



# A novel convex hull method for optimum multi-point 5-axis tool positioning for machining of complex sculptured surfaces

Andreas Marios Tsainis<sup>1,2</sup> · George Papazafeiropoulos<sup>3</sup> · Constantinos Stergiou<sup>1,2</sup> 

Received: 30 December 2018 / Accepted: 29 April 2019 / Published online: 22 May 2019  
© Springer-Verlag London Ltd., part of Springer Nature 2019

## Abstract

The Point-In-Convex-Hull-Control (PICHC) method is presented for the calculation of the optimum tool orientation during the 5-axis simultaneous CNC machining of sculptured surfaces. More specifically, this novel exact multi-point algorithm minimizes the clearance and/or overclosure of the contact surface between the tool and the workpiece and maximizes the material removal rate. It can be used in sculptured surfaces, where conventional algorithms allow up to two points of tool-workpiece contact, leading to suboptimal material removal rate. The proposed method allows a third contact point which makes an ideal matching of the tool-workpiece surface and its application is validated both numerically and experimentally in two case studies that are published in the literature. In the first case study, the PICHC method is compared with an exact algorithm, whereas in the second case study, it is compared with an evolutionary optimization algorithm coupled with a commercial software. The numerical and experimental results, apart from being more accurate and less computationally demanding than other commercial software algorithms, prove the suitability, versatility, and robustness of the developed methodology, which can be extended to describe and control more advanced machining problems, either stand-alone or within an API.

**Keywords** 5-axis CNC milling · Tool path planning · Optimum orientation · Sculptured surface

## 1 Introduction

In recent years manufacturing demands in the aerospace, marine, and automotive industry have increased significantly, on which research has focused on. For example, in the aerospace industry, cast and dies with geometries of various complexities and sizes have to be machined to tight tolerances in order

to match the various components of a structure perfectly. In the marine industry, this applies to machining of propellers, blades in water jets, etc. Similarly, in the automotive industry, a large number of mechanical parts need to be machined with accuracy to be functional. As there is increasing demand for structures with a high degree of complexity which requires further improvements to be made in the quality-to-production time ratio, parts with free-form surfaces (sculptured surfaces) are machined with 5-axis computer numerical control (CNC) machines because of the flexibility that these machines offer in manufacturing. In the past decade, the use of CNC machines with fewer active axes, without the capability of *simultaneously* using the various axes, limited the machining processes (such as the manufacturing of complex sculptured surfaces or curves, like the Bezier curves, or T-splines.) only to the manufacture of simple designs.

During the past few years, several tool positioning algorithms have been developed for five axis CNC machining of sculptured surfaces, which belong to either of two groups. The first one is of the single point machining (SPM) methods where a single tool contact point interacts with the workpiece during machining. The second group uses enhanced multi-point machining algorithms (MPM) that involve more than

---

✉ Constantinos Stergiou  
stergiou@uniwa.gr

Andreas Marios Tsainis  
tsainis@uniwa.gr

George Papazafeiropoulos  
gpapazafeiropoulos@yahoo.gr

<sup>1</sup> Faculty of Science, Engineering and Computing, Kingston University London, Penrhyn Rd, Kingston upon Thames KT1 2EE, UK

<sup>2</sup> Department of Mechanical Engineering, University of West Attica, 12244 Athens, GR, Greece

<sup>3</sup> Institute of Structural Analysis and Seismic Research, National Technical University of Athens, 15780 Athens, GR, Greece

one point between the tool and the workpiece surface. The first investigation in tool orientation for the SPM method was done by Vickers et al. [1] where a fixed tilt angle of the tool was used. This approach is known as the Sturz method.

Afterwards, Jensen et al. [2–4] presented the curvature matching method that calculates the optimal tool orientation angle in the area surrounding the cutter contact point based on the local curvature of an arbitrary free-form surface. Rao et al. [5, 6] proposed a technique for tool orientation and positioning based on local surface properties in order to investigate the effect of feed direction on five axis tool paths.

Wang et al. [7, 8] proposed the curvature catering method which positions the disk cutter on the machined workpiece. The contact surfaces between the tool and the workpiece during the process are supposed to have the same derivatives up to the third order in the section normal to the direction of feed.

The rolling ball method (RBM) presented a few years later by Gray et al. [9, 10] had a more computationally efficient algorithm which inputs the surface coordinates and surface normal. A variable radius ball rolls along the cutter contact points of the tool path generated previously and calculates the position of the cutting tool. Depending on the geometry of the cutting tool, a small area of the ball's surface is used for machining the surface of the workpiece. A grid of points is generated around the surface within the tool and the workpiece at each CC point, in order to calculate the radius of each ball. The radius with the better matching results is used as the rolling ball's radius, immediately after the inspection of a pseudo-radius that is calculated for each point of the grid. The process to select the appropriate radius with the rolling ball method is based on a hierarchical model of surface profiles targeted between convex to concave surfaces.

Gray et al. [11, 12] also proposed the arc intersect method (AIM) to improve the rolling ball's method presented previously. This is achieved by the direct positioning of the cutting tool to contact the surface in contrast to the RBM method. AIM is an area-based method that calculates tool positioning to avoid collisions without using any iterative gouge check or correction algorithms.

Hosseinkhani et al. [13] introduced the penetration elimination method (PEM) that aims both for computational efficiency and optimal tool orientation estimate. Two innovative techniques are used for the calculation of the tool orientation. The first technique develops a quantitative definition for gauge inspection purposes together with numerical root-finding algorithms. The second technique is used to reduce the total calculation time by removing ineffective grid points from the computational model. Compared to AIM, the PEM is 7.5 times faster.

Cha-Soo et al. [14] presented a searching method, the configuration-space search method (CSSM). It estimates the optimal tool orientation by taking into account the local gouging, rear gouging, and global tool collision during machining.

Based on the surface error indications, the initial possible tool orientations are calculated with a search method which eliminates tool-workpiece collision. In order to minimize the machined surface error, the locally optimal tool orientation referring to the C space is determined, with the minimum cusp height providing an objective function in the algorithm.

The second group of tool positioning algorithms that have been developed for five axis CNC machining sculptured surfaces includes multi-point interaction algorithms (MPM) which have been developed in the last decade. Engeli et al. [15] introduced a position strategy called the Hermite method. With this method, the second contact point between the tool and the workpiece is calculated with a Hermite interpolation instead of direct curvature calculations, which maximizes the strip width. The strip width is created by the geometry of the tool and relates to the overall machining time.

Fan et al. [16, 17] developed the rotary contact method (RCM), which is a tool orientation and positioning algorithm that is focused on concave surface machining and uses a torus end mill shape as the cutting tool. The logical flow of the algorithm is based on offset surfaces instead of the actual surface of the model and initializes the first tool location in a single point contact and rotates the tool till the second contact point is calculated. It is a non-conventional technique for gouge-free tool positioning and increased strip width during machining.

A global method associated with the instantaneous cutter position error (ICPE) has been introduced by Li and Chen [18]. It focuses on maximizing the length of the ICPE curve which relates to the tolerance zone. Chen et al. [19, 20] presented the middle point error control method (MPEC) which eliminates fluctuations created by cutting strip width (FCCS) and optimizes the position of the cutting tool depending on the workpiece surface. A toroidal cutting tool is used and is positioned on the effective character curve segment (ECCS) in the midpoint, whereas the other two degrees of freedom of the tool are adjusted appropriately to improve the global position of the tool.

In order to minimize the time travel distance by controlling the smoothness of the tool orientation, a complete gouge free three-dimensional C space method was developed by Lu et al. [21]. This is a quite effective method in cases that collisions must be avoided when compared to the C space methods described earlier but imposes significant computational load. A three dimensional C space method was first proposed by Choi et al. [22], in which the geometric characteristics of the ideal surface and the stock surface of the model are described in C space elements and the tool path generation is calculated in the configuration space. Morishige et al. [23, 24] proposed a C space method to eliminate collisions between the tool and the workpiece. Jun et al. [14] presented an optimization method aiming for the local, rear, and global gouge avoidance. This was achieved by maximizing the scallop height while

minimizing the range of the angles of the tool orientations. Lee [25] introduced the admissible tool orientation control method that focuses on the collision avoidance and rear gouges, by analyzing the local and global surface shapes of the model.

A novel tool orientation algorithm was introduced by Ravinder Kumar et al. [26]. It matches accurately the toroidal cutting tool and the triangulated surface. With this method, the tool is dropped until a single point contact is matched. Then the tool is rotated around the pseudo-insert theoretical axis to find the second point of contact. In this manner, a gouge-free tool position is achieved.

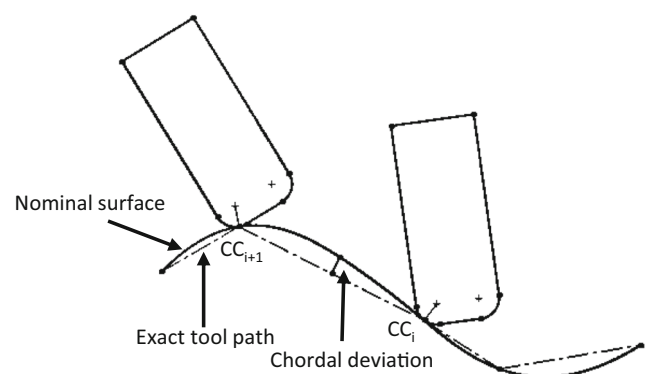
Despite the fact that numerous tool positioning methods have been introduced during the last decades, the need for searching the optimal combination of the parameters which affect the 5-axis CNC milling process is considered crucial. Thus, concerning the optimum tool path generation in conjunction with the optimum tool positioning, a variety of stochastic optimization approaches were developed. Among others, genetic-evolutionary algorithms (Gas-EAs), artificial neural networks (ANNs), and simulated annealing (SA) are the most common used from the researchers. Kersting and Zabel [27] utilized a multi-population GA to resolve complex problems which can be found in 5-axis CNC milling optimization. Afterwards, Ulker et al. [28], in order to reduce the overall machining time for various benchmark case studies, strived to significantly reduce the cutter location points by employing an artificial immune system (AMS). Li et al. [29] introduced a back-propagation ANN with a multi-objective optimization strategy aiming to find the optimum cutting parameters for sculptured surfaces. Among various milling processes, face milling is considered time-consuming having as a result the overall machining operation to be increased. Apparently, special tools exist that can be utilized for feed rate incensement in conjunction with low surface roughness but they are not a general rule of thumb. Raja et al. [30] seek to tackle the aforementioned process by implementing a multi-objective optimization approach utilizing particle swarm optimization (PSO) method. The robustness and versatility of their algorithm tested by conducting a number of experimental runs aiming to attain favorable surface quality of the specimens as well as machining time minimization. Subsequently, Kim [31] dealt with rough machining operation in planar tool trajectories. He was employed a SA method in order to minimize the cycle time and therefore the manufacturing cost by calculating the proper tool trajectory directions. Comparative studies of the various evolutionary algorithms have been carried out for various machining operations (e.g., [32–34]) and hybrid differential evolution algorithms have also been developed regarding the optimization of turning and milling operations [35, 36]. Advanced evolutionary optimization algorithms have also been developed to determine optimal machining parameters in manufacturing processes [37].

It is clear that, to produce high quality products with tight tolerances and superior surface finish still remains a challenge for engineers. The main purpose of this study is a novel method for optimum match of the tool geometry to the final surface of the workpiece and therefore (a) minimization of deviations from required tolerances, (b) minimization of machining times, and (c) maximization of material removal rates without any use of optimization strategies (Figs. 1 and 2).

## 2 Methodology

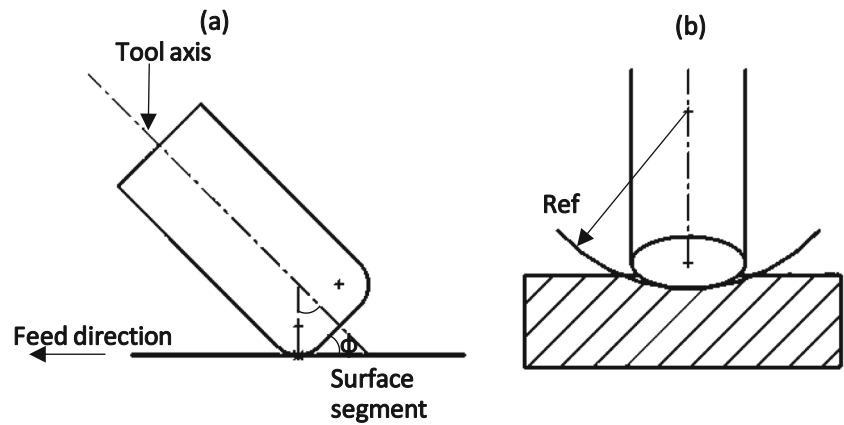
The basic characteristics and limitations of the proposed tool orientation algorithm (PICHC) are the following:

- The clearances and/or overclosures of the contact surface between the tool and the workpiece (see Fig. 3) are as small as possible, for the final surface to be smooth and closest to the initial design.
- The effective radius of the tool (see Fig. 4) and the number of contact points between the tool and the workpiece (see Fig. 3) are maximized. Maximization of the effective radius of the tool is performed in the case of relatively smooth workpiece surfaces, whereas maximization of the contact points (up to 3), is performed for surfaces of increased curvature. The purpose of these is to maximize the material removal rate of the workpiece, and reduce the time required for machining.
- The methodology includes the discretization of the tool and workpiece surfaces, which renders it applicable for any tool and workpiece surface geometries.
- The transition of the tool between successive cutting points (see Fig. 6) must be smooth in order to achieve the desirable surface roughness. For this purpose, extra constraints have been imposed to mitigate high variations of angles between successive CC points.
- The PICHC method can be applied only in workpiece surfaces that do not have cavities or other geometrical



**Fig. 1** Chordal deviation between the actual and approximated cutting tool path

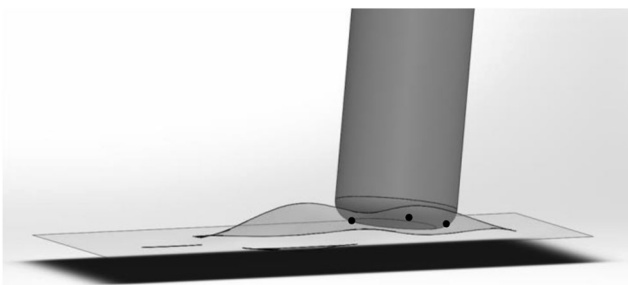
**Fig. 2** Local tool-workpiece engagement: **a** tool inclination obtained in an arbitrary segment of the given cutting surface, **b** effective radius of the cutting tool



irregularities in which the normal vector of the surface intersects the workpiece. In these cases, the PICHC method cannot be applied as it is; it needs to be appropriately extended.

- The thermomechanical response of the workpiece and the tool during the machining process is not taken into account by the proposed algorithm. Inclusion of these effects for the calculation of the optimum tool orientation in 5-axis milling processes with the PICHC method is definitely a good option for future work on this subject.

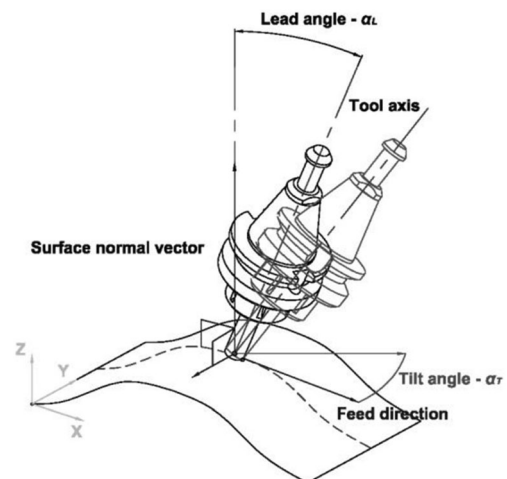
The algorithm that is presented in this study is more of a brutal-force nature, than an evolutionary optimization algorithm. In addition, since there is the need to find the optimum tool orientation regardless of the tool and workpiece geometries, it is necessary to implement numerical geometry point cloud representations of the last two. Therefore, the tool and workpiece surfaces are described through the use of point clouds. The point sets in which the surfaces of the tool and the workpiece are discretized (locally or globally) need to numerically represent the corresponding solids and handle various issues regarding their contact. The simplest shape approximation of a set of points is its convex hull. The convex hull can be informally defined by a rubber band which



**Fig. 3** The concept of the third contact point which leads to an ideal fitting of the tool-workpiece geometry

surrounds a point set and is let to snap tightly around the points. The purpose of using convex hull in this study is to represent a 3D surface of a solid structure as accurately as possible, in order to ensure that the contact between the tool and the workpiece is optimum (i.e., maximizes the performance of the machining process, minimizes deviations between the machined ideal geometries, and at the same time requires a reasonable computational effort). The convex hull of the tool that is used for the calculations in this study is shown in Fig. 5.

For each of the facets (triangles) in which the tool is discretized, there is a corresponding facet on the workpiece surface which is generated by the projection of the former on the workpiece surface, i.e., the vertices  $F_{2,i}$ ,  $i = 1, 2, 3$  of facet  $F_2$  are the projection of the vertices  $F_{1,i}$ ,  $i = 1, 2, 3$  of facet  $F_1$  on the workpiece surface along the global  $z$ -axis (see Fig. 6). The tool facet  $F_1$  is then fitted on the workpiece facet  $F_2$  through geometric transformations (translations and rotations of the tool about the three global axes  $x$ ,  $y$ , and  $z$ ) so that  $F_1$



**Fig. 4** Transition of the tool between successive cutting points using rotational constraints to ensure smoothness of the final workpiece surface (after [38])

and  $F_2$  are coplanar and their centroids coincide. Suppose that  $C_{F1}$  and  $C_{F2}$  are the centroids of  $F_1$  and  $F_2$  respectively. For each facet, the coordinates of its three vertices,  $F_{1,i}$  and  $F_{2,i}$ ,  $i = 1, 2, 3$  and are known and therefore the vectors  $v_{1,i}$ ,  $i = 1, 2$  and  $v_{2,i}$ ,  $i = 1, 2$  are calculated as follows:

$$v_{1,i} = F_{1,i} - C_{F1} \tag{1}$$

$$v_{2,i} = F_{2,i} - C_{F2} \tag{2}$$

where the coordinates of the centroids are given by:

$$C_{F1} = \frac{1}{3} \sum_{i=1}^3 F_{1,i} \tag{3}$$

$$C_{F2} = \frac{1}{3} \sum_{i=1}^3 F_{2,i} \tag{4}$$

The equation of the plane passing through the vertices  $i$  of facet  $j$ ,  $F_{j,i}$ ,  $(F_{j,i,x}, F_{j,i,y}, F_{j,i,z})$  is given by the determinant:

$$\begin{vmatrix} x & y & z & 1 \\ F_{j,1,x} & F_{j,1,y} & F_{j,1,z} & 1 \\ F_{j,2,x} & F_{j,2,y} & F_{j,2,z} & 1 \\ F_{j,3,x} & F_{j,3,y} & F_{j,3,z} & 1 \end{vmatrix} = 0 \tag{5}$$

which results in an equation of the form:

$$cF_j[x \ y \ z]^T + dF_j \tag{6}$$

where  $cF_j$  is a  $[1 \times 3]$  row vector with coefficients:

$$cF_j = \left[ \begin{vmatrix} F_{j,1,y} & F_{j,1,z} & 1 \\ F_{j,2,y} & F_{j,2,z} & 1 \\ F_{j,3,y} & F_{j,3,z} & 1 \end{vmatrix} - \begin{vmatrix} F_{j,1,x} & F_{j,1,z} & 1 \\ F_{j,2,x} & F_{j,2,z} & 1 \\ F_{j,3,x} & F_{j,3,z} & 1 \end{vmatrix} \ \begin{vmatrix} F_{j,1,x} & F_{j,1,y} & 1 \\ F_{j,2,x} & F_{j,2,y} & 1 \\ F_{j,3,x} & F_{j,3,y} & 1 \end{vmatrix} \right] \tag{7}$$

and  $dF_j$  is a scalar given by the determinant:

$$dF_j = \begin{vmatrix} F_{j,1,x} & F_{j,1,y} & F_{j,1,z} \\ F_{j,2,x} & F_{j,2,y} & F_{j,2,z} \\ F_{j,3,x} & F_{j,3,y} & F_{j,3,z} \end{vmatrix} \tag{8}$$

Initially,  $F_1$  is translated so that  $C_{F1}$  and  $C_{F2}$  coincide. The translated  $F_1$  is denoted by  $F_1^t$  to distinguish it from the translated and rotated  $F_1$ , denoted by  $F_1^{tr}$ . If the new centroid is denoted by  $C_F$ , it will be:

$$F_{1,i}^t = F_{1,i} + \Delta C \tag{9}$$

$$C_F = C_{F1} + \Delta C \tag{10}$$

$$C_F = C_{F2} \tag{11}$$

where:

$$\Delta C = C_{F2} - C_{F1} \tag{12}$$

is the translational vector between the centroids of  $F_1$  and  $F_2$ . Equations (1) and (2) become:

$$v_{1,i} = F_{1,i}^{new} - C_F \tag{13}$$

$$v_{2,i} = F_{2,i} - C_F \tag{14}$$

After this, the outer (cross) product between the first point vector  $(v_{1,1}, v_{2,1})$  and the second point vector  $(v_{1,2}, v_{2,2})$  of each facet is calculated and then normalized as follows:

$$\bar{v}_1 = \frac{v_{1,1} \times v_{1,2}}{|v_{1,1} \times v_{1,2}|} \tag{15}$$

$$\bar{v}_2 = \frac{v_{2,1} \times v_{2,2}}{|v_{2,1} \times v_{2,2}|} \tag{16}$$

Each cross product  $(\bar{v}_1, \bar{v}_2)$  has the property that it is vertical to the plane of the corresponding facet. Intuitively it can be said that the angles of rotation required for the two facets to be coplanar are equal to the angles of rotation between  $\bar{v}_1$  and  $\bar{v}_2$ . The above can be seen in Fig. 6.

The new position and orientation of  $F_1$ ,  $F_1^{tr}$ , is determined in matrix form by the following equation:

$$F_{1,i}^{tr} = R F_{1,i}^t = R(F_{1,i} + \Delta C) \tag{17}$$

where  $R$  is the rotation matrix which is used to rotate  $F_1$  after it has been translated so that it becomes coplanar with  $F_2$  in Fig. 6 and is calculated by the following equation:

$$R = \begin{cases} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} + \frac{1 - (\bar{v}_{1,x}\bar{v}_{2,x} + \bar{v}_{1,y}\bar{v}_{2,y} + \bar{v}_{1,z}\bar{v}_{2,z})}{|w|^2} \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix} & |w| > 0 \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & |w| = 0 \end{cases} \tag{18}$$

where  $\bar{v}_{1,x}, \bar{v}_{1,y}, \bar{v}_{1,z}$  and  $\bar{v}_{2,x}, \bar{v}_{2,y}, \bar{v}_{2,z}$  are the components of the vectors  $\bar{v}_1$  and  $\bar{v}_2$  respectively and  $w$  is the cross product of the vectors  $\bar{v}_1$  and  $\bar{v}_2$ , given by the determinant:

$$w = \begin{vmatrix} i & j & k \\ \bar{v}_{1,x} & \bar{v}_{1,y} & \bar{v}_{1,z} \\ \bar{v}_{2,x} & \bar{v}_{2,y} & \bar{v}_{2,z} \end{vmatrix} \tag{19}$$



and  $i, j$ , and  $k$  are the standard basis vectors. Based on Eq. (19), the components of  $w$  appearing in (18) are given by:

$$\begin{aligned} w_1 &= \bar{v}_{1,y}\bar{v}_{2,z} - \bar{v}_{1,z}\bar{v}_{2,y} \\ w_2 &= \bar{v}_{1,z}\bar{v}_{2,x} - \bar{v}_{1,x}\bar{v}_{2,z} \\ w_3 &= \bar{v}_{1,x}\bar{v}_{2,y} - \bar{v}_{1,y}\bar{v}_{2,x} \end{aligned} \quad (20)$$

After the tool is translated and rotated according to the fitting procedure described above, it is checked if any overclosures exist at the contact surface between the tool and the workpiece. If so, the current tool position is rejected since it leads to a sculptured surface different from the desirable one. The algorithm which performs this check is generally found in the literature as “point in polygon” algorithm (PIP). The PIP algorithm implemented in the algorithm presented in this study works as follows:

1. The convex hull of the tool at its initial position (CH) is translated and rotated according to Eq. (17) giving:

$$CH^{tr} = R(CH + \Delta C) \quad (21)$$

where  $\Delta C$  and  $R$  are given by Eqs. (12) and (18) respectively. Any facet  $F_1$  of CH is transformed in the same way as well:

$$F_1^{tr} = R(F_1 + \Delta C) \quad (22)$$

2. The coefficients  $cF_1^{tr}$  and  $dF_1^{tr}$  of the equation of the plane of any facet  $F_1^{tr}$  of  $CH^{tr}$  are found from Eqs. (7) and (8) respectively, where  $j=1$  and the superscript  $tr$  applies in all quantities in these equations.
3. Using these coefficients and the coordinates of the points of the workpiece  $P_{wp}$  ( $x_{wp}$ ,  $y_{wp}$ ,  $z_{wp}$ ), the workpiece points that belong to the interior of the convex hull CH are found by checking the sign of  $S_c$  given by (23):

$$S_c = cF_1^{tr}P_{wp}^T + dF_1^{tr} \leq -tol \quad (23)$$

where  $P_{wp}$  is a matrix with three columns containing the coordinates of the points of the workpiece and  $tol$  is a very small positive tolerance. If any entry of the vector  $S_c$  violates the inequality in (23), the current facet  $F_1^{tr}$  is rejected and the next facet  $F_1^{tr}$  of  $CH^{tr}$  is considered, in which case Eqs. (7), (8), and (23) are applied again, until the inequality in (23) is satisfied for all elements of  $S_c$ , in which case the optimum orientation algorithm stops. The new algorithm presented in this study, i.e., the point-in-convex-hull-control (PICHC) method owes its name to the controlling technique that is described in step 3 above and applied through Eq. (23).

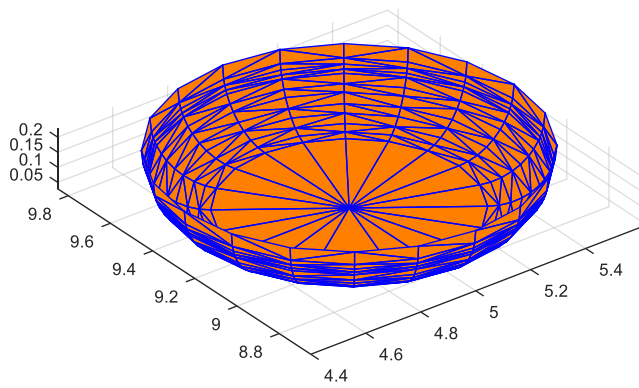
### 3 Numerical implementation

In order to calculate the optimum tool orientation in this study, two commercial software packages (Autodesk Inventor Professional 2018 [40] and CG-TECH VERICUT 7.3 [40]) were used in conjunction with the source code developed in the MATLAB [40] programming language. The Autodesk Inventor Professional 2018 API was used to discretize the workpiece surface and the cutting tool geometry (Fig. 7) in a format developed by the user of the application. The numerical results from the Autodesk Inventor Professional 2018 API were then automatically loaded into MATLAB to calculate the optimum tool positions using the proposed method. Finally, the appropriate G code was automatically produced with MATLAB, and then used in CG-TECH VERICUT 7.3 to simulate the cutting process with the optimum tool orientation method described in this study as illustrated in Fig. 8. It is noted that the Autodesk Inventor Professional 2018 does not have the capability to simulate the extracted G code from MATLAB, and the authors have therefore resorted to the third-party software GC-TECH VERICUT for this purpose.

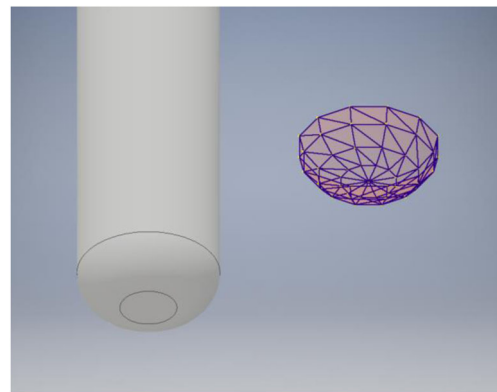
The main Matlab code of the optimum tool orientation algorithm uses various functions. The functions used are **convhull\_nd.m** which calculates the convex hull of the point cloud of the tool surface, **plane\_nd.m** which calculates the coefficients of the equation of the plane that passes through three points in space and is called by **convhull\_nd.m**, **PointSetFitCM.m** which calculates the translation and rotation vector that must be applied to an arbitrary plane in order to become coplanar with a given plane (Eqs. (12) and (18)) and **fcn\_RotationFromTwoVectors.m** which calculates the rotation matrix between two vectors and is called by **PointSetFitCM.m**. The above are illustrated in Fig. 8. The function **convhull\_nd.m** is used as an open source code developed by the authors. It is similar to Matlab’s function **convhulln.m**, but this is an improved version of it as it allows modifications inside the source code for optimum execution depending on the application. In Fig. 7 the initial geometry of the tool as developed with the AUTODESK INVENTOR PROFESSIONAL 2018 is shown, together with its discretized version of a point cloud (49 points) and its convex hull. The point cloud is created with the AUTODESK INVENTOR PROFESSIONAL 2018 API, whereas the convex hull is created with MATLAB.

#### 3.1 Local workpiece-tool discretization algorithm

An algorithm is implemented in Visual Basic for Applications (VBA), which is the programming language of the Autodesk Inventor Professional 2018 API, to perform the discretization of the workpiece and the tool surfaces. This algorithm serves to extract information about the points of the tool that are



**Fig. 5** Discretization of the surface of the tool into triangular facets based on the convex hull algorithm in graphical user interface of Matlab



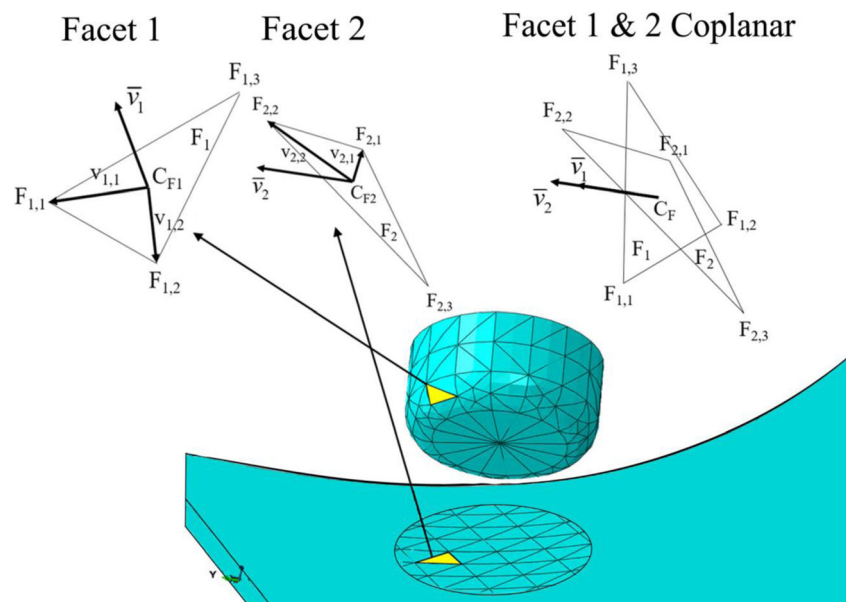
**Fig. 7** Initial tool geometry and discretized tool geometry with the convex hull algorithm

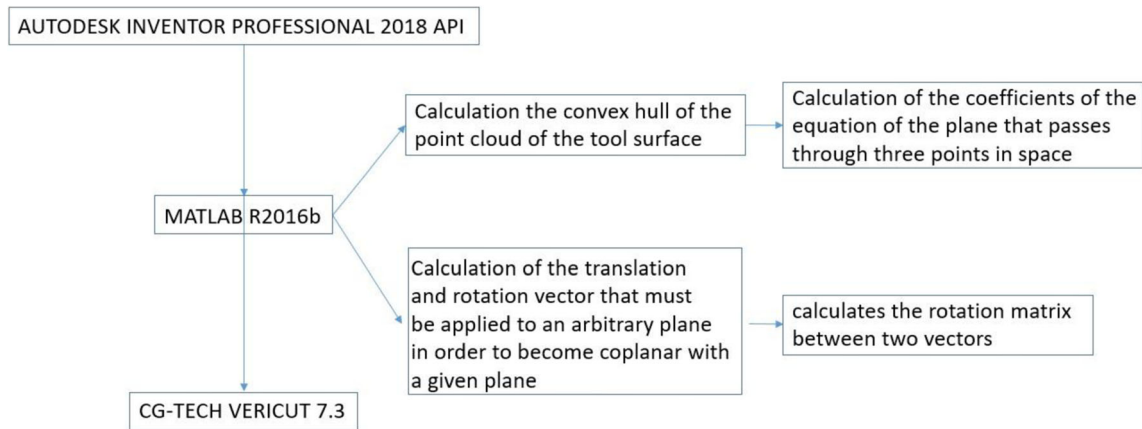
projected onto the workpiece surface in each Cutter Contact (CC) point.

At the beginning, the CC point cloud is created and the surface of the cutting tool is discretized. The CC point cloud is uniformly distributed along the workpiece surface. Following this, the assembly which includes the workpiece and the tool are created. Then the origins of the workpiece and the tool are initialized and the unit vectors are created in order to translate and rotate the parts in the assembly environment. The origins are created in a separate step as the position of the tool origin changes during its movement among the CC points. The user must select the sketch that is created with the point set on the tool, the surface of the workpiece and then the direction of the projection. If the user does not select any projection direction, the algorithm selects automatically from the geometry of the workpiece reference edges which are parallel

to the xyz coordinate system so that the direction of the projection is selected. The tool moves along all CC points based to a translational matrix, in which the coordinates of the tool origin are stored. Following this, the tool is translated and the coordinates of the tool points at their new positions are stored respectively. The points of the tool are projected on the workpiece surface as shown in Fig. 9b. After the calculation of the coordinates of the CC points at the workpiece surface and of the tool point sets projected on the workpiece surface at each CC point, the results are written to an external txt file in order to be post processed by the optimum tool orientation algorithm in MATLAB. It is noted that the optimum tool orientation for a given CC point depends not on the neighboring CC points, but only on the points of the locally discretized workpiece surface, which is directly linked to the discretization of the tool. This can be easily seen in Fig. 9 .

**Fig. 6** Fitting process of a facet of the workpiece ( $F_1$ ) and a corresponding facet of the tool ( $F_2$ ) to become coplanar





**Fig. 8** Diagram of the function dependency in the MATLAB code for the implementation of the PICH method proposed in this study

## 3.2 Optimum tool orientation algorithm

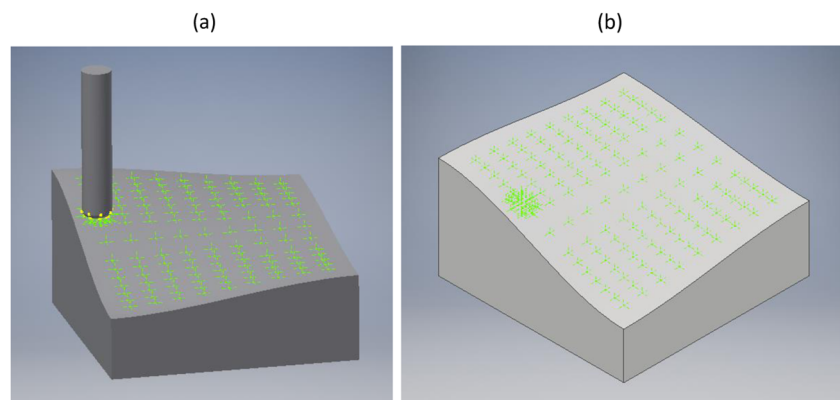
### 3.2.1 Structure of the algorithm

An algorithm is formulated which satisfies all of the criteria mentioned in Section 2 of this study. The idea behind this algorithm is that for each cutting position, the tool position is determined by five parameters (three translations about the global  $x$ ,  $y$ ,  $z$  axes and two rotations about the global  $x$ ,  $y$  axes), as the tool is considered as a rigid body. These five parameters are uniquely defined if the final position of the tool is known. On the other hand, a convenient way to represent the tool geometry numerically is to construct the convex hull of the point cloud in which the tool surface is enclosed. The convex hull virtually discretizes the tool surface into facets each of which is defined by three points. The same happens with the point cloud which defines the workpiece surface. In order to minimize any overclosures between the tool and workpiece surfaces in the area, where they come into contact and reduce the quality of the machined surface, a constraint where the facets of the convex hull of the tool are coplanar with those of the convex hull of the workpiece is imposed. However, this is not possible in practically every case, because of the different geometries of the tool and the workpiece, and

because of discretization errors of the surfaces. To overcome this, the facets of the convex hulls of the tool and the workpiece are scanned one by one through an iterative procedure (loop), and for each iteration, the current facet of the convex hull of the tool is translated and rotated according to Eq. (21) so that it becomes coplanar with the corresponding facet of the convex hull of the workpiece. Following this, any overclosures between the convex hull of the tool and the workpiece are detected by using Eq. (23). If at least one point of the point cloud of the workpiece lies inside the convex hull of the tool, then the current facet is rejected and the iterative procedure goes through the next.

The iterative procedure described above ensures that the contact between the tool and the workpiece minimizes overclosures in a controlled manner, but does not guarantee that the area of the contact surface is the maximum possible. The latter requirement can be satisfied by sorting the facets of the convex hull of the tool, based on their distances from the center of the tool, in ascending order. The iterative procedure starts from the first facet (with the smallest distance) and proceeds to facets with continuously increasing distance, until “perfect” contact is found as outlined above, at which point the iterative procedure stops and the algorithm proceeds to the next cutting position.

**Fig. 9** Local workpiece-tool discretization: **a** placement of the tool at a CC point, **b** projection of the points of the discretized tool on the workpiece surface





### 3.3 Implementation of the algorithm in MATLAB

The pseudocode shown in Listing 1 is programmed with the MATLAB R2016b programming language and determines the optimum orientation of the tool on the workpiece surface, with regard to the position and the lead and tilt angles. At the beginning, the coordinates of the point cloud of the tool and its projection on the workpiece surface for each cutting position are loaded from an external text file that is created by the AUTODESK INVENTOR PROFESSIONAL 2018 API. Then the tool points and the workpiece points are identified. Various variables are initialized to optimize the MATLAB code execution.

All the cutting positions of the workpiece are scanned in a loop. For each cutting position, the coordinates of the points of the workpiece and the coordinates of the points of the tool are calculated and arranged in an easy to handle matrix format.

Then the convex hull of the cutting tool is calculated, after appropriate perturbations of the point cloud of the tool are made to preserve the shape of the convex hull. This perturbation stage is crucial because normality in the convex hull is necessary to ensure optimum contact between the tool and the workpiece (and therefore produce meaningful results of the optimum position of the tool).

The point cloud of the tool must contain points that belong only to the part of its surface that comes (or may come) in contact with the workpiece, in order to reduce the computational effort. The convex hull of the point cloud, due to its convex nature, will enclose the whole point cloud and thus create facets which do not belong to the surface of the tool which participates in the cutting process. These facets do not have any physical meaning and have to be deleted to reduce computational time. This procedure takes place during the initial stages of the MATLAB code run. If the z-coordinates of all points of a facet exceed the maximum z limit, then this facet is deleted.

As it has already been stated, in order to achieve the maximum area at the contact surface, iterations are performed for each cutting position of the workpiece, during which the facets of the convex hull of the tool are scanned starting from

the facet with the smallest distance from the center of the tool in ascending order, and whenever “perfect” contact is achieved, the iterations stop.

In the pseudocode of the PICHC algorithm shown in Listing 1 inside the iterative loop, for each of the facets of the convex hull of the tool, the corresponding facet of the workpiece to which the former comes in contact with is found, and after this, the necessary rigid body translations and rotations of the tool facet to achieve this contact are found. This operation requires three translations about the global x, y, and z axes and two rotations about the global x and y axes, as described in a previous section. Rotation and translation of the tool with the rigid body translation and rotation described earlier is performed from its initial position to the position which corresponds to the best fitting of each pair of facets (tool-workpiece). At the final section of the while loop, there is a check whether all points of the workpiece lie outside the convex hull of the tool (avoid overclosure), in which case the iterations stop; in the opposite case (chordal deviation), the while loop continues. The rigid body parameters of the optimum tool translation and rotation are checked so that they do not exceed certain upper and lower limits, in order to avoid the high variations of the accelerations and jerks that develop during the machining process so that the final machined surface is sufficiently smooth. In case that the above check is not satisfied, the while loop proceeds to the next facet of the convex hull of the tool, even if this satisfies the contact requirements outlined in the preceding sections of this study. After this, they are written in a G code file that is produced by the MATLAB code. This G code is executed by CG-TECH VERICUT 7.3 to simulate the cutting process.

## 4 Results

In order to verify the applicability of the proposed optimum tool orientation method presented in this study, two case studies were considered, that are published in the literature. In the first case study, a comparison is made with the results of another exact algorithm developed in [40], which is proved to be

**Listing 1** Pseudocode of the PICHC algorithm applied for the optimum tool orientation

```

-Load the point cloud of the tool and the workpiece
-Find the convex hull of the tool
-Delete the planes of the convex hull that are at the nonconvex part of the tool point cloud
-Find the distance of all triangles from the center of the tool
-Sort the convex hull triangles based on their distance in ascending order
for each cutting position of the tool on the workpiece
  for the  $i^{\text{th}}$  facet of CH
    - Find  $\Delta C$  and  $R$  according to eq. (12) and (18) respectively
    - Fit the tool point cloud to workpiece according to eq. (21)
    - If eq. (23) is satisfied, exit the loop
  end
-Print optimum tool position in G code
end

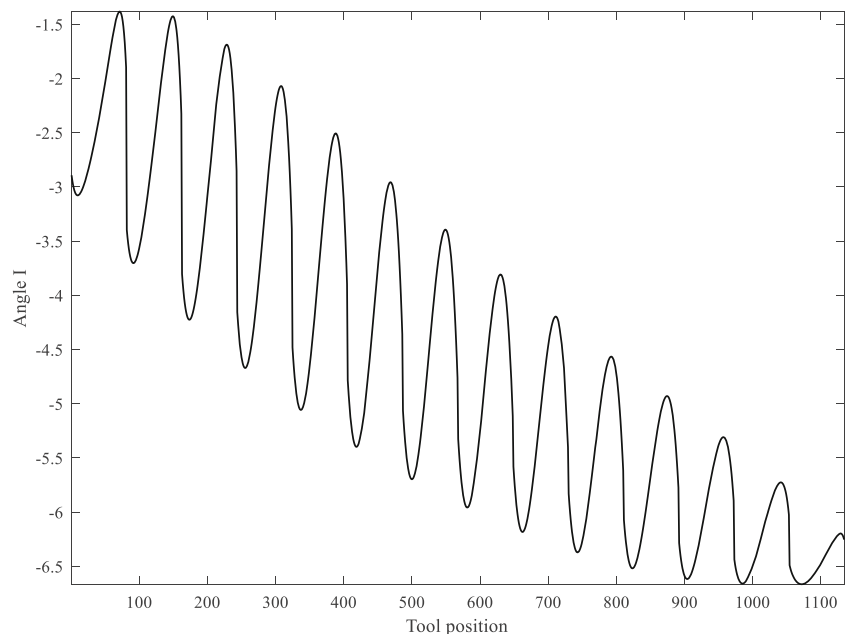
```

**Table 1** Numerical results of two cases of tool point refinement of the PICHC method (present study) and the MPEC method [40] for the test model shown in Fig. 12

	Tool discretization	Maximum excess	Running time
Chen et al. (2017) [40]	–	~0.005	~20 min
Present study, case 1	76 points	~0	~12 min
Present study, case 2	106 points	~0	12.85 min

more efficient than two other basic exact methods (the Middle Point Error Control Method, MPECM, and the mechanical equilibrium method, MEM) that have performed well, according to relevant studies in the literature. In the second case study, a comparison is made with the results of an evolutionary optimization algorithm, coupled with a commercial CAM software (CAD/CAM CATIA V5) [40]. For these two cases, results are obtained both numerically through simulations in CGTech VERICUT and experimentally through the use of a Haas VF4 5-axis CNC machine-center equipped with a Fanuc NC. Detailed presentation and comparison of the various results are shown in the following sections. The numerical experiments conducted in [38] were run in a desktop computer which employed an AMD Athlon 64 processor, 3200+, 2.00 GHz and 1.00 GB RAM supported by WinXP Pro., SP3 2002 version. On the other hand, the authors of the proposed method utilized an AMD Ryzen 51,600 Six-Core Processor, 3.2 GHz and 16 GB RAM supported by Windows 10. The hardware characteristics on which the numerical experiments of [40] were conducted are not mentioned in that study, but based on the publication date it is assumed that last generation hardware was utilized.

**Fig. 10** The variation of the lead angle ( $I$ ) during the machining simulation, according to the algorithm proposed in this study (PICHC method)



## 4.1 Comparison with the exact method

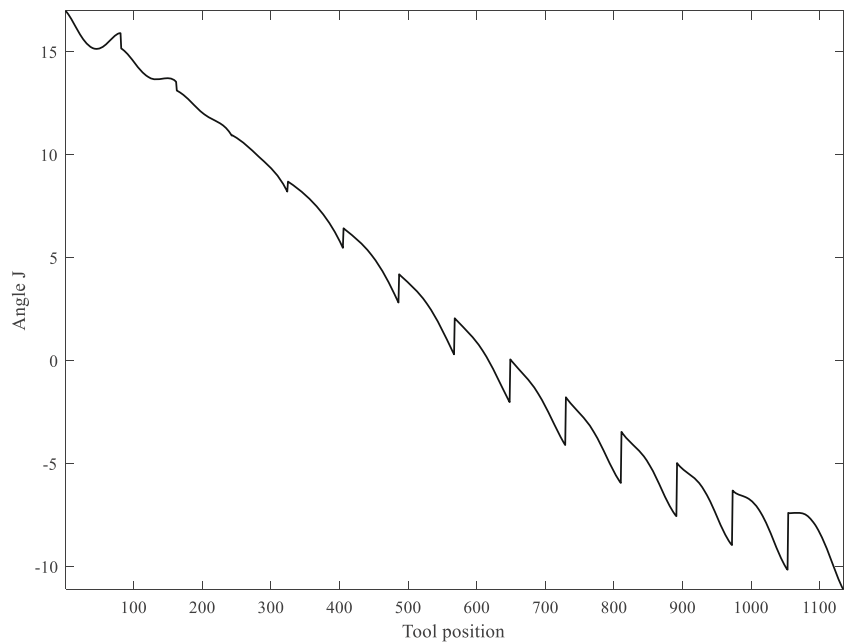
### 4.1.1 Simulation results

In [40], a machining simulation and experiment was performed on a mold surface  $S$ , the parametric equation of which in terms of the global coordinates  $S_x$ ,  $S_y$ , and  $S_z$  is given by the following equation:

$$\begin{aligned} S_x(u, v) &= 5.6v^2 + 88.9v - 94.4 \\ S_y(u, v) &= 28.1u^2 - 131.3u \\ S_z(u, v) &= 5.9(u^2v^2 + u^2v) - 3.9v^2u + 76.2u^2 \\ &\quad + 6.7v^2 - 27.3uv - 50.8u + 25v + 12.1 \end{aligned} \quad (24)$$

Figure 12 shows the simulation results computed by CG Tech Vericut for the above surface both using the algorithm proposed in this study (PICHC method) and using the improved MPEC algorithm used in [40]. For the mold surface, a 5-axis cutting tool path was calculated using a generic 5-axis post-processor. Also, a toroidal cutting tool with major radius 6.5 mm and minor radius 1.5 mm was selected, and a cutting tolerance of 0.1 mm is chosen which is higher than [40]. It is seen that 12 paths have been calculated in both cases, and

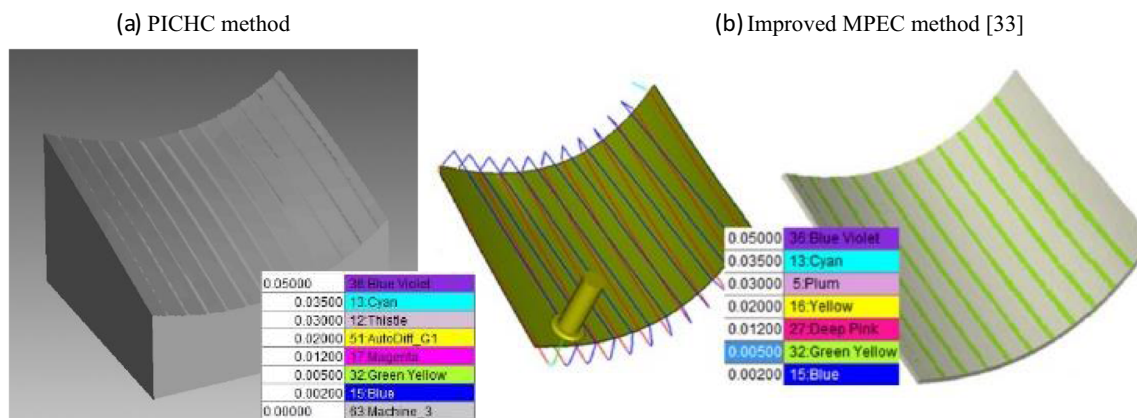
**Fig. 11** The variation of the tilt angle (J) during the machining simulation, according to the algorithm proposed in this study (PICHC method)



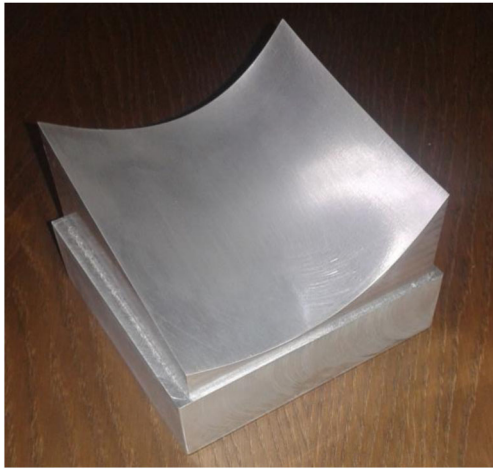
smoothness between successive cutting tool paths was attained. The strip width (radial depth) is fixed and equal to 9.5 mm contrary to that used in [40], which is variable with mean value 9.5 mm. Constant strip width during the machining process is more beneficial than varying strip width, because this leads to the stabilization of material removal rate for each cutting path, constant loading forces on the tool, as well as an increased tool life. The methodology utilized in the present study yields lower excess distribution along the machined workpiece surface, as illustrated in Fig. 12. The computing time of generating the tool path with the proposed algorithm is about 12 min whereas the same time in [40] was about 20 min. The above results are summarized in Table 1. In the same table, two cases of tool point discretization refinement are also examined, i.e., the tool is discretized into 76 points in case 1 and into 106 points in case 2. From the comparison between these two cases in terms of maximum excess and running time, it is concluded that there is no

discretization dependency on the results and therefore, the results are sufficiently accurate.

In [40], the author refers to the comparison results of optimal tilt angles and strip widths that calculated for selected cutting points along the surface of the workpiece. In the proposed algorithm (PICHC method) due to the constant strip width that is selected, it is more purposeful to concentrate on the lead and tilt angle variations during the machining simulation of the overall part. It can be seen from Fig. 10 that lead angle variations of the tool have relatively stable and repeated amplitude as they move smoothly to the maximum magnitude. Although a large number of variations are expected due to the use of constant strip width in areas of the mold surface with high curvature values, it seems that discretization based algorithms using multi-point contact capabilities have robust results. Additionally, in Fig. 11, smaller variations of the tilt angle are observed in contrast with the lead angle. This means that single, double, or three contact point capability of



**Fig. 12** Simulation results of the maximum excess on the mold surface with the PICHC method (left) and the improved MPEC method (right)



**Fig. 13** The experimental result of the machined mold surface considered by Chen et al. [40] and described by Eq. (24)

the PICHC method in combination with appropriate restrictions for smooth angle transitions leads to desirable results (Fig. 12).

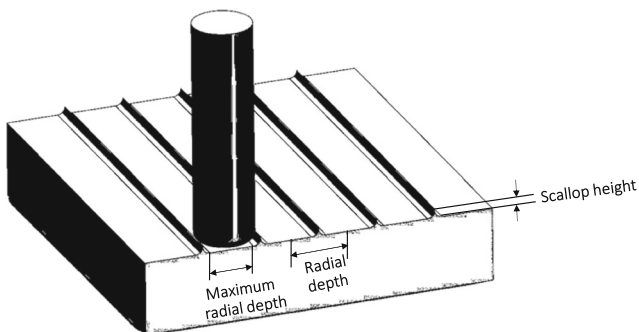
#### 4.1.2 Experimental results

In Fig. 13, the results of the experimental machining process of the mold surface are displayed. The material properties of the stock selected according to [40] and a Haas VF4 5-axis CNC machine-center equipped with a FANUC NC unit was utilized. In order to prevent light reflections due to the flash of the camera so that the final workpiece can be clearly viewed, the machined surface was lightly polished. The resulting specimen has a smooth surface without scallop defects and geometric tolerances which comply with the given specifications (Fig. 14).

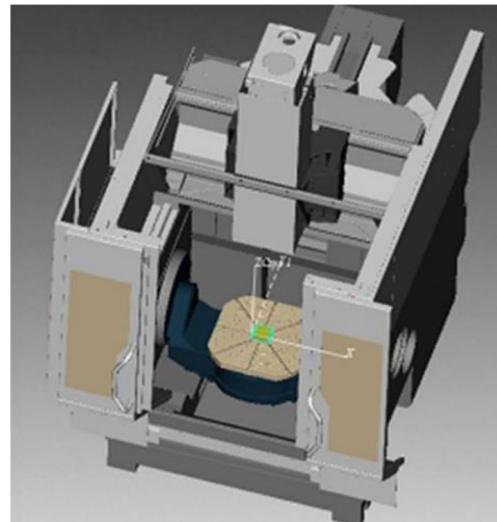
### 4.2 Comparison with genetic algorithm

#### 4.2.1 Simulation results

The tool orientation method was applied to a case study in which a 5-axis cutting tool path is calculated using a Deckel

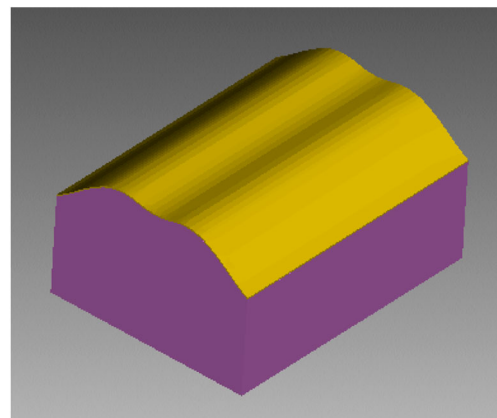


**Fig. 14** Radial depth of Bull nose end mill and the produced scallop geometry



**Fig. 15** Virtual representation of Deckel Maho MH-600C 5-axis CNC machine center in CG-Tech Vericut and setup of the workpiece (after [38])

Maho MH-600C 5-axis CNC machine-center equipped with a TNC-320 Heidenhain® NC unit (see Fig. 15) for a test model shown in Fig. 16. The model contains a NURBS sculptured surface and was developed in Autodesk Inventor Professional 2018 using the exact geometry that is used in [38]. A comparison was made for this model between the results presented in [38] and those produced by the algorithms developed in the present study. In the study of [38], one of the most efficient commercial CAM solvers, namely, CAD/CAM CATIA V5 by Dassault Systèmes [40], was used in combination with a virus-evolutionary genetic algorithm in order to minimize the excess and the overall machining time. The main characteristic of the virus-evolutionary genetic algorithm used in [38] is that it avoids premature convergence by storing solution patterns during the search for the global optimum. For this purpose, the algorithm involves two simultaneously processed populations: the hosts and the viruses. The populations of hosts contain the candidate solutions which aim to minimize one or



**Fig. 16** CAD model with NURBS sculptured surface used for the application of the novel tool positioning method presented in this study



**Table 2** Input parameters and results of two cases of tool point refinement of the PICHC method (present study) and the algorithm used in [38] for the test model shown in Fig. 16

	Tool discretization	Radial depth (mm)	Max. discr. step ( $\times 10^{-3}$ mm)	Max. excess	Machining time	Computational time (h)
Comparison model [38]	–	3.43	8.34	4.59	1.37	5
Present study, case 1	49 points	4	100	3.25	1.05	0.42
Present study, case 2	73 points	4	100	3.24	1.08	0.475

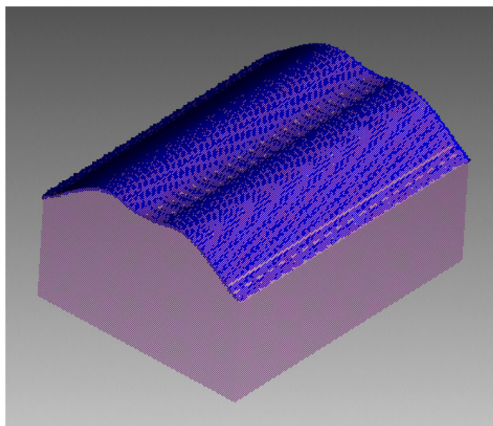
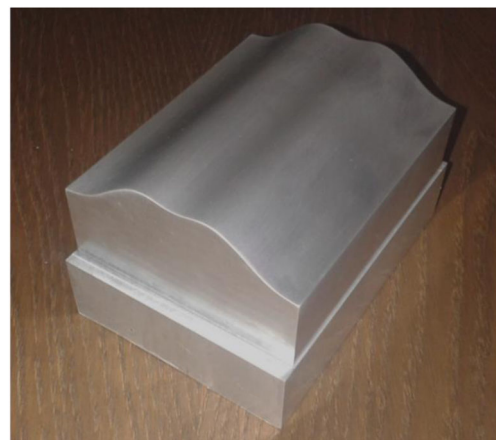
more objective functions, whereas the population of viruses contain those patterns of the host population that offer information to the host population to produce better offspring. Transduction and reverse transcription are the two viral operators that have been used in [38] for the evolution of the virus population. Among the various cases presented in [38], the case with the optimum performance was selected for comparison purposes in this study. The following machining parameters were used:

- Cut tolerance 0.01 mm
- Filleted end mill of diameter 10 mm and cutter corner radius 3 mm
- Radial depth 3.43 mm
- Lead angle is fixed at  $3.208^\circ$
- Tilt angle is fixed at  $7.079^\circ$
- Maximum discretization step along the tool path  $8.339 \cdot 10^{-3}$  mm

It is obvious from Table 2 that while the radial depth and the maximum discretization step used in this study are higher than the corresponding values used in [38], the methodology implemented with the present study yields lower maximum excess and requires lower machining and computation times than the ones achieved in [38]. These results were calculated

in both studies with the simulation of the machining process in CGTech VERICUT. The calculation of the excess in the present study was performed with an accuracy equal to 0.001 units. There is no need for the optimization algorithms to calculate the optimum radial depth (step over), since the algorithm utilized in the present study uses varying lead and tilt angles depending on the local curvature of the surface. The radial depth that is used in the present study is equal to the diameter of the flat part of the tool which is the maximum radial depth that can be used to avoid any undesirable scallops in the final surface (see Fig. 10). The ability of the proposed algorithm to adjust the tool orientation for each CC point separately so that it fits best the local workpiece geometry permits the user to relax the discretization step and decreases computational effort with improved quality.

The Auto-DIFF module of CGTech VERICUT was used to calculate an indicator based on the remaining material after finishing, known as the excess error. The excess error was calculated using the solid comparison method. It is observed that the maximum excess apart from being crucially reduced, is clearly distributed in a more uniform pattern compared to [38], as shown in Fig. 17. A significant reduction in the machining time is also being achieved, which was expected due to the increased discretization step by two orders of magnitude as well as the radial depth employed. Last but not least, a relatively large decrease in the computational time is achieved

**Fig. 17** Simulated cut solid in which the excess error is shown after NC verification conducted in CG-Tech Vericut**Fig. 18** The experimental result of the machined NURBS sculptured surface considered by Fountas et al. [38]



compared to [38], since the optimization process implemented in this study takes less than half an hour, thus accelerating the design process and the production rates. It can be concluded from the above that the optimum tool orientation method proposed in this study is of higher accuracy, without using any optimization algorithms or commercial CAM software. However, simulation-based objective functions such as the one used in [38] could be used in cases where the optimization procedure includes, apart from the geometric characteristics, other machining parameters, such as spindle speed, feed rate, acceleration/deceleration, and jerk.

From the results presented in Table 2, it can be observed that in the CAM software used in [38] the optimum lead and tilt angles that are calculated are fixed for each machining process. This approach cannot exploit the maximum radial depth available. On the contrary, the tool orientation algorithm used in this study for each CC point adjusts the lead and tilt angles so that the maximum radial depth is utilized, leading thus to maximum material removal rate and consequently minimization of the machining time and of the maximum excess.

In the present study two degrees of refinement of the tool discretization have been considered, i.e., the tool is discretized in 49 points and 73 points (Table 2). The purpose of these two separate analyses is to ensure that the mesh refinement of the tool is sufficiently large, so that the results of the algorithm are mesh-independent. And this is the case, indeed, since it is clear from the results presented in Table 2, that there is almost no difference between cases 1 and 2 of the present study. This finding proves that there is no influence due to the discretization of the tool in the results.

#### 4.2.2 Experimental results

In Fig. 18 the results of the experimental machining process of the NURBS sculptured surface are displayed. The material properties of the stock selected according to [38] and a Haas VF4 5-axis CNC machine-center equipped with a Fanuc NC unit was utilized. In order to prevent light reflections due to the flash of the camera so that the final workpiece can be clearly viewed, the machined surface was lightly polished. The resulting specimen seems to have smooth surface without scallop defects in a similar way as in Fig. 13.

## 5 Conclusions

A novel optimum tool positioning algorithm has been developed in this study, which is implemented in part with the Autodesk Inventor Professional 2018 software API and in part with Matlab R2016b and has been verified both numerically by CG-Tech Vericut and experimentally. It has been shown that the algorithm produces results of superior quality while

being robust. The high efficiency of the proposed algorithm is due to the following characteristics:

- The philosophy adopted by the common CAM commercial software is that in order to maintain the cutting error lower than a prespecified tolerance (cutting tolerance), the number of the CC points has to be increased depending on the local curvature of the workpiece surface, which leads to increased computational effort and machining time and decrease in the maximum discretization step. On the other hand, according to the tool orientation algorithm presented in this study, in order to maintain the cutting error lower than the cutting tolerance, instead of increasing the number of CC points along the tool path, the refinement of the tool discretization is increased.
- The option between one-, two-, or three- point contact between the cutting tool and the workpiece is given in order to increase machining performance. Conventional algorithms allow up to two points of tool-workpiece contact (along lead cutting direction), leading thus to suboptimal material removal rate.
- It provides alternative ways to decrease the cutting error (chordal deviation) without necessarily increasing the number of CC points along the tool path. The key idea is the discretization of the surface of the tool and the local contact area of the workpiece, which allows for a relaxed (coarse) distribution of the CC points on the workpiece surface. In this way the computational effort, the machining time and the size of the G code produced are minimized.
- It utilizes an innovative integrated machining optimization environment which is based on a fully parameterized objective function of superior quality, proving that the quality of the result of an optimization procedure depends not only on the optimization algorithm, but also on the proper selection of the objective function. The use of advanced evolutionary strategies and/or other soft computing algorithms does not entail that the result will be of high quality.

The results of the algorithm have been tested with a model from literature. It has been found that it has improved performance compared to commercial CAM software packages in combination with advanced optimization algorithms. The algorithm used in this study is based on advanced and purely numerical procedures, which cover nearly all the theoretically based algorithms and can be applied in machining problems involving workpiece surfaces that do not have cavities or other geometrical irregularities in which the normal vector of the surface intersects the workpiece.

**Acknowledgments** The authors would like to thank Carbie Cut & Co. LP which provided the industrial facilities in which the various experiments presented in this study were performed.

## References

- Vickers GW, Quan KW (1989) Ball-mills versus end-mills for curved surface machining. *ASME J Eng Ind* 111:22–26
- Jensen CG (1993) Analysis and synthesis of multi-axis sculptured surface machining. Dissertation, Purdue University
- Jensen CG, Anderson DC (1993) Accurate tool placement and orientation for finished surface machining. *J Des Manuf* 3(4):251–261
- Jensen CG, Anderson DC, Mullins SH (1993) Scallop elimination based on precise 5-axis tool placement, orientation, and step-over calculations. *Adv Des Automat ASME* 65(2):535–544
- Rao N, Bedi S, Buchal R (1996) Implementation of the principal axis method for machining of complex surfaces. *Int J Adv Manuf Technol* 11:249–257
- Rao N, Ismail F, Bedi S (1997) Tool path planning for five-axis machining using the principal axis method. *Int J Mach Tools Manuf* 37(7):1025–1040
- Wang XC, Ghosh SK, Li YB, Wu XT (1993) Curvature catering—a new approach in manufacture of sculptured surface (part1. Theorem). *J Mater Process Technol* 38(1):159–175
- Wang XC, Li YB, Ghosh SK, Wu XT (1993) Curvature catering—a new approach in manufacture of sculptured surface (part2.methodology). *J Mater Process Technol* 38(1):177–193
- Gray P, Bedi S, Ismail F (2005) Arc-intersect method for 5-axis tool positioning. *Comput Aided Des* 37(7):663–674
- Gray P, Ismail F, Bedi S (2007) Arc-intersect method for  $3\frac{1}{2}\frac{1}{2}$ -axis tool paths on a 5-axis machine. *Int J Mach Tools Manuf* 47(1):182–190
- Gray P, Bedi S, Ismail F (2003) Rolling ball method for 5-axis surface machining. *Comput Aided Des* 35(4):347–357
- Gray P, Ismail F, Bedi S (2004) Graphics-assisted rolling ball method for 5-axis surface machining. *Comput Aided Des* 36(7):653–663
- Hosseinkhani Y, Akbari J, Vafaeseefat A (2007) Penetration elimination method for five-axis CNC machining of sculptured surfaces. *Int J Mach Tools Manuf* 47(10):1625–1635
- Jun CS, Kyungduck C, Lee YS (2003) Optimizing tool orientations for 5-axis machining by configuration space search method. *Comput Aided Des* 35:549–566
- Engeli M, Waldvogel J, Schnider T (2002) Method for processing work pieces by removing material, US patent: US 6:485,236
- Fan WG, Wang XC, Cai YL, Jiang H (2012) Rotary contact method for 5-axis tool positioning. *J Manuf Sci E-T ASME* 134(2):021004.1–021004.6
- Fan WG, Ye PQ, Zhang H, Fang CX, Wang RC (2013) Using rotary contact method for 5-axis convex sculptured surfaces machining. *Int J Adv Manuf Technol* 67:2875–2884
- Li ZQ, Chen WY (2006) A global cutter positioning method for multi-axis machining of sculptured surfaces. *Int J Mach Tools Manuf* 46(12):1428–1434
- Chen ZT, Yue Y, Xu RF (2011) A middle point error control method in strip-width maximization machining. *J Mech Eng* 47(1):117–123 In Chinese
- Xu RF, Chen ZT, Chen WY (2010) Constant scallop tool path planning for 5-axis machining using the end-point error controlling method. *Proc Can Soc Mech Eng Forum*
- Lu J, Cheatham R, Jensen CG, Chen Y, Bowman B (2008) A three-dimensional configuration-space method for 5-axis tessellated surface machining. *Int J Comput Integr Manuf* 21(5):550–568
- Choi BK, Kim DH, Jerard RB (1997) C-space approach to tool-path generation for die and mould machining. *Comput Aided Des* 29(9):657–669
- Morishige K, Takeuchi Y, Kase K (1999) Tool path generation using C-space for 5-axis control machining. *J Manuf Sci Eng, ASME Trans* 121(1):144–149
- Morishige K, Takeuchi Y, Kase K (1997) Collision-free tool path generation using 2-dimensional C-space for 5-axis control machining. *Int J Adv Manuf Technol* 13:393–400
- Lee YS (1997) Admissible tool orientation control of gouging avoidance for 5-axis complex surface machining. *Comput Aided Des* 29(7):507–521
- Duvedi RK, Bedi S, Batish A, Mann S (2015) Numeric implementation of drop and tilt method of 5-axis tool positioning for machining of triangulated surfaces. *Int J Adv Manuf Technol* 78:1677–1690
- Kersting P, Zabel A (2009) Optimizing NC-tool paths for simultaneous five-axis milling based on multi-population multi-objective evolutionary algorithms. *Adv Eng Softw* 40(6):452–463
- Ulker E, Turanalp ME, Halkaci HS (2009) An artificial immune system approach to CNC toolpath generation. *J Intell Manuf* 20:67–77
- Li L, Liu F, Chen B, Li BC (2015) Multi-objective optimization of cutting parameters in sculptured parts machining based on neural network. *J Intell Manuf* 26(5):891–898
- Raja SB, Baskar N (2012) Application of particle swarm optimization technique for achieving desired milled surface roughness in minimum machining time. *Expert Syst Appl* 39(5):5982–5989
- Kim T (2015) Simulated annealing approach with an exhaustive greedy search for the optimal machining region decomposition in CNC roughing tool path generation. *Int J Precis Eng Manuf* 16(2):399–440
- Yildiz AR (2012) A comparative study of population-based optimization algorithms for turning operations. *Inf Sci* 210:81–88
- Yildiz BS, Yildiz AR (2018) Comparison of grey wolf, whale, water cycle, ant lion and sine-cosine algorithms for the optimization of a vehicle connecting rod. *Mater Test* 60(3):311–315
- Yildiz BS (2017) A comparative investigation of eight recent population-based optimization algorithms for mechanical and structural design problems. *Int J Veh Des* 73(1/2/3):208–218
- Yildiz AR (2013) Hybrid Taguchi-differential evolution algorithm for optimization of multi-pass turning operations. *Appl Soft Comput* 13:1433–1439
- Yildiz AR (2013) A new hybrid differential evolution algorithm for the selection of optimal machining parameters in milling operations. *Appl Soft Comput* 13:1561–1566
- Yildiz BS, Yildiz AR (2017) Moth-flame optimization algorithm to determine optimal machining parameters in manufacturing processes. *Mater Test* 59(5):425–429
- Fountas NA, Vaxevanidis NM, Stergiou CI, Benhadj R (2016) A virus-evolutionary multi-objective intelligent tool path optimization methodology for 5-axis sculptured surface CNC machining. *Eng Comput* 33(3)
- CG-Tech Vericut® Ltd., Version 8.1.1, registered in England No: 3579362 VAT No: 717604145
- Chen Z, Li S, Gan Z, Zhu Y (2017) A highly efficient and convergent optimization method for multipoint tool orientation in five-axis 991 machining. *Int J Adv Manuf Technol* 93(5–8):2711–2722

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