Numerical implementation of drop spin and tilt method for five-axis tool positioning for tensor product surfaces

Sandeep Kumar Sharma1 · Ravinder Kumar Duvedi1 · Sanjeev Bedi2 · Stephen Mann2

Received: 16 February 2021 / Accepted: 19 April 2021 / Published online: 19 May 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract
This paper presents an extension of multipoint machining technique, called the Drop Spin and Tilt (DST) method, that spins the tool on two axes, allowing for the generation of multiple contact points at varying distances around the first point of contact. The multiple DST second points of contact were used to manually generate a toolpath with uniform spacing between the two points of contact. The original DST method used a symbolic algebra package to position the tool on a bi-quadratic surface; our extension is a numerical solution that allows positioning a toroidal tool on a tensor product Bézier surface, which we tested on bi-degrees 2, 3, 4, and 5. On the three bicubic surfaces we tested, the average number of initial seeds for our numerical method was less than 5 across all three surfaces, and the average number of Newton iterations was less than 20 except when the curvature of the tool closely matches the local surface curvature (in which case more iterations were required). Furthermore, we investigate the spread of possible second points of contact as the tool is spun around these two axes, demonstrating the feasibility of using the method to control the machining strip width.

Keywords CNC machining · 5-axis machining · Multipoint machining · Bézier surface · DTM

1 Introduction

5-axis machining is a reliable and efficient method to machine geometrically complex parts found in tool and die manufacturing, aerospace, automotive, and marine industries. 5-axis machines may use ball, flat, and toroidal (radiused) end milling cutters, but the cutting characteristics of the toroidal cutters are better at the bottom tip of tool [1] thus offering better surface finish. The geometrical details of a toroidal end milling tool are shown in Fig. 1.

5-axis NC machining offers improved cutting efficiency, such as higher material removal rate and surface finish, over 3-axis machining. These advantages are drawn from the flexibility in the machine kinematics that enables tool positioning in a close proximity to a curved design surface.

Irrespective of the growing complexity of the parts, the focus of most current research has been on improving the machining times while maintaining the surface finish. Several 5-axis tool positioning strategies for machining of sculptured surfaces have been developed by researchers to address these issues. These approaches can be classified into two sets.

In the first set, the tool is positioned so that tool curvature matches the local curvature of the design surface in the area surrounding the single contact point. The curvature matching methods do not guarantee a gouge free toolpath as such. Thus, gouge detection is integrated with the positioning and inclination as a secondary step.

In the second set of solutions, the tool simultaneously touches the workpiece at more than one contact points and thus is termed as Multipoint Machining methods (MPM) [2]. This multipoint contact is accomplished by using enhanced tool positioning and orientation algorithms [3].
Machining times can be significantly reduced by having such tool positions and orientations that yield wider machining strips while meeting the surface finish constraints. In such methods, the first point of contact is relatively easy to obtain, while the second point of contact is more challenging. Several methods have been suggested by researchers for finding the two points of contact; Liang et al. [4] presented a survey of 5-axis tool positioning methods that included an overview of such techniques.

The first MPM technique of interest is the Drop and Tilt Method (DTM), which is a two-step procedure to find gouge-free, 5-axis tool positions on curved surfaces. Duvedi et al. [5] developed DTM for triangulated curved surfaces and later extended it to Bézier tensor product surfaces [6]. Although DTM is computationally robust, it cannot control the distance between the two contact points for a given location on a curved surface. Sharma et al. [7] took DTM further and developed the Drop Spin and Tilt Method (DSTM), which is a three-step method that involves the merger of DTM and DSM [8]. The DSTM method was developed mainly to regulate the distance between the two contact points for given spin angle to generate the 5-axis tool position. Thus, DSTM addressed the key limitation of DTM and enabled the control over the machining strip width by incorporating an intermediate tool-spin step.

As a proof of concept, the DSTM was tested on bi-quadratic Bézier tensor product surfaces within the Maple™ symbolic algebra environment. However, even with bi-quadratic test surfaces, the solution involved higher order transcendental rational equations. The solution of these equations was reasonably challenging in the symbolic algebra environment. Therefore, dealing with higher order surfaces using symbolic solver tools is expected to be even more computationally expensive and infeasible for typical computing facilities available in industry.

The focus of this work is to develop an algorithm to make DSTM a system independent, stand alone application that can be reliably used to generate gouge-free 5-axis tool positioning of radiused end mill tool for machining of higher order tensor product surfaces. The proposed DSTM numerical algorithm has been implemented in C++, and tested on bi-quadratic, bi-cubic, bi-quartic, and bi-quintic Bézier surfaces. In this paper, we give details on the results for the bi-cubic surfaces to allow for comparison to earlier work. We also perform a set of tests to show the feasibility of using the method to control the machining strip width.

2 Background

Most industrial products like dies and molds used in manufacturing of sheet metal parts, automotive and space applications, have curved surfaces. Such curved shapes are generally modelled by Bézier or B-spline surface patches. In the present work, MPM toolpaths for machining a tensor product Bézier patch have been developed by numerical implementation of DSTM.

For generating the toolpath, the Bézier patch is projected on the $z = 0$ plane, also called the tool drop footprint plane. A zig-zag pattern describing the motion of the tool is developed within the rectangular region. Typically, the zig-zag pattern is discretized using two parameters, namely the feed forward step ($f$-step) and the side step ($s$-step), as shown in Fig. 2. The discretized points are termed as tool drop positions. The toolpath is generated by moving the tool between such consecutive tool drop positions ($T$) and applying the three-step DSTM procedure to determine gouge-free tool positions for each position $T$. The machine controller drives the tool between the sequence of tool positions in a toolpath.

To find a single tool position, the DSTM method uses three steps: tool-drop, tool-spin, and tool-tilt. Note that during machining, the machine does not perform these
steps, but instead just uses the final tool positions calculated by these steps.

In the first step, a radiused end mill cutter is vertically dropped on a surface $S$ with the tool axis $\hat{t}$ passing through a given tool drop position $T$ as shown in Fig. 3a. The dropped tool finds the gouge-free highest point of contact on the surface $S$. This is the first point of contact $P$ as shown in Fig. 3a. The point $P$ lies on a circular pseudo-insert that represents the cross-section of the circular cutting edge of the toroidal tool. The surface normal through $P$ is $\hat{n}_p$.

In this configuration, the tool can be rotated about the axis of the pseudo-insert, or about the surface normal $\hat{n}_p$ at the contact point, without losing tangential contact with the surface. Multi-point methods have typically focused on rotating the tool about the pseudo-insert axis; however, in DSTM, the dropped tool is first rotated about the surface normal $\hat{n}_p$ and then rotated about the pseudo-insert axis to get the second point of contact. This additional rotation about the surface normal $\hat{n}_p$ provides a variety of possible second point of contact lying at a desired distance from the first point of contact.

In the second step of DSTM, the dropped tool is rotated about the surface normal $\hat{n}_p$ by an angle $\alpha$, while remaining tangent to the surface $S$ at point $P$ (Fig. 3b). The tool-drop axis $\hat{t}$ generates a conical surface of revolution when the tool is rotated about the spin-axis $\hat{n}_p$. The spin-angle $\alpha$ determines the gap between the two points of contact, thus eventually controlling the cut strip width. A suitable value of spin-angle is chosen, which in turn is used to find the orientation of tool-axis $\hat{t}$ in the tool-spin step.

Finally in the third step of DSTM, the tool is rotated about the tilt-axis $\hat{i}_3$, which passes through the center of the pseudo-insert and is perpendicular to the plane of pseudo-insert as shown in Fig. 3c. The tool is tilted towards the surface until the tool touches the surface at a second, gouge-free tangency point, $Q$. This gives a gouge-free multipoint tool position so that tool is tangent to the surface $S$ at two distinct locations, $P$ and $Q$. The gap between the two contact points is controlled by varying the spin-angle $\alpha$ in the spin-step.

### 3 Mathematical model of drop spin and tilt method

The surfaces used in this work are tensor product Bézier surfaces. The representation of a Bézier tensor-product surface is given by

$$S(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{n} P_{i,j} B_i^n(u) B_j^m(v),$$

![Fig. 3 Three steps of Drop Spin and Tilt Method (DSTM). a Vertical tool-drop. b Rotation of tool about spin-axis $\hat{n}_p$. c Tool-tilt about pseudo-insert axis $\hat{i}_3$](https://example.com/fig3.png)
with $0 \leq u, v \leq 1$, where $P_{i,j}$ are control points, and $B_i^n(u)$ and $B_j^m(v)$ are the Bernstein basis functions given by

$$B_i^n(u) = \binom{n}{i} u^i (1 - u)^{n-i}.$$ 

We assume the $xy$-coordinates of the control points of the Bézier surface to be uniformly distributed in the rectangular footprint domain. This restriction conveniently represents the Bézier surface $S(u,v)$ in the representation $z = f(x,y)$. This facilitates the evaluation of the $z$-coordinate of the surface $S$, which simplifies the gouge checking procedure.

In this section, we derive the mathematical details of our method; additional implementation details along with pseudo-code are given in Appendix A.

### 3.1 Step 1: Tool-drop

In the first step, a toroidal tool is dropped vertically on the Bézier surface $S(u,v)$, along tool-axis direction $\hat{t} = [0, 0, 1]$ while tool-axis also passes through footprint location $T$ as shown in Fig. 3a. The toroidal tool is centered at $T_c$, with radius of insert $R_i$ and offset radius $R_o$. The location of the center $T_c$ of the dropped tool is given by

$$T_c = T + \hat{t}h$$

where $h$ represents the height of the tool center $T_c$ above footprint point $T$.

The vertically dropping toroidal tool touches the surface $S$ at point $P = S(u_1,v_1)$ as shown in Fig. 3a. From the Drop and Tilt Method [9], we know that the first point of contact $S(u_1,v_1)$ for a toroidal tool dropping along $\hat{t}$ through footprint point $T$ on surface $S$ is

$$S(u_1,v_1) + R_i\hat{n}_p + R_o\hat{n}_r - (T + \hat{t}h) = 0,$$

where $\hat{n}_p$ and $\hat{n}_r$ are the surface normal at point $P$ and a projection of the surface normal perpendicular to the tool axis $\hat{t}$:

$$\hat{n}_p(u_1,v_1) = \frac{\partial S(u_1,v_1)}{\partial u_1} \times \frac{\partial S(u_1,v_1)}{\partial v_1} \cdot, $$

$$\hat{n}_r = \frac{\hat{n}_p - (\hat{n}_p \cdot \hat{t}) \hat{t}}{\| \hat{n}_p - (\hat{n}_p \cdot \hat{t}) \hat{t} \|}.$$ 

Equation 2 can be solved for $(u_1,v_1,h)$ to determine the first point of contact $P$, surface normal $\hat{n}_p$ and $\hat{n}_r$. Alternatively, given $u,v$, we can determine the contact point $P$, surface normal $\hat{n}_p$ and $\hat{n}_r$, an approach that we used in our implementation (Appendix A). Furthermore, the location of the center of the pseudo-insert $O_1$ through the first point $P$ is

$$O_1 = P + R_i\hat{n}_p.$$ 

A coordinate frame $F_0 = \{i_0,j_0,k_0\}$ at $O_1$ is used to define the initial position of the dropped toroidal tool centered at $T_c$. This frame is

$$F_0 = \{i_0,j_0,k_0\} = \left\{ \hat{n}_r \times \hat{t}, \hat{n}_r, \hat{t} \right\}$$

if the first point of contact $P$ is on the outside of the toroidal tool; if $P$ is on the inside of the torus, a similar frame is used but with $\hat{i}_0 = \hat{t} \times \hat{n}_r$.

The dropped toroidal tool, $T_{tor0}$, with its center located at $T_c$, is defined in frame $\{i_0,j_0,k_0\}$ as,

$$T_{tor0} = \begin{bmatrix} (R_o + R_i \cos \theta) \cos \phi \\ R_o + (R_o + R_i \cos \theta) \sin \phi \\ R_i \sin \theta \end{bmatrix}.$$ 

where $\theta$ and $\phi$ are the angles used to model the toroidal tool in the frame $\{i_0,j_0,k_0\}$. The orientation of the direction vectors for frame $\{i_0,j_0,k_0\}$ is shown in Fig. 3a.

The drop method has been implemented and tested both in a symbolic algebra environment and in C++ [9].

### 3.2 Step 2: Tool-spin

In the second step, an orthogonal coordinate frame $F_1 = \{i_1,j_1,k_1\}$, called the tool-spin frame, is defined at $O_1$ as shown in Fig. 3b. The tool-spin frame is created by rotating the tool-drop frame $F_0$ about $\hat{i}_0$ so that $\hat{k}_1$ is aligned with the surface normal $\hat{n}_p$, while the dropped tool stays fixed in the frame $\{i_0,j_0,k_0\}$.

The direction vectors of the tool-spin frame $F_1$ are

$$F_1 = \{i_1,j_1,k_1\} = \left\{ \hat{i}_0, \frac{\hat{n}_r \times \hat{i}_0}{\| \hat{n}_r \times \hat{i}_0 \|}, \hat{n}_p \right\}.$$ 

The representation of the dropped toroidal tool with respect to frame $F_1$ can be obtained by using a rotation $R_m^n$,

$$T_{tor1} = R_m^n \cdot T_{tor0} = R_m^n \cdot T_{tor0}$$

where $R_m^n$ is a rotation matrix that maps a point from frame-$n$ to frame-$m$:

$$R_m^n = \begin{bmatrix} i_m \cdot \hat{i}_n \\ j_m \cdot \hat{j}_n \\ k_m \cdot \hat{k}_n \end{bmatrix}.$$ 

All subsequent rotations ensure that the tool definition frame $F_1$ is mobile and it rotates along with the tool in the tool-spin and tool-tilt steps.

In the tool-spin step, the dropped toroidal tool $T_{tor1}$, defined in frame $F_1$, is rotated about the spin-axis $\hat{k}_1$ passing through $O_1$ by spin angle $\alpha$. The desired tool-axis orientation $\hat{t}$ is obtained by using a specified spin-angle $\alpha$, which eventually determines the spacing between the points of tangency between the work surface and the toroidal tool in the third step.
The post-spin orientation of the toroidal tool is defined in frame $\mathcal{F}_2 = \{i_2, j_2, k_2\}$, which is initially assumed to be coincident with $\mathcal{F}_1$ and rotates along with the tool in the tool-spin step as shown in Fig. 3b. Frame $\mathcal{F}_2$ is the tool-tilt frame, and its direction vectors are

$$\mathcal{F}_2 = \{i_2, j_2, k_2\} = \{\hat{i}_1 \cos \alpha + \hat{j}_1 \sin \alpha, \ -\hat{i}_1 \sin \alpha + \hat{j}_1 \cos \alpha, \ \hat{k}_1\}.$$  

### 3.3 Step 3: Tool-tilt

In the third step, the tool in the tool-tilt frame $\mathcal{F}_2$ is further rotated by an angle $\beta$ about the tilt-axis $\hat{i}_3$ until it is tangent to the surface $S$ at a second point of contact $Q$ as shown in Fig. 3c. The resulting tool orientation is defined using another orthogonal coordinate frame $\mathcal{F}_3 = \{i_3, j_3, k_3\}$ with its origin at $O_1$. The frame $\mathcal{F}_3$ is initially coincident with frame $\mathcal{F}_2$, and rotates along with the tool through angle $\beta$ in the tool-tilt step. The direction vectors of the $\mathcal{F}_3$ are given as

$$\mathcal{F}_3 = \{i_3, j_3, k_3\} = \{\hat{i}_2, \ \hat{j}_2 \cos \beta + \hat{k}_2 \sin \beta, \ -\hat{j}_2 \sin \beta + \hat{k}_2 \cos \beta\}.$$  

In the tool-spin and tool-tilt steps, the toroidal tool continues to maintain tangential contact with the surface $S$ at $P$. After the spin and tilt rotations, the tool has a second point of contact with the surface $S$ at point $Q$ as shown in Fig. 3c.

The position and orientation of the rotated toroidal tool with respect to the standard Cartesian frame $\{x, y, z\}$ given as $T'_{tor}$ are

$$T'_{tor} = O_1 + R^3_{0} T_{tor1} = O_1 + R^3_{0} \cdot R^1_{0} \cdot T_{tor0}.$$  

Similarly, the orientation of the rotated tool-axis $\hat{i'}$ and the rotated major circle of torus $C'_{tor}$ (centered at $T'_{c}$) in frame $\{x, y, z\}$ is given as

$$\hat{i'} = R^3_{0} \cdot R^0_{1} \cdot \hat{i},$$

$$C'_{tor} = O_1 + R^3_{0} \cdot R^0_{1} \cdot C_{tor0},$$

$$T'_{c} = O_1 + R^3_{0} \cdot R^0_{1} \cdot T_{c},$$

where $\hat{i} = [0, 0, 1]^T$ and

$$C_{tor0} = [R_{0} \cos \phi, R_{0} \sin \phi + R_{p}, 0]^T.$$  

When the tilted toroidal tool touches the surface $S$ at $Q = S(u_2, v_2)$, the pseudo-insert through $Q$ has its center on the major circle of the rotated torus $C'_{tor}$. Thus,

$$S(u_2, v_2) + R_1 \cdot \hat{n}_q - C'_{tor} = 0,$$  

where $\hat{n}_q$ is the surface normal at $S(u_2, v_2)$ and is defined similar to $\hat{n}_p$.

For a given spin-angle $\alpha$, Eq. 6 is a simultaneous rational equation with three component equations in $x$-, $y$-, and $z$-coordinates but has four unknowns $u_2, v_2, \beta$, and $\phi$. An additional constraint that ensures that the shortest distance between the surface normal $\hat{n}_q$ through the second point of contact $S(u_2, v_2)$ and the rotated tool axis $\hat{i'}$ is zero is used to solve for the four unknowns:

$$\{T'_{c} - S(u_2, v_2)\} \cdot \{\hat{i'} \times \hat{n}_q\} = 0.$$  

We solved Eqs. 6 and 7 for $(u_2, v_2, \beta, \phi)$, and in turn the values of these four parameters are used to compute...
the point $Q$, surface normal $\hat{n}_q$, and direction vectors $\{\hat{i}_3, \hat{j}_3, \hat{k}_3\}$. Frames $F_1, F_2,$ and $F_3,$ and the transformation matrix $R_0^3$ are used to determine the rotated toroidal tool $T_{tor}^1$, tool axis $\hat{i}$, tool center $T_{tor}^c$, and second point of contact $Q$. The resulting tool positions from the numerical solver may gouge the surface. Thus, a gouge checking method [9] was implemented to ensure that the accepted tool positions do not gouge the surface beyond user-specified tolerance, while some gouging may be acceptable for actual machining, requiring no gouging seemed appropriate for our tests.

Pseudo-code for our implementation for this method are given in Appendix A.

### 4 Numerical implementation

The numerical DSTM algorithm computes the toolpath for multipoint tool positions for a tensor product, Bézier surface $S$ (degree $N \times M$) with uniformly spaced control points as shown in Fig. 4. Pseudo-code for our algorithm appears in Fig. 5. The toolpath is generated within the limits of $X_{\text{max}}$ and $Y_{\text{max}}$ along the $X$- and $Y$-directions, which also define the span of the control points of the uniformly spaced Bézier surface $S$. A zig-zag tool-drop trajectory is defined at $z = 0$ within the $X_{\text{max}}$ and $Y_{\text{max}}$ span, and is discretized into $X_{\text{Steps}}$ and $Y_{\text{Steps}}$ along the $X$- and $Y$-directions by using a feed forward step of $f_{\text{step}}$ and a side step of $s_{\text{step}}$.

A toroidal tool, defined by a circular insert of radius $R$, placed at offset radius $R_0$, is dropped vertically on such discretized locations (marked as T in Fig. 3a) to find a non-gouging point of tangency with the surface. Subsequently, the dropped tool is then rotated about the surface normal $\hat{n}_p$ through the user defined spin-angle, $\alpha$. Then finally the tool is tilted about the axis of the pseudo-insert passing through the first point of contact to find a second non-gouging point of contact with the surface.

Newton’s method is used in the tool-drop and tool-tilt steps. To make the numerical solution independent of the degree of the Bézier surface, the partial differentials of the surface and the surface normals are computed using the Richardson extrapolation. In the tool-drop step, Eq. 2 is solved with Newton’s method to compute the first point of contact $S(u_1, v_1)$, and the height of the tool center for the dropped tool. A random initial seed is provided to the Newton’s method to start the solution search. If Newton’s method does not converge within a maximum stipulated number of trials, then another random initial seed is tried. If the tool-drop converges, then the solution is further tested for the gouge-free tangential contact between the tool and the work surface. If the converged solution yields a gouging tool position, then Newton iterations are aborted and another random seed point is reinitialized.

The dropped tool is then rotated about the surface normal $\hat{n}_p$ by an spin angle $\alpha$ and is subsequently tilted about the $\hat{i}_3$ axis through an angle $\beta$ so that the tool has a gouge-free tangency at a second point of contact $Q = S(u_2, v_2)$. Newton’s method is used to compute the tilt solution and the search starts with a random initial seed. In case the tool-tilt solution for the previous/adjacent foot print location is available, then the previous solution is used as the initial seed; if Newton’s method does not converge with this initial seed, then in the next trial, a random initial seed is used. If Newton’s method converges, then the computed tool-tilt solution is tested for its proximity with first point of contact, tool-axis inversion, and a gouge check. These three checks ensure that the first and second points of contact are separated by a user specified gap $\epsilon_{\text{vicinity}}$, the tilt angle $\beta$ is negative, and the tilted tool is not gouging the surface beyond the permitted gouge tilt tolerance ($\epsilon_{\text{Gouge}}$). If the tool-tilt solution fails on any of these three checks, Newton’s method is reset with a new random initial seed. In case no valid tool-tilt solution is found within a maximum number of initial seed solutions ($N_{\text{tilt}}$), then it is construed that for the given spin angle $\alpha$, the tool tilt is infeasible and the first and second points of contact are taken to be identical. Details of the numerical implementation are available in the Appendix A.

### 5 Test surfaces

We implemented the numerical solver for DSTM algorithm in C++ and tested it on bi-quadratic, bi-cubic, bi-quartic, and bi-quintic Bézier surface patches. Here, we give details on three different bi-cubic test surfaces, concave, convex, and...
The tool-drop footprint surface was obtained by projecting the surface patch on the \( z = 0 \) plane. The zig-zag toolpath was generated on the tool-drop footprint plane. This toolpath is composed of 16 machining passes with a side step of 10 mm. In each machining pass, a total of 61 tool positions were computed with a feed forward step of 2.5 mm between two consecutive tool-drop positions. Therefore, in this DSTM study, the toolpath for each test surface was composed of 976 discretized tool positions.

The toroidal tool with an offset radius \( R_o=6.7 \) mm and insert radius, \( R_i=6.0 \) mm was dropped vertically on each discretized location as discussed in Section A.2.

Newton's method was used to obtain the tool-drop and tool-tilt solutions. The partial derivatives required in the computation of the Jacobian matrix were computed using the Richardson extrapolation, with an initial step size \( h_{step} \) of 0.001 mm and a convergence tolerance \( \epsilon_{Richard} \) of \( 10^{-5} \) mm. The seed points required to initialize Newton’s method were randomly generated.

In the tool-drop step, tolerances of \( \epsilon_{Drop} = 10^{-6} \) mm was used as convergence criteria in Newton’s method. Each solution for first point of contact from tool-drop step was examined using gouging check algorithm with a gouge tolerance of \( \epsilon_{Gouge} = 10^{-3} \) mm. The maximum number of random seed points \( N_{Tilt} \) and Newton iterations \( N_{Iter} \) in the tool-drop step was kept as 500 and 150 respectively.

Furthermore, the dropped tool is rotated by spin-angle \( (\alpha) \) about surface normal \( \vec{n}_p \) passing through first point of contact at \( P \). In this work, the toolpaths were generated with values of spin-angle varying from 0 to 180° with uniform increments of 15°.

Finally in the tool-tilt step, the number of random trial seed points \( N_{Tilt} \) and number of Newton iterations \( N_{Iter} \) was taken as 50000 and 150 respectively. A relatively higher value of \( N_{Tilt} \) was employed as a preventive measure so that a feasible tilt-solution is computed at each drop location. For the tool-tilt step, tolerances of \( \epsilon_{Tilt} = 10^{-4} \) mm and \( \epsilon_{Gouge} = 10^{-3} \) mm were used as convergence criteria in Newton’s method and for gouge check respectively. The tolerances used for convergence in tool-drop and tool-tilt steps are chosen to ensure that the resulting 5-axis tool positions are accurate in the range of \( 10^{-3} \) mm.

DSTM algorithm statistics are given in Table 4. This table shows the maximum and the average number of the random seed points, along with maximum and the average number of iterations taken for Newton’s method in the tool-drop and tool-tilt steps for all three test surfaces corresponding to various values of spin-angle.

The maximum number of randomly generated initial seeds taken for the tool-drop step across all surfaces, at different values of spin-angle \( \alpha \) is less than 100. Moreover, the average number of initial seeds is also observed to be less than 5 when averaged over 976 tool positions for each spin-angle. These results indicate that in a tool path (comprising 976 tool-drop locations) except for few hard tool-drop locations, the majority of the locations need only a few initial seeds to converge to the acceptable solution. Similarly, the maximum number of Newton’s method iterations in the tool-drop step is found to be less than 20, except for the saddle surface. In the tool-drop step, the maximum value of \( N_{Iter} \) for the saddle surface is relatively higher when compared to the other test surfaces. This may be attributed to the inflection regions on the saddle surface where the curvature transitions exist. However, the average number of Newton iterations for the saddle surface is marginally higher as compared to the other two surfaces.

The tool-tilt step of the DSTM algorithm is computationally more expensive than the tool-drop step. This extra expense is indicated by the relatively higher values of the

### Table 1 Control points for 150mm×150mm Bi-cubic concave Bézier surface patch

| \( P_{0,0} \) | \( P_{0,1} \) | \( P_{0,2} \) | \( P_{0,3} \) |
|---|---|---|---|
| (0.0, 0.0, 90.0) | (0.0, 50.0, 80.0) | (0.0, 100.0, 80.0) | (0.0, 150.0, 90.0) |
| \( P_{1,0} \) | \( P_{1,1} \) | \( P_{1,2} \) | \( P_{1,3} \) |
| (50.0, 0.0, 80.0) | (50.0, 50.0, 70.0) | (50.0, 100.0, 70.0) | (50.0, 150.0, 80.0) |
| \( P_{2,0} \) | \( P_{2,1} \) | \( P_{2,2} \) | \( P_{2,3} \) |
| (100.0, 0.0, 80.0) | (100.0, 50.0, 70.0) | (100.0, 100.0, 70.0) | (100.0, 150.0, 80.0) |
| \( P_{3,0} \) | \( P_{3,1} \) | \( P_{3,2} \) | \( P_{3,3} \) |
| (150.0, 0.0, 90.0) | (150.0, 50.0, 80.0) | (150.0, 100.0, 80.0) | (150.0, 150.0, 90.0) |

### Table 2 Control points for 150mm×150mm Bi-cubic convex Bézier surface patch

| \( P_{0,0} \) | \( P_{0,1} \) | \( P_{0,2} \) | \( P_{0,3} \) |
|---|---|---|---|
| (0.0, 0.0, 90.0) | (0.0, 50.0, 100.0) | (0.0, 100.0, 100.0) | (0.0, 150.0, 90.0) |
| \( P_{1,0} \) | \( P_{1,1} \) | \( P_{1,2} \) | \( P_{1,3} \) |
| (50.0, 0.0, 100.0) | (50.0, 50.0, 110.0) | (50.0, 100.0, 110.0) | (50.0, 150.0, 100.0) |
| \( P_{2,0} \) | \( P_{2,1} \) | \( P_{2,2} \) | \( P_{2,3} \) |
| (100.0, 0.0, 100.0) | (100.0, 50.0, 110.0) | (100.0, 100.0, 110.0) | (100.0, 150.0, 100.0) |
| \( P_{3,0} \) | \( P_{3,1} \) | \( P_{3,2} \) | \( P_{3,3} \) |
| (150.0, 0.0, 90.0) | (150.0, 50.0, 100.0) | (150.0, 100.0, 100.0) | (150.0, 150.0, 90.0) |
maximum number of initial seeds used in the Newton solver in the tool-tilt step. Specifically, in the tool-tilt step for the saddle surface, at spin-angles 15°, 30°, and 150°, the maximum number of initial seeds used was exceptionally higher (greater than 1000). This need for more initial seeds is attributed to the conditional check incorporated in the tool-tilt step to determine the second point of contact \( Q \) away from the first point of contact \( P \). This spacing between \( P \) and \( Q \) is ensured by specifying a minimum parametric spacing between the two contact points \( P \) and \( Q \) using the condition that \( \{ \| u_2 - u_1 \| \} \& \{ \| u_2 - v_1 \| \} \geq \varepsilon_{\text{vicinity}} \), where \( \varepsilon_{\text{vicinity}} = 0.001 \). However, when the curvature of the tilting tool closely matches the local surface curvature, there is hardly any scope for the tool to tilt because as the tool is tilted, the tool curvature decreases, and to have a gouge free tool position, the tool curvature can not be smaller than the local maximum surface curvature. In such situations, the second point of contact tends to lie close to the first point of contact and thus it is hard to have a large gap between the two points of contact. In such cases, when the second contact point is not achieved within first 1000 initial seeds, the tolerance for gap between two points of contact, given by \( \varepsilon_{\text{vicinity}} \), is relaxed to 0.0001. In all cases, as the vicinity check condition is relaxed, the tool-tilt solutions converge within fewer initial seeds.

For each test surface, the 5-axis multipoint tool position and orientation data at each tool-drop location along the all zig-zag machining passes was generated using the DSTM.

The five-axis tool positioning data computed from numerical DSTM solver was verified by plotting the entire toolpath for each of the test surfaces in a custom developed 3D toolpath simulator in MATLAB. The vector format five-axis tool positioning data for the toroidal tool (at each toolpath location) is defined by the location of the first point of contact \( S(u_1, v_1) \), surface normal at first point of contact \( \hat{n}_p \) and the tool axis orientation \( \hat{f} \). Figures 6, 7, and 8 show the positioning of the toroidal tool on the test surfaces as viewed from top and bottom of the surfaces. The simulation results showed that the toroidal tool touches the surface at two points for each tool position. As discussed above, the toolpath for each surface comprises a total of 976 tool positions, but in Figs. 6b, 7b, and 8b, only every eighth tool position along each of the 16 machining passes is presented for the spin-angle value of 15°. In the 3D plots, the bottom surface of the toroidal tool can be seen touching the three test surfaces tangentially which is visible as the circular shadow of the tool as seen in the bottom view. These circular impressions disappear as we zoom in, which confirms the gouge-free tool positioning. It is clear from the bottom views for the three test surfaces that the tool does not contact the toolpath simulator in MATLAB. The vector format five-axis tool positioning data computed from numerical DSTM solver was verified by plotting the entire toolpath for each of the test surfaces in a custom developed 3D toolpath simulator in MATLAB.
not protrude out of the surface in any case. Furthermore, the circular impression of the tool in the bottom views validates that the toroidal tool has a gouge-free tangential contact with the surfaces. The top views in Figs. 6a, 7a, and 8a demonstrate that the transition of tool-axis orientation along each of the machining passes is also smooth which is confirmed by the smooth orientation change of the tori along each machining pass.

6 Discussions

DSTM is a multipoint five-axis tool positioning method which comprised three distinct steps resulting in multiple gouge-free tool positions and orientations. Each DSTM solution has at least two gouge-free points of contact between the toroidal tool and the curved surface wherein the gap between the contact points can be altered by choosing a suitable value of the spin-angle. Three steps in the DSTM were mathematically established and solved using Maple™ [7].

The tool-drop and tool-tilt steps involve solving sets of three and four higher order rational equations respectively. In the present work, Newton’s method was employed to numerically solve the simultaneous non-linear equations. A generalized method for numerically computing various partial derivative terms required in Newton’s method was developed using the Richardson extrapolation technique.

The initial seeds for Newton’s method were generated using a random number generator. It was observed that it is relatively easier to solve the tool-drop algorithm as it takes relatively fewer initial seeds as well as lesser number of iterations in Newton method solver as compared to the tool-tilt step. This difference is attributed to the tool-drop step that require solution to only three simultaneous equations as compared to tool-tilt step that need solution to four simultaneous equations. To accelerate the computation of the second point of contact, the initial seed for the next tool
position was generated by using an offset from the previous tool position. The tilt solution easily converged using such an initial seed in those cases where the surface curvature does not change significantly between two adjacent tool positions. In the event of not succeeding with such initial seeds derived from previous tool position, the random initial seeds were then used to find a solution using the numerical solver.

For a given first point of contact $P$, the location of the second point of contact $Q$ in DSTM depends on the specified spin-angle. Rotating the toroidal tool by spin-angle $\alpha$ about the surface normal $\hat{n}_p$ results in a new orientation of the pseudo-insert axis given by $\hat{i}_3$. This results in the possibility of having a number of different gouge-free solutions for the second point of contact when the tool is tilted about the new pseudo-insert axis $\hat{i}_3$ in the tool-tilt step. As an example, for tool-drop location $T = [50, 30, 0]$ on the concave test surface, the first point of contact ($P$) is shown as a large red color sphere in Fig. 9, while a trail of second points of contact corresponding to spin-angle ($\alpha$) varying from 0 to 360° through an increment of 15° is shown by small yellow color polyhedrons on either side of the first point of contact. The spread in the polyhedrons in Fig. 9, representing the second point of contact, is influenced by the local surface curvature and effective curvature of tilted tool at first point of contact. Such a distribution is possible in relatively open surfaces. Therefore, rotation of the dropped tool through the spin-angle in the DSTM opens up the possibility of having larger side-steps of uniform cut-width between machining passes, thereby increasing the machining efficiency through reduced machining time. A gouge check algorithm [9] is used to confirm the gouge-free tangency at both points of contact.

For the tests of the DSTM algorithm for the three bi-cubic Bézier test surfaces, the toolpaths corresponding to different values of spin-angle ($\alpha$) varying from 0 to 180° in increments of 15° were generated, where each toolpath comprised 976 tool positions. Figure 10 shows a subset of the tool positions (every eighth tool position) along machining passes at $T_s = 40$ mm, 80 mm, and 120 mm for all three test surfaces. At each tool position, the first point of contact is highlighted by a sphere and the multiple solutions for the second point of contact corresponding to varying values of spin-angle are represented by an arch of polyhedra.

The shape of the distribution pattern of the second points of contact varies along the machining pass and is also dependent on the type of surface. In particular, the shape is dependent on the local surface properties. For example, in case of a concave test surface, in Fig. 10a, the arch is wide and open at the outer periphery of the surface. In this case, as the tool approaches the mid-span of the surface, the arch tends to align along the tool feed direction. This
change in orientation is due to the fact that near the middle of the concave patch, the surface is relatively lower, and the tool gets the second point of contact at higher locations. Moreover, the direction of principal curvature tends to align with the feed direction as the tool proceeds towards the middle of the concave patch starting from the boundary edge. In such cases, a suitable spin-angle may help in attaining a little transverse width between the two points of contact.

Similarly, in case of the convex patch, shown in Fig. 10b, the machining pass corresponding to $T_y=80$ mm, the trajectory of the second points of contact (at varying spin-angles) is aligned along the feed direction throughout the machining pass and the trajectory of the second points of contact (around the first point of contact) has little transverse spread. This lack of transverse spread is because the direction of maximum principal curvature of the surface is aligned along the feed direction in this machining pass. This phenomenon is attributed to the fact that the convex surface patch is relatively higher in the middle than the adjacent regions. Therefore, at a given spin-angle, the tilted tool finds the gouge-free second point of contact along the feed direction only.

In case of Saddle patch, shown in Fig. 10c, the spread pattern changes its orientation as the tool progresses along a machining pass. This change in orientation is driven by changes in the local surface curvature along the machining pass.

It is observed in this study that the DSTM method is capable of achieving two gouge-free points of contact where the tool is positioned tangential to the surface. Moreover, the gap between the two tangential point of contact on the surface can be controlled by choosing a suitable value of spin-angle. Thus, using the DSTM method, a consistent cut-width can be achieved for five-axis machining of a given parametric Bézier tensor product surface using a toroidal tool.

### 7 Conclusions

The numerical implementation of the Drop Spin and Tilt method of multipoint five-axis tool positioning for tensor product Bézier surfaces is presented in this work. The method is successfully tested on a variety of bi-quadratic, bi-cubic, bi-quartic, and bi-quintic Bézier surfaces with uniform control points. This paper highlights the unique capability of the DSTM method to generate two gouge-free points of contact between surface and toroidal tool, while the gap between two contact points can be controlled using the spin-angle.

The developed method is found to be robust as well as reliable for computing multipoint five-axis toolpath data.
which is demonstrated using three bi-cubic Bézier test surfaces. On the three bicubic surfaces we tested, the average number of initial seeds for our numerical method was less than 5 across all three surfaces, and the average number of Newton iterations was less than 20 except when the curvature of the tool closely matches the local surface curvature (in which case more iterations were required).

The DSTM method can be extended to other parametric surfaces with non-uniform control points. This work also opens up the avenues for future work to create efficient algorithms to develop the uniform cut width toolpaths for machining of open surfaces while simultaneously ensuring the smooth transition of tool from one tool position to the next.

Appendix A: Numerical implementation of DSTM

The DSTM method comprises three steps: tool-drop, tool-spin, and tool-tilt. In the present work, Newton’s method was used to determine the solutions for the tool-drop and tool-tilt steps. In the Newton’s algorithm, first- and second-order partial differentials of the surface and surface normal need at each tool-drop and tool-tilt steps. To have a numerical solution that is independent of the degree of the Bézier surface, we use the Richardson extrapolation to compute these partial differentials. The following sections discuss the implementation of the Richardson extrapolation method and Newton’s method to compute the gouge free tool-drop and tool-spin solution at each tool drop location.

A.1 Richardson extrapolation for numerical differentiation

The goal of numerical differentiation is to compute an accurate estimate of the partial derivative of a function. This accuracy can be improved by decreasing the interval step size \( h \) or by using a higher order formula that employs more points. The Richardson extrapolation employs a center difference approach and offers better convergence as the truncation error drops by a factor of 4 as the step interval drops from \( h \) to \( \frac{h}{2} \). The test surfaces \( S \) used in DSTM algorithm are function of parameters \( u \) and \( v \). Figure 11 shows the procedure developed to numerically compute the partial derivative of surface \( S \) with respect to parameter \( u \). Similar, procedure is used to determine the partial derivative of surface \( S \) with respect to parameter \( v \). The initial value of the interval size \( h_{\text{step}} \) is set to 0.001 and \( h_{\text{step}} \) is reduced by half in each subsequent iteration. The iterations continues till the progressive truncation error is less than the specified error, given by \( \epsilon_{\text{Richard}} = 10^{-5} \). The numerical values of the computed derivatives were compared with the exact values computed symbolically in MAPLE\textsuperscript{TM} and both were found to be correct to the sixth decimal place.

A.2 Tool-drop algorithm

As a first step in DSTM implementation, the toroidal tool is dropped vertically through tool footprint location \( T \) (Fig. 12 gives pseudo-code for this algorithm). Equation 2 is solved by employing Newton’s method to compute the location of first point of contact \( S(u_1, v_1) \), and the height of the dropped tool \( h \). Newton’s method needs an initial seed to start the solution search. In the first step, a random initial solution, \( [X_{\text{Drop}}] = [u, v, h] \), is generated and the partial derivatives required in the Newton’s method are computed using the Richardson extrapolation method. Depending upon the local surface properties and the tool-drop location \( T \), first point of contact \( P \) may lie on the outer-side or the inner-side of the cutting edge of the pseudo-insert, i.e., \( O_1 \) (see Fig. 1).

If the first point of contact \( P \) lies on the outer-side of the pseudo-insert, Eq. 3 is used to compute \( \vec{n}_r \). However, when the point of contact \( P \) lies on the inner side of the pseudo-insert, as shown in Fig. 13, the surface normal \( \vec{n}_p \) is oriented in a direction away from the tool-axis \( \hat{i} \) and Eq. 3 can not be used to compute the radial vector \( \vec{n}_r \). In such scenarios, \( \vec{n}_r \) is computed alternatively by using the following equation,

\[
\vec{n}_r = \frac{T \vec{P} - (T \vec{P} \cdot \hat{i}) \hat{i}}{\|T \vec{P} - (T \vec{P} \cdot \hat{i}) \hat{i}\|}.
\]  

(8)

Terms of the Jacobian matrix\( (J_D) \), required in the Newton’s method, comprises the partial derivatives of Eq. 2 with
Fig. 12 Pseudo-code for numerical implementation of tool-drop step

1. Initialize Flag\_Newton = 0, Flag\_gouge = 0, i = 0,
2. While (i < N\_Iter)
   Randomize [X] = [u, v, h]
   For j = 0, j < N\_Iter, j +
   Compute P\_i = \partial S\_i / \partial u, \partial S\_i / \partial v, \partial S\_i / \partial h, \partial h\_i, \partial h\_i / \partial u, \partial h\_i / \partial v
   If (Point Of Contact is outside of O\_1) Then
     \( \hat{n}_r = \frac{\hat{n}_p - (\hat{n}_p \cdot \hat{t}) \hat{t}}{||\hat{n}_p - (\hat{n}_p \cdot \hat{t}) \hat{t}||} \)
   Else
     \( \hat{n}_r = \frac{\hat{t} P - (\hat{t} P \cdot \hat{t}) \hat{t}}{||\hat{t} P - (\hat{t} P \cdot \hat{t}) \hat{t}||} \)
   Compute \( \frac{\partial n_p}{\partial u}, \frac{\partial n_p}{\partial v} \)
   Compute \( J_D = \begin{bmatrix} \frac{\partial E_2}{\partial u} & \frac{\partial E_2}{\partial v} & \frac{\partial E_2}{\partial h} \\ \frac{\partial E_2}{\partial u} & \frac{\partial E_2}{\partial v} & \frac{\partial E_2}{\partial h} \\ \frac{\partial E_2}{\partial u} & \frac{\partial E_2}{\partial v} & \frac{\partial E_2}{\partial h} \end{bmatrix} \)
   Compute \( F[X] = S(u, v) + R_i \hat{n}_p + R_o \hat{n}_r - (T + h \hat{t}) \)
   If (Flag\_Newton = 1) Then
     Compute \( T_c = S(u, v) + R_i \hat{n}_p + R_o \hat{n}_r \)
     \( \text{Flag} \_\text{gouge} = \text{Gouge\_Check}(T_c, i, \hat{n}_p) \)
     If (Flag\_gouge = 0) Then
       PRINT: Valid Drop Solution Found
       Return [X, \hat{n}_r, \hat{n}_p, T_c]
     Else
       Flag\_Newton = 0, Flag\_gouge=0
       PRINT: Tool is Gouging
       Break For Loop
   Else Next j
   If (j = N\_Iter) Then
     PRINT: Non Converging Solution. Try new seed
     Next i
   3. If (Flag\_Newton = 0 & Flag\_gouge = 0) Then
     PRINT: Drop Solution Not feasible

Fig. 13 Computing \( \hat{n} \) when the first point of contact \( P \) lies towards inner-side of the circular cutting edge of toroidal tool

The partial derivatives of surface \( S \), surface normal \( \hat{n}_p \), and radial vector \( \hat{n}_r \) are numerically computed by employing the Richardson extrapolation method as shown in Fig. 11.

Newton’s method starts with a randomly generated initial seed for \{u, v, h\} and attempts to converge to the solution

\[
\begin{align*}
\frac{\partial E_2}{\partial u} &= \frac{\partial S(u, v)}{\partial u} + R_i \frac{\partial \hat{n}_p(u, v)}{\partial u} + R_o \frac{\partial \hat{n}_r(u, v)}{\partial u} \\
\frac{\partial E_2}{\partial v} &= \frac{\partial S(u, v)}{\partial v} + R_i \frac{\partial \hat{n}_p(u, v)}{\partial v} + R_o \frac{\partial \hat{n}_r(u, v)}{\partial v} \\
\frac{\partial E_2}{\partial h} &= -\hat{t}
\end{align*}
\]

Newton(X, F[X], J, \epsilon)

1. Compute \( J^{-1} \)
   If \( F[X] < \epsilon \) Then
     Return[TRUE, X]
   Else
     \( X_{n+1} = X - J^{-1} F[X] \)
     \( X = X_{n+1} \)
     Return[FALSE, X]

Fig. 14 Pseudo-code for the implementation of the Newton method within \( N\_Iter \) iterations. If Newton’s method (Fig. 14) converges within the specified tolerance of \( \epsilon\_Drop \), then the solution is tested for gouging. If the solution yields a non-gouging tool position, this is accepted as the final drop solution. If the tool is gouging with the surface or Newton’s method does not converge within the specified number of iterations \( N\_Iter \), then the solution is aborted and Newton’s method is re-initialized with another random seed. In the gouge check step, the tool bottom is discretized into 48 \( (N_\phi) \) and 36 \( (N_\theta) \) divisions along \( \phi \) and \( \theta \) directions respectively as shown in Fig. 17.

Thus, the tool-drop algorithm yields the first non-gouging point of contact \( P = S(u_1, v_1) \) between the tool and the surface, the unit surface normal \( \hat{n}_p \) through \( P \), the unit radial vector \( \hat{n}_r \), center of the Pseudo-Insert \( O_1 \), and the center of the dropped toroidal tool \( T_c \).
A.3 Tool-spin and tilt algorithm

In the tool-spin and tool-tilt steps, the dropped tool is first rotated about the spin-axis (\( \hat{n}_s \)) through the specified spin-angle (\( \alpha \)). Subsequently, the tool is rotated about the tilt-axis \( \hat{i}_3 \) through an angle \( \beta \) so that the tool touches the surface at second point of contact, \( Q = S(u_2, v_2) \). In the tool-tilt algorithm, Newton’s method is employed to solve Eqs. 6 and 7 for four unknowns \( u_2, v_2, \beta, \) and \( \phi \). The procedure to determine the coordinate frames \( \{ \hat{i}_0, \hat{j}_0, \hat{k}_0 \}, \{ \hat{i}_1, \hat{j}_1, \hat{k}_1 \}, \{ \hat{i}_2, \hat{j}_2, \hat{k}_2 \} \) and \( \{ \hat{i}_3, \hat{j}_3, \hat{k}_3 \} \) is explained in Sections 3.1, 3.2, and 3.3 respectively.

If the tool-tilt solution of the previous footprint location is available, then the previous solution is taken as the initial seed for Newton’s method. Since the control points of the Bézier surface are uniformly distributed, this approximation works at some locations and the tool-tilt solution from Newton method converges quickly. Otherwise, the initial seed \( [X_{Tilt}] \) for Newton method is determined using random seed generation. Next step in tool-tilt algorithm is to compute the values of the surface normal \( \hat{n}_q \), tilted tool-axis \( \hat{t}' \), tool-center \( T'_c \), and circle of torus \( C_{tor} \) from the initial seed.

The Jacobian matrix \( J_T \) used in the tool-tilt step is given as

\[
J_T = \begin{bmatrix}
\frac{\partial E_{qX}}{\partial u} & \frac{\partial E_{qX}}{\partial u} & \frac{\partial E_{qX}}{\partial \beta} & \frac{\partial E_{qX}}{\partial \phi} \\
\frac{\partial E_{qY}}{\partial u} & \frac{\partial E_{qY}}{\partial u} & \frac{\partial E_{qY}}{\partial \beta} & \frac{\partial E_{qY}}{\partial \phi} \\
\frac{\partial E_{qZ}}{\partial u} & \frac{\partial E_{qZ}}{\partial u} & \frac{\partial E_{qZ}}{\partial \beta} & \frac{\partial E_{qZ}}{\partial \phi} \\
\frac{\partial E_{qT}}{\partial u} & \frac{\partial E_{qT}}{\partial u} & \frac{\partial E_{qT}}{\partial \beta} & \frac{\partial E_{qT}}{\partial \phi}
\end{bmatrix}
\]

The elements of Jacobian matrix \( J_T \) are computed using the partial derivatives of Eqs. 6 and 7 with respect to the initial seed \( [X] = (u, v, \beta, \phi) \).

\[
\frac{\partial E_{q_6}}{\partial u} = \frac{\partial S(u, v)}{\partial u} + R_i \frac{\partial \hat{n}_q(u, v)}{\partial u}
\]

\[
\frac{\partial E_{q_6}}{\partial v} = \frac{\partial S(u, v)}{\partial v} + R_i \frac{\partial \hat{n}_q(u, v)}{\partial v}
\]

\[
\frac{\partial E_{q_6}}{\partial \beta} = R_o (1 + \sin \phi) (\hat{k}_3 \cos \gamma - \hat{j}_3 \sin \gamma)
\]

\[
\frac{\partial E_{q_7}}{\partial u} = M(\beta) \times \frac{\partial \hat{n}_q(u, v)}{\partial u} \cdot [N(\beta) - S(u, v)]
\]

\[
\frac{\partial E_{q_7}}{\partial v} = M(\beta) \times \frac{\partial \hat{n}_q(u, v)}{\partial v} \cdot [N(\beta) - S(u, v)]
\]

\[
\frac{\partial E_{q_7}}{\partial \beta} = [M(\beta) \times \frac{\partial \hat{n}_q(u, v)}{\partial u} \cdot [N(\beta) - \frac{\partial S(u, v)}{\partial u}]
\]

\[
\frac{\partial E_{q_7}}{\partial \phi} = M(\beta) \times \frac{\partial \hat{n}_q(u, v)}{\partial \phi} \cdot [N(\beta) - \frac{\partial S(u, v)}{\partial \phi}]
\]

where, \( \gamma = \text{arccos}(\text{dot}(\hat{j}_0, \hat{j}_1)) \),

\[
M(\beta) = \hat{k}_3 \cos \gamma - \hat{j}_3 \sin \gamma
\]

\[
N(\beta) = O_1 + R_o (\hat{j}_3 \cos \gamma + \hat{k}_3 \sin \gamma)
\]

Newton’s method in Tool-Tilt step converges when the solution for \( [X_{Tilt}] \) is found within the user specified tolerance given by parameter \( \epsilon_{vicinity} \). In case of convergence, the \( \text{Flag}_{Newton} \) is set to be \( \text{TRUE} \) as shown in Fig. 15. The converged tool-tilt solution \( [X_{Tilt}] \) is then subjected to the following three checks before it is accepted as a valid solution:

1. The first check is to ensure that the second point of contact \( Q = S(u_2, v_2) \) should not lie too close to the first point of contact \( P = S(u_1, v_1) \). To avoid this, a rectangular boundary for parameters \( u \) is taken around point \( P \) to force the solution of second point of contact away from point \( P \) as shown in Fig. 16. The limits of the rectangular boundary are taken as an offset from point \( P \) using a tolerance of \( \epsilon_{vicinity} \). In case a converged solution for second contact point \( Q \) lies within the boundary defined around point \( P \), then the tilt solution is discarded and a new trial with a new random seed is initiated.

2. If the first check is satisfied, then the next check is implemented to ensure that the converged tool-tilt solution has tool-axis orientation pointing in the desired direction. It is possible that the tool-axis solution can be such that the tilted tool may have its Shank intersecting with the surface. This corresponds to the tool-angle \( \beta \) being positive and the axis of the tilted tool points towards the negative \( z \)-direction. Such tilt solutions are also discarded and the new trial is initiated with new random seed.

3. A tilt solution that satisfies the above two checks is finally tested for gouging. For gouge checking, the bottom surface of the toroidal tool, which is formed by surface of revolution of the cutting edge, is discretized into a polar grid as shown in Fig. 17. The grid is defined with circular rings that are normal to the tool axis and centered on the tool axis. Since the bottom arc of the cutting tool is expected to possibly gouge the surface, only a portion bounded between \( \theta_{min} \) and \( \theta_{max} \) is considered for the gouge check. This part of the cutting edge is divided into \( N_\theta \) divisions and each ring in the grid corresponds to these divisions on the cutting edge. Each circular ring, defined in a plane normal to
Tool Tilt \( (\alpha, \text{Prev-Sol}, u, v, X_{\text{Max}}, Y_{\text{Max}}, \epsilon_{\text{vicinity}}, i, \hat{n}_p, N_\phi, O_1, T_0, N_{\text{Tilt}}, N_{\text{Iter}}, \epsilon_{\text{Tilt}}, R_o) \)

1. \(UL_u = u + \epsilon_{\text{vicinity}}, \quad LL_u = u - \epsilon_{\text{vicinity}}\)
   \(UL_v = v + \epsilon_{\text{vicinity}}, \quad LL_v = v - \epsilon_{\text{vicinity}}\)

2. Initialize \(\text{Flag}_{\text{Newton}}, \text{Flag}_{\text{Gouge}}, i = 0\)

3. Compute \((i_0, j_0, k_0, \beta, \phi, \epsilon_{\text{Tilt}}, \epsilon_{\text{Gouge}}, \hat{n}_p, \hat{n}_q, \eta_c, \eta_c')\)

4. While \((i < N_{\text{Tilt}})\)
   - If \((\text{Prev-Sol} = \text{TRUE})\) Then
     \[ |X| = |X_{\text{Tilt}}(i-1)| \]
   - Else Randomize \([X] = [u, v, \beta, \phi]\)
   - If \((i > 1000)\) Then
     \[ \epsilon' = 0.001 \times \epsilon_{\text{vicinity}} \]
     \(UL_u = u + \epsilon', \quad LL_u = u - \epsilon'\)
     \(UL_v = v + \epsilon', \quad LL_v = v - \epsilon'\)
   - For \(j = 0, j < N_{\text{Iter}}, j++\)
     - Compute \(i_j, j_3, k_3, \frac{\partial\phi}{\partial u}, \frac{\partial\phi}{\partial v}, \frac{\partial^2\phi}{\partial u^2}, \frac{\partial^2\phi}{\partial v^2}, \frac{\partial^2\phi}{\partial u \partial v}, \hat{n}_q, \frac{\partial u}{\partial x}, \frac{\partial v}{\partial y}, \eta_c, a_3, b_3, r_3, C'_{\text{tor}}\)
     - Compute \(j'\)
     - For \((k = 0, k < 4, k++\)
       - If \((k < 3)\)
         - Compute \(F[X_k] = S(u, v) + \hat{n}_p R_1 - C'_{\text{tor}}\)
       - Else \(F[X_k] = \{T'_1 - S(u, v)\} \cdot \{\hat{r}_1 \times \hat{n}_2\}\)
     - If \((\text{Flag}_{\text{Newton}} = 1)\) Then
       - Print: Second Point is too close.
       - \(\text{Flag}_{\text{Newton}} = 0, \text{Prev-Sol} = \text{FALSE}, \text{Break}\)
     - Else if \((\hat{r}'[2] < 0)\) Then
       - Print: Inverted Tool Position.
       - \(\text{Flag}_{\text{Newton}} = 0, \text{Prev-Sol} = \text{FALSE}, \text{Break}\)
     - Else \(\text{Flag}_{\text{Gouge}} = \text{Gouge-Check}(T'_1, \hat{r}_1, \hat{n}_p)\)
       - If \((\text{Flag}_{\text{Gouge}} = 1)\) Then
         - Print: Tool is Gouging.
         - \(\text{Flag}_{\text{Newton}} = 0, \text{Prev-Sol} = \text{FALSE}, \text{Break}\)
       - Else
         - Print: Valid Tilt Solution Found
         - \(\text{Prev-Sol} = \text{TRUE}\)
         - Return \([X, \hat{r}', \text{Prev-Sol}]\)
   - Else \(N_{\text{Next j}}\)
     - If \((j = N_{\text{Iter}})\) Then
       - Print: Non Converging Solution. Try new seed.
       - \(\text{Prev-Sol} = \text{FALSE}\)
     - Next i

5. If \((i = N_{\text{Tilt}})\) Then
   - Print: No Feasible DSTM Solution Exists
   - \(\text{Prev-Sol} = \text{FALSE}\)
   - \(u = \frac{\hat{F}[0]}{X_{\text{Max}}}, \quad v = \frac{\hat{F}[1]}{Y_{\text{Max}}}, \quad \beta = 0, \phi = 0\)
   - Compute \(\hat{r}'\)
   - Return \([X, \hat{r}', \text{Prev-Sol}]\)

Fig. 15 Pseudo-code for the numerical implementation of tool-tilt

the tool axis, is divided into \(N_\phi\) divisions as shown in Fig. 17. Since the Bézier surface is uniformly defined in the \(X-\) and \(Y-\)directions, each grid point on the tool \(T_{i,j}\) can be mapped to the corresponding point on the Bézier surface and the \(Z\) component of the surface points is computed.

The \(Z-\)axis component of the grid point on tool-surface should not lie more than \(\epsilon_{\text{Gouge}}\) below the corresponding point on Bézier surface. If all the points on the tool grid meet this criteria, then the tool-tilt solution is accepted as a valid solution. However, if any point on the tool surface lies below the corresponding point on the Bézier surface by a magnitude more than \(\epsilon_{\text{Gouge}}\), then the tool is considered to be gouging with surface. If the tool is found to be gouging then the solution is discarded and new trial is initiated with a new random seed. The gouge checking procedure is identical to [10] and therefore pseudo-code for the same is not given here.

Fig. 16 Boundary around first point of contact \(P\) for implementing the vicinity check in the tool-tilt step to locate the second point of contact

Fig. 17 Discritization of the surface of toroidal tool for implementation of Gouge-Check algorithm
At any given tool-drop location $T$, it is also possible that a valid tool-tilt solution is not found despite considering $N_{Tilt}$ random seed points. In such cases, we assume that the local maximum principal curvature of the surface and the tool curvature are identical, making it infeasible to tilt the tool anymore. Therefore, in absence of a valid tilt solution, the first and second point of contact are considered to be at the same location.

Author contribution The authors contributed the work in this paper.

Funding This research was supported by NSERC.

Availability of data and material Some of the surfaces used in this work are given in the paper. The remaining surfaces and other data are available on request.

Code availability Pseudo-code for the work in this paper appears in the Appendix; the actual code is available on request.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Warkentin A, Ismail F, Bedi S (2000) Multi-point tool positioning strategy for 5-axis machining of sculptured surfaces. Comput Aided Geomet Design 17(1):83–100
2. Warkentin A, Ismail F, Bedi S (2000) Comparison between multipoint and other 5-axis tool positioning strategies. Int J Mach Tools Manuf 40(2):185–208
3. Gray P, Bedi S, Ismail F (2005) Arc-intersect method for 5-axis tool positioning, Computer-Aided Design, vol 37. Elsevier, Amsterdam, pp 663–674
4. Liang F, Kang C, Fang F (2020) A review on tool orientation planning in multi-axis machining. International Journal of Production Research, Int. J. Prod. Review, online 1–31
5. Duvedi RK, Bedi S, Batish A, Mann S (2014) A multipoint method for 5-axis machining of triangulated surface models. Comput Aided Des 52:17–26
6. Duvedi RK, Singh M, Bedi S, Mann S (2020) Multipoint tool positioning of a toroidal end mill for five-axis machining of generalized tensor product Bézier surfaces. Int J Adv Manuf Technol 113:495–503
7. Sharma SK, Duvedi RK, Bedi S, Mann S (2019) A method for generating multiple solutions for multipoint five-axis tool positioning. Int J Adv Manuf Technol 100:2509–2520
8. Sharma SK, Duvedi RK, Bedi S, Mann S (2019) A multipoint tool positioning method for five-axis machining in the region of two intersecting tensor product Bezier surfaces. Int J Machine Tools Manufact 142:42–53
9. Duvedi RK, Bedi S, Mann S (2017) Numerical implementation of drop and tilt method of 5-axis tool positioning for tensor product surfaces. In: International journal of advanced manufacturing technology. Springer, London, pp 1–14. https://doi.org/10.1007/s00170-017-1193-1
10. Duvedi RK, Bedi S, Mann S (2018) An efficient multipoint 5-axis tool positioning method for tensor product surfaces. In: International journal of advanced manufacturing technology. Springer, London, pp 279–295. https://doi.org/10.1007/s00170-018-1940-y

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.