Toroidal Point Milling: the *how to* basics

Juan Zaragoza Chichell

Somewhere in time

Contents

1 Preliminaries

In the next pages we deal with the basic concepts of toroidal point milling from a mathematical viewpoint. For that matter, we will define the YA ESCRIBIRÁS BIEN LA INTRO

1.1 Free-form surfaces and contact curves

Let

$$S(u,v) = (\boldsymbol{x}(u,v), \boldsymbol{y}(u,v), \boldsymbol{z}(u,v)),\ (u,v) \in \Omega \subset \mathbb{R}^2$$

be a surface. We will denote the locus of $S(u,v)$ by $\boldsymbol{s}$, i.e.,

$$\boldsymbol{s} = \{(x,y,z) \in \mathbb{R}^3 \colon \exists (u,v) \in \Omega \colon (x,y,z) = S(u,v)\}.$$

Since we want $\boldsymbol{s}$ to be our objective surface, we will require of its parametrization $S(u,v)$ to be both smooth and orientable. Having $S(u,v)$ as a parametrization of class C^2 should suffice us.

Denote $S_u = \dfrac{\partial S(u,v)}{\partial u}$ and $S_v = \dfrac{\partial S(u,v)}{\partial v}$. Hence, the outwards normal field of $S(u,v)$ is

$$\boldsymbol{n}(u,v) = \frac{S_u \times S_v}{\|S_u \times S_v\|}.$$

Recall that the tool (which we will deal with later) is meant to move tangentially to to $\boldsymbol{s}$, and it is meant to do so by following some *contact curves*. A *contact curve* on $\boldsymbol{s}$ is a curve

$$c(t) := (\boldsymbol{x}(u(t),v(t)), \boldsymbol{y}(u(t),v(t)), \boldsymbol{z}(u(t),v(t))),\ t \in [t_0,t_1]$$

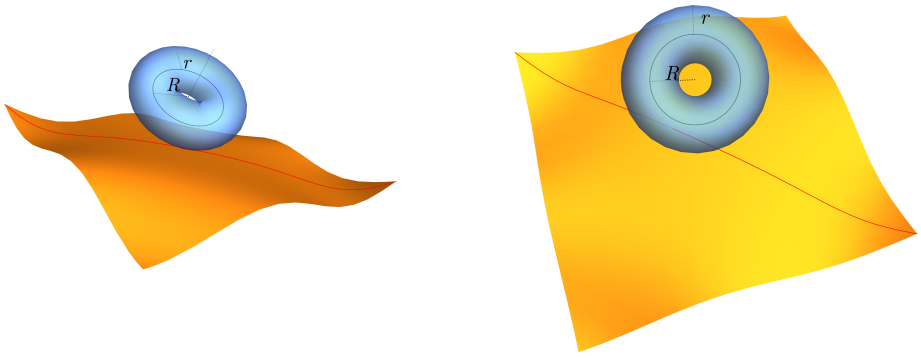
and we can consider, therefore, the Frenet frame along $c(t)$ defined by

$$\boldsymbol{T}(t) = \frac{\dot{c}(t)}{\|\dot{c}(t)\|},\ \boldsymbol{N}(t) = \boldsymbol{n}(u(t),v(t)),\ \boldsymbol{B}(t) = \boldsymbol{T}(t) \times \boldsymbol{N}(t)$$

1.2 Toroidal tools

The kind of tool we deal with are the so called *toroidal tools* or *toroidal cutters*, which, because the cut by revolving around an axis, can be modeled as a solid of revolution and, in particular, as a torus. GRÁFICAS DE LAS HERRAMIENTAS.

That being said, we denote the big radius of the torus by R and the small radius by r. See **??** for a sketch. We want to control tangential contact between the torus and the objective surface $\boldsymbol{s}$ along a (family of) contact curve $c(t)$ and not only that, but doing it while both rotating and tilting the tool, procedure that we will explain later on the text. However, controlling this tangential movement (as well as the collision control, which we will also explain later on the text) is not easy to do straight ahead: doing so to the whole torus involves dealing with a system of non-linear equations at every $t \in [t_0,t_1]$. A much more elegant approach to this issues is done by following the *offsetting argument*, which we will now deal with.



(a) (b)

Figure 1: Diagram of torus radii. (a) Side view of torus. (b) Perpendicular view of torus.

1.3 The offsetting argument

Consider our objective surface $\boldsymbol{s}$, which is parametrized by

$$S(u, v) = (\boldsymbol{x}(u, v), \boldsymbol{y}(u, v), \boldsymbol{z}(u, v))$$

as we had previously stated. Observe that, if a torus $\mathbb{T}$ with big radius R and small radius r is tangent to $\boldsymbol{s}$ at $\boldsymbol{p} \in \boldsymbol{s}$, there is always going to be a point on the inner circle that will be, at most, at distance r from $\boldsymbol{p}$.

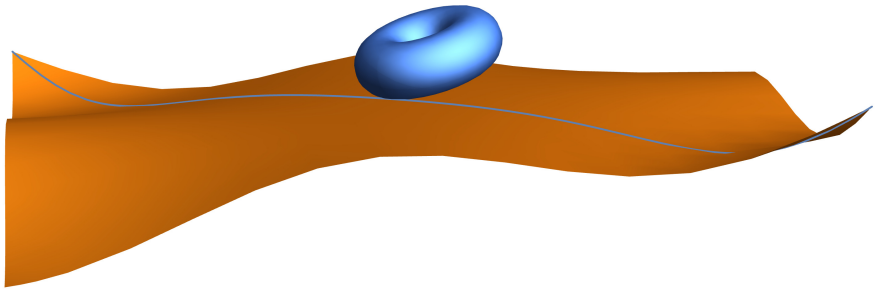


Figure 2: Surface, contact curve and a torus tangent to the curve.

This fact suggests something very interesting: the tangentiallity between the torus and the surface can be reduced to the tangentiallity between the

medial circle of $\mathbb{T}$, which we denote by $\mathbb{D}$, and the offset surface by distance r of $\boldsymbol{s}$. Let us define this properly.

Definition 1.1 (Offset of a surface)**.** *Let $\boldsymbol{s}$ be a smooth orientable surface parametrized by*

$$S(u,v) = (\boldsymbol{x}(u,v), \boldsymbol{y}(u,v), \boldsymbol{z}(u,v)),\ (u,v) \in \Omega \subset \mathbb{R}^2\,.$$

Then, we define the offset of $\boldsymbol{s}$ by distance $d \in \mathbb{R}$, and we denote it by $\bar{\boldsymbol{s}}_d$, as the locus of the following parametrized surface:

$$\bar{S}_d(u,v) = S(u,v) + d \cdot \boldsymbol{n}(u,v)$$

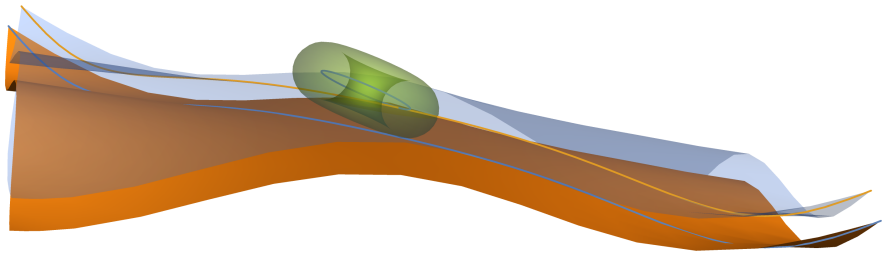


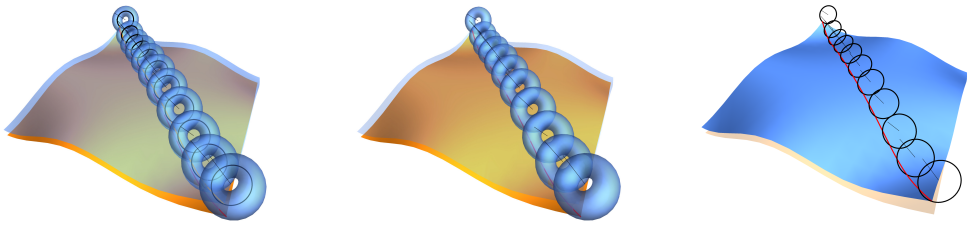
Figure 3: Surface (orange), its offset, (light purple), contact curves at the surface and at the offset, torus being tangent to the surface and its medial circle being tangent to the offset.

Recall that the distance d can be negative: it is a signed distance and it is considered as such intentionally, since there can be applications in which we might be interested in working on the inside of the surface. However, in the following, d will be considered positive unless specified otherwise.

Now, the tangential motion of the torus following the contact curve $c(t)$ is reduced to the tangential motion of the medial circle $\mathbb{D}$ along a contact curve on the offset $\bar{\boldsymbol{s}}_r$,

$$\bar{c}(t) = c(t) + r \cdot \boldsymbol{N}(t),\ t \in [t_0, t_1]$$

It is obvious that $\bar{S}_d(u,v)$ has the same normal field as $S(u,v)$ for every $d \in \mathbb{R}$, and in particular so it does for $d = r$. Still, that is not necessarily



(a) (b) (c)

Figure 4: (a) Multiple torus and their medial circle. Surface in orange and offset surface in transparent blue. Contact curve on surface and on offset both represented. (b) Multiple torus and their respective shank tangent to the surface along contact curve. Surface in orange and offset surface in transparent blue. (c) Multiple medial circles and their respective shank tangent to the offset surface (blue) along offset contact curve (red). Surface in transparent orange.

true for the rest of the elements on the Frenet frame along $\bar{c}(t)$. For that matter, we denote

$$\bar{\boldsymbol{T}}(t) = \frac{\dot{\bar{c}}(t)}{\|\dot{\bar{c}}(t)\|}, \ \bar{\boldsymbol{N}}(t) = \boldsymbol{N}(t), \ \bar{\boldsymbol{B}}(t) = \bar{\boldsymbol{T}}(t) \times \bar{\boldsymbol{N}}(t)$$

were we employ the notation $\bar{\boldsymbol{N}}(t)$ for consistency. We now have all that is required to deal with the rotating and tilting of the torus on the surface.

1.4 Positioning the medial circle

From now on, we will only deal with the medial circle and the offset surface. Positioning the medial circle tangentially to the $\bar{s}_r$ at $\boldsymbol{p} \in \bar{s}_r$ is easy, but doing it in a particular manner is kind of sophisticated. Because of gauging issues and error minimization, rotating and tilting the medial circle (torus) is key to obtaining satisfactory results. The rotation will be done around the normal vector of $\bar{s}_r$ at $\boldsymbol{p}$ and the rotation function is denoted by $\varphi(u, v)$. This rotation produces a new direction d which will act as the tilting axis (this will be discussed in this subsection); we will denote the tilting function by $\theta(u, v)$. Let us give all of this a little more sense.

1.4.1 Rotation

Consider that we want to place the circle $\mathbb{D}$ tangentially to $\bar{s}$ at $\boldsymbol{p}$, where $\boldsymbol{p} = \bar{c}(t^*)$ for some $t^* \in [t_0, t_1]$, and we want it to be rotated an angle φ around $\bar{\boldsymbol{N}}(t^*)$. Consider $\bar{\boldsymbol{N}} = \bar{\boldsymbol{N}}(t^*)$ and respectively with $\bar{\boldsymbol{T}}$ and $\bar{\boldsymbol{B}}$ for simplicity.

Since we want the circle $\mathbb{D}$ to lie tangentially to $\bar{s}$ at $\boldsymbol{p}$, a good starting position is to place it in the the tangent plane to $\bar{S}(u, v)$ at $\boldsymbol{p}$, $T_{\boldsymbol{p}}\bar{S} = \langle \bar{\boldsymbol{T}}, \bar{\boldsymbol{B}} \rangle$, being tangent to $\boldsymbol{p}$ and with center in the axis defined by $\bar{\boldsymbol{B}}$, i.e.,

$$\mathbb{D}(\alpha) = O + R\cos(\alpha)\,\bar{\boldsymbol{T}} + R\sin(\alpha)\bar{\boldsymbol{B}}$$

where $O = \boldsymbol{p} + R\bar{\boldsymbol{B}}$ is the center of $\mathbb{D}$. Rotating around $\bar{\mathbb{N}}$ an angle φ transforms $\bar{T}$ and $\bar{B}$ in the vectors

$$\bar{\boldsymbol{T}} \mapsto \boldsymbol{d} = \sin\varphi\bar{\boldsymbol{B}} + \cos\varphi\bar{\boldsymbol{T}},\ \ \bar{\boldsymbol{B}} \mapsto \boldsymbol{d}^{\perp} = -\sin\varphi\bar{\boldsymbol{T}} + \cos\varphi\bar{\boldsymbol{B}} = \boldsymbol{d}\times\bar{\boldsymbol{N}}$$

hence now $\mathbb{D}$ is parametrized as

$$\mathbb{D}(\alpha) = O' + R\cos(\alpha)\,\boldsymbol{d} + R\sin(\alpha)\,\boldsymbol{d}^{\perp}$$

where $O' = \boldsymbol{p} + R\,\boldsymbol{d}^{\perp}$.

Observe that $\mathbb{D}$ is tangent to $\bar{S}$ at $\boldsymbol{p}$ and that any rotation of $\mathbb{D}$ around $\boldsymbol{d}$ will keep this true. We have, therefore, found our *tilting around* direction.

1.4.2 Tilt

Let us assume we want to tilt our circle θ radians around $\boldsymbol{d}$. By construction, the circle will keep being tangential to $\bar{s}$ at $\boldsymbol{p}$ as we have stated previously. We only need to find the parametrization of our circle in that aspect. For that, we only need to find the center of the final circle, let us abuse of notation and call it O, as well as the image of $\boldsymbol{d}^{\perp}$, let us denote it by $\boldsymbol{\rho}$, under this rotation, since the circle will be such that lies in the plane spanned by $\langle \boldsymbol{d}, \boldsymbol{\rho} \rangle$, has center O and radius R.

It is obvious that $\boldsymbol{\rho} = \cos(\theta)\,\boldsymbol{d}^{\perp} + \sin(\theta)\bar{\boldsymbol{N}}$ and, with that, $O = \boldsymbol{p} + R\boldsymbol{\rho}$. Hence, $\mathbb{D}(\alpha) = O + R\cos(\alpha)\,\boldsymbol{d} + R\sin(\alpha)\boldsymbol{\rho}$ is our final circle. However, we would like to have $\alpha = 0$ corresponding to the contact point. Hence, the parametrization

$$\mathbb{D}(\alpha) = O + R(\sin(\alpha)\,\boldsymbol{d} - \cos(\alpha)\boldsymbol{\rho})$$

is considered. We will use $\boldsymbol{a}$ to denote the corresponding rotation of $\bar{\boldsymbol{N}}$, which corresponds to the tool axis.

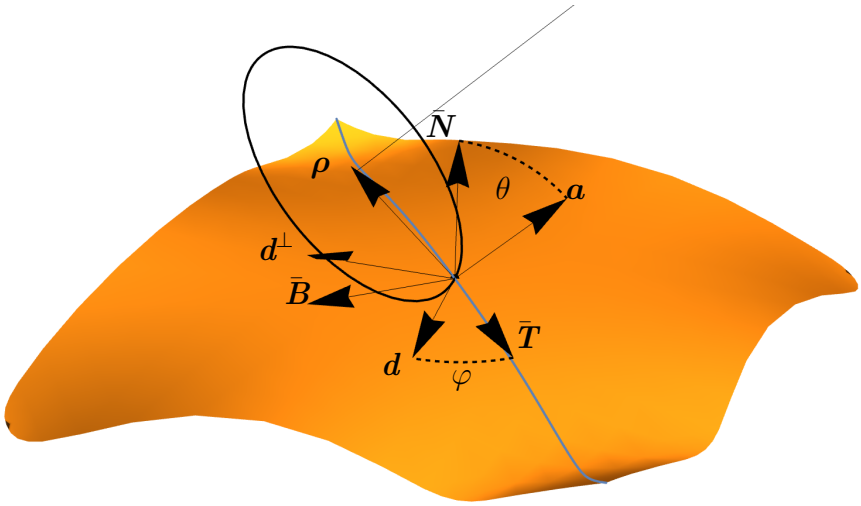


Figure 5: Scheme of offset surface, contact curve and circle tangent along contact curve.

2 Selecting the contact paths

Assume $N \in \mathbb{N}$ contact paths on the objective surface are desired. Those paths can be chosen to be the preimage of the level sets of a function $G : \Omega\,\mathbb{R}$. Of course, the set of solutions of the equation $G(u, v) - z = 0,\ z \in \mathbb{R}$ is formed either by a single point (if there is a single pair (u^*, v^*) for which $G(u^*, v^*) - z = 0$) or a finite number of curves. Let us describe how to obtain such level sets.

First, assume G to be a Bézier bi-variate function

$$G(u, v) = \sum_{i=0}^{3}\sum_{j=0}^{3} B_i^3(u)B_j^3(v)P_{i,j},\ \ P_{i,j} \in \mathbb{R}$$

and let $z^* \in \mathbb{R}$ be such that the level set $G(u, v) - z^* = 0$ is formed by, at least, one curve. We will first detect the roots on $\partial\Omega$ and start tracing the curves from there. In a general form, $\Omega = [u_0, u_{fin}] \times [v_0, v_{fin}]$, but, because it does only require a reparametrization, we can assume $\Omega = [0, 1]^2$ for this matter. Let us explain how to detect roots on the border line $u = 0$.

Because of properties of Bézier bi-variate surfaces,

$$G(0, v) = \sum_{j=0}^{3} B_j^3(v)P_{0,j}$$

i.e., it is a degree 3 polynomial. Denote a, b, c, d as $P_{0,j},\ j = 0, 1, 2, 3$ respectively and hence, we get that

$$\frac{\partial G}{\partial v} = -3v^2(a - 3b + 3c - d) + 6v(a - 2b + c) - 3a + 3b$$

and thus the relative extrema of $G(0, v)$ are found at

$$v^* = \frac{a - 2b + c \pm \sqrt{b^2 - c(a + b) + d(a - b) + c^2}}{a + 3(c - b) - d}$$

Of course, it could happen that neither of the roots (if they exist) lie on Ω or that only one of them did; control over this behavior must be taken since the detection of roots on the border will play a key role in the level set computation as we will now show. To detect the root on the border, we take advantage of the knowledge of v^*: if both values lie in Ω we search for the root in the interval defined by the two values $v^*_-, \quad v^*_+$.

The equation $G(0, v) - z^* = 0$ has solution if and only if $(a - z^*)(d - z^*) \leq 0$. In that case, there is, at least, one intersection point and three at most. To find them, denote $v_0 = \min\{v^*_-, v^*_+\},\ v_1 = \max\{v^*_-, v^*_+\}$ and so consider the set $\mathcal{T} = \{0, v_0, v_1, 1\}$. Assume both $v_0, v_1 \in [0, 1]$. Hence, we check for possible roots on the subintervals defined by the elements of $\mathcal{T}$ by checking the sign of $(G(0, \nu_i) - z^*)(G(0, \nu_{i+1})^* - z^*)$ for $(\nu_0, \ldots, \nu_3 = 0, v_0, v_1, 1)$ and, if the sign is nonpositive, we can look for the root employing a Newton-Raphson method with initial estimation $\bar{v} = \dfrac{\nu_i + \nu_{i+1}}{2}$. This process will detect every possible root on the border $u = 0$ and, by extending it to the other borders, we can compute every root found on the border.

Assume now $r_0 = (u_0, v_0)$ is a border root of $G(u, v) - z^* = 0$. Let us define the curve tracing algorithm that we employ. The intuitive notion is that we would like to move tangentially to the curve; $\nabla G(r_0)$ gives us the direction of maximum increase; $\nabla G(r_0)^\perp$ gives, hence, the direction in which the increase is null. Denote by $w_0 = \dfrac{\nabla G(r_0)^\perp}{\|\nabla G(r_0)^\perp\|}$ and $x_0 = r_0 + \delta w_0$; if $x_0 \notin \Omega$ that means that we are moving "outside" of the region, hence the sign of w_0 should be switched. We will consider that w_0 has the appropriate sign. Now, x_0 is a point outside of the curve that lies on the tangent line to the curve that passes throw r_0. Let L_0 be the line that passes throw x_0 and that is perpendicular to w_0, i.e., $L_0(u, v) \equiv \langle w_0, (u, v)^T - x_0 \rangle$. Provided that δ is small enough, the system of equations

$$G(u, v) - z^* = 0, \quad L_0(u, v) = 0$$

does have a solution; if more than one solution is found, keep the one closest to x_0 and call such root r_1; of course r_1 satisfies $G(u,v) - z^* = 0$, therefore being a point on the curve. Repeat the process until you exit Ω (making sure that, at every iteration, the adequate sign of w_k is given)

3 Selecting the tilt and rotation functions

Previously, the values for φ and θ where selected as scalar values at the given contact points. However, such scalar values must be selected somehow. Our approach considers two bi-variate functions, namely $\varphi, \theta \colon \Omega \longrightarrow \mathbb{R}$ with images in $[0, 2\pi)$ and $[0, \frac{\pi}{2}]$ respectively. Because we wish to control the values of $\varphi(u,v)$ and $\theta(u,v)$, we consider them to be polynomials of degree 3 defined as Bézier surfaces, therefore each one being controlled by a mesh of 16 control points $P^{\varphi}_{i,j}, P^{\theta}_{i,j}, i, j = 0, \ldots, 3$, i.e.,

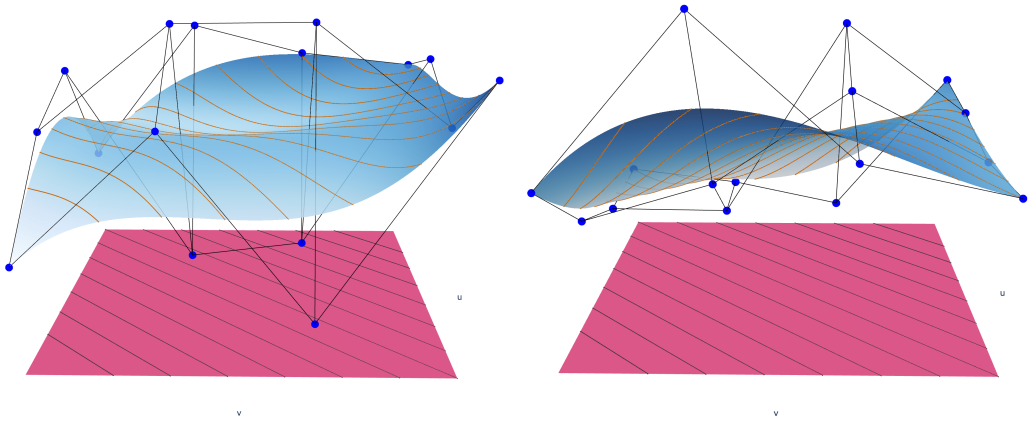
$$\varphi(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} B^3_i(u)B^3_j(v)P^{\varphi}_{i,j}, \quad \theta(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} B^3_i(u)B^3_j(v)P^{\theta}_{i,j}$$

where in this definition Ω is being considered to the unit square; if not, it must be reparametrized as such. By doing this, we have assigned both a rotation and tilt angles for every point in the domain, therefore the position of the torus (medial circle) is completely determined along the contact curve.

3.1 Selecting $\theta(u,v)$ in terms of $\varphi(u,v)$

While both functions can be selected randomly as a starting position for the iterative process, we could help the system by providing a much sensible starting position. For that matter, we will considered $\varphi(u,v)$ fixed but arbitrary and we will provide an appropriate choice of $\theta(u,v)$ such that it minimizes the error in the first place: the values of $\theta(u,v)$ will be determined by the Meusnier sphere. Let us explain this properly.

Again, consider the offset surface $\bar{\boldsymbol{s}}$ and assume we have already rotated the medial circle around $\bar{\boldsymbol{N}} = \bar{\boldsymbol{N}}(t^*)$ at a point $\boldsymbol{p} = \bar{c}(t^*), t^* \in [t_0, t_1]$, therefore we have both $\boldsymbol{d}$ and $\boldsymbol{d}^{\perp}$ well defined. Now, Meusnier's theorem tells us that every curve on $\bar{\boldsymbol{s}}$ passing through $\boldsymbol{p}$ and with tangent vector at $\boldsymbol{p}$ equal to $\dot{\bar{c}}(t^*)$ have the same normal curvature and that their osculating circles all lie in a sphere; such sphere is called *Meusnier's sphere* and has radius $r_p = \frac{1}{k_N}$ and center $\boldsymbol{p} + r_p\bar{\boldsymbol{N}}$, where k_N is the normal curvature of $\bar{\boldsymbol{s}}$ at $\boldsymbol{p}$ in the direction $\boldsymbol{d}^{\perp}$. Then, a second order line contact is achieved when the medial circle is



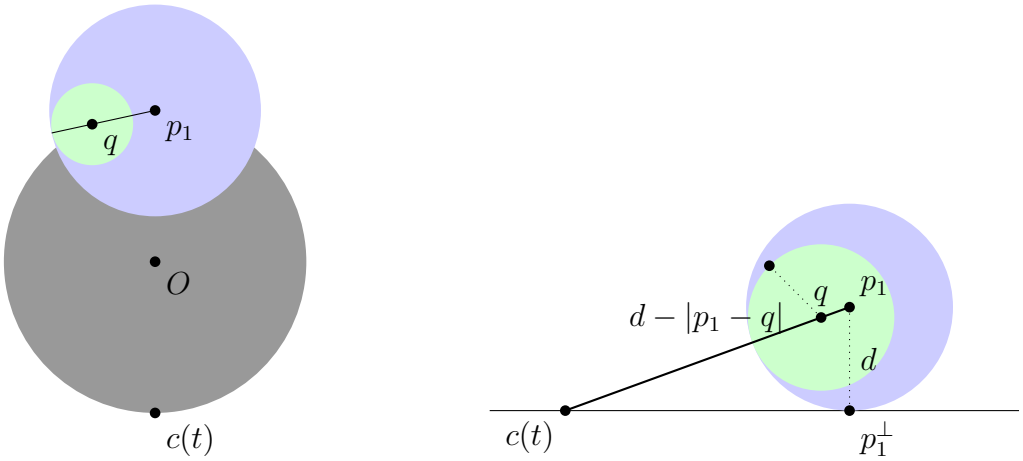
(a) Graphic of rotation function $\varphi(u, v)$. (b) Graphic of tilt function $\theta(u, v)$.

Figure 6: Example of rotation / tilt functions. Plane $z = 0$ in gray; black lines on $z = 0$ plane represent multiple parametric curves $(u(t), v(t))$. Surface in blue shows the graphic of $\varphi(u, v)$ $(\theta(u, v))$ as a bicubic Bézier surface determined by the shown control net (blue points). Black lines connecting blue points represent the control polygon. Red line on blue surface represent $\varphi(u(t), v(t))$ $(\theta(u(t), v(t)))$ curves.

tilted so that it is placed in contact with the sphere, and that is achieved when the tilting angle is $\theta = \sin^{-1}(\frac{R}{r_p})$. Because r_p depends on $\boldsymbol{d}^{\perp}$ and that depends on the value of $\varphi(u, v)$, this procedure defines θ as a function of (u, v). IMAGEN ESQUEMA THETA EN MEUSNIER

4 Global Collision Control

In this section we describe the procedure for collision detection. Collision detection plays a really important role in all of this procedure; detecting properly colliding positions will allow us to rectify positions of the tool that penetrate on the surface. However, if we were to consider the smooth case for both the surface and the tool in order to detect the maximum penetration point, we would have to deal with a system of non-linear equations for every point in the contact curve, which would be slow for execution. In contrast, we deal with a discrete version of both the surface and the tool, which reduces the problem of collision finding to a system of linear equations, much faster to solve and, given a fine enough mesh for both the surface and the tool, also accurate enough.



(a) Top view

(b) Side view.

Figure 7: Head Collision Scheme

In the following, consider $\bar{\boldsymbol{s}}$ to be a fine mesh.

In the following, the next definition will play a key role:

Definition 4.1 (Signed distance)**.** *Let S be our oriented surface and $q \in \mathbb{R}^3$ be any point. Denote by $q^\perp$ q's footpoint on S. Assume $q^\perp = S(u^*, v^*)$ for some particular $(u^*, v^*) \in \Omega$. We define the signed distance between q to $q^\perp$ by*

$$d_{sgn}(q, q^\perp) := sgn(\langle q - q^\perp, \boldsymbol{n}(u^*, v^*)\rangle)|q - q^\perp| \tag{1}$$

where $|q - q^\perp|$ denotes the Euclidean distance between q and $q^\perp$.

Observe that the previous definition is not symmetrical, since

$$d_{sgn}(q, q^\perp) = -d_{sgn}(q^\perp, q)$$

, hence it is not a *distance* in the strict sense.

THIS NEEDS BETTER PHRASING.

4.1 Head Collision

Denote by

$$\mathbb{D}(\alpha) = O + R(\sin(\alpha)\,\boldsymbol{d} - \cos(\alpha)\boldsymbol{\rho}),\ O = p + R\boldsymbol{\rho} \tag{2}$$

the medial circle tangent to the surface S at point $p \in S$ of radius $R > 0$. Recall that Equation 2 provides a parametrization of the medial circle with

center O, lying in the plane spanned by $\langle \boldsymbol{d}, \boldsymbol{\rho} \rangle$ such that for $\alpha = 0$, the obtained point is the contact point with the surface. In that sense, we need the following definition:

Definition 4.2 (Head collision)**.** *Let $q = \mathbb{D}(\alpha^*),\ \alpha^* \in (0, 2\pi)$, be a point of the medial circle and denote $q^\perp$ its footpoint on the surface. We say that q causes a head collision if $d_{sgn}(q, q^\perp) < 0$.*

Proposition 4.1 (Head collision criteria)**.** *Let $p = \mathbb{D}(\alpha),\ \alpha \in (0, 2\pi)$ be a point of the medial circle and let $p^\perp$ be its footpoint and assume $d = d_{sgn}(p, p^\perp) > 0$. Hence, if $q = \mathbb{D}(\beta),\ \beta \in (0, 2\pi) \backslash \alpha$ satisfies $|p - q| < d$, then $d_{sng}(q, q^\perp) > 0$.*

Proof. Observe that if $d_{sng}(q, q^\perp) < 0$, that would mean that the surface needs to pass between p and q, but that cannot happen since the open ball $B_p^\circ(d)$ is a save (open) sphere, meaning that no point of the surface lies inside of it. We conclude that $d_{sng}(q, q^\perp) > 0$. □

The previous result suggests us that, in order to detect head collisions, we do not need to compute footpoints on every point of the head but rather pick one, compute its footpoint and select the points of the head that lie at exactly that same distance. It is easy to verify that $|p - q| = 2R|\sin(\frac{\alpha-\beta}{2})|$, and since we look for $\beta \in (0, 2\pi)$ satisfying $|p - \mathbb{D}(\beta)| = d$, we have that $\beta = \alpha - 2\arcsin(\frac{d}{2R})$. This gives us a new point to check and proceeding, we can check until a collision is found or we are close enough to $c(t)$.

4.2 Shank Collision

In order to detect a shank collision, we proceed on a similar manner. Let us provide the following definition:

Definition 4.3 (Shank collision)**.** *Let $[p_0, p_N]$ be the -medial line of the-shank of radius $R > 0$ and let*

$$p = (1 - \mu)p_0 + \mu p_N, \mu \in [0, 1]$$

be any point on the shank. The point p is said to be colliding if both of the following conditions are satisfied:

1. $d_{sng}(p, p^\perp) < R$
2. $\langle p_N - p_0, p^\perp - p_0 \rangle > 0$

Because footpoint evaluation is expensive, we want to find (similar to the head collision) a test that allows us to minimize the number of points to evaluate its colliding condition. In that sense, let us give the following propositions.

Proposition 4.2 (Maximal Save Distance)**.** *Let $p \in [p_0, p_N]$ and assume $d = d_{sng}(p, p^\perp) > R$. Let $q \in [p_0, p_N]$ be another point and let $\Delta = |p - q|$. If $d - \Delta > R$ then $d_{sng}(q, q^\perp) > R$.*

Proof. Let $[p_0, p_N]$ be a shank, $p, q \in [p_0, p_N]$ be two points in the shank and $p^\perp, q^\perp$ their footpoints respectively and $d = d_{sng}(p, p^\perp) > R$, $\Delta = |p - q|$ and $\bar{d} = d_{sng}(q, q^\perp)$. It is clear that $|p - q^\perp| \geq d$ and $\bar{d} \leq |q - p^\perp|$ by construction and the main definition of footpoint.

Hence, $q^\perp \in \bar{B}_q(|q - p^\perp|) \backslash B_p^\circ(d)$. Denote by $c_1 = \partial B_p(d)$, $c_2 = \partial B_q(|q - p^\perp|)$ and

$$r(\lambda) = p + \lambda \frac{\vec{qp}}{\|\vec{qp}\|}$$

be the line of the shank. It is obvious that $P_1 = r(\lambda) \cap c_1$ exists and satisfies

$$|P_1 - q| = d - \Delta.$$

Likewise, $Q_1 = r(\lambda) \cap B_q(\bar{d})$ exists too.

Finally,

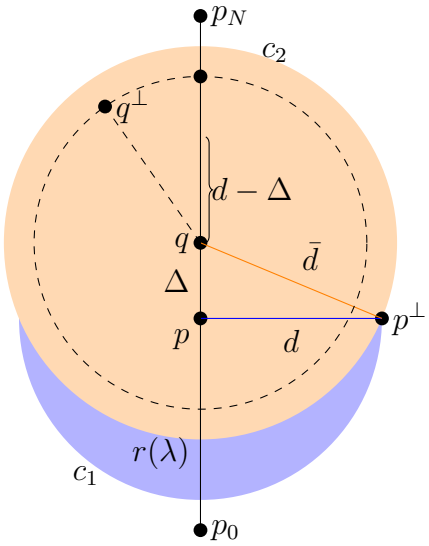
$$|q - q^\perp| = |q - Q_1| \geq |q - P_1| = d - \Delta > R$$

. It only remains left to see that the sign is positive, which is obvious since for $d_{sng}(q, q^\perp)$ and $d_{sng}(p, p^\perp)$ to have different signs, the surface would have to pass between p and q, and that is not possible since that would imply that there is a point of the surface closer to p than $p^\perp$ and that is a contradiction with our hypothesis.
[1]

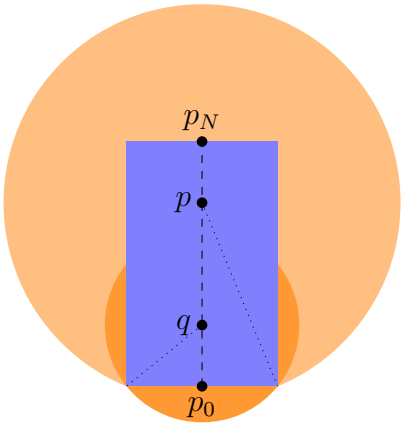
□

Proposition 4.3 (Bad Tool Side Criteria)**.** *Let $p \in [p_0, p_N]$ be a point in the shank and let $p^\perp$ be its footpoint. If the footpoint lies on the "bad side" of the tool, then, so does every point $q \in [p_0, p]$.*

[1] ASK MICHAEL IF ABATEMENT ARGUMENT IS BETTER AND THEN PROCEED FROM DRAWING.



(a) Scheme of proof for Proposition 4.2.



(b) Scheme of proof for Proposition 4.3

Figure 8: Schemes of "proofs" for the claims. In both, shank is the segment $[p_0, p_N]$.

Proof. Let $p \in [p_0, p_N]$ be such that its footpoint $p^\perp$ lies on the bad side of the tool and let $q \in [p_0, p]$. Recall that

$$d_{sng}(p, p^\perp) \leq \sqrt{|p - p_0|^2 + R^2}, d_{sng}(q, q^\perp) \leq \sqrt{|q - p_0|^2 + R^2}$$

since those are the corresponding distance to the contact point of the tool and the surface from p and q respectively. Hence, if $q^\perp$ lied on the good side of the tool it would mean that the distance from p to $q^\perp$ would be less than the one to $p^\perp$, which is not possible. We conclude that $q^\perp$ lies on the bad side of the tool too.

□

This two results combined provide us with a way of optimally checking for shank collisions: usually, the shank would be close-to-normal to the surface, hence the point of the shank that will be the farthest away from the surface will, usually, be p_N. From Proposition 4.3 we get that if $p_N^\perp$ lies on the bad side of the tool, then so does every point in the shank and the shank is non-colliding. Otherwise, Proposition 4.2 tells us that the biggest save distance will be achieved at precisely that point and, denoting by d the distance from p_N to its footpoint, it will be $d_{save} = d - R$ and the process is then repeated at

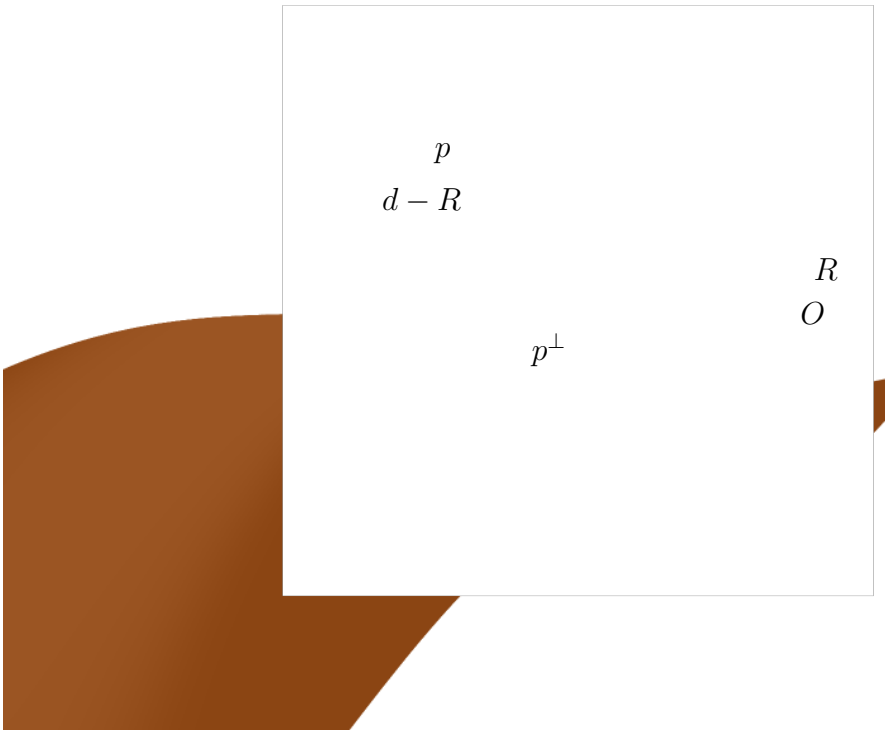


Figure 9: Scheme of Proof 8a with maximum save distance.

the point of the shank that is precisely at that distance from p_N. Having that $p = (1-\mu)p_N + \mu p_0$ tells us that $d(p, p_N) = \mu|p_0 - p_N|$ therefore the point that lies at distance $d - R$ from p_N in the shank is that of parameter $\mu = \frac{d-R}{|p_0-p_N|}$. Observe that p will lie in the shank if and only if $0 \leq \mu \leq 1$, otherwise, we can conclude that the shank is non-colliding. See Figure 10. This way, we have an optimal and error-free procedure to detect shank collisions. It could, of course, occur that the shank had an almost tangential point, that is, $\exists p \in [p_0, p_N]$:

1. $R < |p - p^{\perp}| < R + \epsilon,\ \epsilon > 0$
2. $\langle p_N - p_0, p^{\perp} - p_0 \rangle > 0$

If such a point is found, due to safety reasons, we will say that such point is colliding, since any small perturbation of the machine could cause a shank collision.

Reciprocally to 4.2, we have that, if $p \in [p_0, p_N]$ and $|p - p^{\perp}| < R$, any other point of the shank $q \in [p_0, p_N]$ will satisfy

$$|q - p^{\perp}| \geq |q - q^{\perp}|$$

hence verifying that, if $|q - p^{\perp}| < R$, q is warrantied to be colliding.

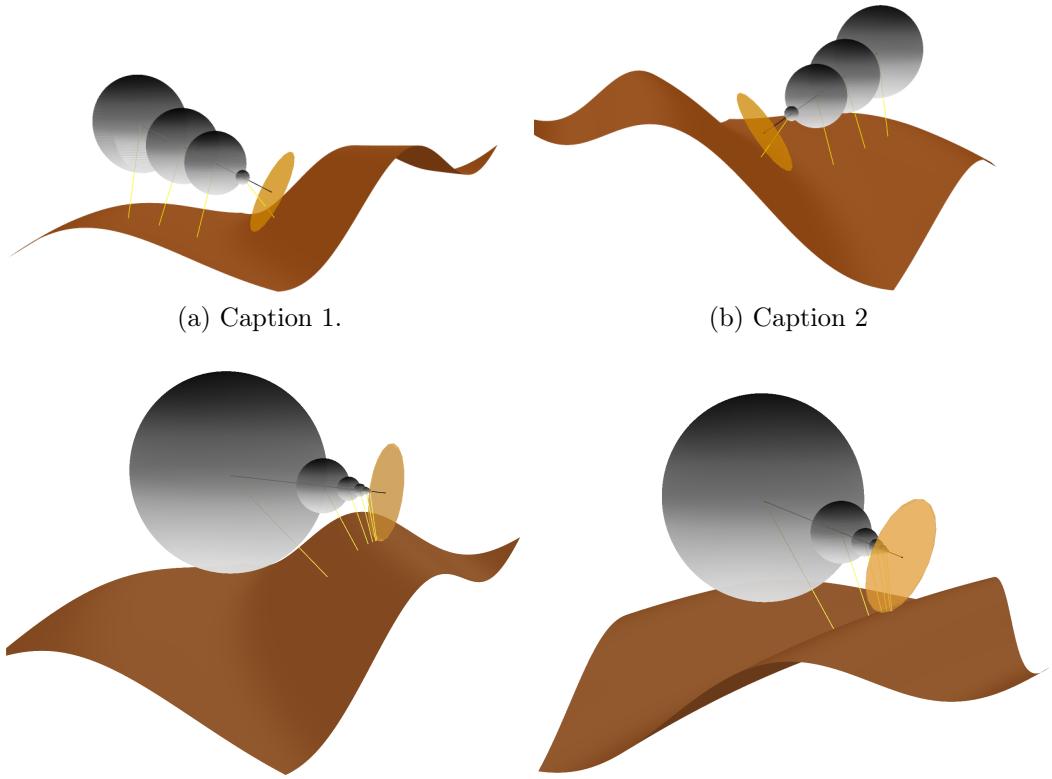
5 Modified shank collision detection

The robustness of the above described method does not come from taking the points that lie exactly on the boundary of the sphere, but rather of assuring that the spheres of maximum save distance of two points intersect. If such thing happens, we can assure no points between the two evaluated points is colliding. Therefore, our aim now is to detect the intersection of spheres in the most efficient way, Allow us to describe the protocol:

Let $[p_N^0, p_0]$ be our shank of radius R, parametrized as

$$r(\mu) = (1 - \bar{\mu})p_N^0 + \bar{\mu}p_0$$

and let ${p_N^0}^{\perp}$ be p_N's footpoint on S. Denote by $d_0 = d_{sgn}(p_N^0, {p_N^0}^{\perp}) - R$ the save distance of p_N^0 and by $L = |p_N^0 - p_0|$ the length of the shank. According to our previously discussed method, the next point to chose would be $p_N^1 = r(\mu_0)$ where $\mu_0 = \frac{d_0}{L}$. However, let $\bar{\mu}_0 = \frac{d_0+h}{L},\ h > 0$ and let $p_N^1 = r(\bar{\mu}_0)$, ${p_N^1}^{\perp}$ its footpoint and $d_1 = d_{sgn}(p_N^1, {p_N^1}^{\perp}) - R$ its save distance. Of course, the sphere



(a) Caption 1.

(b) Caption 2

(c) Caption 3.

(d) Caption 4

Figure 10: Earth toned carpet represents objective surface. Orange circle (ellipse) represents medial circle of the cutting tool. Black lines represent the shank. Foot point from the top of the shank is computed, represented by the furthest yellow line to the circle. Grey sphere represents the sphere of radius the safe distance from the top of the shank to the surface and its intersection with the shank provides the next point to evaluate. Consecutive grey spheres and yellow lines represent the respective save distances and footpoints.

of save distance of center p_N^1 and radius d_1 has two intersections with $r(\mu)$, those being at

$$\mu_1^+ = \frac{d_0 + h + d_1}{L},\ \mu_1^- = \frac{d_0 + h - d_1}{L}$$

hence, if $\mu_1^- \leq \mu_0$, there is intersection of spheres and we can assure that all the segment between p_N^0 and $r(\mu_1^+)$ is collision free. If, on the contrary, $\mu_1^- > \mu_0$, there would be a segment between $r(\mu_1^-)$ and $r(\mu_0)$ where we would not know if a collision is taking place. Hence, that segment shall be inspected on itself.

Assume there was intersection of spheres. Let $\mu_1 = \mu_1^+$ and $\bar{\mu}_1 = \mu_1 + \frac{h}{L}$, $p_N^2 = r(\bar{\mu}_1)$, ${p_N^2}^\perp$ its footpoint and d_2 its save distance. Repeating the process, denote by $\mu_2^- = \frac{d_0+d_1+2h-d_2}{L}$. If $\mu_2^- < \mu_1$, then we have intersection of spheres, otherwise, the line segment between $r(\mu_1)$ and $r(\mu_2^-)$ should be inspected.

In general, if we have m consecutive intersecting spheres, the next point we want to evaluate is $p_N^{m+1} = r(\mu_m + \frac{h}{L})$. Assume it has save distance d_{m+1}. Hence, parameter $\mu_{m+1}^- = \frac{\sum_{i=0}^{m} d_i + (m+1)h - d_{m+1}}{L}$ is the one we have to compare with μ_m to detect intersection of spheres: if $\mu_{m+1}^- \leq \mu_m$, we have intersection of spheres.

6 Determining Milling Error

Consider a circle of radius $R > 0$ at a given position as we had described previously, i.e., $\mathbb{D}(\alpha) = O + R(\sin(\alpha)\,\boldsymbol{d} - \cos(\alpha)\boldsymbol{\rho})$, where $O = \boldsymbol{p} + R\boldsymbol{\rho}$. Observe that $\boldsymbol{p}, \boldsymbol{d}, \boldsymbol{\rho}$ can be described in terms of the time parameter: if $\varphi(t) := \varphi(u(t), v(t))$, $\theta(t) := \theta(u(t), v(t))$, we have

- $\boldsymbol{p} \longrightarrow \bar{c}(t)$
- $\boldsymbol{d} \longrightarrow \boldsymbol{d}(t) = \sin(\varphi(t))\bar{\boldsymbol{B}}(t) + \cos(\varphi(t))\,\bar{\boldsymbol{T}}(t)$
- $\boldsymbol{d}^\perp \longrightarrow \boldsymbol{d}^\perp(t) = \boldsymbol{d}(t) \times \bar{\boldsymbol{N}}(t)$
- $\boldsymbol{\rho} \longrightarrow \boldsymbol{\rho}(t) = \cos(\theta(t))\,\boldsymbol{d}^\perp(t) + \sin(\theta(t))\bar{\boldsymbol{N}}(t)$

  therefore,

$$\mathbb{D}(\alpha, t) = O(t) + R(\sin(\alpha)\,\boldsymbol{d}(t) - \cos(\alpha)\boldsymbol{\rho}(t)),\ O(t) = \bar{c}(t) + R\boldsymbol{\rho}(t)$$

describes the motion of the circle following a tangential contact with $\bar{S}$ at every $t \in [0, 1]$ (for $\alpha = 0$ we have $\mathbb{D}(\alpha, t) = \bar{c}(t)$), with its center moving

along $O(t)$ and being rotated and tilted at each time t according to $\varphi(t)$ and $\theta(t)$ respectively.

Such description of $\mathbb{D}(\alpha, t)$ is, as a matter of fact, the parametric representation of the envelope of the circle in motion. The milling error of that circle is 0 along $\bar{c}(t)$ due to tangentiality and it has its maximum along $\mathbb{D}(\pm\frac{\pi}{2}, t)$ except for particular scenarios.

Now, consider two different curves $\bar{c}_1(t), \bar{c}_2(t)$ on $\bar{S}$. Such two curves describe the motion of two circles of radius $R > 0$ and, therefore, they define two envelopes,

$$E_i(s,t) = O_i(t) + R(\sin(s)\,\boldsymbol{d}_i(t) - \cos(s)\boldsymbol{\rho}_i(t)),\ O_i(t) = \bar{c}_i(t) + R\boldsymbol{\rho}_i(t),\ i = 1, 2$$

where $\boldsymbol{d}_i(t), \boldsymbol{\rho}_i(t)$ are the corresponding to each $\bar{c}_i(t)$.

If the distance between $\bar{c}_1(t)$ and $\bar{c}_2(t)$ is of the appropriate size, the intersection of the locus of both envelopes will be a continuous curve, along which the error will be maximal. We need, then, to have explicit knowledge of such intersection. In the general scenario, both $E_i(s,t), i = 1, 2$ are *tube-like*, therefore, two intersection curves would be expected, however, it is obvious that we only need the information regarding the intersection that is closest to the surface, that is, the intersection of the bottom-half of the envelopes, which corresponds to $E_i(s,t),\ s \in [-\frac{\pi}{2}, \frac{\pi}{2}],\ i = 1, 2$.

Recall that, although defined in the same parametric space,

$$(s,t) \in \Omega = [-\frac{\pi}{2}, \frac{\pi}{2}] \times [0, 1]$$

they must not be confused between each other, since they serve to define different creatures. Having said that, observe that the intersection of E_1, E_2, despite being a 3D curve, will be defined by a family of 4D vectors, namely

$$C_{E_1 \cap E_2} = \{(s,t,u,v) \in \Omega^2 \colon E_1(s,t) = E_2(u,v)\}$$

The effective computation of $C_{E_1 \cap E_2}$ comes with one problem: its definition states a system of three non-linear equations with four unknowns, hence not being well determined.

Surface-surface intersection is a common problem in CAGD and so literature on the topic is bast LITERATURA ON THE TOPIC.

However, due to our specific scenario, the problem can be simplified just by making use of some *a priori* knowledge of the envelopes: since we know that both parameters $t, v \in [0,1]$, if we want to obtain $N+1 \in \mathbb{N}$ number of points in the curve, we can solve for $N+1$ systems of nonlinear equations

$$\Sigma_k \equiv \{E_1(s,t) - E_2(u,v_k) = (0,0,0)\},\ v_k = \frac{k}{N},\ k = 0,\ldots,N.$$

By fixing the value of the v parameter, each of the above $N+1$ systems of equations has three equations and three unknowns, therefore no more additional conditions must be added so that a solution can be found. Still, appropriate choice of the initial approximation must be provided so that a numerical scheme can be applied. Such appropriate initial approximation to the first root can be given as $x_0^* = (s_0^*, t_0^*, u_0^*)$ where the particular values will be set as follows:

- If $\bar{c}_1(t), \bar{c}_2(t)$ have the same orientation, i.e., $\langle \dot{\bar{c}}(t)_1, \dot{\bar{c}}(t)_2 \rangle > 0$, then $t_0^* = \frac{1}{N}$. Otherwise, $t_0^* = 1 - \frac{1}{N}$.
- If $\bar{c}_1(t), \bar{c}_2(t)$ have the same orientation, $s_0^* = -u_0^*$ and the value of u_0^* will be either $\frac{\pi}{4}$ or $-\frac{\pi}{4}$, depending on which one lies on which side. Otherwise, $s_0^* = u_0^*$ with same particular values following the same choice criteria.

Now, solving Σ_0 with inital approximation x_0^* yields the solution $r_0 = (s_0, t_0, u_0)$. Because of the continuous (and smooth) behavior of both envelopes, we can expect parameters s, u to change smoothly too, hence, we can set $x_1^* = (s_0, t_0 + \frac{1}{N}, u_0)$ (or $x_1^* = (s_0, t_0 - \frac{1}{n}, u_0)$ depending on the orientation) as the next initial approximation and proceed until r_N is computed.

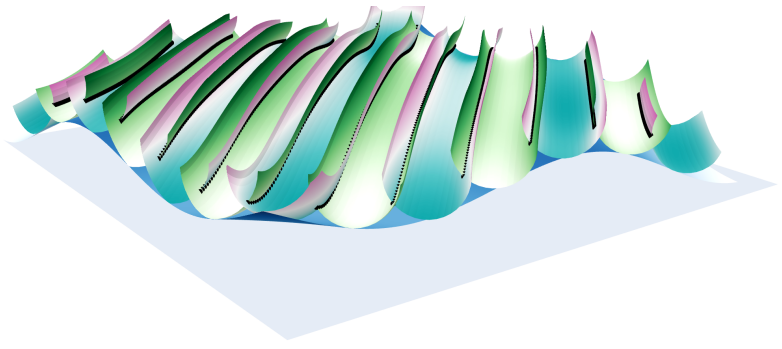


Figure 11: Surface, envelopes on surface and points on intersection curves obtained by the algorithm described.