

Geometry and tool motion planning for curvature adapted CNC machining

MICHAEL BARTOÑ, Basque Center for Applied Mathematics & Ikerbasque, Spain

MICHAL BIZZARRI, University of West Bohemia, Czech Republic

FLORIAN RIST, KAUST & TU Vienna, Saudi Arabia & Austria

OLEKSII SLIUSARENKO, Basque Center for Applied Mathematics, Spain

HELMUT POTTMANN, KAUST, Saudi Arabia

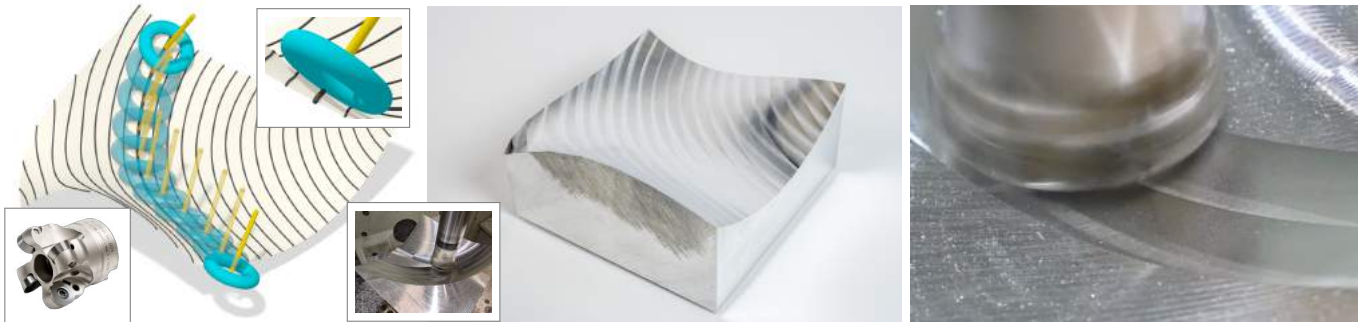


Fig. 1. Left: Given a cutting tool (bottom framed) and a reference free-form surface (light), the tool is conceptualized as a moving torus and its highly-accurate positions (top framed) as well as the milling paths (black) are computed by our optimization-based framework to result in an accurate, collision-free milling process. Middle: The final machined workpiece whose physical realization was achieved by 5-axis CNC machining (bottom framed). The milling strip density is intentionally set coarse to visually distinguish the neighboring strips. Right: The tool spinning with high-speed and two paths of high-quality surface finish.

CNC machining is the leading subtractive manufacturing technology. Although it is in use since decades, it is far from fully solved and still a rich source for challenging problems in geometric computing. We demonstrate this at hand of 5-axis machining of freeform surfaces, where the degrees of freedom in selecting and moving the cutting tool allow one to adapt the tool motion optimally to the surface to be produced. We aim at a high-quality surface finish, thereby reducing the need for hard-to-control post-machining processes such as grinding and polishing. Our work is based on a careful geometric analysis of curvature-adapted machining via so-called second order line contact between tool and target surface. On the geometric side, this leads to a new continuous transition between “dual” classical results in surface theory concerning osculating circles of surface curves and osculating cones of tangentially circumscribed developable surfaces. Practically,

it serves as an effective basis for tool motion planning. Unlike previous approaches to curvature-adapted machining, we solve locally optimal tool positioning and motion planning within a single optimization framework and achieve curvature adaptation even for convex surfaces. This is possible with a toroidal cutter that contains a negatively curved cutting area. The effectiveness of our approach is verified at hand of digital models, simulations and machined parts, including a comparison to results generated with commercial software.

CCS Concepts: • **Computing methodologies** → **Modeling and Simulation**; *Optimization algorithms*.

Additional Key Words and Phrases: Computer-Aided Manufacturing, CNC machining, curvature adapted machining, computational fabrication, path planning, motion planning

ACM Reference Format:

Michael Bartoñ, Michal Bizzarri, Florian Rist, Oleksii Sliusarenko, and Helmut Pottmann. 2021. Geometry and tool motion planning for curvature adapted CNC machining. *ACM Trans. Graph.* 1, 1, Article 180 (April 2021), 16 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

Authors' addresses: Michael Bartoñ, Basque Center for Applied Mathematics & Ikerbasque, Alameda Mazarredo 14, Bilbao, 48009, Spain, mbarton@bcamath.org; Michal Bizzarri, University of West Bohemia, Univerzitní 8, Pilsen, 301 00, Czech Republic, bizzarri@kma.zcu.cz; Florian Rist, KAUST & TU Vienna, P.O. Box 2187, 4700 Thuwal, 23955-6900, Saudi Arabia & Austria, florian.rist@kaust.edu.sa; Oleksii Sliusarenko, Basque Center for Applied Mathematics, Alameda Mazarredo 14, Bilbao, 48009, Spain, osliusarenko@bcamath.org; Helmut Pottmann, KAUST, P.O. Box 2187, 4700 Thuwal, 23955-6900, Saudi Arabia, helmut.pottmann@kaust.edu.sa.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

0730-0301/2021/4-ART180 \$15.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 INTRODUCTION

Geometric modeling has been motivated by industrial needs at the advent of computer-aided manufacturing, aiming at increased productivity via a completely digital workflow from design to production. Nowadays, the possibilities for digital shape design are almost unlimited, but there is still room for improvement in the area of geometric computing for manufacturing.

One of the most important fabrication techniques is *Computer numerically controlled (CNC) machining*, where a piece of raw material is cut into a final desired shape by a controlled material removal process. Material is cut away by a moving tool (see Fig. 1 and Fig. 2, top row) that spins with high speed around its axis. CNC machining (aka milling) is the leading subtractive manufacturing technology and a key part in the production of molds, usually from metal, but it can also be used on materials such as wood, plastic, ceramic and composites.

Remarkably, CNC machining has received much less interest within Computer Graphics than the more recent additive manufacturing technologies. However, as we will show, there is a variety of methods from classical geometry, geometric modeling and optimization, which can be effectively used for advances in CNC machining.

The initial stage of CNC machining, called *roughing*, starts with a solid material block from which most of redundant material is removed. This proceeds in parallel layers, leading to staircase effects near freeform surfaces. Roughing may be followed by *semi-finishing* where the most prominent staircase effects are removed. We are interested in the final *finishing* stage, which has the goal of producing nearly perfectly smooth surfaces.

Our focus is on *path planning for curvature adapted CNC machining of freeform surfaces*. The main idea here is to move the cutting tool such that it optimally adapts to the local geometry of the surface to be produced, thereby achieving a high quality surface finish. This concept has been first proposed by G. Jensen [1993] under the name *curvature matched machining*. Unfortunately, in its original formulation, it removes too much material. Subsequently published improvements partially fixed this problem by a careful analysis of the contact situation between tool and target surface. However, to the best of our knowledge, the integration into an algorithm for tool motion planning is still far from what is achievable with properly designed algorithms that exploit the available degrees of freedom.

The difficulties in curvature adapted CNC-machining lie in the interdependence of the individual tasks to be solved. These include the optimized selection of a cutting tool T , the computation of collision-free positions of T which adapt well to the target surface S , and the integration of optimized tool positions into tool motions which lead to a high quality surface finish.

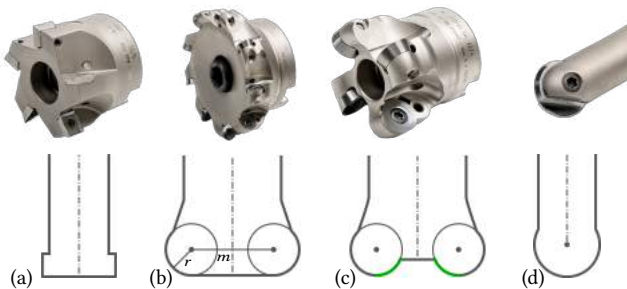


Fig. 2. Cutting tools. Various types of real cutting tools (top) and their geometric simplification as rotational solids (bottom): (a) Cylindrical face mill. (b) Toroidal cutter with small recess. (c) Toroidal cutter with significant recess and the possibility to mill with the negatively curved part (green). (d) Ball-end cutter.

1.1 Overview and contributions

- The best possible tool positions are close to those which provide so-called second order line contact. Hence, we carefully study second order line contact between the most common tools and a given surface (Sec. 2).

- We study criteria for local collision avoidance (Sec. 2.2). Here we include the so far apparently neglected use of the negatively curved part of a torus cutter (Fig. 2(c)) to machine a convex surface. This removes a previous limitation of curvature adapted machining.

- In Sec. 3, we discuss the variety of good tool positions and tool motion directions at a given contact point.

- Restricting to only those positions which are free of collisions (Sec. 4) and using a well selected tool (Sec. 5), we formulate the integration of tool positions into tool motions as a geometric optimization problem (Sec. 6).

- Our path planning algorithm aims at short production times through a small number of paths and achieves a high quality surface finish through adaption to the surface geometry. This is demonstrated at hand of machined parts in Sec. 7, where we also compare with results achieved with commercially available CAM software.

1.2 Related Work

CNC machining techniques can be categorized in several ways. The most common classification criteria are: (i) The *degrees of freedom* of the milling machine: 3-axis and 5-axis machines are the most frequently used [Jensen et al. 2008]. (ii) The shape of the *milling tool* (see also Fig. 2): flat-end [Fan and Ball 2014; Li and Jerard 1994], ball-end [Ikua et al. 2001], toroidal (aka bull-nose) [Campa et al. 2007; Roth et al. 2001], and general [Engin and Altintas 2001; Machchhar et al. 2017] cutting tools. (iii) The *type of contact* between the milling tool and the reference surface: single- (multi-) point milling in the case where the tool touches the surface at a single (several) point(s), and *flank* milling where the tool is in contact with the target surface along a curve (see e.g. [Calleja et al. 2018; Harik et al. 2013; Zhu et al. 2010]). Our research belongs to 5-axis point milling with toroidal tools, including special and limit cases, but it easily extends to tools composed of toroidal ones. For an introduction into the central concepts and algorithms for CNC machining we point the readers also to the monograph [Choi and Jerard 1998], and for machining of freeform surfaces to the proceedings [Olling et al. 1999] and the survey article [Lasemi et al. 2010].

Compared to flank milling, single-point contact approaches offer more freedom in adapting the tool to the surface [Warkentin et al. 2000]. However, the approximation quality is usually high only in the very neighborhood of the contact point, resulting in a vast number of milling paths to achieve a highly-accurate finish. This motivated research on *curvature matched machining* which uses curvature analysis to compute tool positions that are in optimized contact with the reference surface. That shall reduce the number of milling paths necessary for achieving a high quality surface finish [Cao et al. 2007; Jensen et al. 2008, 2002; Kim et al. 2015; Wang et al. 1993a,b; Yoon et al. 2003].

The problem of curvature matched positioning of a toroidal cutter can be reformulated via the offsetting argument to a problem of higher order contact between a circle (medial circle of the torus)

and an offset surface of the target geometry [Cao et al. 2007]. Unfortunately, such a configuration implies local penetration (gouging). Even though third order contact is possible (and avoids local penetration), it does not offer sufficient degrees of freedom for tool motion planning [Wang et al. 1993a]. However, it can be used to initiate algorithms searching for double-tangential contact between the tool and the reference surface [Kim et al. 2015]. In most curvature adaptation methods, the contact curves are computed first (see next paragraph) and afterwards the positions of the tool are computed [Cao et al. 2007]. We believe that this machining paradigm should be changed, and the milling paths and tool orientations shall be simultaneously computed in an optimized way.

Path planning often refers to the computation of those curves along which the tool will be in contact with the target surface. These cutter contact curves are roughly equidistant and chosen so that the maximum deviations between adjacent paths (scallop heights) are below a given threshold or even nearly constant. This task is greatly simplified for a ball cutter. That type of path planning has recently been addressed by [Zhao et al. 2018]. Based on an accessibility analysis, the object to be machined is decomposed into patches. These patches are then filled with spiral-like curves whose layout is tuned to achieve nearly constant scallop heights. The use of a ball cutter simplifies tool positioning and accessibility analysis, but does not allow for curvature adaptation. We also point to [Zou et al. 2014], which uses a ball cutter, but shares a similarity with our approach, since the authors formulate path planning as a global optimization problem with paths represented as level sets of a scalar function defined on the target surface. However, our problem is harder since we align the paths with good directions resulting from curvature adaptation and also consider collision detection already in the path-planning stage.

Recently, subtractive manufacturing received attention also in the graphics community. [Mahdavi-Amiri et al. 2020] decompose a 3D object into patches that are carvable with a single continuous path using 3-axis CNC machining. [Rivers et al. 2012] deals with accurate tool positioning and introduces a semi-automatic path-planning strategy where the user selects several tool positions that are subsequently optimized to interactively minimize the tool-surface error. For soft materials, such as foam, hot-wire cutting is the dominant subtractive fabrication technique. A unified pipeline for hot-wire cutting of free-form objects using a dynamic elastic rod is presented in [Duenser et al. 2020].

There are many other important issues related to path-planning for CNC machining such as velocity and acceleration control [Beudaert et al. 2012; De Lacalle et al. 2006; Sorby et al. 2000], feed rate distribution into tangential and axial directions [Beudaert et al. 2012], motion smoothness [Pechard et al. 2009], or vibrations [De Lacalle et al. 2006]. Even though our machining experiments are physically validated, our main objectives are geometric aspects of highly accurate path-planning and we refer the reader, e.g., to [Altintas 2012] for more details on physical aspects of CNC machining. Another topic related much closer to Computer Graphics are problems of accessibility and collision avoidance. As we are not contributing to this topic, we just refer to the survey [Tang 2014].

2 GEOMETRY

2.1 Basic geometric model

We address the task of manufacturing a part from a solid block of material by successive material removal. As mentioned above, we focus on finishing and thus can assume that most material has been removed already. However, one has the staircase effects which are typical for the result of the initial major material removal phase. Those staircases shall be removed with the additional aim of producing a high quality surface finish. This means that we want to get very close to the ideal target surface S , which we assume to be smooth. In practice, S may be given as a spline surface. If that is not the case, we can work with triangle meshes S as well, but would finally compute local spline approximations for enhanced smoothness.

2.1.1 Cutting tool. The cutting tool itself is complicated (see Fig. 2), but under fast rotation about its axis A it generates a rotational solid \mathcal{T} . We assume that its profile consists of straight lines and circles, since this is true for almost all available tools. The active part of the profile, i.e., the one which is designed for cutting, is either a circle or a straight line segment. Under rotation about A it generates a rotational surface T which we call *tool* or *cutter* henceforth. However, even when talking about a surface, we always have to keep in mind that the actual tool is a solid \mathcal{T} . A circular profile c_p of radius r leads to a part of a torus or sphere as a tool surface T . The center of c_p shall lie at distance m from A ; under rotation about A it generates the medial circle c_m of radius m . For $m = 0$, the cutter T becomes part of a sphere; see Fig. 2(d). Such a ball cutter is the least useful tool for curvature adapted machining since one cannot improve the local contact by changing the axis direction. The case of $r = 0$ is also considered here and called cutting circle (Fig. 2(a)), but we note that in practice r will be just very small. We also include $r = \infty$, i.e., machining with the developable flank surface of a conical or cylindrical cutter T .

2.1.2 Tool motion. When we talk about path planning we actually mean *tool motion planning*. Although in practice both the tool and the part to be produced may move, the essential information lies in the relative motion of T with respect to S . We plan this motion T/S , neglecting the fast rotation of the tool about its axis A ; its action has already been considered through the simplification of the tool to a rotational solid.

As our focus is on surface quality, we want to adapt the cutter motion optimally to the curvature behavior of the boundary surface S of \mathcal{S} . We assume a CNC machine with at least 5 axes. For our purposes this implies that the tool axis directions relative to the part are variable. Although possible in our algorithm, we impose no constraints on axis directions that are realizable with the CNC machine.

The target surface S bounds a solid \mathcal{S} which is the final part to be produced. S shall be oriented by a unit normal vector field \mathbf{n} which points towards the outside S^+ of the solid \mathcal{S} , i.e., the side from which the surface shall be machined. The material in \mathcal{S} has to remain; cutting into the interior of \mathcal{S} is called *overcutting*, and it is a more severe problem than *undercutting* that leaves some residual material and corresponds to a (typically tiny) gap between T and

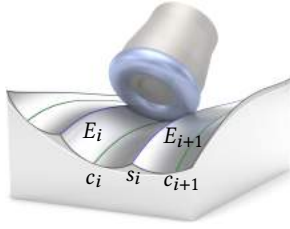


Fig. 3. Tool envelopes. Three paths of a milling tool through a material block are shown. The neighboring envelopes of the tool's motion, E_i and E_{i+1} , intersect in a curve, s_i , where the milling error is typically maximum. This phenomenon is called scalloping and the distance error of s_i from the desired surface S is known as *scalloping heights*.

S . During the machining process, the tool T accesses S from the positive side S^+ and moves under a sequence of continuous one-parameter motions along S , thereby always being tangent to S along *cutting contact curves (milling paths)* c_i , $i = 1, \dots, n$.

During the motion of T along a contact curve c_i , the tool envelopes a surface E_i (machined surface), which ideally is fitting very well to S in a sufficiently broad surface strip around c_i . Neighboring contact curves c_i, c_{i+1} generate envelopes whose intersection curve $s_i = E_i \cap E_{i+1}$ will in general exhibit tangent discontinuities and contain the largest deviations between target S and the produced surface S_p , see Fig. 3. These so-called *scalloping heights* shall be as small as possible in order to obtain a good surface finish of S_p . In that case, it will require minimal post-processing through grinding and polishing processes which are not easily controlled.

We exploit the available degrees of freedom in tool motion planning and adapt the tool positions to the local shape of S . However, as we will see below, we cannot simply aim at second order contact between the generated surfaces E_i and S , since this implies local penetration and thus overcutting.

2.2 Local contact situation and second order line contact

2.2.1 Osculating paraboloid and Dupin indicatrix. To analyze the local contact situation between a tool position T and the target surface S at a contact point \mathbf{p} , we introduce the principal frame of S at \mathbf{p} . The axes (x, y, z) of this orthonormal frame are aligned with the two principal curvature directions $\mathbf{t}_1, \mathbf{t}_2$ and the normal vector \mathbf{n} . Recall that \mathbf{n} and thus the positive z -axis point to the outside of the solid \mathcal{S} bounded by S . In this frame, a local second order approximation of S is given by the osculating paraboloid \mathcal{P}_S ,

$$2z = \kappa_1 x^2 + \kappa_2 y^2,$$

where κ_1, κ_2 are the corresponding signed principal curvatures. It is useful to view the osculating paraboloid as boundary of a solid \mathcal{P}_S whose points satisfy $2z \leq \kappa_1 x^2 + \kappa_2 y^2$. We can see \mathcal{P}_S as a local approximation of \mathcal{S} at \mathbf{p} .

In surface theory, we do not have a solid attached to S , but here it matters. Hence, the distinction of cases does not only concern the sign of Gaussian curvature $K = \kappa_1 \kappa_2$. We can assume $|\kappa_1| \leq |\kappa_2|$ and arrive at the following cases:

- $\kappa_1 < 0, \kappa_2 < 0$. *Convex elliptic point.*
- $\kappa_1 > 0, \kappa_2 > 0$. *Concave elliptic point.*
- $\kappa_1 = 0, \kappa_2 < 0$. *Convex parabolic point.*
- $\kappa_1 = 0, \kappa_2 > 0$. *Concave parabolic point.*
- $\kappa_1 \kappa_2 < 0$. *Hyperbolic point.*
- $\kappa_1 = \kappa_2 = 0$. *Flat point.*

The osculating paraboloid \mathcal{P}_S contains information about the variation of normal curvatures. A normal section of \mathcal{P}_S with a plane

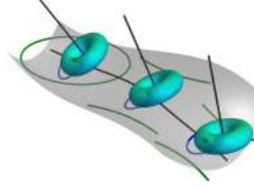


Fig. 4. Three positions of a tool along a milling path (black) are shown. The tool traverses from a concave elliptic to a hyperbolic region; the Dupin indicatrices of the tool (blue) and the surface (green) are shown.

spanned by \mathbf{n} and the tangent vector $\mathbf{t} := \mathbf{t}_1 \cos \phi + \mathbf{t}_2 \sin \phi$ is a parabola (or straight line), whose curvature $\kappa_n(\phi)$ at \mathbf{p} satisfies Euler's formula,

$$\kappa_n(\phi) = \kappa_1 \cos^2 \phi + \kappa_2 \sin^2 \phi. \quad (1)$$

For $\kappa_n < 0$ the parabolic section lies on the negative side of the tangent plane $z = 0$.

The distribution of normal curvatures is nicely visualized with the *Dupin indicatrix*. We enrich its information adapted to the present application. For that, we intersect the solid \mathcal{P}_S with the planes $z = \pm 1/2$ and project the intersection regions into the tangent plane $z = 0$. This yields the regions

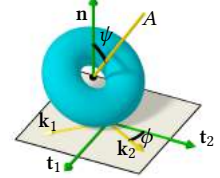
$$i_S^+ : \kappa_1 x^2 + \kappa_2 y^2 \geq 1, \quad i_S^- : \kappa_1 x^2 + \kappa_2 y^2 \leq -1.$$

They are bounded by the usual Dupin indicatrix of the surface S ,

$$i_S : \kappa_1 x^2 + \kappa_2 y^2 = \pm 1,$$

which is a radial diagram of the function $1/\sqrt{|\kappa_n(\phi)|}$. It is an ellipse for an elliptic point, a pair of parallel lines for a parabolic point, and a pair of hyperbolae for a hyperbolic point, see Fig. 4. Indicatrix regions for the most important cases in our application are shown in Fig. 6. The case of a flat point is not interesting: \mathcal{P}_S is the negative half-space $z \leq 0$, i_S^+ is empty and i_S^- is the entire plane.

Analogously, we investigate the curvature behavior of the tool surface T and associated solid \mathcal{T} at \mathbf{p} . The principal directions of a rotational surface are in the profile planes and orthogonal to those. Hence, the projection A' of the axis A onto the tangent plane is one principal direction \mathbf{k}_2 of T . In the principal frame of S it shall be given by $\mathbf{k}_2 := -\mathbf{t}_1 \sin \phi + \mathbf{t}_2 \cos \phi$. The other principal direction $\mathbf{k}_1 = \mathbf{t}_1 \cos \phi + \mathbf{t}_2 \sin \phi$ of T is tangent to the rotational circle on T . We call ϕ the *rotational angle of the tool*, and the angle ψ between tool axis A and the surface normal \mathbf{n} at the contact point \mathbf{p} the *tilt angle of the tool*.



Everything which has been said for the curvature of the target solid \mathcal{S} applies to the tool solid \mathcal{T} as well. We prefer to use the principal frame of S at \mathbf{p} and the normal \mathbf{n} , which points to the inside of \mathcal{T} , and we consider the signs of normal curvatures of T with respect to \mathbf{n} . The solid \mathcal{P}_T associated with the osculating paraboloid \mathcal{P}_T of T is of the form

$$\mathcal{P}_T : 2z \geq a_0 x^2 + 2a_1 xy + a_2 y^2 =: p_T(x, y).$$

The indicatrix region i_T^+ is obtained by projecting $\mathcal{P}_T \cap (z = 1/2)$ into $z = 0$, and analogously arises i_T^- from $\mathcal{P}_T \cap (z = -1/2)$.

Most tool surfaces are convex and thus belong to cases (a) and (c) above (with respect to T 's principal frame). There is, however, a toroidal cutter where part of the negatively curved area of the torus at the bottom of the tool is operating as a cutting surface (see

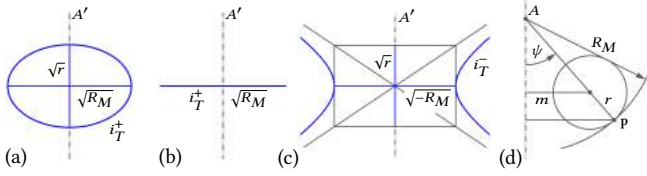


Fig. 5. Tool indicatrix for a convex elliptic point on a toroidal cutter (a), the limit case of a cutting circle, i.e. $r = 0$ (b) and the negatively curved part of a toroidal cutter (c). In (d) we illustrate the geometry leading to the normal curvature radius R_M .

Fig. 2 c). This is case (e) and very important for machining convex regions of S . For a cutting flat tool bottom we have case (f).

Let us provide more details on the *tool indicatrix* i_T (see Fig. 5). One axis direction k_2 lies in the projection A' of the tool axis A onto the tangent plane at p . For a toroidal cutter, the corresponding normal curvature radius is r . The other principal direction k_1 is tangent to the rotational circle c_T on T through p . The corresponding principal curvature center is on the axis A , i.e., its normal curvature radius is equal to the radius R_M of a sphere (Meusnier sphere; see below) which touches T along c (Fig. 5 d). R_M is computed from tilt angle ψ , medial circle radius m , and profile radius r as

$$R_M = r + \frac{m}{\sin \psi}. \quad (2)$$

2.2.2 Local millability condition. In an arbitrarily small neighborhood around the contact point p , the corresponding solids \mathcal{T} and \mathcal{S} shall only have the contact point p in common. Here we first exclude the (unfortunately) very rare case that the two solids are tangent to each other along a curve; this is discussed below and related to 2nd order line contact. Excluding this curve tangency, no intersection curve $S \cap T$ should pass through p . If that was the case, the normal curvatures of the intersection curve with respect to T and S (and the same choice of \mathbf{n}) at p were the same and the osculating paraboloids and Dupin indicatrices would intersect. Hence, *local millability requires that the intersection $\mathcal{P}_S \cap \mathcal{P}_T$ consists of the single point p (P_T lies above P_S except at p ; $p_T(x, y) > \kappa_1 x^2 + \kappa_2 y^2$ for $(x, y) \neq (0, 0)$). Equivalently, the corresponding Dupin indicatrix regions derived from planes $z = \pm 1/2$ should have no common point.*

Prior work deals with convex cutters only, and there one has the condition that the tool indicatrix i_T (ellipse) must lie in the interior of the target surface indicatrix i_S (see [Yoon et al. 2003] and the example in Fig. 4).

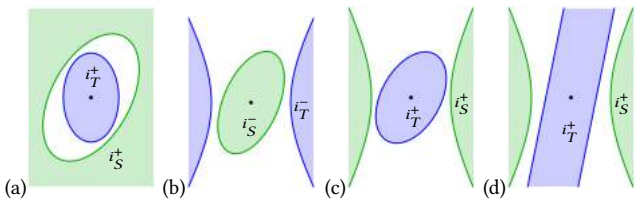


Fig. 6. Locally collision free positions visualized via indicatrix regions for tool and target shape. (a) A concave point of S milled with a convex cutter. (b) A convex point of S milled with a negatively curved cutter. (c) A hyperbolic point of S milled with a convex cutter and (d) milled with the flank of a cylindrical or conical cutter.

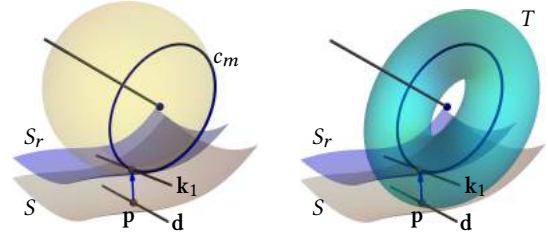
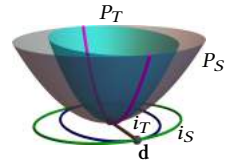


Fig. 7. Circle c_m in 2nd order line contact with S_r lies on the Meusnier sphere (left). The corresponding torus T possesses 2nd order line contact with S (right).

Fig. 6 illustrates the discussed concepts at hand of the typical local contact situations in milling the various types of surface points. Here we exclude points $p \in S$ with $K = 0$ for brevity, as they may be seen as limits of the discussed cases. In subsection 2.3.3, we discuss the machining of developable surfaces S , where $K = 0$ everywhere.

2.2.3 Second order line contact. Rotating the tool about the contact normal \mathbf{n} may bring the osculating paraboloids P_T and P_S closer together until in a limiting position they are tangent to each other along a parabola in a plane D through the contact normal. Equivalently the indicatrices i_T, i_S are tangent to each other at the end points of a common diameter with direction vector \mathbf{d} . This situation would occur in the ideal case where the tool surface T is tangent to the target S along a curve with tangent \mathbf{d} at p . However, in general it implies that there are curves $c_S \subset S$ and $c_T \subset T$ (e.g. the intersection curves $c_S = D \cap S, c_T = D \cap T$) which have second order contact at p . This situation is called *second order line contact* and can be characterized with help of the ruled surfaces N_S, N_T formed by the surface normals of S along c_S and normals of T along c_T ([Pottmann and Wallner 2001], pp. 457): *Two surfaces S, T are in second order line contact at p if there are curves $c_S \subset S$ and $c_T \subset T$ whose normal surfaces N_S and N_T are tangent to each other along the entire common normal at p .*



Since offset surfaces possess the same surface normal at corresponding points, we conclude that offsetting keeps second order line contact. We are mostly dealing with toroidal cutters T which are offsets of their medial circle at distance r , and thus we conclude:

PROPOSITION 2.1. *A torus T with profile circle radius r is in second order line contact with a surface S , if its medial circle c_m has second order contact with the offset S_r of S at distance r , see Fig. 7.*

Based on this and the fact that offsetting keeps second order contact between surfaces, it is easy to prove the following result: *A cutter T which moves tangentially along a curve $c \subset S$ by keeping in each position second order line contact with S in a direction transversal to c generates an envelope surface E which is in second order contact with the target surface S along the contact curve c .*

This shows the good local approximation one gets with these limit positions of cutters, but unfortunately second order contact implies in general local penetration (*gouging*). Thus, we have to move the cutter minimally away from such positions so that there is no collision with the target surface S anymore.

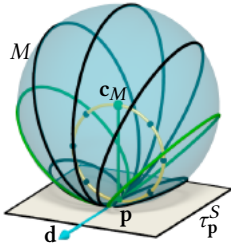


Fig. 8. The Meusnier sphere. The one-parameter family of circles (black) that are in second order contact with S at point \mathbf{p} in the direction \mathbf{d} form the Meusnier sphere M . The locus of their centers is another circle (yellow) that lies in a plane passing through \mathbf{p} and is perpendicular to \mathbf{d} . Given a specific radius, two solutions generically exist (green).

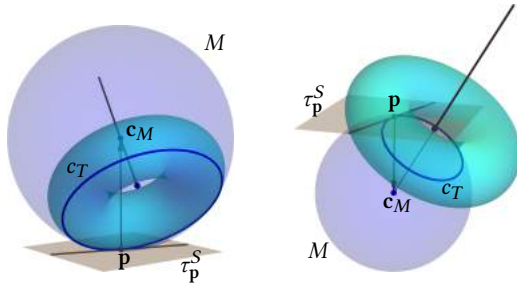


Fig. 9. Tool Meusnier spheres at an elliptic point (left) and a hyperbolic point (right) of the tool.

2.3 Generalized Meusnier spheres and Euler formula

In the following, we aim at a better understanding of second order line contact positions between cutter T and surface S . This yields a new smooth transition between two classical geometric results, namely the spheres of Meusnier and Mannheim, and the formulae of Euler and Blaschke. These classical results have recently been used in connection with principal meshes and a proposal to employ those for CNC machining with a flat end cutter or a conical cutter [Pellis et al. 2020]. Moreover, we present a simple relation between tool axis direction and direction \mathbf{d} of the 2nd order line contact.

2.3.1 Tool Meusnier sphere. We start with the case $r = 0$ of a cutting circle $T = c_m$. Second order line contact means that the circle c_m is in second order contact with S . All circles which have second order contact with a surface S at a point \mathbf{p} in tangent direction \mathbf{d} lie on the so-called *Meusnier sphere* M . Its radius $R_M = \frac{1}{\kappa_n}$ equals the normal curvature radius to direction \mathbf{d} and its center is $\mathbf{c}_M = \mathbf{p} + \frac{1}{\kappa_n} \mathbf{n}$ (see, e.g., [Blaschke and Leichtweiß 1973, p. 105]). The considered circles possess axes which pass through \mathbf{c}_M and lie in a plane, see Fig. 8. Its angle with the first principal direction \mathbf{t}_1 at \mathbf{p} is the rotation angle ϕ of the tool c_m .

Consider now all tools with the same rotation angle and profile radius r . We apply Prop. 2.1 and offsetting with distance r to S and the tool T . This brings us to the above situation of a circular tool c_m (Fig. 7). There is a Meusnier sphere for the offset, and reversing the offsetting operation, we obtain the following nice local situation:

PROPOSITION 2.2. *All tools with a fixed profile radius r which are in second order line contact with a surface S at a point \mathbf{p} at a rotation angle ϕ , possess axes which pass through the same point \mathbf{c}_M on the surface normal at \mathbf{p} . Each of these tools is tangent to a sphere $M(r, \phi)$ along a circle. This tool Meusnier sphere M touches the target surface at the contact point \mathbf{p} , see Fig. 9.*

Note that the tool Meusnier sphere $M(r, \phi)$ deserves this name also because it is the Meusnier sphere of the torus T at all points of the contacting circle c_T of T and M , for directions tangent to c_T .

While our derivation uses a finite radius r , we can pass to the limit $r \rightarrow \infty$ and obtain an analogous result for rotational cones or cylinders which are in second order line contact with S . The tool Meusnier sphere M for this case is known as *Mannheim sphere* and has so far been seen as a dual counterpart to the Meusnier sphere (see, e.g., [Blaschke and Leichtweiß 1973, p. 157]). We can now view it as a limit of a continuous sequence of tool Meusnier spheres which belong to tori that are in second order line contact with S .

Let us now derive the radius $R_M(r, \phi)$ of the tool Meusnier sphere $M(r, \phi)$. This is easily done by noting the following facts: A surface S and all its offsets S_r possess the same principal curvature centers. This means that the principal curvature radii of S_r are computed from the principal curvature radii $R_i = 1/\kappa_i$ of S as $R_i - r$. Likewise, we know that the Meusnier sphere radius for c_m equals $R_M - r$, but it also equals the normal curvature radius for the offset surface S_r . Applying Euler's formula (1) for the variation of normal curvatures to the offset S_r , we obtain

$$\frac{1}{R_M - r} = \frac{1}{R_1 - r} \cos^2 \phi + \frac{1}{R_2 - r} \sin^2 \phi, \quad (3)$$

and the following generalization of Euler's formula for the radii R_M :

PROPOSITION 2.3. *The radii R_M of the tool Meusnier spheres $M(r, \phi)$ satisfy the generalized Euler relation*

$$R_M = \frac{R_1(R_2 - r) \cos^2 \phi + R_2(R_1 - r) \sin^2 \phi}{(R_2 - r) \cos^2 \phi + (R_1 - r) \sin^2 \phi}. \quad (4)$$

The case $r = 0$ yields Euler's formula, since there $R_M = 1/\kappa_n$. In the limit $r \rightarrow \infty$, we obtain

$$R_M = R_1 \cos^2 \phi + R_2 \sin^2 \phi,$$

which is known as Blaschke's formula for the variation of Mannheim sphere radii (see, e.g., [Blaschke and Leichtweiß 1973, p. 157]).

If the tool axis lies in a principal plane, i.e. $\phi = 0$ or $\phi = \pi/2$, the corresponding spheres $M(r, \phi)$ are the principal spheres: $R_M = R_1$ or $R_M = R_2$, respectively, independently of the radius r . In a concave elliptic point, R_M lies in between R_1 and R_2 .

For a hyperbolic point \mathbf{p} of S this is different. Note that our radii come with a sign, dependent on whether the sphere lies on the positive or negative side of the tangent plane. Assume $R_1 > 0$ and $R_2 < 0$. Then, increasing ϕ , the sphere radii grow from R_1 to $\infty = -\infty$. This case $R_M = \infty$ belongs to an asymptotic direction on the offset S_r . It happens for the rotational angle ϕ_∞ ,

$$\phi_\infty = \pm \arctan \sqrt{\frac{r - R_2}{R_1 - r}}. \quad (5)$$

Going further, the spheres lie on the negative side of the tangent plane and radii vary from $-\infty$ to R_2 . As we mill a hyperbolic point \mathbf{p} with the convex part of a toroidal cutter, this does not concern us.

However, convex surface points \mathbf{p} ($R_1 \leq R_2 < 0$) are of great interest to us, since we can machine them with the negatively curved part of a toroidal cutter. Due to $r > 0$, (4) yields a negative radius $R_M < 0$ of the tool Meusnier sphere. Its center is on the negative side of the tangent plane and the torus T touches it from the outside.

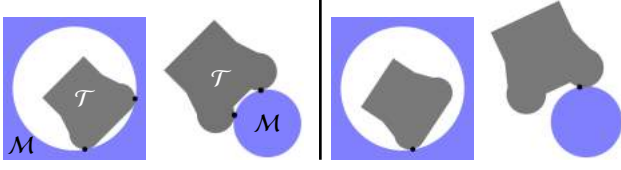


Fig. 10. Removing gouging, shown in planar sections through the cutter axis. Left column: Configuration with second order line contact at an elliptic point (left) or hyperbolic point (right) of the tool. Tool solid \mathcal{T} and Meusnier sphere solid \mathcal{M} touch along a circle. Right column: Tilting the tool so that $\mathcal{T} \cap \mathcal{M}$ consists locally only of the contact point, removes the local penetration of tool and target surface.

Dealing with signs of R_M follows a simple principle. We assign a solid \mathcal{M} with the tool Meusnier sphere M : For $R_M > 0$, the material is at the outside of the sphere, for $R_M < 0$ in the interior (ball). Let the tool solid \mathcal{T} be just the torus surface materialized in its interior. Then, at a locally penetration free position, \mathcal{M} and \mathcal{T} only touch at a single point, and for second order line contact they are tangent along a circle, see Fig. 10, left. Changing the tilt angle of the tool to move away from second order line contact and thus remove the local penetration must achieve $\mathcal{M} \cap \mathcal{T} = \{\mathbf{p}\}$, see Fig. 10, right.

2.3.2 Direction of second order line contact. So far, we did not talk about the direction \mathbf{d} in which second order line contact between tool T and target surface S happens. This direction is the diameter line through the contact points of the two indicatrices and can be computed as follows. By Prop. 2.1, second order line contact of T and S implies second order contact of the medial circle c_m of T and S_r . The characterization of second order line contact with help of normal surfaces which are tangent to each other along the entire contact normal at \mathbf{p} (see [Pottmann and Wallner 2001], pp. 457, and the proof of Prop. 2.1) shows: The tangent \mathbf{d} of this contact on S is mapped to the tangent of the medial circle c_m of the tool at its contact point with S_r . The mapping from S to the offset S_r via common surface normals is most easily understood if we use a principal curvature parameterization $\mathbf{s}(u, v)$ of S . With the unit normal field $\mathbf{n}(u, v)$, the first partial derivatives of \mathbf{s} and \mathbf{n} are eigenvectors of the shape operator and satisfy

$$\mathbf{n}_u = -\kappa_1 \mathbf{s}_u, \quad \mathbf{n}_v = -\kappa_2 \mathbf{s}_v.$$

The tangent vector \mathbf{d} is of the form $\mathbf{d} = u_1 \mathbf{s}_u + v_1 \mathbf{s}_v$. The offset at distance r is $\mathbf{s}_r = \mathbf{s} + r\mathbf{n}$, and thus the derivative of the map $\mathbf{s}(u, v) \mapsto \mathbf{s}_r(u, v)$ is given by

$$u_1 \mathbf{s}_u + v_1 \mathbf{s}_v \mapsto u_1(1 - r\kappa_1) \mathbf{s}_u + v_1(1 - r\kappa_2) \mathbf{s}_v.$$

The 2-dimensional tangent vector spaces at corresponding points of S and S_r are parallel and can be identified. We can assume that $\mathbf{s}_u, \mathbf{s}_v$ at the considered contact point $\mathbf{p} = \mathbf{s}(u_0, v_0)$ form a Cartesian frame. In this frame, the vector (u_1, v_1) at S is mapped to (\bar{u}_1, \bar{v}_1) at S_r , with

$$\bar{u}_1 = (1 - r\kappa_1)u_1, \quad \bar{v}_1 = (1 - r\kappa_2)v_1. \quad (6)$$

The direction (\bar{u}_1, \bar{v}_1) is tangent to the medial circle c_m of T and therefore orthogonal to the projection (u'_1, v'_1) of the tool axis into

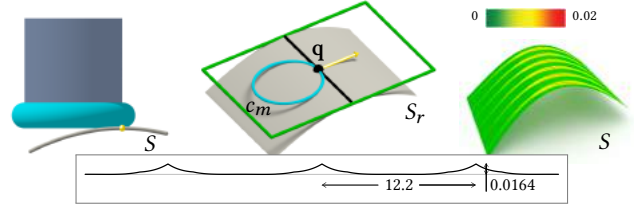


Fig. 11. High quality milling of a convex developable surface S with a toroidal cutter (left) can be achieved by positioning the tool such that the medial circle of the torus, c_m , moves tangentially to the offset S_r in a feed direction (yellow) transversal to the rulings (middle). The whole surface S is approximated by only seven paths of the tool and color-coded by the distance error (right). The third order contact is well seen by the scallop heights in a plane containing the ruling (bottom framed).

the tangent plane at \mathbf{p} . Hence, we can write (6) also as

$$\frac{\bar{u}_1}{\bar{v}_1} = -\frac{v'_1}{u'_1} = \frac{(1 - r\kappa_1)u_1}{(1 - r\kappa_2)v_1}.$$

This may be expressed as follows:

PROPOSITION 2.4. *In the tangent plane at the contact point \mathbf{p} between tool T and target surface S , the relation between the direction $\mathbf{d} = (u_1, v_1)$ of the second order line contact and the vector (u'_1, v'_1) of the projected tool axis is given by*

$$(1 - r\kappa_1)u_1 u'_1 + (1 - r\kappa_2)v_1 v'_1 = 0. \quad (7)$$

Here, coordinates are with respect to the principal frame at \mathbf{p} .

Of course, (7) expresses orthogonality for $r = 0$. In the limit $r \rightarrow \infty$, we obtain

$$\kappa_1 u_1 u'_1 + \kappa_2 v_1 v'_1 = 0.$$

Here the tool T is a cylinder or cone and we obtain the well-known fact that the ruling of the tool (= projection of its axis) must be conjugate to the direction \mathbf{d} of the contact curve. Again, our setting provides a continuous blend between two simple relations: orthogonality and conjugacy. Note that the only directions which are orthogonal and conjugate are the principal directions $(1, 0)$ and $(0, 1)$, which satisfy (7) for any r . This is a result of symmetry.

If we denote by δ the angle of \mathbf{d} against the first principal direction, we have $(u_1, v_1) = (\cos \delta, \sin \delta)$, and with the tool angle ϕ , one has $(u'_1, v'_1) = (-\sin \phi, \cos \phi)$ and therefore

$$\cot \phi = \frac{1 - r\kappa_1}{1 - r\kappa_2} \cot \delta. \quad (8)$$

2.3.3 Milling a developable surface. Finally, we address the special situation of machining a *developable surface* S . It has $K = 0$ everywhere and carries a family of straight line segments (rulings) along which the tangent plane is constant. If accessibility is given, one can mill S by flank milling with a cylindrical or conical cutter T . One just has to make sure that T remains tangent to the rulings of S (see e.g. [Stein et al. 2018]). However, if this is not possible due to collisions, there are still other ways to solve the problem if the points of S are *convex parabolic* (see Fig. 11).

If a cutter T with a flat cutting bottom is available, one leads the cutter so that the flat cutting part remains in contact with S along a ruling. If one has just a toroidal cutter with a negatively curved cutting part, one can use the bottom circle c_b of T . One mills S so

that c_b is always tangent to a ruling of S (Fig. 11, left) and proceeds transversal to the rulings. This strategy produces *envelopes* E which are in third order contact with S along the cutter contact curve. By the offsetting argument, it is sufficient to consider the ringed surface S_m generated by the medial circle c_m of T . On S_m , the ruling direction is an asymptotic direction since c_m lies in the tangent plane of \mathbf{q} , see Fig. 11, middle. Intersecting the normal plane of the developable surface S_r along the ruling with S_m yields a curve c^n with vanishing curvature at \mathbf{q} . Thus, we have at least contact order two, but it must be three since there is no local penetration. Fig. 11, bottom, shows the $O(h^4)$ behaviour of the error.

3 CURVATURE ADAPTED POSITIONS

In this section we discuss the variety of good tool positions at a given surface point \mathbf{p} . At first we look at second order line contact which is then corrected to get rid of local penetration. We assume that the tool T is given; its choice is discussed in Section 5.

3.1 Positions with second order line contact

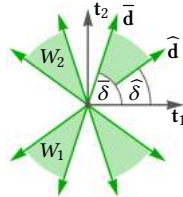
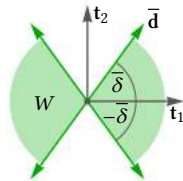
We now describe all possible positions of T providing 2nd order line contact with S at a fixed point \mathbf{p} . The basics are well prepared: For a given rotational angle ϕ , equations (3) or (4) yield a radius R_M of the tool Meusnier sphere M and the tilt angle ψ , cf. (2). If we mill with the convex part of T , M has to be sufficiently big so that the torus T can be inscribed (see Prop. 2.2). This requires $R_M \geq r + m$. If we mill with the hyperbolic part of T , the torus T must touch M (with radius $R_M < 0$) from the outside. Geometrically, this requires $R_M \leq r - m$. In practice, $R_M = r - m$ will not work since the toroidal cutter cannot cut with its smallest circle (see Fig. 2) and thus there will be a limit value $R^* < 0$ such that $R_M < R^*$.

We now discuss all possible situations. Let us start with concave elliptic points and w.l.o.g. $R_1 \geq R_2$. If $R_2 \geq r + m$, all rotation angles ϕ provide a 2nd order line contact, whereas if $R_1 < r + m$ it is not possible for any ϕ . For $R_2 < r + m \leq R_1$ we have a wedge W of admissible directions \mathbf{d} along which 2nd order line contact of T with S can be obtained. The utmost directions of the wedge correspond to $R_M = m + r$. Together with (8) it yields

$$\bar{\delta} = \arctan \left(\left| \frac{R_2}{R_1} \right| \sqrt{\frac{(R_1 - r)(R_1 - m - r)}{(R_2 - r)(m + r - R_2)}} \right). \quad (9)$$

For convex elliptic points, equation (9) remains valid, however we need to consider $-m$ instead of m . Nevertheless in practise we employ a limit value $R^* < 0$.

The situation is slightly more complicated at hyperbolic points since one cannot obtain 2nd order line contact along the directions with a negative value of $R_M - r$. The directions where $R_M = \pm\infty$ correspond to the asymptotes of the Dupin indicatrix (hyperbola) of S_r at \mathbf{q} , see (5). Hence two wedges W_1, W_2 ($W = W_1 \cup W_2$) of admissible directions



with utmost directions are given by (9) and

$$\widehat{\delta} = \arctan \left(\left| \frac{R_2}{R_1} \right| \sqrt{\frac{R_1 - r}{r - R_2}} \right).$$

Once \mathbf{d} is chosen in W , the rotational and tilt angles of T are given by (8) and (2), respectively, and a direct computation gives the center \mathbf{s} and the axis vector \mathbf{a} of T as

$$\mathbf{a} = \mathbf{k}_2 \sin(\psi) + \mathbf{n} \cos(\psi)$$

and

$$\mathbf{s} = \mathbf{q} + m\mathbf{N}_p,$$

where \mathbf{N}_p is the unit principal normal of c_m , i.e.,

$$\mathbf{N}_p = \mathbf{n} \sin(\psi) - \mathbf{k}_2 \cos(\psi).$$

REMARK 1. Once an admissible direction \mathbf{d} , $\mathbf{d} \in W$, is selected, the position of T w.r.t S is fully determined and, generically, there are two solutions. This fact is a direct consequence of the Meusnier theorem which assigns to the given radius the tilt angle guaranteeing osculation, recall Fig. 8. Therefore, we speak about “wedges” rather than “angles” of admissible directions as we encode the tilt inclination (positive or negative) into the orientation of \mathbf{d} .

3.2 Correction of local penetration

The analysis described in Section 3.1 results in positions of T that are in 2nd order line contact with S . Such a contact means a very good local approximation as the distance error between T and S in \mathbf{d} behaves as $O(h^3)$, h being a parameter that parametrizes the medial circle c_m , see Fig. 13. However, the cubical behaviour of the error function corresponds to a sign change at zero (contact point \mathbf{p}) and consequently local penetration in the very neighborhood of \mathbf{p} . This phenomenon, called “gouging” in the machining literature, is very undesirable as it directly implies overcutting at \mathbf{p} .

One can fix this phenomenon by moving the torus as follows:

- (1) Rotate the tool towards the right principal direction. For an elliptic point of S , this is the direction with the smaller principal curvature (major axis of the indicatrix i_S). For a hyperbolic point, this is the direction with the negative principal curvature (minor axis of the hyperbola), see Fig. 14, middle.
- (2) Increase the tilt angle ψ . This is the same as working with slightly smaller tool Meusnier spheres and considering tools tangential to those, see Fig. 14, right and Fig. 10, right.
- (3) Combination of both (1) and (2) in situations where global collision occurs.

Operations (1) and (2) fix penetration locally, i.e., close to \mathbf{p} , but may lead to global collision, discussed next.

4 ACCESSIBILITY AND COLLISION AVOIDANCE

To access S with the cutter T , one has to avoid both local and global collisions. The analysis described in the previous section takes care of local collisions. To control global collisions, we perform collision tests as follows. The *head* of the milling cutter is conceptualized as a torus while the *shank* is represented by a cylinder (the real shank is a more complex object, but its cylindrical bound is taken for the sake of simplicity). Our collision detection algorithm relies on fast computation of point-mesh projections [Chang et al. 2010;

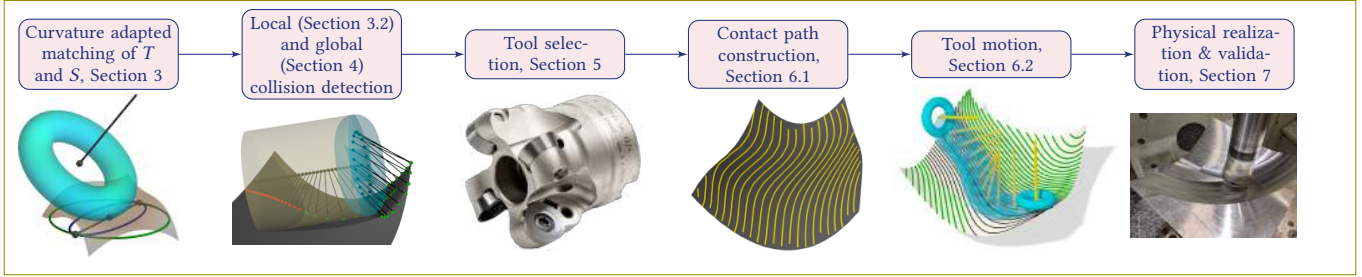


Fig. 12. Algorithm overview.

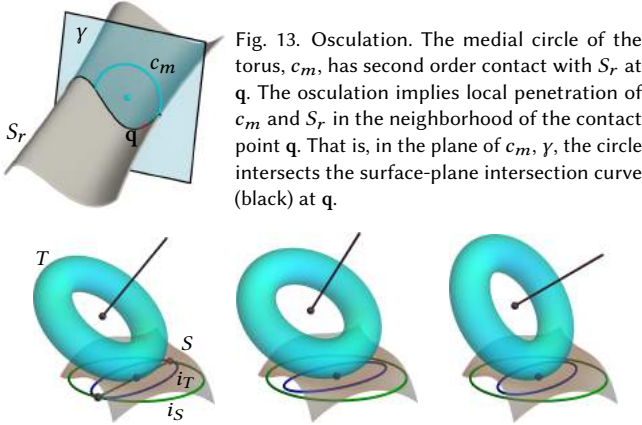


Fig. 13. Osculation. The medial circle of the torus, c_m , has second order contact with S_r at q . The osculation implies local penetration of c_m and S_r in the neighborhood of the contact point q . That is, in the plane of c_m , γ , the circle intersects the surface-plane intersection curve (black) at q .

Fig. 14. Correction of local penetration. Left: A toroidal milling tool T in second order line contact with a given surface S at an elliptic point p — this corresponds to a tangential contact of the Dupin indicatrices of the tool, i_T (blue), and of the surface, i_S (green). This configuration results in local penetration (cf. Fig. 13) and can be corrected by decreasing the rotational angle (middle) or by increasing the tilt angle (right).

Redon et al. 2005]. The input geometry is represented as a fine mesh, which is then loaded and manipulated with the Trimesh [Dawson-Haggerty et al. 2019] Python library. Being capable of building the bounding volume hierarchy (BVH) of the mesh, the latter assures fast computation of closest points (aka footpoints), as well as other convenient routines like navigation through adjacent faces, mesh alignment etc.

Both the head and the shank can collide with S globally. To detect a collision of the head, we sample the medial circle c_m and compute the footpoints on S . For a sample point p_i , the projection algorithm returns its footpoint p_i^\perp , see Fig. 15, and a distance check $\|p_i - p_i^\perp\| < r$ reveals the collision (up to sampling of c_m). For the cylindrical part, we proceed analogously: we sample the axis A , compute the footpoints on S , and if lying in the half-space of the shank part, see Fig. 15 right, a distance check $\|p_i - p_i^\perp\| < r + m$ is evoked. If the collision is realized at some point, the collision routine is terminated (and the corresponding tangent direction d is labeled as colliding).

5 TOOL SELECTION

We aim at efficient milling and look for a tool as “big” as possible in order to maximize the width of the milling strips, and consequently to minimize the number of paths needed for machining the workpiece. As the second order line contact is strongly correlated with

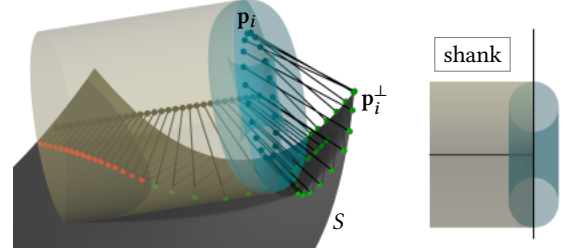


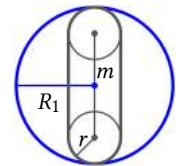
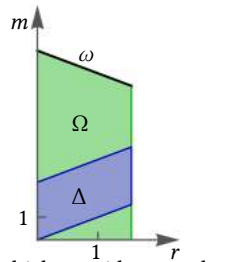
Fig. 15. Global collision detection test. Left: the medial circle and the axis of the tool are sampled and their footpoints on S are computed. The collision-free footpoints (green) and those that correspond to collision (red) are returned. Right: In case of the footpoints of the axis (shank), only the footpoints in the half-space of the shank are considered, and tested by $\|p_i - p_i^\perp\| < m + r$. The shank is typically curved (cf. Fig.2), but for the collision test it is simplified as a cylinder (that contains the real shank).

the width of the milling strips, we postulate that a good tool shall be able to offer second order line contact in large parts of S .

The space of possible tools is part of the rm -plane, where each point (r, m) represents the two radii of a torus tool T . We denote by Ω the subset characterizing tools that can possess 2nd order line contact at all (concave elliptic and hyperbolic) points of S (see inset). Ω turns out to be a trapezoidal domain; its upper side $\omega \subset \Omega$ yields tools with the maximal value of $r + m$. The (possibly empty) subset $\Delta \subset \Omega$ characterizes tools which provide second order line contact also at all convex elliptic points.

We first consider a concave elliptic point p ($R_1 \geq R_2 > 0$). Milling with the convex part of T , it can possess 2nd order line contact at p if it can be inscribed into the maximal tool Meusnier sphere at p , i.e., if $r + m \leq R_1$. At a hyperbolic point p ($R_1 < 0, R_2 > 0$), the tool Meusnier sphere associated to the asymptotic directions is infinite and hence there is no constraint on $r + m$. As $r > R_2$ implies local penetration of T and S (i_T is not inside i_S), a necessary condition for a locally penetration-free position at a (concave elliptic or hyperbolic) point is $r < R_2$.

Now looking at all concave elliptic and hyperbolic points of S (we suppose $\kappa_1 \leq \kappa_2$), there is a minimal value \bar{M} of R_1 (over all



concave elliptic points) and a minimal value \bar{R} of R_2 (over all concave elliptic and hyperbolic points). Finally, we identify Ω as the following trapezoidal domain (note $\bar{R} < \bar{M}$)

$$\Omega : r \in [0, \bar{R}] \quad \text{and} \quad m \in (0, \bar{M} - r).$$

The largest tools may be seen as those with maximal $m + r$. They correspond to the line segment ω ,

$$\omega : m = \bar{M} - r, \quad \text{where} \quad r \in [0, \bar{R}].$$

At a convex elliptic point ($R_1 \leq R_2 < 0$), we mill with the hyperbolic part of T (we suppose $m > r$) and obtain second order line contact for $r - m \geq R_1$. Hence we have

$$\Delta : (K \leq r - m < 0) \cap \Omega,$$

where $K = 1/\max_{p \in S_{convex}} \kappa_1$. However, in practice we employ a limit value $R^* < 0$. Let us note that Δ may be empty.

The above described approach considers tools that enable second order line contact at *all* (concave elliptic and hyperbolic) sampled points. However, when some (possibly tiny) part of S is very curved, this approach forces the tool to be small even though the rest of S admits a larger tool. In such a case one may consider to machine the workpiece with two different cutters.

Finally, to select the optimal tool from Ω , we consider those points in Ω which correspond to the available tools and for each such a tool T perform the collision detection of S and T . For that, we compute the number of points (and positions of the tool) at which the tool does not intersect S . Then we choose the best tool as the one with the maximal coverage in terms of non-colliding points and directions. Hence the selection process yields the largest possible tool (from the set of all available tools) providing penetration-free positions (with possible 2nd order line contact) at a maximal part of S .

6 TOOL MOTION PLANNING

With the information about the admissible directions of second order line contact (Section 3) and the fast global collision test (Section 4), we are now ready to design milling (aka contact) paths $c_j \in S$, $j = 1, \dots, n$. There are several requirements on these paths: (i) At every time instant, the second order line contact shall be as orthogonal as possible to the tangent direction of c_j in order maximize material removal using a single sweep of the tool. (ii) The paths as well as the tool motion along them should be smooth and fair to prevent chattering of the tool. (iii) The paths shall be aligned one to another in quasi parallel fashion with the milling error occurring between two neighboring paths c_j, c_{j+1} below a predefined machining error, see Fig. 16. (iv) The motion of the tool must be locally and globally penetration-free. With the objectives (i-iv) in mind, we describe the construction of the contact paths.

6.1 Contact path construction

We represent the milling paths $c_j \in S$, $j = 1, \dots, n$, as level sets of a scalar function $G : S \rightarrow \mathbb{R}$ defined on S , i.e., G is constant along each path c_j . The function G is computed within an optimization-based framework as follows.

In our discrete setup, S is represented by a triangle mesh with N vertices \mathbf{p}_i . G is assumed to be linear in each triangle and thus defined by its values x_i at \mathbf{p}_i , i.e., $G(\mathbf{p}_i) = x_i$, $i = 1, \dots, N$. The vector

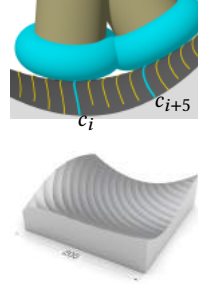


Fig. 16. Scalloping. Top: After fixing the local penetration, the tool-surface error in the neighborhood of the milling paths (blue) is tiny as it mimics the $O(h^3)$ behaviour due to the nearby osculation. The material left between two neighboring sweeps of the tool and the surface (blank) forms so called scallops that determine the milling accuracy. Here, for visualization purposes, a position of a tool along an i -th and $(i + 5)$ -th milling path are shown. Bottom: Material block with milling strips and scallops.

$\mathbf{x} = \{x_1, \dots, x_N\}$ of function values at vertices is the unknown of a minimization problem. Its objective function (11) is composed of several parts which are now described.

Good tangent directions. The milling error vanishes along a milling path and increases when moving away from it, see Fig. 16. The maximum error between two neighboring tool envelope strips occurs along their intersection curve. As milling paths are required to be quasi parallel (requirement (iii)), one aims at having the contact direction orthogonal to the paths. This places the intersection curve of the two strips at nearly equal distance to the two corresponding milling paths, which helps to minimize the milling error.

At each \mathbf{p}_i , we may have several wedges in which T possesses second order line contact w.r.t. S and the configuration is globally penetration-free, see Fig. 17(b). Let k_i be the number of wedges W_i^j at \mathbf{p}_i , \mathbf{u}_i^j the central vector of each wedge, and α_i^j the angle between \mathbf{u}_i^j and the corresponding wedge boundary. The objective that the level-sets of G are orthogonal to (either of) the wedges is formulated as

$$F_{ortho}(\mathbf{x}) = \sum_i^N \prod_j^{k_i} \left(\mu(\alpha_i^j) \langle \nabla G(\mathbf{p}_i), \mathbf{u}_i^{j\perp} \rangle^2 \right)^{1/k_i}, \quad (10)$$

∇ being the gradient, and μ being a weight function defined as

$$\mu(\alpha) = \frac{(\pi - 4\alpha)\beta}{\pi(\alpha + \beta)}.$$

It maps the opening angle α of the wedge to a real number in $[0, 1]$. The purpose of this function is to assign higher weights to narrow wedges as there is little choice for maneuvering of the tool to be in second order line contact with S . Observe that $\mu(0) = 1$ and $\mu(\pi/4) = 1$ and the function is decreasing. The parameter β controls the decay of μ and was set experimentally to $\beta = 0.05$ in all our examples. Therefore, for small values of α , one obtains large weights (higher priority), while for wedges with large opening angles, the orthogonality constraint is more relaxed as many good directions exist.

Fairness of the paths c_j is closely related to fairness of G , which is achieved by the energy

$$F_{fair}(\mathbf{x}) = \sum_i^N \Delta G(\mathbf{p}_i)^2,$$

where Δ is the Laplace–Beltrami operator, discretized with the cotangent formula.

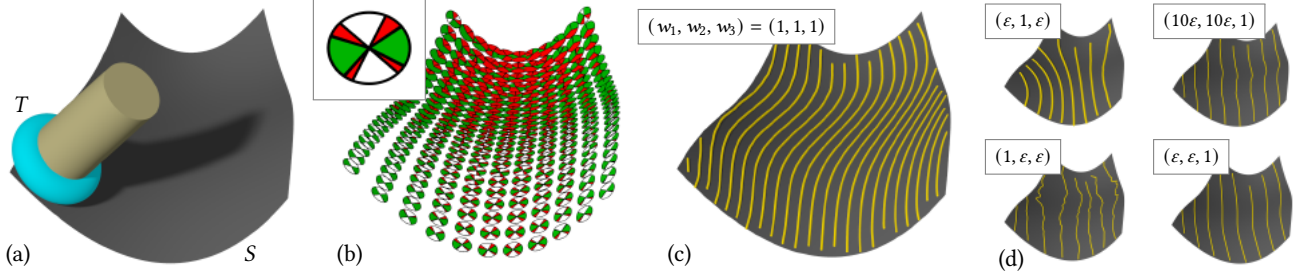


Fig. 17. Contact path construction. Given a reference surface S and a toroidal tool T (a), the analysis conducted in Section 3.1 categorizes tangent directions into three types of tangent wedges: second order line contact between S and T is possible (green); second order contact causes global collision (red); first order contact (blank). (c) The contact paths are computed as level-sets of a scalar function G , defined on S , computed by minimizing the objective function in Eq. (11); here all three objective terms are represented equally. (d) The contact paths are defective if only one term dominates; various triplets of weights (w_1, w_2, w_3) with $\epsilon = 10^{-2}$ are shown.

Parallelism of paths. Finally, we aim at level sets c_j which are close to geodesic offsets of each other. In other words, G shall not deviate much from a distance function on S , characterized by $\|\nabla G\| = 1$. This leads to the term

$$F_{parallel}(\mathbf{x}) = \sum_i^N (\|\nabla G(\mathbf{p}_i)\| - 1)^2.$$

Altogether, we obtain the following objective function

$$F(\mathbf{x}) = w_1 F_{ortho}(\mathbf{x}) + w_2 F_{fair}(\mathbf{x}) + w_3 F_{parallel}(\mathbf{x}), \quad (11)$$

where w_1 , w_2 , and w_3 are weights to control orthogonality, fairness, and parallelism of the milling paths, respectively. The impact of these weights on the milling paths is shown in Fig. 17. One can see that setting a dominant weight for F_{ortho} results in meandering milling paths, while a dominating fairness term produces fair curves, but without a geodesically parallel pattern. Finally, the parallel term alone does not yield a nice (orthogonal) path through the wedges, see Fig. 17(d). Therefore a combination of all terms is needed and the weights were set to $(w_1, w_2, w_3) = (1, 1, 1)$ in all our examples.

The objective function in Eq. (11) is non-linear and therefore an initial guess $\mathbf{x}^0 = \{x_1^0, \dots, x_N^0\}$ is needed. We initialize G as distance function of an appropriately chosen plane Γ , i.e., we set $x_i^0 = \text{dist}(\mathbf{p}_i, \Gamma)$. Thus the initial level sets lie in planes parallel to Γ which is determined as follows. We compute a central point \mathbf{m} as closest point on S to the barycenter $(\sum \mathbf{p}_i)/N$. Γ is spanned by the normal line of S at \mathbf{m} and a further vector \mathbf{w} whose computation is based on our requirement that milling paths shall be perpendicular to the green wedges, see Fig. 17(b). At every vertex \mathbf{p}_i , we take the central wedge direction \mathbf{u}_i , if any¹, compute its orthogonal tangential complement \mathbf{u}_i^\perp , and project it to the tangent plane of S at \mathbf{m} . Then \mathbf{w} is computed as median of these projections.

Finally, having the initial guess, minimization uses the gradient-based solver Adam [Kingma and Ba 2014] implemented in TensorFlow, without the stochastic aspect.

REMARK 2. Typically, path-planning methods prescribe and/or require the contact paths as input and deal only with the orientation of the tool. However, this reduces the search space and consequently makes

¹Points with no admissible direction (empty wedge) and points with too many good directions (half discs and larger) do not contribute.

the solution sub-optimal. In contrast, our approach looks for contact paths such that the milling strips are as wide as possible (milling time minimization) and their construction is a part of the path-planning algorithm.

6.2 Tool motion planning

In this section we describe how the smooth and penetration-free motion of a given tool is computed. In our discrete setup, a particular contact path c is a polyline $\mathbf{c}_1, \dots, \mathbf{c}_\ell$ on a triangular mesh. Let us first consider the case when an underlying B-spline surface S is available. We proceed as follows. We pull back the cutter contact path $\mathbf{c}_1, \dots, \mathbf{c}_\ell$ to the parameter domain and approximate it with a cubic spline $\bar{\mathbf{c}}(t)$, $t \in [0, 1]$ with a parametric speed as constant as possible. Mapping $\bar{\mathbf{c}}(t)$ back to S yields the smooth contact curve $\mathbf{c}(t)$. When no underlying spline is available, we proceed analogously by computing locally B-spline approximations of the triangle mesh along the cutter contact polyline.

Once the smooth contact paths are computed, the motion construction inheres in finding smooth rotation $\bar{\phi}(t)$ and tilt $\bar{\psi}(t)$ functions along each path $\mathbf{c}(t)$. Since we aim at parallel envelope strips with equal widths, we do not further optimize the rotation angle and choose the tool axis in the plane spanned by the path tangent and the surface normal at $\mathbf{c}(t)$.

Thus, the tool motion is optimized only via the tilt function $\bar{\psi}(t)$, which is constructed as follows: We sample parameter values $t_0, \dots, t_k \in [0, 1]$ and for each t_i :

- (i) Compute the tilt angle ψ_i corresponding to the 2nd order line contact (if there is no 2nd order line contact possible, $\psi = 0$).
- (ii) Sample values for tilts from $[\psi_i + \epsilon, \pi]$ and perform the collision detection. Penetration-free positions correspond to the interval $[\psi_i^0, \psi_i^1]$, see Fig. 18, top.

Intervals $[\psi_i^0, \psi_i^1]$ determine a region of admissible tilts, see Fig. 18, middle. Thus, tool motion planning is translated to the construction of a function $\bar{\psi}(t)$ through the green region, which is (a) close to the bottom boundary ψ_i^0 (yielding tool orientations close to 2nd order line contact) and (b) possesses a nice distribution of curvature (yielding a fair motion of the tool), see Fig. 18, bottom. We proceed as follows. The tilt angle function is represented as a cubic B-spline function $\bar{\psi}(t) = \sum b_i N_i^3(t)$ with coefficients $b_0, \dots, b_n \in \mathbb{R}$ and

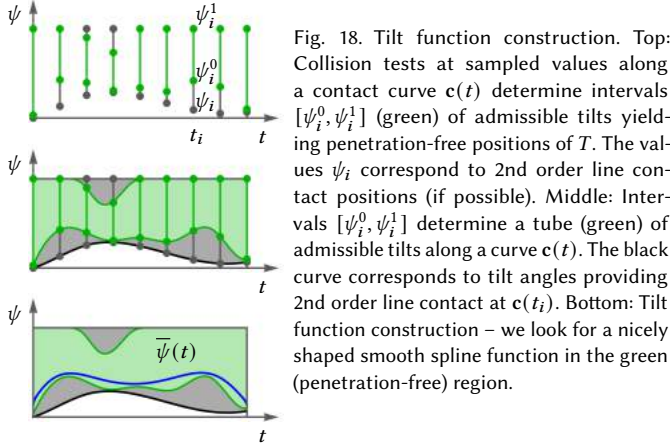


Fig. 18. Tilt function construction. Top: Collision tests at sampled values along a contact curve $c(t)$ determine intervals $[\psi_i^0, \psi_i^1]$ (green) of admissible tilts yielding penetration-free positions of T . The values ψ_i correspond to 2nd order line contact positions (if possible). Middle: Intervals $[\psi_i^0, \psi_i^1]$ determine a tube (green) of admissible tilts along a curve $c(t)$. The black curve corresponds to tilt angles providing 2nd order line contact at $c(t_i)$. Bottom: Tilt function construction – we look for a nicely shaped smooth spline function in the green (penetration-free) region.

normalized cubic B-spline basis functions $N_i^3(t)$ over a uniform knot sequence K . At first, we interpolate values ψ_i^0 and ψ_i^1 by cubic B-spline functions $\psi^0(t)$ and $\psi^1(t)$ (with knot sequence K and B-spline coefficients b_0^0, \dots, b_n^0 and b_0^1, \dots, b_n^1 , respectively). Then $\bar{\psi}(t)$ is found by minimizing the quadratic objective function

$$F_{\text{tilt}}(b_0, \dots, b_n) = F_{\text{close}} + wF_{\text{fair}},$$

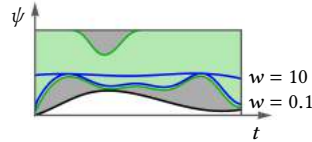
subject to the constraints $b_i^0 + \varepsilon \leq b_i \leq b_i^1$, $i = 1, \dots, k$. The term F_{close} forces $\bar{\psi}(t)$ to lie close to the bottom boundary $\phi_0(t) + \varepsilon$, where $\varepsilon = 10^{-2}$ is used as a safety threshold,

$$F_{\text{close}}(b_0, \dots, b_n) = \frac{1}{n+1} \sum_{i=0}^n (b_i - b_i^0)^2.$$

The second term F_{fair} votes for the fair motion of the tool,

$$F_{\text{fair}}(b_0, \dots, b_n) = \frac{1}{n-1} \sum_{i=1}^{n-1} (b_{i-1} - 2b_i + b_{i+1})^2.$$

The weight w controls the trade-off between closeness to 2nd order line contact and fairness. Setting $w = 0$ yields exactly the function $\psi_0(t) + \varepsilon$ and $w = \infty$ a straight line (if compatible with the constraints). In our examples, we set $w = 1$ as a suitable compromise, see Fig. 18, bottom.



REMARK 3. Let us emphasize that the above described motion planning approach automatically handles the transition between concave and convex points of S by tilting the tool and using its elliptic or hyperbolic part appropriately, see Fig. 19.

7 SIMULATION RESULTS AND PHYSICAL VALIDATIONS

The proposed methodology has been tested on a variety of free-form surfaces. Given a reference surface S , Fig. 20 shows milling paths for a specific milling tool T .

Another example where the milling paths are computed for two specific tools on a single target surface is shown in Fig. 21(a,c). The view is upside-down to demonstrate a collision-free configuration of S and T throughout the surface. The motion of the toroidal head (blue) of the tool is visualized as a swept volume in Fig. 21(b,d).

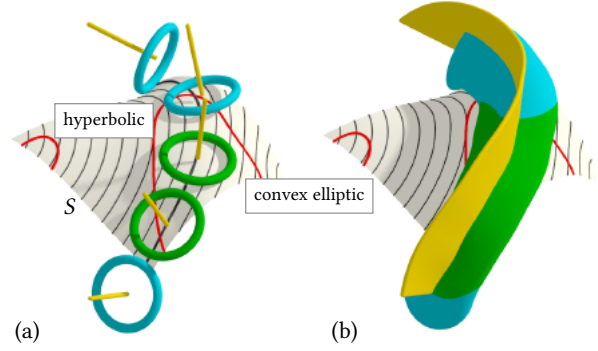


Fig. 19. Milling with the right part of the tool. The parabolic points on S form curves (red) that separate regions with positive and negative Gaussian curvature. In hyperbolic regions, the tool touches S with its elliptic part (blue positions), while in the convex elliptic regions, the tool touches by its hyperbolic part. Hyperbolic-elliptic and elliptic-hyperbolic match of the tool and the surface is essential for highly-accurate milling. (b) A smooth motion of the tool with involvement of its elliptic (blue) and hyperbolic (green) part.

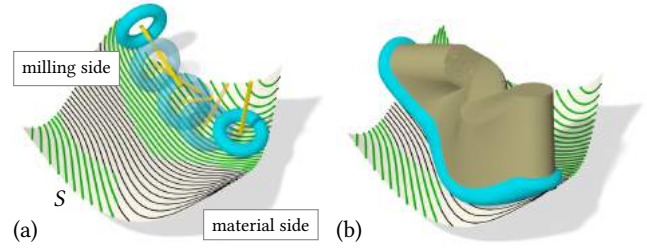


Fig. 20. Collision-free motions. (a) A reference surface S can be accessed by a toroidal cutter only from the milling side. The analysis described in Section 6 computes the contact paths. The green curves correspond to positions of the tool where second order line contact is possible and no global collision occurs. Along the black curves (b), only first order contact can be achieved. There we aim at smooth motions of the tool, where neither the toroidal part nor the shank collide with S .

When the tool T moves along a contact curve c_i , it cuts a surface into the material block, which is part of the envelope E of the torus T under that motion. The nature of E is simple and thus it can be computed with high accuracy: The medial circle c_m of T sweeps out a ringed surface, and the envelope E is its offset at distance r , where r is the profile radius. We prefer to use this direct geometric construction over alternatives proposed in the literature; see e.g. [Roth et al. 2001].

For our curvature adapted tool motion planning, the surface E generated by the moving tool approximates the target surface S really well in the proximity of the milling path. This can be seen in Fig. 22(b). The two neighboring milling paths, that lie (exactly) on S , are within a fine distance error of $\varepsilon \doteq 10^{-4}$ of the size of the bounding box of the surface. Typically, the machining tolerance is several micrometers for an object of tens of centimeters (factor ca 10^{-5}). Hence, the distribution of the milling paths shown in Fig. 22(a) roughly corresponds to the desired accuracy. Observe that the distance error ε is not the scallop height, but the distance of the next path to the envelope (i.e., an upper bound of the milling

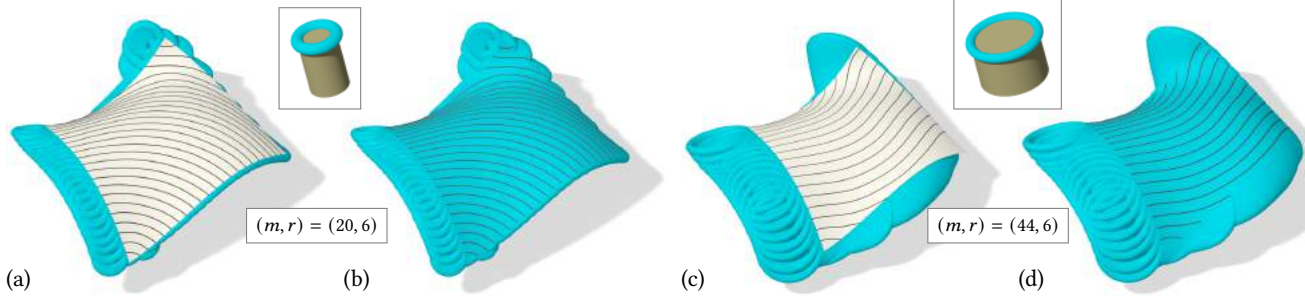


Fig. 21. Tool-dependent milling paths. (a,c) Given the reference surface S (light) and a specific milling tool T (framed), the milling paths that correspond to motions perpendicular (in the least square sense) to the second order line contact between S and T are computed (black). (b,d) The two swept volumes of the bottom of the tool (blue) that approximate the input surface within fine machining tolerances are shown. Note that the viewpoint is from the material side in order to show the part of the envelope that approximates S .

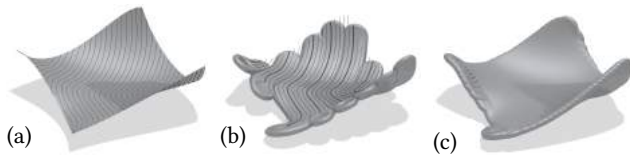


Fig. 22. Approximation by swept volumes. (a) A reference surface with a set of milling paths. Its approximation by a coarse (b) subset of toroidal swept volumes. (c) The whole surface approximated by 38 swept volumes. For visualization purposes, the objects are upside-down.

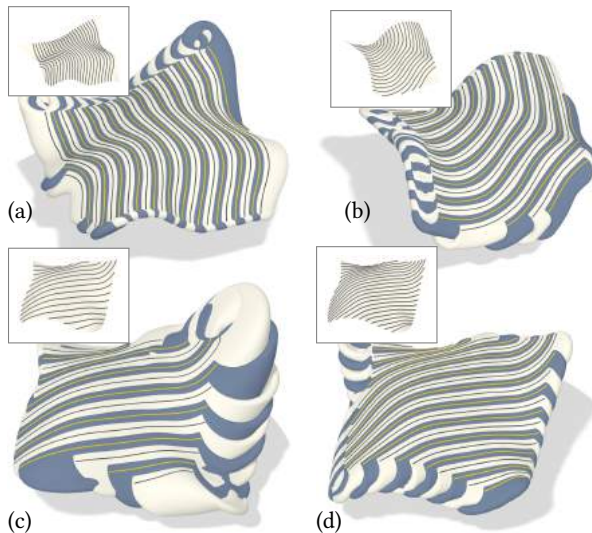


Fig. 23. Envelope patterns. A moving tool defines an envelope surface and the machining quality is governed by the error of the intersection curve of two neighboring envelope strips. The machining error is (theoretically) zero along the milling paths and maximal along the intersection curves. Therefore, a “bisector-like” pattern is desirable. The reference surfaces with the computed milling paths are showed framed.

error.) Such a high accuracy can be intuitively felt by looking at the shadows of the neighboring paths cast on the swept volume.

To mill S with high accuracy, the tool has to approach it with the appropriate part. Fig. 19(a) shows configurations of the tool in hyperbolic (blue) and convex elliptic (green) regions. While the

hyperbolic regions require the elliptic part of the tool (positive tilt), convex elliptic regions need to be approached by the hyperbolic part of the tool (negative tilt). The smooth transition of the tool throughout these regions is shown in Fig. 19(b).

Milling paths and some tool positions at a convex elliptic surface S are shown in Fig. 24. Typically, in state-of-the-art path planning CAM software such as Siemens NX or SprutCAM, a toroidal tool is navigated such that the bottom circle of the tool is in contact with a convex elliptic surface. In contrast, our approach poses the tool such that it touches the surface with its hyperbolic part, leading to higher approximation quality, and consequently better surface finish, see Fig. 25. Note that we use fewer number of milling strips than needed in real life applications to better demonstrate the difference between our approach and the commercial software.

Another important issue of highly-accurate milling are the widths of the machined strips. Even though it is possible to relax the constraint on parallel strips by decreasing w_3 in Eq. (11), we prioritize parallel strips. The reason for that is the fact that the milling error is zero along the milling paths and increases as one moves away from those. The maximum error between two neighboring strips occurs along the intersection of the two tool envelopes. Therefore, one aims at intersection curves at equal distance to the two milling paths. Fig. 23 shows the pattern of milling strips for various surfaces and milling tools. One can observe this parallel-like behaviour among the milling paths and the intersections of neighboring strips.

A comparison with the commercial software SprutCAM is shown in Fig. 26. SprutCAM requires user intervention to tune several parameters. We experimented with these parameters to optimize the performance and compare our results against the best result we could achieve with SprutCAM. A machined part exhibiting a general free-form surface is shown in Fig. 28. We again emphasize that the coarse use of milling paths is intentional, to visually demonstrate the high approximation quality and minimal scalloping effects with just a few milling paths.

A quantitative comparison with ball-end milling is shown in Fig. 27. Machining the convex surface of Fig. 24 is simulated by our approach and compared with ball milling while maintaining a given distance error. Our approach needs less than a third of the milling paths and still has significantly smaller kink angles at the intersections of neighboring strips, see Fig. 27(c,d).

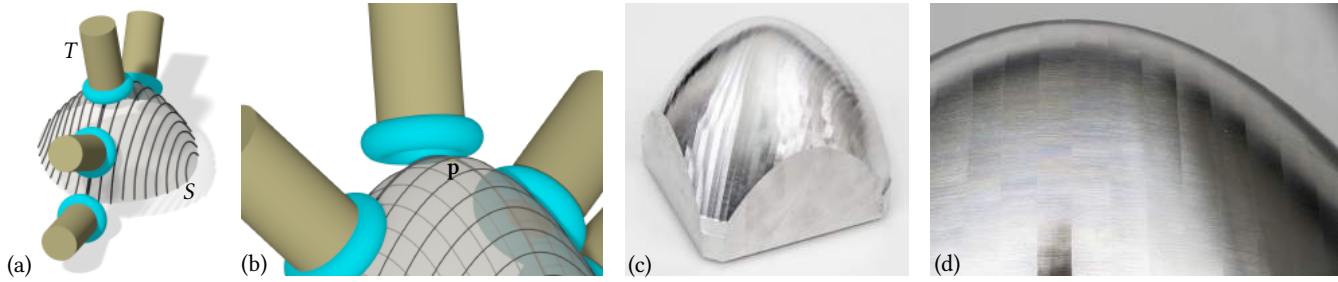


Fig. 24. High quality machining of a convex elliptic surface. (a) Motion of a milling tool along a milling path (bold) on a convex surface S . (b) The tool touches the surface with its negatively curved part at the contact point p . (c) Machined part using $n = 38$ milling paths. (d) High resolution of the surface silhouette.

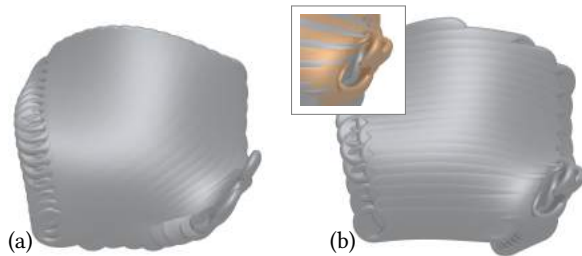


Fig. 25. Smooth surface finish. The surface shown in Fig. 21 is approximated by $n = 24$ milling strips using our algorithm (a) and the commercial software SprutCAM (b). Observe the smooth finish as well as equally distributed milling strips in (a) while (b) suffers by irregularities. A zoom-in of the bottom-right corner of (b) that highlights the irregular strip pattern is shown framed.

Another qualitative comparisons against toroidal cutting (non curvature-adapted) and ball-end cutting are shown in Fig. 29 via reflection lines. One can see that the smoothest reflection lines appear for our curvature-adapted method with a toroidal cutter.

7.1 Implementation details

The algorithms corresponding to mesh processing (computation of wedges, milling paths and collision detection) were implemented in Python 3.6.11, and the tool motion planning in Mathematica 12.2. Parallelized libraries like Trimesh, TensorFlow, and igl [Jacobson et al. 2018] were used to speed up the computation, including multi-threaded realizations.

The quality of the final tool paths depends on the resolution of the mesh. We worked with two sets of meshes: the fine ones (cca 50-150K vertices), that were pre-processed and used for collision detection, and the coarse ones (1-3K) that were used for the milling path construction, in particular for the optimization in Eq. (11). With that, we achieved a good balance in time-vs-performance, being able to run all the example optimizations on a low-end desktop PC Intel NUC6CAYB (Intel Celeron J3455) (below referred to as NUC), or a DELL XPS laptop with Intel i7-10510U. A substantial speed boost was achieved when running the code on a computer cluster (4 cores of Intel Xeon Gold 6140) and/or a GPU (NVIDIA Tesla K40c). A summary of simulation timings of all other examples shown in the paper, computed on a cluster, are summarized in Table 1.

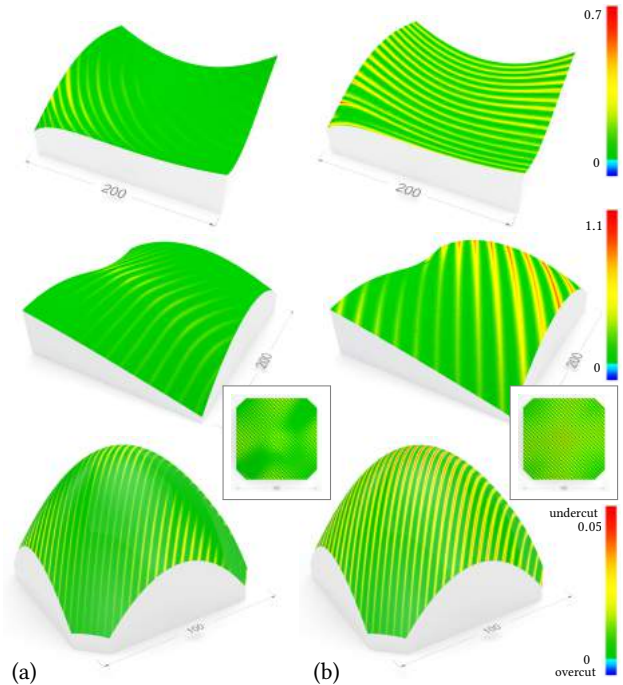


Fig. 26. Comparison with a commercial software. The surfaces shown in Fig. 21, Fig. 19, and Fig. 24 (from top to bottom) are approximated by $n = 28$, $n = 14$, and $n = 34$ milling strips, respectively, using our algorithm (a) and the commercial software SprutCAM (b). The machined envelopes are color-coded by the error to the reference surface. While both algorithms are penetration free (no overcutting) our algorithm produces a much smaller machining error (undercutting) and thus a better surface finish. As the convex surface is not fully visible, the top views are also shown (framed).

7.2 Discussion and limitations

We aimed at highly-accurate collision free machining. Milling paths c are constructed with a wish that second order line contact is possible in a direction perpendicular to the tangent direction t_c of c . Due to the size of the tool or global collision, this is not always possible and then one would need to rotate the tool w.r.t. t_c . This is not done in our current implementation; we only optimize the tilt angle. For mold-like reference geometries like ours, we always found a tilt angle that gives a globally collision-free position. This approach also generates nice milling patterns with maximum milling

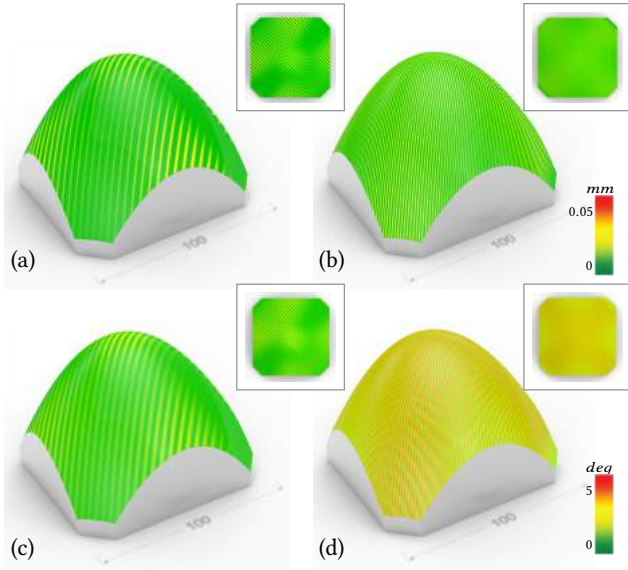


Fig. 27. Quantitative comparison with ball-end milling. Using the testing surface of Fig. 24, to achieve the same machining error $0.05mm$, our algorithm with a toroidal cutter ($r = 6mm$, $m = 21mm$) needs 38 milling paths (a), while ball-end cutting ($r = 10mm$) requires 143 milling paths to get the same maximum scallop height (b). Our results (c) outperform ball-end cutting (d) also in terms of angular difference (kink angle) between neighboring milling strips. In (c,d) the color coding represents this angular deviation for the threshold 0-5 degrees. The framed images are the top views to show the error on the whole surface.

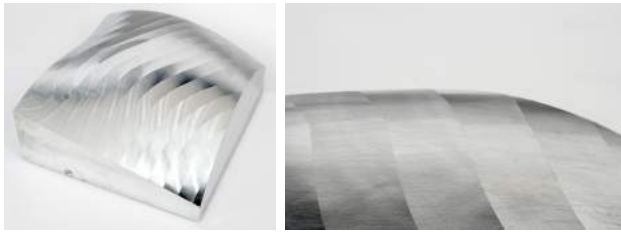


Fig. 28. Physical validation. Left: The surface shown in Fig. 26 (middle row) has been 5-axis milled. As the real machining is time demanding, a sparser set of the milling paths was used. The surface contains all types of points (convex elliptic, hyperbolic, concave elliptic), yet the transition between these regions is very smooth. Right: A zoom-in figure of a fine finish with a marginal scalloping effect; note the smoothness of the silhouette.

errors attained in the middle between neighboring milling paths, as seen in Fig. 23. It is an open question whether optimizing the rotation angle would significantly improve the machining quality.

Another place for possible improvements may be the global collision detection test. There, we consider a cylindrical shank instead of an eventually existing tighter approximation, cf. Fig. 15. This approach slightly restricts the search space of solutions. However, it is a conservative strategy that makes the computation more efficient, and it has not been an obstacle in finding non-colliding positions.

Our approach supports both discrete (triangle mesh) and smooth (B-spline) target surfaces S . If S is smooth, we still compute a triangulation and use it for path planning and point-surface projections.

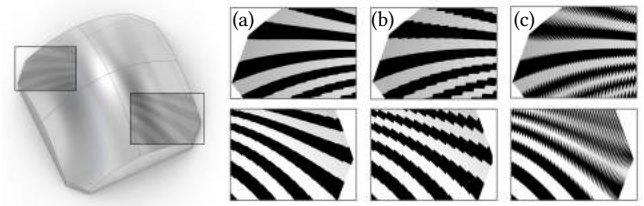


Fig. 29. Reflection lines on two patches of the surface from Fig. 24 are shown. (a) shows the result of our algorithm, (b) is based on the commercial software SprutCAM using a cylindrical center-cutting end mill, which is the best one can do on a convex surface without using a negatively curved cutter, and (c) is achieved by a ball-end cutter.

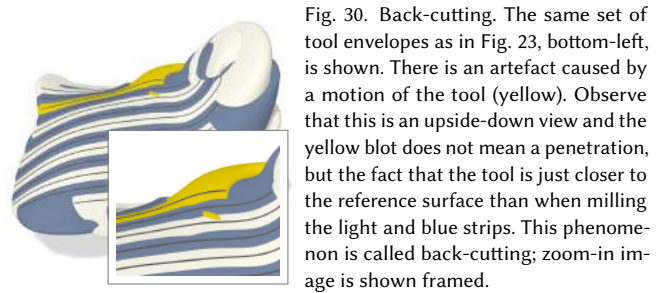


Fig. 30. Back-cutting. The same set of tool envelopes as in Fig. 23, bottom-left, is shown. There is an artefact caused by a motion of the tool (yellow). Observe that this is an upside-down view and the yellow blot does not mean a penetration, but the fact that the tool is just closer to the reference surface than when milling the light and blue strips. This phenomenon is called back-cutting; zoom-in image is shown framed.

Curvature computation is performed on the smooth representation, if available, otherwise we apply the jet-fit method [Cazals and Pouget 2003].

8 CONCLUSIONS & FUTURE RESEARCH

We have introduced a path-planning pipeline for curvature adapted 5-axis CNC machining of free-form surfaces. We conduct curvature analysis to compute tangent directions along which second order line contact between a toroidal cutting tool and the reference surface is possible, followed by a global collision detection test to remove positions that are not physically realizable. We use this analysis to design collision free tool motions which yield nearly parallel milling strips and generate surfaces which well approximate the target geometry and reduce kink angles at intersections of neighboring strips. As a result, our approach offers highly accurate machining that needs a lower number of milling paths than other techniques, and consequently reduces the machining time. We have tested the proposed methodology on various free-form surfaces, compared it

Table 1. Timings of three computation stages: 1. Tool positioning and collision detection on a coarse mesh; 2. Path optimization on a coarse mesh; 3. Tool motion planning and validation on a fine mesh, for different models used in the paper (run on 4 cores of Intel Xeon Gold 6140)

Example	Time per stage			Total time (min:sec)
	st. 1	st. 2	st. 3	
Fig. 21(a)	2:18	6:13	1:57	10:28
Fig. 21(c)	4:53	10:21	8:07	23:21
Fig. 22	6:18	12:55	15:54	35:07
Fig. 24	2:51	5:17	1:35	9:43
Fig. 23(a)	4:50	12:53	4:26	22:09
Fig. 23(b)	1:59	6:13	2:07	10:19
Fig. 23(c)	7:27	21:27	6:06	35:00
Fig. 23(d)	8:57	21:17	22:19	52:33

to state-of-the-art CAM path-planning software, and validated it by real machining experiments.

A direction for future research is the avoidance of artefacts created by *back-cutting*. One can observe this artefact in Fig. 23, bottom-left, where one of the light envelopes interferes the blue neighbor. This is caused by another sweep of the tool; see the yellow envelope in Fig. 30. While the manufacturing would still be collision-free, the impact on the surface finish is undesirable.

Another thread for future research is incorporating the physics of the milling process into the path-planning loop. We postulated that wide milling strips reduce machining time. However, there are other motion parameters, such as feed rate, that need to be aligned according to the milling paths. A related issue is incorporating feed parameters (e.g., velocity, acceleration, and jerk drive limits) of a particular milling machine into the path-planning process.

ACKNOWLEDGMENTS

This work was supported by the Spanish Ministry of Science, Innovation and Universities, grant No PID2019-104488RB-I00, BCAM “Severo Ochoa” accreditation (SEV-2017-0718), and by the European Union’s Horizon 2020 program under grant agreement No 862025. Michael Bartoň was supported by the Ramón y Cajal fellowship RYC-2017-22649. Michal Bizzarri was funded by KAUST under BRF grant No 3989 and by the project LO1506 of the Czech Ministry of Education, Youth and Sports. Florian Rist was supported by KAUST baseline funding.

REFERENCES

Yusuf Altintas. 2012. *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*. Cambridge Univ. Press.

Xavier Beudaert, Sylvain Lavernhe, and Christophe Tournier. 2012. Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path. *International Journal of Machine Tools and Manufacture* 57 (2012), 73–82.

Wilhelm Blaschke and Kurt Leichtweiß. 1973. *Elementare Differentialgeometrie*. Springer, Berlin-New York. Fünfte vollständig neubearbeitete Auflage von K. Leichtweiß, Die Grundlehren der mathematischen Wissenschaften, Band 1.

Amaia Calleja, Pengbo Bo, Haizea González, Michael Bartoň, and Luis Norberto López de Lacalle. 2018. Highly accurate 5-axis flank CNC machining with conical tools. *The International Journal of Advanced Manufacturing Technology* 97, 5 (2018), 1605–1615.

Francisco J. Campa, Luis Norberto López de Lacalle, Aitzol Lamikiz, and José Antonio Sanchez. 2007. Selection of cutting conditions for a stable milling of flexible parts with bull-nose end mills. *Journal of Materials Processing Technology* 191, 1 (2007), 279–282.

Li-Xin Cao, Hu Gong, and Jian Liu. 2007. The offset approach of machining free form surface: Part 2: Toroidal cutter in 5-axis NC machine tools. *Journal of Materials Processing Technology* 184, 1-3 (2007), 6–11.

Frederic Cazals and Marc Pouget. 2003. Estimating differential quantities using polynomial fitting of osculating jets. In *Symp. Geometry processing*, L. Kobbelt, P. Schröder, and H. Hoppe (Eds.). Eurographics, 177–178.

Jung-Woo Chang, Wenping Wang, and Myung-Soo Kim. 2010. Efficient collision detection using a dual OBB-sphere bounding volume hierarchy. *Computer-Aided Design* 42, 1 (2010), 50–57.

Byoung K. Choi and Robert B. Jerard. 1998. *Sculptured Surface Machining: Theory and Applications*. Kluwer.

Dawson-Haggerty et al. 2019. *trimsh*. <https://trimsh.org/>

LN López De Lacalle, C Angulo, A Lamikiz, and JA Sanchez. 2006. Experimental and numerical investigation of the effect of spray cutting fluids in high speed milling. *Journal of Materials Processing Technology* 172, 1 (2006), 11–15.

Simon Duenster, Roi Poranne, Bernhard Thomaszewski, and Stelian Coros. 2020. Robo-Cut: hot-wire cutting with robot-controlled flexible rods. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 98–1.

Serafettin Engin and Yusuf Altintas. 2001. Mechanics and dynamics of general milling cutters.: Part I: helical end mills. *International Journal of Machine Tools and Manufacture* 41, 15 (2001), 2195–2212.

Jianhua Fan and Alan Ball. 2014. Flat-end cutter orientation on a quadric in five-axis machining. *Computer-Aided Design* 53 (2014), 126–138.

Ramy Harik, Hu Gong, and Alain Bernard. 2013. 5-axis flank milling: A state-of-the-art review. *Computer-Aided Design* 45, 3 (2013), 796–808.

Bernard W Ikua, Hisataka Tanaka, Fumio Obata, and Satoshi Sakamoto. 2001. Prediction of cutting forces and machining error in ball end milling of curved surfaces-I theoretical analysis. *Precision Engineering* 25, 4 (2001), 266–273.

Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. <https://libigl.github.io/>.

C. Greg Jensen and D. G. Anderson. 1993. Accurate tool placement and orientation for finish surface machining. *J. Design and Manufacturing* 3, 4 (1993), 251–261.

C. Greg Jensen, W. Edward Red, and C. Ernst. 2008. Machining Free-Form Surface Cavities Using a Combination of Traditional and Non-Traditional Multi-Axis Machining Methods. *Computer-Aided Design & Applications* 5 (2008), 241–253.

C Greg Jensen, W Edward Red, and J Pi. 2002. Tool selection for five-axis curvature matched machining. *Computer-Aided Design* 34, 3 (2002), 251–266.

Yong-Joon Kim, Michael Bartoň, Gershon Elber, and Helmut Pottmann. 2015. Precise gouging-free tool orientations for 5-axis CNC machining. *Computer-Aided Design* 58 (2015), 220–229.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Ali Lasemi, Deyi Xue, and Peihua Gu. 2010. Recent development in CNC machining of freeform surfaces: A state-of-the-art review. *Computer-Aided Design*, 42, 7 (2010), 641–657.

S. X. Li and R. B. Jerard. 1994. 5-axis machining of sculptured surfaces with a flat-end cutter. *Computer-Aided Design* 26, 3 (1994), 165–178.

Jinesh Machchhar, Denys Plakhotnik, and Gershon Elber. 2017. Precise algebraic-based swept volumes for arbitrary free-form shaped tools towards multi-axis CNC machining verification. *Computer-Aided Design* 90 (2017), 48–58.

Ali Mahdavi-Amiri, Fenggen Yu, Haisen Zhao, Adriana Schulz, and Hao Zhang. 2020. VDAC: volume decompose-and-carve for subtractive manufacturing. *ACM Transactions on Graphics (TOG)* 39, 6 (2020), 1–15.

Gustav J. Olling, Byoung K. Choi, and Robert B. Jerard. 1999. *Machining Impossible Shapes*. Kluwer, Boston.

Pierre-Yves Pechard, Christophe Tournier, Claire Lartigue, and Jean-Pierre Lugarini. 2009. Geometrical deviations versus smoothness in 5-axis high-speed flank milling. *International Journal of Machine Tools and Manufacture* 49 (2009), 453–461.

Daive Pellis, Hui Wang, Martin Kilian, Florian Rist, Helmut Pottmann, and Christian Müller. 2020. Principal symmetric meshes. *ACM Trans. Graphics* 39, 4 (2020), 127:1–127:17.

Helmut Pottmann and Johannes Wallner. 2001. *Computational Line Geometry*. Springer.

Stephane Redon, Ming C Lin, Dinesh Manocha, and Young J Kim. 2005. Fast continuous collision detection for articulated models. *Journal of Computing and Information Science in Engineering* 5, 2 (2005), 126–137.

Alec Rivers, Ilan E Moyer, and Frédo Durand. 2012. Position-correcting tools for 2D digital fabrication. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–7.

David Roth, Sanjeev Bedi, Fathi Ismail, and Steven Mann. 2001. Surface swept by a toroidal cutter during 5-axis machining. *Computer-Aided Design* 33, 1 (2001), 57–63.

Knut Sorby, Kjell Tonnessen, Jan Erik Torjusen, and Finn Ola Rasch. 2000. Improving high speed flank milling operations in multi-axis machines. *CIRP Annals-Manufacturing Technology* 49, 1 (2000), 371–374.

Oded Stein, Eitan Grinspun, and Keenan Crane. 2018. Developability of Triangle Meshes. *ACM Transactions on Graphics* 37, 4 (2018), 1–14. Proc. SIGGRAPH.

Tran Duc Tang. 2014. Algorithms for collision detection and avoidance for five-axis NC machining: A state of the art review. *Computer-Aided Design* 51 (2014), 1 – 17. <https://doi.org/10.1016/j.cad.2014.02.001>

Xiaochun Wang, S. K. Ghosh, Y. B. Li, and Xutang Wu. 1993a. Curvature catering – a new approach in manufacture of sculptured surfaces (Part 1. theorem). *Journal of Materials Processing Technology* 38, 1–2 (1993), 159–175.

Xiaochun Wang, S. K. Ghosh, Y. B. Li, and Xutang Wu. 1993b. Curvature catering – a new approach in manufacture of sculptured surfaces (Part 2. methodology). *Journal of Materials Processing Technology* 38, 1–2 (1993), 177–193.

Andrew Warkentin, Fathi Ismail, and Sanjeev Bedi. 2000. Comparison between multi-point and other 5-axis tool positioning strategies. *Journal of Machine Tools & Manufacture* 40 (2000), 185–208.

Joung-Hahn Yoon, Helmut Pottmann, and Yuan-Shin Lee. 2003. Locally optimal cutting positions for five-axis sculptured surface machining. *Computer-Aided Design* 35 (2003), 69–81.

Haisen Zhao, Hao Zhang, Shiqing Xin, Yuanmin Deng, Changhe Tu, Wenping Wang, Daniel Cohen-Or, and Baoquan Chen. 2018. DSCarver: Decompose-and-Spiral-Carve for Subtractive Manufacturing. *ACM Transactions on Graphics (Special Issue of SIGGRAPH)* 37, 4 (2018), Article No. 137.

L. Zhu, G. Zheng, H. Ding, and Y. Xiong. 2010. Global optimization of tool path for five-axis flank milling with a conical cutter. *Computer-Aided Design* 42, 10 (2010), 903–910.

Qiang Zou, Juyong Zhang, Bailin Deng, and Jibin Zhao. 2014. Iso-level tool path planning for free-form surfaces. *Computer-Aided Design* 53 (2014), 117 – 125. <https://doi.org/10.1016/j.cad.2014.04.006>