Generalization of the Imprint Method to General Surfaces of Revolution for NC Machining

Stephen Mann

University of Waterloo, Department of Computer Science, Waterloo, Ontario, Canada. N2L 3G1. (519) 888-4567, Fax (519) 885-1208

Sanjeev Bedi

University of Waterloo, Department of Mechanical Engineering, Waterloo, Ontario, Canada. N2L 3G1. (519) 888-4567, Fax (519) 888-6197

Abstract

This paper presents a method of determining the shape of the surface swept by a generalized milling tool that follows a 5-axis tool path for machining curved surfaces. The method is a generalization of an earlier technique for toroidal tools that is based on identifying grazing points on the tool surface. We present a new proof that the points constructed by this earlier method are in fact grazing points, and we show that this previous method can be used to construct grazing points on (and only on) the sphere, the cone, and the torus. We then present a more general method that can compute grazing points on a general surface of revolution. The advantage of both methods is that they use simple, geometric formulas to compute grazing points.

Key words: 5-axis machining, grazing curves, tool path verification

1 Introduction

Automation of the manufacturing process from the nominal part geometry on a CAD system to the final machined part offers the opportunity for huge gains in productivity and cost savings. The advent of 5-axis machining and

[1] Email: smann@cgl.uwaterloo.ca
[2] Email: sbedi@surya.uwaterloo.ca

methods for generating NC tool paths has already offered the opportunity to reduce machining time by up to 85% [1]. However, this added flexibility also brings added complexity. Research efforts have concentrated on generating interference free NC tool paths that also produce machined parts free from excessive gouging or under cutting. Central to these ideas is the generation of the swept volume of the tool along its programmed NC tool path, and the ideas of the simulation, verification and correction of NC tool path programs [2].

Envelope theory and SDE [3,4,5] provide a general framework for computing the volume swept by a tool undergoing 5-axis motion, where the surface is found by solving a system of implicit equations. While further work has refined this method (see [6,7,8] for example), the basic approach still requires solving a system of implicit equations. One variation that is computationally less expensive is the SEDE [9,10], which requires fewer solutions of the systems of implicit equations and instead finds the swept volume by using Runge-Kutta techniques. Vector methods offer an approximation based on envelope theory, the SDE, or on static instances of the tool at various locations [11,12,13]. While these vector methods can approximate the tool movement by using many instances of the tool at intermediately interpolated positions, the computational cost is prohibitive.

A few other methods have been proposed for 3-axis and 4-axis generation of swept volumes by finding explicit equations for curves on the surface (which reduces the computation cost by avoiding the need to solve systems of implicit equations), namely using silhouettes and generating curves [14,1]. This paper presents a simple, geometric method for computing grazing points on any surface of revolution, generalizing the earlier 5-axis work of Roth et al. [15], which was restricted to toroidal tools. In Section 2, the concept of a grazing curve (the curve that leaves its imprint behind as the tool moves from one programmed location to the next) for 5-axis machining and the cross product method of Roth et al. are reviewed. Further, we then show that the cross product method can compute grazing points on spheres, cylinders, and tori, but that it can compute grazing points on those surfaces only. Then in Section 4, we extend the work of Roth et al. to general NC tools, using the conical tool as our example.

Our focus has been on NC machining, and thus surfaces of revolution, and we have not looked at more general surfaces. The advantage of our method over other variations of envelope theory is simplicity. By exploiting the special properties of surfaces of revolution, we derive a simple equation for directly computing points on the swept surface without needing to solve systems of equations or to use numerical techniques.

2 Grazing Curves and the Cross Product Imprint Method

Chung, Park, Shin and Choi [14] present silhouette curves in a method for determining the surface swept by a generalized APT cutter for 3-axis machining. The generalization of Chung et al.'s method to 5-axis machining is non-trivial. In 5-axis machining, a *grazing curve* is the set of points on the rotating tool surface at which the direction of motion lies in the tangent plane of the cutter [9]; these are the points on the cutter surface that remain on the swept surface unless milled away in a different pass of the cutter. The silhouette curve used in 3-axis machining is a special case of a grazing curve where there is no rotation. (Note: in an earlier paper, we referred to these points as *imprint points* [15]; as Blackmore, Leu, and Wang's definition pre-dates ours, we adopt their terminology. However, for continuity with our previous paper, we will refer to our geometric construction as the *imprint method*.)

In SDE, the grazing points are found by solving implicit equations for the surface and the tangent condition on the derivative. The SEDE reduces the computational cost by using these points as the starting points for a Runge-Kutta solver for differential equations. However, the observation made in the Roth et al. paper is that the grazing points can be directly and quickly computed for a toroidal surface, thus avoiding both solving the implicit equations and the Runge-Kutta computation. Like many SDE methods, Roth et al. connected these points to form a piecewise linear approximation to the grazing curve, and then connected these grazing curves to form the swept surface. The critical step in this algorithm is the computation of grazing points, which we discuss in this section and in Section 4.

The Roth et al. method computes grazing curves on toroidal cutters [15]. The approach in that paper (illustrated in Figure 1) is to slice the toroidal cutter with planes through the tool axis. Each plane cuts the torus into two circles. The grazing points on each circle are then computed by forming the vector $\vec{r}$ that is the cross-product of the direction of motion $\vec{d}$ of the center of the circle with the normal $\vec{n}$ to the plane of the circle. The line from the center of the circle in direction $\vec{r}$ is intersected with the circle; these intersection points (P_0 and P_1) are the grazing points.

Although the Roth et al. paper illustrated this method and showed its validity using experimental methods, the justification did not allow for a generalization to arbitrary surfaces of revolution. We now present an alternative justification for why the cross product method produces grazing points, which we will then use to extend the method to general surfaces of revolutions.

The idea behind the cross product method is that if we know the direction of motion of a point at the center of a circular slice of a torus, then it is easy to

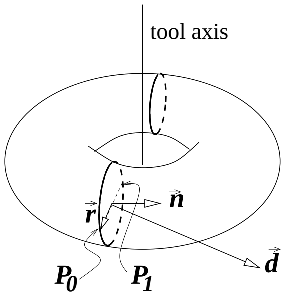

Fig. 1. Cross product method for torus.

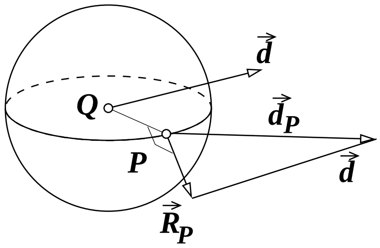

Fig. 2. In a rigid body motion, $\vec{d}_P = \vec{d} + \vec{R}_P$, where $\vec{R}_P$ lies in the tangent plane at P.

compute the grazing points that lie on the circle. To understand why the cross product method works, we first note that for any rigid body transformation T and for any point Q in space, we can describe T as a translation of Q followed by a rotation around Q. Now consider a point P on a sphere with center Q (Figure 2). If the instantaneous motion of Q under a rigid body transformation is $\vec{d}$, then the motion $\vec{d}_P$ of P differs from $\vec{d}$ only by a rotation around Q, i.e., only by a vector $\vec{R}_P$ in the tangent plane of the sphere at P:

$$\vec{d}_P = \vec{d} + \vec{R}_P.$$

Since $\vec{R}_P$ lies in the tangent plane of the sphere at P, $\vec{d}_P$ will lie in the tangent plane at P if and only if $\vec{d}$ lies in the tangent plane at P. Therefore, the grazing points will be those points on the sphere where $\vec{d}$ lies in the tangent plane at the point. One final property to note is that at a grazing point R on the sphere, $R - Q$ is perpendicular to both the tangent plane at R and to $\vec{d}$.

The idea behind the cross product method is that if we know the direction of motion of a point on the center Q of a circular slice C of the torus is $\vec{d}$, then for any point P on the circle, we know P has a motion that differs from $\vec{d}$ only by a vector lying in the tangent plane at P on a sphere centered at Q having radius equal to that of C. However, along C such a sphere has the same tangent planes as the torus. Thus, the cross product of $\vec{n}$ (the normal to the plane of C) and $\vec{d}$ yields a vector $\vec{r}$ that is perpendicular to $\vec{d}$ and perpendicular to the tangent plane of the torus at P_0 and P_1. Therefore, $\vec{d}$ lies in the tangent plane at P_0 and P_1, and these points are grazing points.

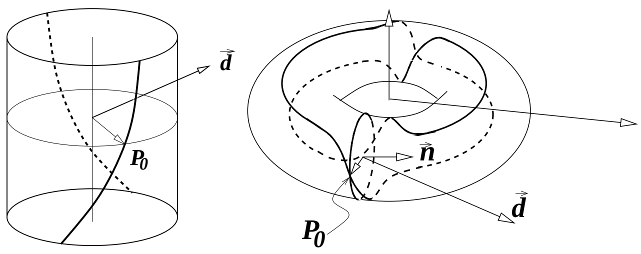

Fig. 3. Grazing curves on the cylinder and the torus.

It is clear that the cross product method can be used for any surface that can be defined as the swept surface of a sphere of fixed radius. For machining surfaces, this restricts this method to the torus, the cylinder, and the sphere.

3 Discussion and Examples of Grazing Curves

Using the cross product method, we can compute grazing curves for cylinders, tori, and spheres. However, there is a simpler way to compute the grazing curve for a sphere: it is a great circle lying in the plane through the sphere's center perpendicular to the direction of motion of the center. Examples of grazing curves on the cylinder and the torus appear in Figure 3; note that for circular slices on these surfaces, grazing points occur in pairs, 180 degrees apart on the circle. Further, as discussed by Roth et al., for machining purposes, only about half of one of the two grazing curves on the torus is in contact with the machined surface, and the remaining portions need not be considered for computing the surface (although they could be used for gouge detection).

The situation is more complex for general surfaces of revolution. As a representative shape, we use the cone, although the ideas apply to general surfaces of revolution. In Figure 4, we see some examples of the grazing curves for different motions of a conical tool (the details of computing these motions are discussed in the next section). Figures (a)-(d) are translational motions only, while figures (e) and (f) are a translation and a rotation of the cone about a line through the tool tip. In figure (a), the motion vector is perpendicular to the tool axis, and there are two grazing points on each circular slice, which are 180 degrees apart on the circle. However, when the motion vector is no longer perpendicular to the tool axis (b), the grazing curves are no longer 180 degrees apart, although they are still lines on the cone. When the motion vector is parallel to a line on the cone (c), the two grazing curves meet, resulting in a single grazing curve on the cone. When the motion vector nearly aligns with the tool axis (in (d), whenever the motion vector lies within the small cone that is parallel to the conical tool), the larger circle of the truncated cone becomes the grazing curve.

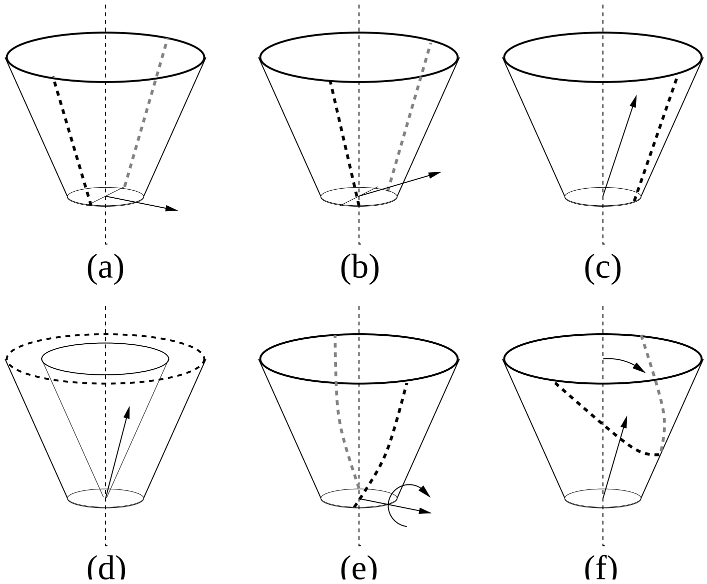

Fig. 4. Dotted lines show the grazing curves for different motions.

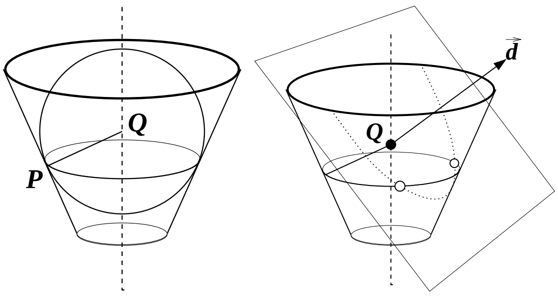

Fig. 5. Computing grazing points on a cone.

When we rotate the tool, the grazing curves will usually appear as those in figure (e); i.e., there are two curves, but they are no longer straight lines on the cone. However, for some motions, the grazing curve becomes a single curve on the cone (f). In the next section, we describe how to compute grazing points on a circular slice of a generalized milling tool.

4 Generalization of the Cross Product Imprint Method

The cross product method works because we have circular slices of a surface where the tangent planes of the surface are identical to those of a sphere centered at the circle center, with radius equal to that of the circle. This restricts the cross product method to the torus, the cylinder, and the sphere.

We can generalize the cross product method to general surfaces of revolution as follows (which we illustrate for a cone in Figure 5): for any point P on the generating curve for the surface of revolution, find the point Q on the axis of revolution such that $P - Q$ is parallel to the normal to the generating curve at P. Now a sphere centered at Q of radius $|P - Q|$ will be tangent to the surface of revolution at all points generated by rotating P around the axis of revolution. In particular, note that as we translate/rotate the surface, the motion of each point on this circle will differ from the motion $\vec{d}$ of Q only by a rotation around Q. Thus, the points on the circle for which $\vec{d}$ lies in the tangent plane of the surface are grazing points.

Our method to compute the grazing points (illustrated in Figure 5, right) is:

> Take the plane through Q perpendicular to $\vec{d}$, and then intersect this plane with the circle of revolution through P. This will yield zero, one, or two points. By design, the vector from Q to each of these points is perpendicular to both the tangent plane at each point and to $\vec{d}$. Thus, $\vec{d}$ lies in the tangent plane at each point, and the points are grazing points.

In general, the method for computing grazing points described in the previous paragraph will give two grazing points for each circular slice of the tool. This is the case for a wide variety of motions (Figure 4, (a), (b), and (e)). However, for some motions, there may be zero or one grazing point, or the entire circular slice may be a grazing curve. These special cases occur for *plunging motions*, or motions that are nearly plunging motions.

A plunging motion occurs when the direction of motion of the tool is in the direction of the tool axis or more generally, when the only points not milled away in a differential time step following (or proceeding) the current location are the circular slice(s) of the tool of locally largest radius. To complicate matters for our method, for some tool positions/motions, some circular slices of the tool have grazing points while others do not.

Regardless, to compute the swept surface by a truncated surface of revolution, we commonly have to add a portion of the circle in the truncated region to the grazing curves, or in some cases, connect grazing curves on two surface. This addition of edge effects is well known in the swept surface community; we mention it here only to point out that our method has to handle this problem as a special case.

For example, the cylinder is part of both a cylindrical tool and of a ball-end mill. In general, two grazing curves run from the top circle of the cylinder, along the sides of the cylinder, down to the bottom circle of the cylinder. At the bottom of the cylinder, these two curve must be connected to finish the curve that mills the surface. For the ball-end mill, we connect the two grazing curves on the cylinder with the grazing curve computed for the sphere (Figure 6, right). In the case of a cylindrical tool, the bottom edge of the cylindrical tool will also machine the swept surface (Figure 6, left), and we

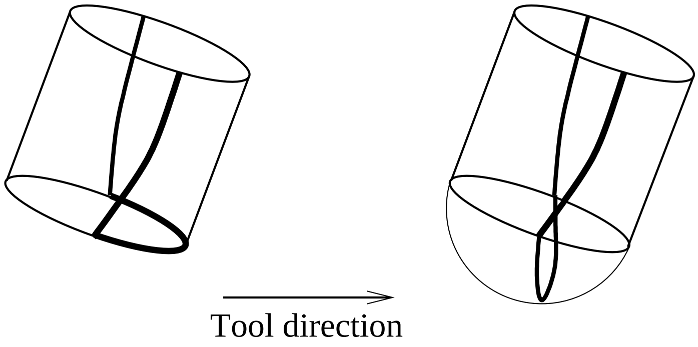

Fig. 6. Grazing curve (the thick line) will contain curve on tool bottom.

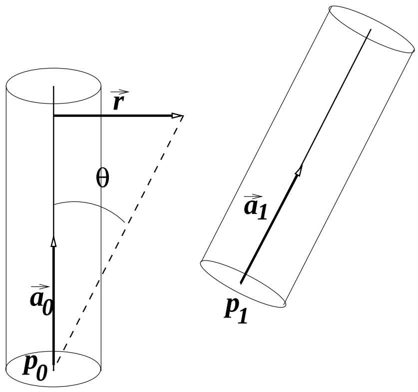

Fig. 7. A tool in two positions.

need to select between the two edges of the tool bottom to connect these two grazing curves. For the complete swept surface, the top edges of the grazing curves should also be connected; however, this is often not required for NC-machining, since that edge is never in contact with the work piece.

4.1 Computing the Tool Motion

To compute the grazing points, we need to know $\vec{d}$, the direction of motion of a point on the axis of the tool. This vector will depend on the mechanics of the machine being used. If the mechanics are unknown, then we will have to make assumptions about the tool motion. One approach is to take two tool positions and assume a piecewise linear motion, with the rotation taking place around the tool tip.

For a piecewise linear motion, assume we have two tool positions, with the tool tip at p_0 at the first tool position and at p_1 at the next tool position (Figure 7). Let $\vec{r}$ be the vector perpendicular to the tool axis $\vec{a}_0$ in the plane containing p_0, $\vec{a}_0$, and $\vec{a}_1$ (if $\vec{a}_0$ and $\vec{a}_1$ are parallel, then choose the plane containing p_0, p_1, and $\vec{a}_0$). Then the linear motion M parameterized over time t (over $[0, 1]$) and along the tool (parameterized by u, with $u = 0$ being the tool tip) gives us

$$M(t, u) = T(t) + R(t, u),$$

where $T(t) = (1 - t)p_0 + tp_1$ and $R(t, u) = u(\sin(t)\vec{r} + \cos(t)\vec{a}_0)$. Note that M is the sum of a linear motion and a circular arc. Differentiating with respect to t gives us

$$M'(t, u) = p_1 - p_0 + u(\cos(t)\vec{r} - \sin(t)\vec{a}_0),$$

which can be used as $\vec{d}$ in the calculation of grazing points.

However, if the mechanics of the machine are known, they should be used to compute the motion vectors. A further discussion of tool motion may be found in a companion paper [16].

4.2 From Grazing Points to a Surface

The idea of our method for approximating the swept surface is to slice the tool into planar slices, and use either the cross product method or the method described in this paper to compute grazing points on the tool. We then connect corresponding grazing points on adjacent tool positions with line segments, and triangulate to obtain a piecewise linear approximation to the swept surface. Details of the algorithm can be found in the paper of Roth et al. [15]. This construction of a piecewise linear surface is similar to that of many SDE/SEDE, and the SDE/SEDE papers should be consulted for further issues that arise such as trimming.

We tested our ideas in the symbolic algebra package Maple [17], using Maple to generate points on the swept surface using the methods described in this paper, and then plotting the surface swept by a single movement. In Figures 8 and 9, we show the surfaces swept by a cylindrical ball-end tool and by a conical tool. In both figures, there is a subfigure of the swept surface and a subfigure of the swept surface together with the tool at three positions and the surface swept by the tool axis. On the sides of the tool in dark lines are the grazing curves on the tool in each of the three positions. In the conical tool, we did not sweep the surface generated by the bottom edge of the tool.

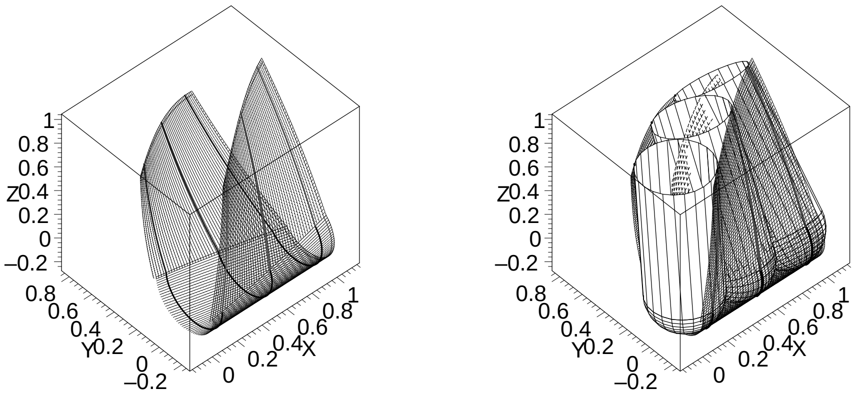

Fig. 8. Surface swept by cylindrical tool.

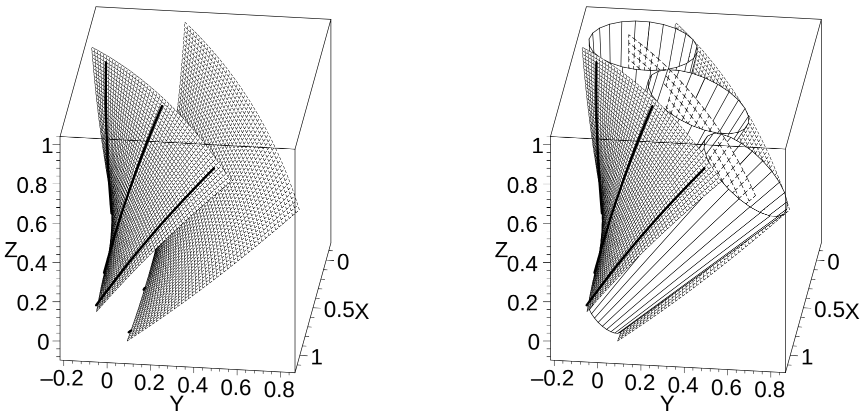

Fig. 9. Surface swept by conical tool.

5 Conclusions

In this paper, we have extended the imprint method to simulate the surfaces cut with mills made of general surfaces of revolution. The main contribution is a simple, geometric formula for computing the grazing points. Our focus has been on NC machining, and thus surfaces of revolution, and we have not looked at more general surface. However, our method should generalize to surfaces that have simple planar slices.

Although our new method is a generalization of the cross product method, the cross product method is simpler and it may make sense to use it in the special cases of the cylinder and the torus. While we could also use the cross product method to compute grazing points on the sphere, we note that the grazing curve on a sphere will always be a great circle, and is easily calculated and sampled without using the cross product method.

References

[1] K. Sheltami, S. Bedi, and F. Ismail. Swept volumes of toroidal cutters using generating curves. *International Journal of Machine Tools and Manufacture*, 38:855–870, 1998.

[2] R.B. Jerard, S.Z. Hussaini, R.L. Drysdale, and B. Schaudt. Approximate methods for simulation and verification of numerically controlled machining programs. *The Visual Computer*, 5(6):329–348, 1989.

[3] W.P. Wang and K.K. Wang. Geometric modeling for swept volume of moving solids. *IEEE Computer Graphics and Applications*, 6(12):8–17, 1986.

[4] R.R. Martin and P.C. Stephenson. Sweeping of three-dimensional object. *Computer-Aided Design*, 22(4):223–234, 1990.

[5] D. Blackmore, M.C. Leu, and F. Shih. Analysis and modelling of deformed swept volumes. *Computer-Aided Design*, 26(4):315–326, 1994.

[6] K. Abdel-Malek and H.-J. Yeh. Geometric representation of the swept volume using jacobian rank-deficiency conditions. *Computer-Aided Design*, 29(6):457–468, 1997.

[7] K. Abdel-Malek, H.-J. Yeh, and S. Othman. Swept volumes: void and boundary identification. *Computer-Aided Design*, 30(13):1009–1018, 1998.

[8] K. Abdel-Malek, W. Seaman, and H.-J. Yeh. An exact method for nc verification of up to 5-axis machining. In *Proceedings of DETC99*. ASME, September 1999.

[9] D. Blackmore, M.C. Leu, and L.P. Wang. Applications of flows and envelopes to nc machining. *Annals of the CIRP*, 41(1):493–496, 1992.

[10] D. Blackmore, M.C. Leu, and L.P. Wang. The sweep-envelope differential equation algorithm and its application to NC machining verification. *Computer-Aided Design*, 29(9):629–637, 1997.

[11] K.C. Hui. Solid sweeping in image space - application in NC simulation. *The Visual Computer*, 10(6):306–316, 1994.

[12] R. B. Jerard, R. L. Drysdale, K. Hauck, B. Schaudt, and J. Magewick. Methods for detecting errors in numerically controlled machining of sculptured surfaces. *IEEE Computer Graphics and Applications*, 9(1):26–39, 1989.

[13] I.T. Chappel. The use of vectors to simulate material removed by numerically controlled milling. *Computer-Aided Design*, 15(3):156–158, 1983.

[14] Y. C. Chung, J. W. Park, H. Shin, and B. K. Choi. Modeling the surface swept by a generalized cutter for NC verification. *Computer-Aided Design*, 30(8):587–594, 1998.

[15] D. Roth, S. Bedi, F. Ismail, and S. Mann. Surface swept by a toroidal cutter during 5-axis machining. *Computer-Aided Design*, 33(1):57–63, 2001.

[16] S. Mann and S. Bedi. Tool motion for 5-axis machining. In preparation.

[17] M.B. Monagan, K.O. Geddes, G. Labahn, and S. Vorketter. *The Maple Programming Guide*. Springer-Verlag, 1996.