Recall Equations (10) and (12). Again, considering these and Equation (25) we obtain the following expressions:

$$S_a^2(a_0,t)\cdot n^1(a_1,t) = -r_a(a_0), \tag{27}$$

$$S_t^2(a_0,t)\cdot n^1(a_1,t) = 0. \tag{28}$$

Since Equations (26), 27 and (28) are equivalent to Equations (24), 25, they also present the necessary conditions for tangent continuity. Moreover, Equation (28) is superfluous since it can be derived from (26) as follows:

$$
\begin{aligned}
\frac{\partial}{\partial t}S^2(a_0,t) &= \frac{\partial}{\partial t}(S^1(a_1,t) + (r(a_1) - r(a_0))n^1(a_1,t)),\\
S_t^2(a_0,t) &= S_t^1(a_1,t) + (0)n^1(a_1,t) + (r(a_1) - r(a_0))n_t^1(a_1,t),\\
S_t^2(a_0,t) &= S_t^1(a_1,t) + (r(a_1) - r(a_0))n_t^1(a_1,t).
\end{aligned}
$$

Now, taking the product $S_t^2(a_0,t)\cdot n^1(a_1,t)$ the equation above implies

$$
\begin{aligned}
S_t^2(a_0,t)\cdot n^1(a_1,t) =& S_t^1(a_1,t)\cdot n^1(a_1,t) + (r(a_1) - r(a_0))n_t^1(a_1,t)\cdot n^1(a_1,t),\\
S_t^2(a_0,t)\cdot n^1(a_1,t) =& 0 + 0,\\
S_t^2(a_0,t)\cdot n^1(a_1,t) =& 0,
\end{aligned}
$$

since n_t^1 and S_t^1 are perpendicular to n^1.

Theorem 5.1. *For the enveloping surfaces of conical tools $X^1(a,t) = S^1(a,t) + r(a)n^1(a,t)$ and $X^2(a,t) = S^2(a,t) + r(a)n^2(a,t)$, the following are necessary and sufficient conditions for G^1-continuity:*

$$S^2(a_0,t) = S^1(a_1,t) + (r(a_1) - r(a_0))n^1(a_1,t), \tag{29}$$

$$S_a^2(a_0,t)\cdot n^1(a_1,t) = -r_a(a_0). \tag{30}$$

Proof. We have already shown that Equations (29) and (30) are necessary conditions for tangent continuity, it remains to show that they are sufficient. That is, we want to show that (29) and (30) imply $n^1(a_1,t) = n^2(a_0,t)$ and $X^1(a_1,t) = X^2(a_0,t)$.

We begin by showing that they imply $n^1(a_1,t) = n^2(a_0,t)$. For this, recall how we reached Equation (21). We can express $n^1(a_1,t)$ as a linear combination of S_a^2, S_t^2 and n_s^2 because the latter span $\mathbb{R}^3$:

$$n^1(a_1,t) = \alpha S_a^2(a_0,t) + \beta S_t^2(a_0,t) + \gamma n_s^2(a_0,t). \tag{31}$$

Taking the inner products $S_a^2(a_0,t)\cdot n^1(a_1,t)$ and $S_t^2(a_0,t)\cdot n^1(a_1,t)$, we obtain a system of equations similar to that in Equation (15).

For $S_a^2(a_0,t)\cdot n^1(a_1,t)$ we have

$$\begin{aligned} S_a^2(a_0,t)\cdot n^1(a_1,t) =& \alpha S_a^2(a_0,t)\cdot S_a^2(a_0,t) + \beta S_a^2(a_0,t)\cdot S_t^2(a_0,t) + \gamma S_a^2(a_0,t)\cdot n_s^2(a_0,t),\\ -r_a(a_0) =& \alpha S_a^2(a_0,t)\cdot S_a^2(a_0,t) + \beta S_a^2(a_0,t)\cdot S_t^2(a_0,t). \end{aligned}$$

Here, we used Equation (30) and the perpendicularity of S_a^2 and n_s^2. This is the first equation of our system.

For $S_t^2(a_0,t)\cdot n^1(a_1,t)$ we have

$$\begin{aligned} S_t^2(a_0,t)\cdot n^1(a_1,t) =& \alpha S_t^2(a_0,t)\cdot S_a^2(a_0,t) + \beta S_t^2(a_0,t)\cdot S_t^2(a_0,t) + \gamma S_t^2(a_0,t)\cdot n_s^2(a_0,t),\\ 0 =& \alpha S_t^2(a_0,t)\cdot S_a^2(a_0,t) + \beta S_t^2(a_0,t)\cdot S_t^2(a_0,t), \end{aligned}$$

where we used Equation (30) and the perpendicularity of S_t^2 and n_s^2. This is the second equation of our system.

Let

$$\begin{bmatrix} A_{11}^2 & A_{12}^2 \\ A_{21}^2 & A_{22}^2 \end{bmatrix} = \begin{bmatrix} 1 & S_a(a_0,t)\cdot S_t(a_0,t) \\ S_t(a_0,t)\cdot S_a(a_0,t) & 1 \end{bmatrix}^{-1}.$$

Then,

$$\alpha = -A_{11}^2 r_a(a_0), \tag{32}$$

$$\beta = -A_{21}^2 r_a(a_0). \tag{33}$$

We can plug in Equations (32) and (33) into Equation (31) resulting in

$$n^1(a_1,t) = -A_{11}^2 r_a(a_0) S_a^2(a_0,t) - A_{21}^2 r_a(a_0) S_t^2(a_0,t) + \gamma n_s^2(a_0,t).$$

Computing $n^1(a_1,t)\cdot n^2(a_0,t)$ we can find an expression for γ:

$$\begin{aligned} n^1(a_1,t)\cdot n^2(a_0,t) =& -A_{11}^2 r_a(a_0) S_a^2(a_0,t)\cdot n^2(a_0,t)\\ &- A_{21}^2 r_a(a_0) S_t^2(a_0,t)\cdot n^2(a_0,t)\\ &+ \gamma n_s^2(a_0,t)\cdot n^2(a_0,t). \end{aligned}$$

We know that $S_a^2(a,t) \cdot n^2(a,t) = -r_a(a)$ and $S_t^2(a,t) \cdot n^2(a,t) = 0$ from Equations (10) and (12). Using this and $n_s^2(a_0,t) \cdot n^2(a_0,t) = \gamma$ which we derive from Equation (19), we obtain the following expression:

$$n^1(a_1,t) \cdot n^2(a_0,t) = -A_{11}^2 r_a(a_0)^2 + \gamma^2. \tag{34}$$

Since from Equation (20) we have that $1 = -A_{11}^2 r_a(a_0)^2 + \gamma^2$, this and Equation (34) imply that

$$n^1(a_1,t) = n^2(a_0,t).$$

Since we have now shown $n^1(a_1,t) = n^2(a_0,t)$, we can use this in combination with Equation (29) to reach $X^1(a_1,t) = X^2(a_0,t)$:

$$\begin{aligned}
&S^2(a_0,t) = S^1(a_1,t) + (r(a_1) - r(a_0))n^1(a_1,t),\\
\Rightarrow &S^2(a_0,t) = S^1(a_1,t) + r(a_1)n^1(a_1,t) - r(a_0)n^1(a_1,t),\\
\Rightarrow &S^2(a_0,t) = S^1(a_1,t) + r(a_1)n^1(a_1,t) - r(a_0)n^2(a_0,t),\\
\Rightarrow &S^2(a_0,t) + r(a_0)n^2(a_0,t) = S^1(a_1,t) + r(a_1)n^1(a_1,t),\\
\Rightarrow &X^2(a_0,t) = X^1(a_1,t).
\end{aligned}$$

We have now shown that (29) and (30) are the necessary and sufficient conditions. □

Note that Theorem 5.1 assumes that the envelopes are generated by the same tool, namely they use the same function $r(a)$. The result above can be generalized by re-naming said function in each envelope to $r^1(a)$ and $r^2(a)$ respectively. Thus we can define the necessary and sufficient conditions for continuity between envelopes generated by different milling tools.

Corollary 5.2. *For the enveloping surfaces of conical tools $X^1(a,t) = S^1(a,t) + r^1(a)n^1(a,t)$ and $X^2(a,t) = S^2(a,t) + r^2(a)n^2(a,t)$, the following are necessary and sufficient conditions for G^1-continuity:*

$$S^2(a_0,t) = S^1(a_1,t) + (r^1(a_1) - r^2(a_0))n^1(a_1,t), \tag{35}$$

$$S_a^2(a_0,t) \cdot n^1(a_1,t) = -r_a^2(a_0). \tag{36}$$

Furthermore, we can take the derivative of $S^2(a,t)$ with respect to a

$$S_a^2(a,t) = \frac{\partial}{\partial a}(P^2(t) + aA^2(t) - a_0A^2(t)) = A^2(t), \tag{37}$$

and rewrite Equation (30) as

$$A^2(t) \cdot n^1(a_1,t) = -r_a(a_0). \tag{38}$$

Corollary 5.3. *For the enveloping surfaces of conical tools* $X^1(a,t) = S^1(a,t) + r(a)n^1(a,t)$ *and* $X^2(a,t) = S^2(a,t) + r(a)n^2(a,t)$*, where* $S^i(a,t) = P^i(t) + (a - a_0)A^i(t)$*,* $i = 1, 2$*, the following are necessary and sufficient conditions for* G^1*-continuity:*

$$S^2(a_0,t) = S^1(a_1,t) + (r(a_1) - r(a_0))n^1(a_1,t), \tag{39}$$

$$A^2(t) \cdot n^1(a_1,t) = -r_a(a_0). \tag{40}$$

Remark 5.4. In Section 4 we parametrized the surface generated by the sphere centers by assuming that they are positioned along a straight line or axis with respect to a, see Equation (4). This assumption is used in Corollary 5.3 but not in Theorem 5.1.

At some point we might only require positional continuity between envelopes. In this case, we can use the following, weaker, result:

Corollary 5.5. *For the enveloping surfaces of conical tools* $X^1(a,t) = S^1(a,t) + r(a)n^1(a,t)$ *and* $X^2(a,t) = S^2(a,t) + r(a)n^2(a,t)$*, where* $S^i(a,t) = P^i(t) + (a - a_0)A^i(t)$*,* $i = 1, 2$*, the following is a necessary and sufficient condition for* G^0*-continuity:*

$$S^2(a_0,t) = S^1(a_1,t) + r(a_1)n^1(a_1,t) - r(a_0)n^2(a_0,t). \tag{41}$$

Remark 5.6. Theorem 5.1 and its corollaries make use of the assumption that the tool is rigid (thus the radius doesn't change w.r.t. time), which should be the case for any real life milling tool. However, it can be generalized by changing $r(a)$ for a two-parameter function $R(a,t)$.

To better understand the results obtained above, let us discuss and illustrate their significance. Note that $S^2(a_0,t) = P^2(t)$, where $P^2(t)$ is the guiding curve of the tool path that generates the enveloping surface $X^2(a,t)$ (see Figure 15). Moreover, Equation (39) has no free variables which means that the guiding curve of the second surface is completely defined by the first surface.

On the other hand, Equation (40) indicates how closely the vectors $n^1(a_1,t)$ and $A^2(t)$ must align. Since rotating $A^2(t)$ around $n^1(a_1,t)$ doesn't affect the angle between said vectors, see Figure 16, there is a choice for the second orientation of the tool axis, namely any vector from the family of vectors that form the required angle with $n^1(a_1,t)$. The angle, θ, between $n^1(a_1,t)$ and $A^2(t)$ depends on the rate of change of radii w.r.t a, namely $-r_a(a_0)$. In the case of a cone with opening angle φ, this relation can be expressed as $A^2(t) \cdot n^1(a_1,t) = -\sin\varphi$. In the case of a cylinder, the opening angle being null (namely $r(a) = R_0$) means that $r_a = 0$. Therefore for a cylindrical tool, the tool axis $A^2(t)$ must be perpendicular to $n^1(a_1,t)$.

Note that both $n^1(a_1,t)$ and $A^2(t)$ are unit vectors, hence the angle, θ, between $n^1(a_1,t)$ and $A^2(t)$ is

$$\arccos(-r_a(a_0)) \tag{42}$$

We have derived the conditions for positional and tangent continuity for the envelopes of a wide variety of surface families. These surface families are able to represent numerous tools which are commonly used in CNC-machining. If we recall Remark 5.4, we may

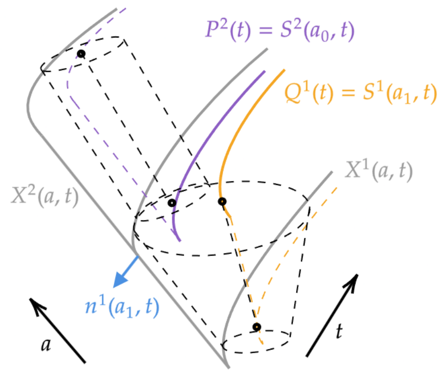

Figure 15: The guiding curve of the second envelope, $P^2(t)$, is defined with respect to the parameters of the first envelope, namely $Q^1(t)$, $n^1(a_1, t)$ and the respective radii of the tools.

note that for tools parametrized as the envelope of a family of circles which don't lie in a straight line, we have to parametrize the surface generated by the sphere centers differently form Equation (4). This parametrization might look similar since one could define the tool orientation using the tangent vector of the curve defining the axis of spheres. Hence, we can claim that these results can be easily expanded on to allow for more tool shapes.

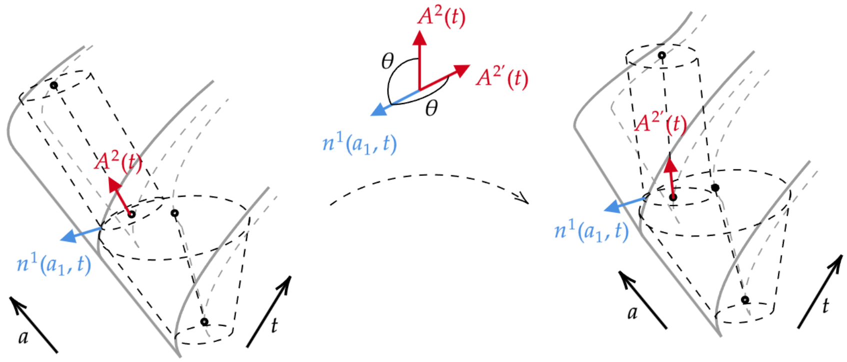

Figure 16: Tangent continuity requires the axis vector of the tool of the second envelope, $A^2(t)$, to be oriented at an angle with respect to the normal vector of the bounding curve of the first envelope, $n^1(a_1, t)$. Rotating the axis vector around said normal vector conserves this angle, giving us certain freedom over the axis orientation.

6 Application

We developed a simple application[1] in order to apply our results and provide the reader with an interactive tool to better visualize and understand the concepts discussed. Said user interface was is written in C++, using the Qt framework and expanding on an application that was made for the Computer Graphics course at Rijksuniversiteit Groningen.

6.1 Architecture and logic

In this section we go over the main structure of our program, which is broadly illustrated in the class diagram in Figure 17. Our aim was to structure the application following the model-view-controller design pattern where the view is able to display the main elements used in the definition of the tool envelope in Section 4 with little to no knowledge of the model logic.

Naturally, the `MainView` class renders the elements chosen by the user, while the `MainWindow` class provides a comprehensive input control menu, which we go over in Section 6.2. The `MainView` enlists the help of the rendering classes in the `renderers/` folder. To allow for easy incorporation of new renderers in the future, all renderers extend the abstract class defined in `renderers/renderer.h` which defines the basic functionality applicable to every renderer. All renderers contain vertex arrays to store, in discrete form, the elements that they need to render and, of course, they then use vertex array objects and vertex buffer objects along with the methods of the `OpenGL` graphics library to draw the objects. These components are not included in the abstract class since some renderers draw more than one object, however all renderers have a reference to the settings

[1] The codebase for this application can be found on Github and Gitlab at the following urls: https://github.com/c-aranda-bassegoda/moving_cylinders, https://git.lwp.rug.nl/svcg/carolina-bscthesis2024

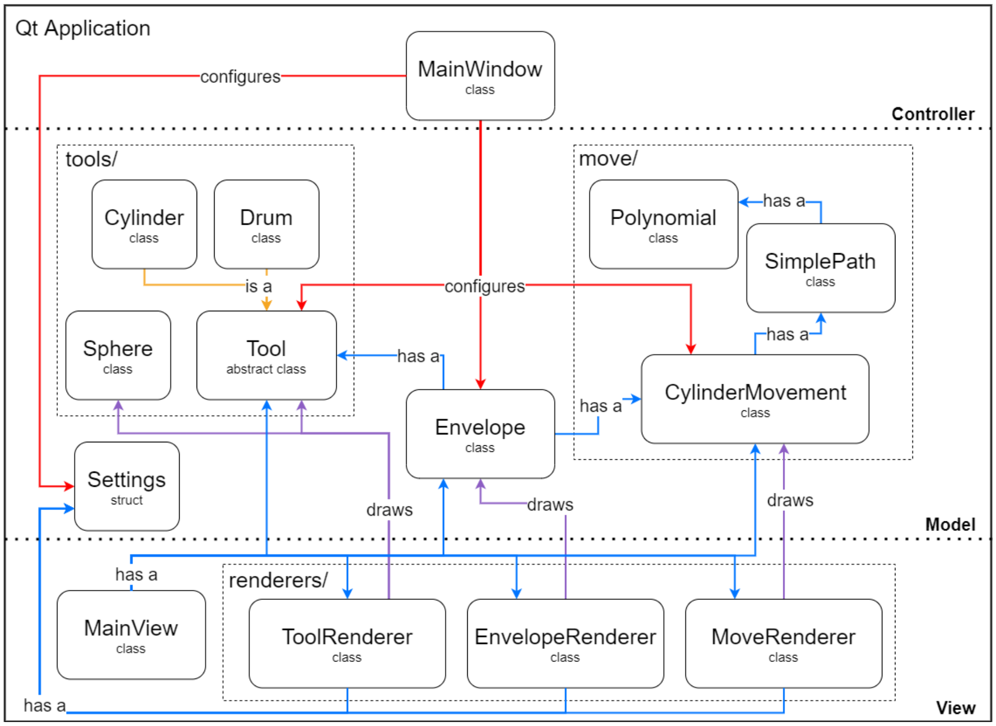

Figure 17: Class diagram of the application

such that they know what elements to draw. Here, `Settings` is a struct which allows for communication between the controller and the view.

As follows, we go over the elements that each renderer is tasked with displaying by following the order in which the components were presented in Sections 4, 3 and 5.

The elements that are essential to every tool are defined in `tool.h`. This abstract class must define the methods that provide `renderers/toolrenderer.h`, and `envelope.h` the information needed in order to render the tool and compute the envelope respectively. The main attributes of the `Tool` class are the vertex array of the tool and its position. Moreover, it must also include attributes concerning the axis domain and methods that allow other classes to fetch the information concerning the family of spheres that define the tool, namely, a method to evaluate the radius function and its derivative at a point in the axis. Every class that defines a milling tool should inherit from `tool.h` in order to ensure that they are compatible with the classes that require them.

An important element in the definition of a tool is the family of spheres described in Section 3. Hence, we want to provide the user with an illustration of how the spheres relate to the tool and, in turn, the envelope. For this purpose we define the `Sphere` class which is a simple class defining the vertex array of a sphere with a given radius. One could also make this class part of the `Tool` class using composition. This class, `tools/sphere.h`, is only used by `renderers/toolrenderer.h` in order to compute and draw the spheres in the family of spheres that define the tool envelope as described in Section 3.

Next, the `renderers/moverenderer.h` class draws and updates the path that is defined in the header file `movement/cylindermovement.h`. Said path is a `SimplePath`, namely, a curve parametrized as in Definition 2.1 using three cubic polynomials which are defined in `movement/polynomial.h`. This class provides all members necessary to set, update and retrieve its members, namely, the polynomials, the bounds of the interval and the vertex array where the path is stored in discrete form. The tool movement consists not only of the path the tool takes, but also the orientation of the tool axis at each point along the path. Said directions are stored in `axisDirections` and can be set to a default orientation or computed by interpolating the intermediate orientations from the initial and final ones. Furthermore, to facilitate the computation of the tool transformations, the `CylinderMovement` class provides a method to easily retrieve the rotation matrix that needs to be applied to the tool transformation. This transformation is computed using the previously defined `axisDirections` and `rotationVectors`. The latter is computed by taking the cross product of `axisDirections` and the tool axis vector. Note that all vectors that represent directions must be normalized.

Finally, we have the `renderers/enveloperenderer.h` which draws all the remaining elements which are defined in `envelope.h`. The `Envelope` class is naturally the most important class; it contains a pointer to a `Tool` and a `CylinderMovement`. These two fields allow the `Envelope` class to compute the vertex arrays that store the information of all the components of an envelope in discrete form. The main component that this class is concerned with is the envelope itself. Furthermore, we also store the grazing curves, the normal vectors of the envelope and the sphere centers (or tool axes) of the family of surfaces that define the envelope. The latter geometric elements are essential to the computation of the envelope and rendering them allows us to better visualize the interplay of the parameters that define an envelope as described in Section 4.

Additionally, we want the user to be able to define an envelope with respect to another envelope and give them certain continuity constraints. Hence, the `Envelope` class also has a pointer to another envelope called `adjEnv` as well as booleans `positToAdj` and `contToAdj` which indicate whether the envelope is positionally continuous only or also tangentially continuous to `adjEnv`. If an envelope is positionally continuous to its adjacent envelope, the path is computed according to Equation (41). Additionally, if it is tangentially continuous, the axis orientation of the tool movement is rotated using our understanding of Equation (40), see Figure 16.

The correct rotation of the second tool is achieved through the following algorithm: The axis of the tool is first rotated such that it is equal to the normal vector of the adjacent envelope at `a1` and the corresponding `time`. Then a second rotation is applied which rotates the axis vector such that it forms the desired angle with respect to the normal vector (see Equation (40)). This rotation turns the axis vector **ra** degrees about the tangent vector of the guiding path at the corresponding `time`. The variable **ra** is computed according to equation (42). Finally, the axis vector is rotated about the normal vector according to the user input.

6.2 User interface

In this section, we go over the functionalities offered by the input fields in the input menu. The application provides multiple input controls which allow the user to manipulate the components described above. We kept the transformation widgets, consisting of a scaling slider and three rotation knobs each of which indicates the rotation of the model about the x-, y- or z-axis that were already implemented in the application that we expanded. We also added new input fields to give the user ample freedom in defining the numerous possible enveloping surfaces that we discuss in earlier sections.

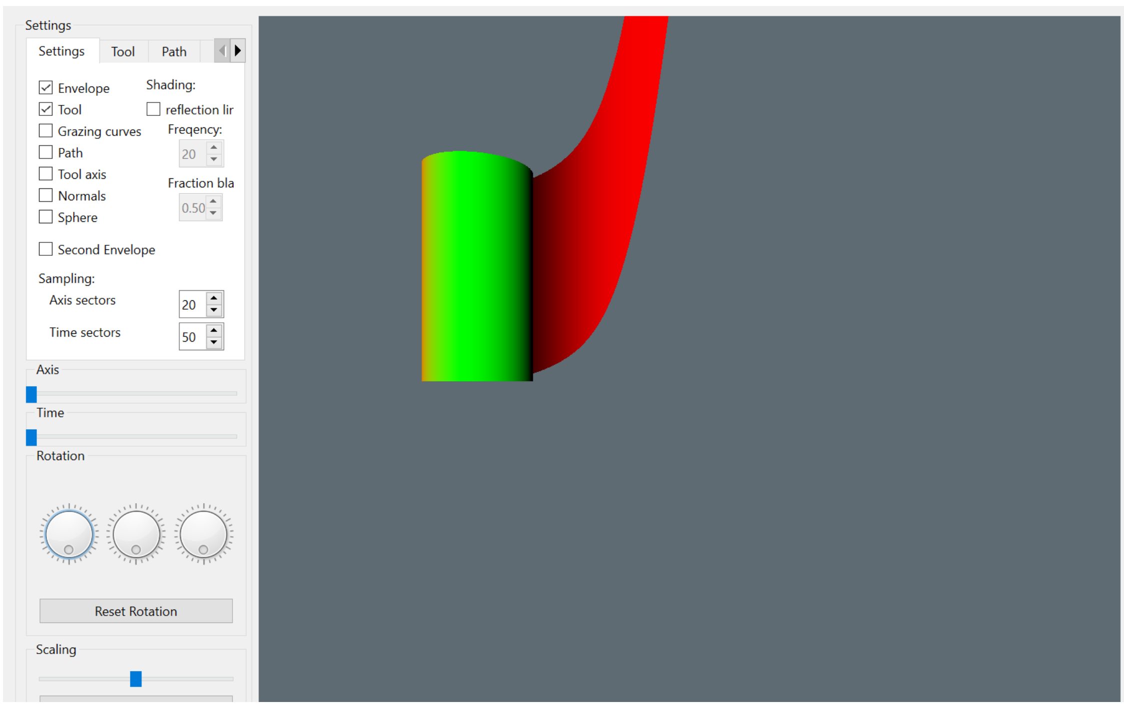

Figure 18: View of the application upon initialization or the user interface

The time slider defines the position of the tool in the display with respect to the cylinder movement. This slider should illustrate the cylinder moving through time, it does this by updating the `time` attribute of the `Settings` struct, updating the tool transform and re-drawing the tool. Similarly we have the axis slider which allows the user to visualize the family of spheres that define the tool envelope by moving it along the tool axis. This slider works in a similar fashion to the time slider. Right at the top of the input menu we find a `tabWidget` with four tabs, see Figure 19.

The first tab allows the user to select which elements should appear on the view by clicking on the relevant check boxes. Moreover, here the user can edit the resolution of the displayed meshes by writing the desired number of sectors on the spin boxes; see Figure 20. The default shading of the envelopes is normal shading, however, in order to better visualize the continuity of the envelopes a shading option was added which simulates reflection lines on the surface; see Figure 21. This shading is a simple procedural texture

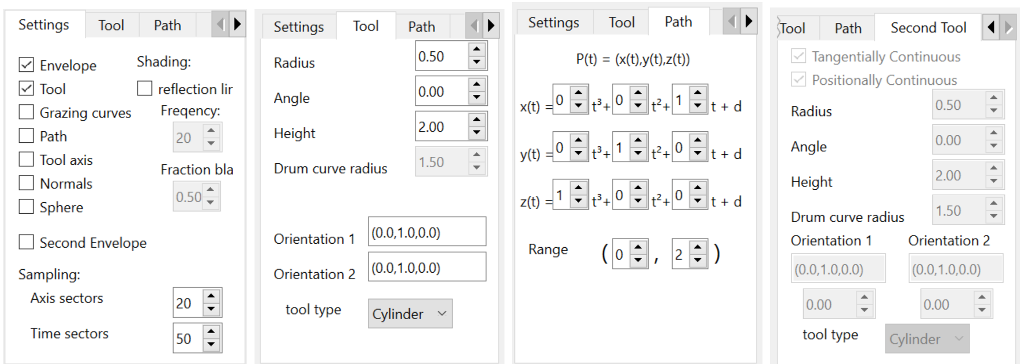

Figure 19: Tab widget of the input menu

computed by mapping the angles between the normal vectors and a view vector to either black or white.

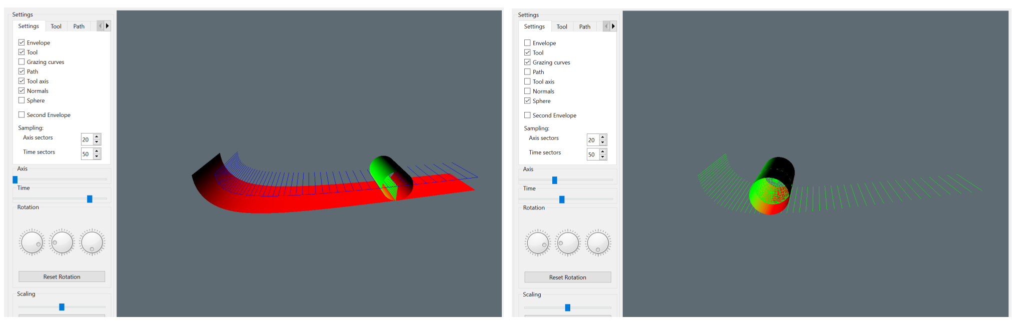

Figure 20: View of the model with different selections

The Tool tab allows the user to define the dimensions of the tool as shown in Figure 22. Additionally, when the second envelope check box is selected, any changes to these settings will be applied to both tools. If the user wishes to modify the dimensions of the second tool only, this can be done in the last tab. The Tool tab updates both tools such that the user doesn't have to switch between the Tool and Second Tool tabs if they want both tools to have the same parameters. In this tab the user may also select a different type of tool using the drop box at the bottom and according to the tool selected the spin boxes relevant to the tool parametrization get enabled or disabled. Moreover, the user can define the orientation of the tool axis at the beginning and the end of the path through the Orientation 1 and Orientation 2 text fields respectively. The application uses a validator to prevent the user from entering an invalid expression and displays an error message if the user enters a trivial vector, namely $(0, 0, 0)$.

The third tab indicates to the user that the path is parametrized using three cubic polynomials and allows them to modify their coefficients as well as the range for the variable, t. The user is not given the option to modify the constant term because it would simply translate the displayed path.

Finally, there's the Second Tool tab. This tab is very similar to the second tab, but it

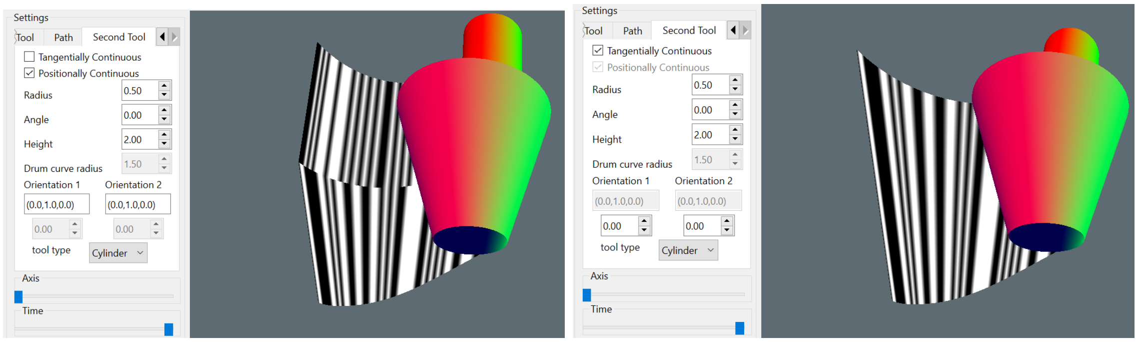

Figure 21: View of both envelopes with reflection-lines-shading selected. Positional continuity on the left, Tangent continuity on the right.

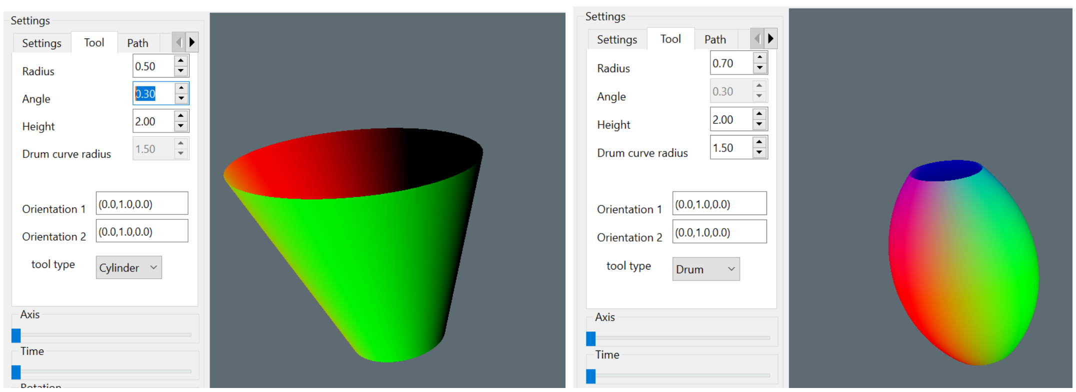

Figure 22: Tool settings

provides some additional settings. The two check boxes allow the user to choose the type of continuity between the envelopes. See Figure 23 for some illustrations of possible resulting envelopes. Checking the Tangentially Continuous box will check and disable the Positionally Continuous box. This is because tangential continuity requires positional continuity. Moreover, as we saw in Section 5, tangential continuity imposes constraints on the axis direction, therefore, while with positional continuity (or no continuity conditions) the user should be able to define the orientation of the tool axis freely through a vector, with tangential continuity there is less freedom to the tool orientation. Given this, when the tangential continuity option is selected the spin box is enabled instead of the text field. Here, the spin box represents the angle that the computed vector should be spun around the normal vector of the first envelope, thus conserving tangential continuity as seen in Figure 16.

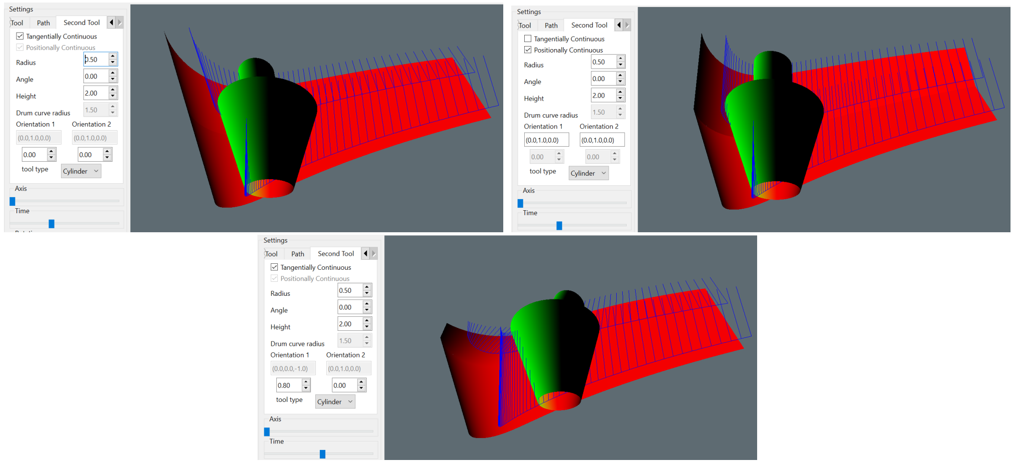

Figure 23: Views given different continuity settings

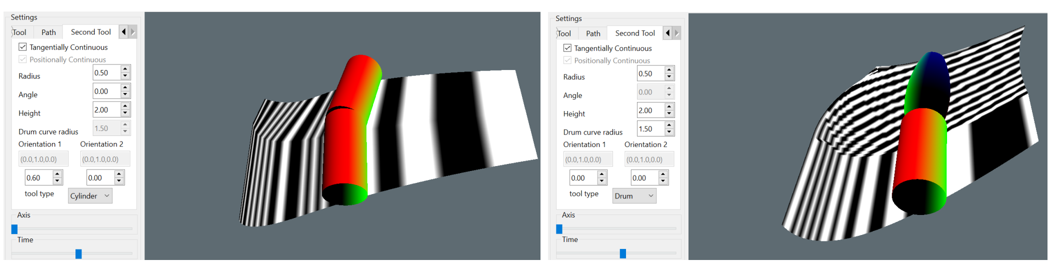

Figure 24: Views of tangent continuous envelopes with reflection lines given different tool settings

7 Conclusion

In this thesis we discussed the parametrization of the envelope of a revolving tool as that of a family of spheres. This method is useful for representing the most common tool shapes in CNC machining. Furthermore, it has the advantage that this kind of parametrization allows us to represent numerous different tools by simply changing the radius function. This allows us to study the envelopes of numerous tools by making no assumptions regarding the exact shape of the tool.

Our parametrization of the tool allowed us to define the envelope of said tool as the envelope of a two parameter family of spheres regardless of the shape of the particular tool. Having achieved this we were able to derive two equations which entail the necessary and sufficient conditions for tangent continuity between two such envelopes. These equations were chosen to help us define an envelope tangent continuous to a previously defined envelope. As such they help us define the path and orientation of the second envelope.

In fact, as we discussed in Section 5, we found that the curve which defines path of the second envelope is fully determined by the first envelope, however, recall that the tool movement involves both the path the tool orientation. Regarding the tool orientation we found that there was a relation between the orientation of the second tool axis and the normal of the bounding curve of the first envelope at a point (this is defined in Equation (30)). However, this relation doesn't uniquely define the orientation of the tool at every point, since there are infinitely many vectors that fulfill this condition.

Having derived the conditions for tangent continuity in this way allowed us to use these as an algorithm in our user interface. Our user interface allows the user to visualize all the elements that come into play when defining enveloping surfaces using families of spheres. Moreover, it allows the user to visualize both positional and tangential continuity between envelopes and experiment with the free variables that each type of continuity allows.

Since this was a simple visualization tool and was not foreseen as a bigger design application, there was little emphasis given to the scalability of the program. This means that there are a lot of improvements that can be done regarding computational efficiency. Because of this same reason, we have not built it with the intention of allowing the user to define more than two envelopes, however it remains a possibility to expand it by allowing any number of envelopes. Adding more tool shapes also remains a possibility, given our motivation we limited ourselves to the most obvious ones.

Finally, the limitations of this project did not allow for analysis of the validity of the surfaces generated. By validity we mean verifying that there are are no elements preventing the defined tool movement from being traced and thus the surfaces can be milled. In order to tackle this problem, future efforts could focus on implementing and illustrating the role of collision detection in the design of enveloping surfaces.

References

Bartoň, M., Bizzarri, M., Rist, F., Sliusarenko, O., & Pottmann, H. (2021). Geometry and tool motion planning for curvature adapted CNC machining. *ACM Transactions on Graphics*, *40*, 1–16. https://doi.org/10.1145/3450626.3459837

Bo, P., Bartoň, M., Plakhotnik, D., & Pottmann, H. (2016). Towards efficient 5-axis flank CNC machining of free-form surfaces via fitting envelopes of surfaces of revolution. *Computer-Aided Design*, *79*, 1–11. https://doi.org/https://doi.org/10.1016/j.cad.2016.04.004

Bo, P., Bartoň, M., & Pottmann, H. (2017). Automatic fitting of conical envelopes to free-form surfaces for flank CNC machining. *Computer-Aided Design*, *91*, 84–94. https://doi.org/https://doi.org/10.1016/j.cad.2017.06.006

DoCarmo, M. P. (2016). *Differential geometry of curves & surfaces* (second edition). Dover Publications, INC.

Farin, G. E. (1993). *Curves and surfaces for computer aided geometric design : A practical guide* (W. Rheinbolt, Ed.; Third edition). Academic Press, INC.

Montiel, S., & Ros, A. (2000). *Curves and surfaces* (D. Babbitt, Ed.; second edition, Vol. 69). Editorial Board of Graduate Studies in Mathematics.

Radzevich, S. P. (2020). *Geometry of surfaces a practical guide for mechanical engineers* (second edition). Springer Nature Switzerland.

Rajaina, K., Bizzarri, M., Lávička, M., Kosinka, J., & Bartoň, M. (2023). Towards G^1-continuous multi-strip path-planning for 5-axis flank CNC machining of free-form surfaces using conical cutting tools. *Computer-Aided Design*, *163*, 103555. https://doi.org/https://doi.org/10.1016/j.cad.2023.103555

Skopenkov, M., Bo, P., Bartoň, M., & Pottmann, H. (2020). Characterizing envelopes of moving rotational cones and applications in CNC machining. *Computer Aided Geometric Design*, *83*, 101944. https://doi.org/https://doi.org/10.1016/j.cagd.2020.101944

Zhang, W. (2020, November). Geometry of curves and surfaces.

Zhu, L. M., Zhang, X. M., Zheng, G., & Ding, H. (2009). Analytical expression of the swept surface of a rotary cutter using the envelope theory of sphere congruence. *Journal of Manufacturing Science and Engineering*, *131*, 0410171–0410177. https://doi.org/10.1115/1.3168443

Zhu, L., & Lu, Y. (2015). Geometric conditions for tangent continuity of swept tool envelopes with application to multi-pass flank milling. *CAD Computer Aided Design*, *59*, 43–49. https://doi.org/10.1016/j.cad.2014.07.008