CNC double spiral tool-path generation based on parametric surface mapping

Bo Zhou a, b, Jibin Zhao a, *, Lun Li a

a Shenyang Institute of Automation, Chinese Academy of Sciences, Liaoning, 110016, China
b University of Chinese Academy of Sciences, Beijing, 100049, China

HIGHLIGHTS
- We study a new double spiral tool-path generation algorithm for 5-axis HSM.
- We call a smoothness optimization method during solving PDEs to smooth curves.
- Machining parameters (e.g., path interval and step length) can be guaranteed.
- Mapping rules are based on the same ranges of NURBS curve and parametric domain.
- Our tool-paths have a self-complementary structure and can be suitably linked.

ABSTRACT
High-speed machining (HSM) has been an important method for machining complex parametric surface. Tool-path planning for HSM has a significant impact on processing efficiency and surface quality. A new double spiral tool-path generation algorithm for HSM is proposed in this paper. First, the isothermal lines which satisfy the machining parameters in the mapping parametric domain are computed by means of constructing a thermal conductivity model and solving partial differential equations (PDEs). Furthermore, a smoothness optimization method is proposed to improve the smoothness of the isothermal lines and avoid taking up too much memory. Then, the mapping rules are constructed and the trajectory is planned out in the standard parametric domain in order to generate double spiral trajectory in the corresponding parametric domain. Finally, the trajectory is mapped onto the parametric surface to obtain the tool-paths, and the tool-paths linking method is planned for complex multi-domains. This method can realize the precision milling of complicated parametric surface without tool retraction, and meanwhile it improves the uniformity of the tool-paths and machining efficiency. Our method has been experimented in several simulations and validated successfully through the actual machining of a complicated pocket. The results indicate that this method is superior to other existing machining methods, and it can realize HSM of complex-shaped pocket based on parametric surface.

1. Introduction
Pocket components with aspheric surfaces have been extensively applied in different industrial sectors, including aerospace, space exploration, mold, powertrain system and so on [1]. At present, HSM has emerged as a key technology in machining components with aspheric surfaces and other manufacturing applications, which possesses obvious superiority in complicated surface machining with much higher rotational speed and feedrate of the spindle than traditional milling methods. So HSM has become one of the most widely adopted approaches to increase machining efficiency, improve machining quality and reduce cost [2].

HSM has not only brought higher requirements for the lathe, clamp, cutter, etc., but also proposed further requests for the topology geometric shape and dynamics performance of the tool-path. Spiral tool-path is an ideal tool-path planning method for HSM. Generally speaking, splinterly machining trajectories rarely occur during machining, and furthermore, load fluctuation of the cutter can be effectively restrained and the times of cutter lifting can be reduced so as to achieve higher machining quality. In order to effectively avoid sudden direction changes of the velocity...
vectors, tool-path with smooth transitions at sharp corners will be needed. In all the transitions between feed and retract, etc., motions need to be as smooth as possible. Since the spiral tool-path starts on the outside border and stops in the center (or vice versa), undesired artifacts may occur in starting or stopping operation. This effect gets even more intensified when there is a high curvature at the start or the end of the tool-path, especially inside of the spiral [3].

Generally, tool load fluctuations, tool wear and the number of tool retractions can be reduced through generating a smooth continuous spiral tool-path, which plays a particularly important role for HMS.

2. Related work

Existing algorithms for creating spiral tool-paths can be subdivided into several classes: The first class, called the class of “Offset” algorithms, creates offset curves on the surface and connects the offset curves to generate a spiral, which has been studied deeply, including several typical algorithms: Pateloup [4], et al. offset the external contour line of the planar pocket part by means of iso-parametric method and made B-Spline interpolation, respectively. Smooth connection can be realized on each corner for every offset trajectory to generate the further spiral trajectory. Li [5], et al. presented an efficient 2D invalid loops removal algorithm to obtain valid cutter contact (CC) contours and cutter location (CL) contours. Zhu [6], Chen [7], Lee [8] and Hauth [3], et al. proposed contour offset approach to spiral tool-path generation with constant scallop height method, respectively, in order to minimize the processing time, as redundant machining in iso-parametric and iso-planar methods were minimized since the scallop height was kept constant. Then, the spiral tool-paths were similarly generated as a set of the diagonals between the offset curves. But all these above-mentioned algorithms needed to make self-intersection type judgments and reduce operations. Except for Pateloup [4] and Hauth’s [3] algorithms, other algorithms all come with sharp corners, which means these spiral tool-paths were not smooth enough and sudden direction changes of the velocity vectors were more likely to occur. So these methods were not available for general actual machining, let alone HMS.

The second class, called the class of “Mapping” algorithms, takes an exact Archimedean spiral or space filling curves, like the Hilbert curve [9], and projects it to the workpieces surface. Several algorithms are typical within this class: Ren [10], et al. proposed reparameterization-based tool-path generation methods to generate an optimal mean value coordinate mapping between the designed surface and a specified planar circular domain with relatively less distortion both in length and in angle, and constructed spiral guide path with maximum path interval in the circular domain. Sun [11], et al. proposed a new tool-path regeneration approach for polyhedral machining. The method used a harmonic map to parameterize the triangular meshes. Both above-mentioned methods were based on mesh surfaces, which were not available for parametric surface machining. As for parametric surface machining, boundary interpolation methods and PDE methods are two major classes of techniques for the parameterization of a 2D parametric surface or trimmed surface, which are exactly the same as a 2D numerical grid generation to the same given boundaries [12]. The first type cannot be applied to complex boundaries-shape surface machining, because the skew, crowdingness, or over-crossing problems are not inevitable if the given boundaries have relatively high degree of irregularity [13]. The second type generates the contours by numerically solving a system of PDEs, such as Laplace and Poisson equations. However, the Poisson equation need to be solved via a trial and error process [14], it is not suitable for automatic tool-path generation. Oulee [12] adopted a Laplace-based equation method which based on the 2D Laplace parameterization of pocket contours and the redistribution of the original Laplace iso-parametrics, continuous tool-paths are generated. However, the method was not suitable for 3D parametric surface machining.

Currently, there have already been several novel algorithms based on the study of parametric surface: Held [15] introduced a new algorithm for generating a spiral tool-path which was generated by interpolating growing disks placed on the medial axis of the pocket. The pocket might be an arbitrary simply-connected 2D and was not considered machining parameters, so it was not available for actual machining. Bieterman [16] and Banerjee [17] aimed at solving the boundary value problem with MATLAB™ PDE Toolbox and obtaining a series of contours, respectively. However, the spiral tool-path generation method based on solving ellipse PDEs has difficulty in controlling the distance between two adjacent contours. This meant that it was hard to guarantee machining parameters during actual machining and only applicable for 3-axis plane machining of flat areas. Lu [18], et al. proposed ‘feature-based’ approach that based on the equipotential lines principle with the consideration of scallop height. The contours were generated with calculated density and these contours were then interpolated to form spiral or zig-zag tool path. However, this method was based on detected the critical edges i.e., top and bottom edges, and generated contours between them, so it might not be suitable for complex surfaces machining.

To solve the above-mentioned tool-path generation problems, the spiral tool-path must be one intersection-free curve, particularly with a starting and an ending point both located on the boundary of the workpiece. This paper presents a new double spiral tool-path generation algorithm without offset algorithm and self-intersection. The contours with smooth corners are obtained by solving PDEs and will further ensure the smoothness of the spiral tool-path. With the aim of reconciling two conflicting objectives: precision and efficiency for HSM. For a given precision tolerance (i.e., the scallop height and chord deviation), a desirable tool path is always supposed to be as smooth and short as possible. In this paper, the smoothness of tool path is measured by its curvature analysis of the CL path in the 3D space. If the tool path is smooth enough, there is less repeated acceleration/deceleration, which makes it possible to maintain a high federate [19]. Meanwhile, a shorter tool path means less machining time-consuming. The machining quality can be controlled by the following two major factors: the machining path interval is determined by means of the scallop-height method based on the selection of initial isothermal lines; the step length is determined by means of the chord deviation method. Both are common and efficient ways in actual machining.

The remainder of this paper is organized as follows. In Section 3, the principle, characteristics and overview are presented. In Section 4, a dichotomy method is presented to determine the initial isothermal lines. In Section 5, the mapping rules between standard parametric domain and parametric domain are established and the spiral tool-path is generated. In Section 6, generation and linking of the double spiral tool-paths are conducted in arbitrary free form surface. Simulation and experiments are conducted in Section 7, and the conclusions are given in Section 8.

3. Overview

The steps of the algorithm are as shown in Fig. 1: ① planning out the initial tool-paths: a set of scallop-height tool-paths are planned out on the parametric surface. ② Calculating the initial isothermal lines: the thermal conductivity model is constructed in the parametric domain formed by the surface projection maximum contour and boundary conditions. Then, the isothermal lines are generated.
It can be solved by PDEs. The heat equation is a parabolic PDE that describes the distribution of heat (or variation in temperature) in a given region over time.

$$\frac{\partial T}{\partial t} = K_0 \left( \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$  \hspace{1cm} (1)$$

where, \( K_0 = \lambda/\rho C \), \( \lambda \) represents thermal conductivity, other parameters are as follows: \( \rho \) represents density, \( C \) represents specific heat capacity (heat capacity per unit of volume), which is defined as the product value of density and specific heat.

The numerically computing solution of the 2D heat equation is based on FEM. In mathematics, it is a numerical technique for finding approximate solutions of boundary value problems for PDEs. A brief description of each of these steps is as follows.

First, we divide the complex-shaped domain into non-overlapping elements (e.g., triangles, quadrilaterals). In this paper, the finite element discretization is considered as a typical linear triangular element with 3 nodes ‘i’, ‘j’ and ‘k’ with the corresponding temperature being denoted by \( T_i, T_j \) and \( T_k \) respectively. The temperature distribution can be written as

$$T = N_iT_i + N_jT_j + N_kT_k$$  \hspace{1cm} (2)$$

where \( N_i, N_j \) and \( N_k \) are called shape functions.

Then, the matrix equations that express the properties of the individual elements by forming an element Left Hand Side (LHS) matrix \([K]_e\) and load vector \([f]_e\) can be determined, where the subscript \( e \) represents an element.

To find the properties of the entire system, all the matrix equations of each element must be combined in an appropriate way, and the resulting matrix represents the behavior of the entire solution domain. The boundary conditions must be incorporated after the assemblage of the individual element contributions. The final matrix form of the finite element equations is

$$[K] [T] = [f]$$  \hspace{1cm} (3)$$

where \([K]\) is the global LHS matrix, which is the assemblage of the individual element LHS matrices, \([f]\) is the global load vector, which is the assemblage of the individual element load vectors, and \([T]\) is the global unknown vector.

The resulting set of algebraic Eqs. (3) can be solved to obtain the nodal values of the temperature field variable. Using the nodal temperature values, the temperature at any other location within an element can be determined by using linear interpolation. Furthermore, the calculation of the temperature at any arbitrary location can also be achieved. The line joining a sequence of points with the same temperature will be the isothermal line. The detailed step-by-step procedures can be referred to the related literature [20].

In Fig. 2, let the temperature value of outer boundaries have the same value and without other heat sources located inside the domain, we draw isothermal lines of approximately equal temperature in order to observe the thermal energy invading the interior of this domain.

There are some common rotational parts with circle structures, such as cylindrical and spherical cap. Herein, we introduce the temperature distribution method for the disc. It is supposed that one thin disc which possesses the designated temperature on the boundary. That is to solve the symmetric solution of circle plate using separation of variables method [21]. It is a technique commonly used to solve first-order ordinary differential equations which rearrange the equation to be solved such that all terms involving the dependent variable appear on one side of the equation, and all terms involving the independent variable appear on the other. This method will be used for machining rotational parts in ensuing chapters.
4.2. Contours smoothness optimization method

As illustrated in Fig. 2(b), there still exist some problems that should by no means be neglected and should be further optimized in order to give some recommendations for actual machining:

1. the closer the contour near to the outer boundaries, the smoother it can be; Conversely, the smoothness of contours is relatively low near to the innermost contour, some contours are even with sharp corners;
2. there inevitably exists a blank domain in the innermost contour, the smaller amount of contours are generated, the larger the blank domain will be;
3. the distribution of contours is uneven.

These problems will result in a series of problems such as: unevenly distribution and poor continuity of contours. For the quality of spiral tool-path is determined by the smoothness of the contours directly, we must optimize contours generation method for better performance.

For the first problem, the possible reasons are analyzed in different aspects and the main reason is found out, the method for smoothing the contours will be given in this chapter. We obtain the contours by FEM, so we first implement the mesh refinement method. Fig. 3 shows a comparison of unit square where mesh refinement is implemented for 5 and 7 times, respectively. The smoothness of simulation result can prove the utility effect of the mesh refinement technique through the contrast test.

However, there is very limited support for smoothness and a lot of computing time will be consumed if we continue operating mesh refinement, not to mention “out of memory” errors may be detected during mesh refinement operation. Other methods such as partial thinning mesh can improve the quality of mesh, but the memory-consuming problem still exists because it need to implement remeshed in these partial domains. If we only generate smooth contours by these methods, we will encounter a typical bottleneck related to contemporary computer architecture. It is caused by the fact that the speed of the CPU has become faster and faster over the last years, whereas the memory could not keep pace with that development [22]. Of course, this conclusion sounds to some extent like a commonplace. However, the truth is that the problem of organizing the data in adequate memory structures in order to avoid the described bottleneck is not trivial. Some scholars proposed the method so-called “cache hit rate” which is a technique for solving a specific type of PDEs to avoid taking up too much...
memory during calculating [22,23]. If we compare simple pattern to pattern with more complex outer boundaries, we find the latter will be a very more time- and memory-consuming problem. Take the pattern in Fig. 2 for example, based on the same hardware configuration, the domain can be implemented mesh refinement for only 5 times, while the simple domain like unit square can be implemented mesh refinement for 8 times. According to the problems mentioned above, we propose an algorithm which is memory saving. Under the same number of remesh times, we can generate smoother contours with high continuity.

It is a major point in every existing fairing algorithm to find suitable smoothness criteria, as most “definitions” of smoothness are subjective and mathematically not unique [24,25]. Therefore, we are interested only in practical concepts which are useful for the problem considered here. One of the basic tasks that our method can be implemented mesh refinement for 8 times. Aiming to confirmatory tests: If \(E_0 = 1.15E_\alpha < 0\) is valid or not. If \(E_0\) exceeds 1.15\(E_\alpha\), we will retain the contours on the outside of the current contour and construct the closest inside one as the new outer boundaries, based on which the new contours are generated in limited quantities as the number of reserved ones to be subtracted from total ones; Implementing recursively the above steps until all the curvature value of discrete points not to exceed the curvature threshold value. It is notable that, as mentioned above, the contours which near to the outer boundaries are smoother. These contours need not be optimized. Generally, after each time of implementing optimized operation, we can reserve directly 15%-20% of total amount of contours without implementing smoothness optimization. As illustrated in Fig. 4(b), under the conditions of the remesh times are both equal to 4 and numbers of generated contours are just the same, the drawing of partial enlargement clear contrast between the original contours and optimized contours: the smoothness of contours has been greatly improved.

However, from the drawing of partial enlargement, we can find some transition segments in the portion of blue contours are somewhat long. These transition segments are located at narrow strips of the pattern which is the so-called trunk part. Although the curvature values of these discrete points are not exceeding the preset value, the distance between the adjacent points may result in exceeding chord errors and affect the generation of spiral tool path. So it is necessary to implement length judgment in order to further improve the smoothness of the contours at narrow strips.

After setting the length threshold value, we implement the similar algorithm mentioned above. Through comparative figures of Figs. 4(b) and 5, the before- and after-contours can demonstrate the availability of this algorithm: it can rectify and reform the blue contours into 3 sets of contours which are in cyan–blue, green and blue, respectively. Among them, the blue contours are just the same as the original partial of blue contours, on which the length of adjacent points is not exceeding the setting of length threshold value; while the smoothness of newly generated cyan–blue and green contours is much smoother than the original blue contours (see Figs. 4(d) and 5(b)).

Let the definition of “C Key Loop” be the contour which is determined by curvature threshold value, while let the definition of “L Key Loop” be the contour which is determined by length threshold value. An algorithm flowchart is shown in Fig. 6 and is summarized as follows:

(1) let the most lateral contour of off-reserve contours be the “C Key Loop”, set the boundaries condition and generate contours with the total number of \(N_1\); set the curvature threshold value \(C = 1.15E_\alpha\) as the first judgment criterion; In an inward direction, each contour is individually checked whether the curvature energy exceeds their own \(C_e\) or not:

(1) if \(mth\) contour is the first one whose original curvature energy \(E_0\) exceeds the setting value \(C_e\); let the \((m – 1)th\) contour be the new “C Key Loop” and reserve the contours with the quantities \(N_1 = m – 1\) whose original curvature energy does not exceed their respective value \(C_e\); clears all meshes from the vector and removes useless contours entirely.

In this paper, we use \(\alpha = 1.15 \text{ and } i = 4\). That is, we let \(1.15E_\alpha\) be the curvature threshold value and judge whether \(E_0 = 1.15E_\alpha < 0\) is valid or not. If \(E_0\) exceeds 1.15\(E_\alpha\), we will retain the contours on the outside of the current contour and construct the closest inside one as the new outer boundaries, based on which the new contours are generated in limited quantities as the number of reserved ones to be subtracted from total ones; Implementing recursively the above steps until all the curvature value of discrete points not to exceed the curvature threshold value. It is notable that, as mentioned above, the contours which near to the outer boundaries are smoother. These contours need not be optimized. Generally, after each time of implementing optimized operation, we can reserve directly 15%-20% of total amount of contours without implementing smoothness optimization. As illustrated in Fig. 4(b), under the conditions of the remesh times are both equal to 4 and numbers of generated contours are just the same, the drawing of partial enlargement clear contrast between the original contours and optimized contours: the smoothness of contours has been greatly improved.

After setting the length threshold value, we implement the similar algorithm mentioned above. Through comparative figures of Figs. 4(b) and 5, the before- and after-contours can demonstrate the availability of this algorithm: it can rectify and reform the blue contours into 3 sets of contours which are in cyan–blue, green and blue, respectively. Among them, the blue contours are just the same as the original partial of blue contours, on which the length of adjacent points is not exceeding the setting of length threshold value; while the smoothness of newly generated cyan–blue and green contours is much smoother than the original blue contours (see Figs. 4(d) and 5(b)).

Let the definition of “C Key Loop” be the contour which is determined by curvature threshold value, while let the definition of “L Key Loop” be the contour which is determined by length threshold value. An algorithm flowchart is shown in Fig. 6 and is summarized as follows:

(1) let the most lateral contour of off-reserve contours be the “C Key Loop”, set the boundaries condition and generate contours with the total number of \(N_1\); set the curvature threshold value \(C = 1.15E_\alpha\) as the first judgment criterion; In an inward direction, each contour is individually checked whether the curvature energy exceeds their own \(C_e\) or not:

(1) if \(mth\) contour is the first one whose original curvature energy \(E_0\) exceeds the setting value \(C_e\); let the \((m – 1)th\) contour be the new “C Key Loop” and reserve the contours with the quantities \(N_1 = m – 1\) whose original curvature energy does not exceed their respective value \(C_e\); clears all meshes from the vector and removes useless contours entirely.

\[
E_0 - \alpha E_i < 0
\]

where \(E_i\) is the energy value after \(i = 3\) iterations, and \(\alpha = 1.10 \text{–} 1.20\) is a rational coefficient that is obtained empirically according to confirmatory tests: If \(\alpha\) is relatively small, it will result in relatively large number of “C Key Loops” and many of them are original ‘visually pleasing’ contours; while \(\alpha\) is relatively large, there may remain contours with sharp corners. The smoothness criterion can efficiently determine sharp corners and reduce energy value, so the iteration number should be small.

As mentioned above, contours with low smoothness are with sharp corners, and the sharp corners in the curvature plot correspond to large discontinuities in the derivative of curvature. In this paper, the smoothness criterion for each contour is computed by the method in [25], including local and global smoothness. The smoothness algorithm we determined can automatically identify the point at which the contour to be smoothed (local smoothness). The fairing process will be organized iteratively by modifying only one point of the given contour in each iteration. Modifying locally the contour, the algorithm gradually yields a curve with improved smoothness. We can conclude that the energy values and iteration times are the most significant characteristics of a smoothed curve. Therefore, our initial idea is to evaluate the decreasing rapidity of the energy values along with the increase of iterations.

Now to get practical and suitable smoothness criteria in the case of judging contours smoothness, we proceeded as follows:

0. compute the initial energy value \(E_0\) of current contour.
1. call smoothness criterion and modify one point, compute the new energy value \(E_i\).
2. justify whether the relationship \(\frac{E_{i-1} - E_i}{E_i} \leq \varepsilon\) can be established.
3. if the above relationship has been achieved exit else replace the changed values and goto step 1, where \(i\) is number of the iterations, \(\varepsilon\) is the desired decreasing rapidity of the energy values along with the increase of iterations.

However, this method may not always be necessary to work well in all situations. Although the energy values are always strictly monotone decreasing, the decreasing rapidity of the energy values along with the increase of iterations may not always be strictly monotone decreasing. It is summarized that the decreasing value of the energy values must follow the global smoothness law: for curve with low smoothness, \(E\) may drastically change during the first few iterations. Conversely, if the original curve is smooth enough, the first few changes of \(E\) occur slowly and are constrained within relatively tight bounds. Therefore, we use an estimated value that is given by empirically testing. When the energy value of the initial contours satisfies the following condition, these contours would be classified as “C Key Loops”:

\[
E_0 - \alpha E_i < 0
\]
Fig. 4. Comparison chart of before and after optimized operations. (a) Contours before optimized operations; (b) contours after optimized operations; (c) partial enlargement of original contours; (d) partial enlargement of optimized contours. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 5. Further optimized operations. (a) Contours after further optimized operations; (b) partial enlargement of further optimized contours. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(2) if all the curvature energies are below their own $C_1$, reserve these contours directly. Go to step 3.

(2) set the boundaries condition and generate contours with the total number of $N - N_1$. The process described above can be executed and judged recursively until a set of new contours with quantities $N$ are obtained. During this process, all of the “C Key Loop” will be found individually. The contours which cannot meet the first judgment criterion of smoothness will be substitutes by the new generated ones which are sufficiently smooth.

As illustrated in Fig. 6, after determining the $(i - 1)$th “C Key Loop” and generating corresponding new contours, red solid line indicates the first contour which cannot meet the smoothness demand. The $i$th “C Key Loop” is the first black solid line outside it.

(3) To furtherly improve the smoothness of contours at narrow strips, set a length threshold value $d_0$ as the second judgment criterion and implement the similar algorithm in step 2 to find the so-called “L Key Loop”. In order to accurately describe this step, we combine the description with Fig. 7: take $k$th
and (k + 1)th “C Key Loop” for example, there are contours with quantities $N_k$ between them; From the exterior toward the interior, every contour is individually checked whether the length of adjacent points exceeds value $d_e$ or not:

1. if $(m')$th contour is the first one on which there exist adjacent points whose length values exceed the setting value $d_e$; let the $(m - 1)$th contour be the first “L Key Loop” and reserve the contours with the quantities $N_{m} = m' - 1$ on which all the length values do not exceed the $d_e$; similarly, clears all meshes from the vector and removes useless contours entirely.

2. if all the length values are below the value $d_e$, reserve these contours directly. Go to step 5.

4. let the “L Key Loop” and (k + 1)th “C Key Loop” be the inner and outer boundaries, or vice versa, respectively; set the inner and outer boundaries conditions with different temperature values and generate contours with the total number of $N_k - (m' - 1)$; The process described above can be executed and judged recursively until all the contours which are located at narrow strips can meet the length judgment criterion of smoothness. During this process, all of the “L Key Loop” will be found individually. The contours at narrow strips which cannot meet the second judgment criterion of smoothness will be substitutes by the new generated ones which are sufficiently smooth.

5. Store the data of the points of each reserved contours, respectively.

As for the second and third problems, we will propose an algorithm in the next chapter.

5. Tool-path generation

The determinations of step length and path interval are critical to tool-path generation, because they will determine the machining-quality and efficiency [26]. After the contour of the surface projection domain is determined, next step is to select the isothermal lines in the parametric domain and construct the mapping rules between standard parametric domain and parametric domain.

5.1. Isothermal lines selection in parametric domain

In this paper, the machining path interval is determined by means of the scallop-height method. The scallop height is defined as the maximum allowable height of the crests of material left uncut after machining. Conventionally, the algorithm begins with the boundary contour offset procedure along the sculptured surface being machined [8]. Afterwards methods are mainly concentrated on improving the computing efficiency [27,28] and accuracy [29–31]. Special attention should be paid that the algorithm described in this paper is also applicable to other types of cutters such as flat-end cutter [32]. Only the following two machining parameters need to be changed in the corresponding equations: the scallop-height $h$ and path interval $L$ [33].

The tool-path interval $L$ is the distance between the two adjacent paths. If $r \gg h, L$ can be replaced with path interval $L$. $L$ on a CL point and the $R_c$ of a CL surface along the normal direction to a tool-path for machining can be expressed as

$$L = \sqrt{8h^2 - \frac{R_c}{r} + r} \quad \text{or} \quad L = \sqrt{8h^2 - \frac{R_c}{r} - r}$$

where $R_c$ is the radius of normal curvature along path interval direction, $h$ is the scallop height and $r$ donates the ball-end cutter radius. The former expression is for convex surfaces and the latter is for concave ones.

According to this formula, we can get the relationship between scallop height $h$ and path interval $L$: $h \propto L^2$. Under ideal conditions of spiral $B$ can divide spiral $A$ equally, the path interval $(L_1)$ between Tool path $A$ and Tool path $B$ is approximately equal to $L/2$. So is the case with Tool path $C$ and Tool path $B$, the path interval $(L_2)$ is also approximately equal to $L/2$. Therefore, after implementing double spiral machining, the ideal scallop height $h_0$ should be approximately equal to $h/4 (\frac{h_0}{h} = \frac{1}{4})$. In this paper, we have to make the statement that the processing requirements of preset scallop height is $h_0$. The set of path interval points $P_0', P_1', \ldots, P_n'$ which are obtained according to scallop-height method are determined by $h$.

5.1.1. Determinations of guide line and initial isothermal lines

First, the scallop–height method is used to get a set of machining trajectory $A$, among which $A_i$ is one of the elements, where $A_i \in A (1 \leq i \leq n - 1)$. Then a guide line $L$ is determined on the surface, which intersects each machining trajectory $A_i$ to obtain a set of intersections $P_0, P_1, \ldots, P_n$ (and $P_0$ and $P_n$ are, respectively, the two endpoints of $L$). The guide line $L$ and intersection points $P_0, P_1, \ldots, P_n$ are mapped onto the parametric domain to obtain the corresponding mapping line $L'$ and a set of the mapping points $P_0', P_1', \ldots, P_n'$ (see Fig. 8(b)).

The principles for determining the guide line $L$ are to ensure the distribution of intersections $P_0', P_1', \ldots, P_n'$ as even as possible and reduce the processing repeat as much as possible to avoid leaving unprocessed domains reside simultaneously. The densities of isothermal lines in the domain of obtuse and acute boundaries are sparse and dense, respectively. So the guide line $L$ shall be the one which start at the midpoint of the longer boundary and end in the
boundaries, respectively, and let the temperature interval be \( P \), which means that the temperature value of mapping point \( P \) is assumed to be maximum temperature \( T_{\text{max}} \). Let the temperature interval be \( T' \) °C, then \( T_{\text{max}}/T' + 1 \) sets of isothermal lines are generated including outer boundaries. There will always be two adjacent isothermal lines \( r'(k-1) \) and \( r'(k+1) \) existing and the interval between them is \( d'(i) \) (equal to \( |P'(k+1)P'(k-1)| \)) along the direction of line segment \( P'(k+1)P'(k-1) \). Their temperature values are \( T'_i(k-1) \) and \( T'_i(k+1) \), respectively, and the mapping point \( P'_i \) is just located between them, which means that the temperature value of mapping point \( P'_i \) is just between the range of \( T'_i(k-1) \leq T'_i(i) \leq T'_i(k+1) \), \( T'_i(k+1) - T'_i(k-1) = T' \) °C; \( \therefore \) Judging the relationship between \( d'_i \) and \( \varepsilon_d \):

If (1) \( d'_i > \varepsilon_d \): Let \( r'_i(k+1) \) and \( r'_i(k-1) \) be the outer and inner boundaries, respectively, and let the temperature interval be \( T'/2 \). By implementing dichotomy method, the above-mentioned interval will be gradually shortened until the condition \( d'_i \leq \varepsilon_d \) can be satisfied. It is assumed that the finally determined two isothermal lines are \( r'_i(k) \) and \( r'_i(k+1) \) and their corresponding temperature values are \( T'_i(k) \) and \( T'_i(k+1) \), respectively. Moreover, the conditions \( T'_i(k) \leq T'_i(i) \leq T'_i(k+1) \), \( T'_i(k+1) - T'_i(k) = T'/(w \times 2) \) °C should be satisfied, where \( w \) is the times of recursive computation.

\( r'_i(k) \) and \( r'_i(k+1) \) intersect the line segment \( P'(k+1)P'(k-1) \), respectively, at points \( P'_i(k) \) and \( P'_i(k+1) \), then the temperature value on mapping point \( P'_i(k) \) can be obtained by the following proportion Eq. (4):

\[
T'_{(i)} = |P'_{(i+1)}P'_{(i)}| \times T'_i + \left( 1 - \frac{|P'_{(i+1)}P'_{(i)}|}{|P'_{(i+1)}P'_{(i)}|} \right) \times T'_{(i+1)}. \tag{4}
\]

If (2) \( d'_i \leq \varepsilon_d \), there is no need to make recursive computation. The temperature value on mapping point \( P'_i(k) \) can be calculated directly by the following proportion Eq. (5):

\[
T'_{(i)} = |P'_{(k+1)}P'_{(k)}| \times T'_k + \left( 1 - \frac{|P'_{(k+1)}P'_{(k)}|}{|P'_{(k+1)}P'_{(k)}|} \right) \times T'_{(k+1)}. \tag{5}
\]

The corresponding isothermal line with temperature value \( T'_i \) is obtained according to the algorithm in this chapter, as illustrated in Fig. 9.

Special attention should be paid if some mapping points still cannot be obtained within the isothermal line with the value 0 °C by means of the difference calculation method described in the previous chapter, then let the isothermal line with the value 0 °C be the outside boundaries to obtain isothermal lines with negative temperature values. The obtaining method of the negative temperature values to the corresponding mapping points is similar to the positive values, which will not be described in detail.

5.1.2. Determination of initial interpolated isothermal lines

First, the corresponding isothermal line \( r'(i) \) is mapped onto the parametric surface to obtain the mapping line \( r'_{(i)} \); for each \( r'_{(i)} \), the step length is calculated by means of the equi-chord deviation method, then the corresponding points \( P'_{(i)}, P'^1_{(i)}, \ldots, P'^m_{(i)} \) are obtained. The variable \( m \) stands for the total number of the
corresponding points. Then these points are mapped onto the isothermal line $r_{i0}$. Let these mapping points be $p_{i0}^1, p_{i0}^2, \ldots, p_{i0}^m$.

Machining error in multi-axis CNC milling operation is due to the approximation of surface curves by linear tool motions [34]. If the curve is to be approximated by a line segment (a machining segment, namely chord) the chord-height error is defined as the maximum distance from the curve to the chord. The equi-chord height error method for step length estimation is adopted to satisfy the precision request at the discrete points which located on the tool-path. The current CC point is obtained by means of the uneven recursive interpolation method, based on which, the next CC point can be determined. This algorithm is simple, reliable and highly efficient. It processes the curvature adaptability along the machining direction for the cutting step length. The method can be referred to the related literature for details [35].

According to the method in previous chapter, the adjacent isothermal lines $r_{i0}$, $r_{i0}+1$, and $r_{i0}-1$ are taken as the boundaries of the temperature field, respectively. The interpolated number is the larger one of the mapping points on the two isothermal lines, respectively (see Fig. 10). If a set of isothermal lines (quantity $m$) are to be interpolated between isothermal lines $r_{i0}$ and $r_{i0}+1$, the temperature values $T_{i0}$ and $T_{i0+1}$ (corresponding to isothermal lines $r_{i0}$ and $r_{i0}+1$), respectively.

$$T_{i0}^k = T_{i0} - \frac{T_{i0} - T_{i0+1}}{m - 1} \times k$$

where $1 \leq k \leq m - 1$.

5.2. Parametric domain mapping

In this paper, a simplified parametric mapping method is proposed. As a result, a re-standardized parametric domain will match with the mapping boundaries of the parametric surface. The method enables the tool-path and the trimmed surface matched strictly, as a result of which, the redundancy of machining tool-paths is reduced so as to achieve consistent surface quality more efficiently.

5.2.1. Constructing mapping rules between the parametric curve in the parametric domain and the line segment of "Unit Length" in the standard parametric domain

In this paper, positive $X$ direction of the workpiece coordinate system is assumed to be positive $u$ parameter direction and positive $Y$ direction is assumed to be positive $v$ parameter direction. The correspondences between $s/t$ direction of standard parametric domain and $u/v$ direction of parametric domain can be similarly constructed. The initial parametric domain is indicated in Fig. 11(a). The parametric mapping curve in the $u\sim v$ parametric domain is obtained by projecting the parametric surface onto the $UV$ plane, as indicated in Fig. 11(b). By establishing mapping rules, this paper is to make the parametric domain which is obtained by using Boolean operation (see Fig. 11(b)) corresponding to the standard parametric domain (see Fig. 11(c)). The corresponding isothermal lines will be planned out in the parametric domain, as indicated in Fig. 11(d).

Firstly, we give the method for determining the relation between a straight line segment of length 1 in standard parametric domain and an approximation contour in the parametric domain:

As illustrated in Fig. 11(c), each straight line is kept at a $t$ constant value and the $s$ value is varied in the range between 0 and 1. While in Fig. 11(d), each contour is relatively complex and consisted of a set of short straight line segments. The quantity of the segments is determined by the above-mentioned methods, including the finite element method and the smoothness optimization method. So the quantity may be a large one. The method of directly constructing the relationship seemed to be complex. However, method can be made simpler by a wise choice of substituting the short line segments into a parametric curve. The parameter value, like $s$ value, can vary in the range between 0 and 1. So their parameter values can change asynchronously in their respective parametric domain.

Bezier, B-Spline and non-uniform rational B-Spline (NURBS) are representative of these parametric curves. Especially, NURBS curves have the local modification property [36] and can achieve complex contour representation without increasing the order of equation, etc. So in this paper, we use NURBS curves to describe the initial isotherm lines.

5.2.2. Constructing one-to-one mapping rules between the points in the parametric domain and the corresponding points in the standard parametric domain

Secondly, if an arbitrary point $(s, t)$ is selected in standard parametric domain, the unique corresponding point $(u, v)$ is required to be found in the parametric domain. Therefore, $(s, t)$ and $(u, v)$ get into a certain mapping relationship, which will be described later in the following text.

The mapping rule (1):

The serial number of the corresponding isothermal line can be obtained by $t$ value.

$$t' = \lfloor N_t \times t \rfloor.$$
This equation is used to calculate the serial number of the selected isothermal line, where \( N_t \) is the total number of the initial isothermal lines and the interpolation isothermal lines. One unit along the \( t \) direction is divided into \( N_t \) aliquots in the standard parametric domain. Each \( t \) value is an integer multiple of \( \frac{1}{N_t} \). The \( t \) value varies from small to big corresponding to the isothermal lines distribution from the external to internal in the parametric domain. By this way, all the isothermal lines will be associated and numbered.

The calculated floor rounding result is used to determine the serial number \( t' \) of isothermal line.

The mapping rule (2):

As previously described, the point \( p^k_{(u,v)} \) is calculated by equi-chord deviation method on each initial isothermal line \( r_i(t) \) and the corresponding point \( p^k_{(u,v)} \) can be obtained in the standard parametric domain. The obtained quantities of corresponding discrete machining points \( m \) on each \( r_i(t) \) are usually unequal. Let NURBS curve \( C(s) \) pass through all the points \( p^k_{(u,v)} \) and the corresponding control points can be computed [37]. As illustrated in Fig. 12(a), we take two adjacent initial isothermal lines \( p_k \) and \( p_{k+1} \) are the corresponding control points for example, there are \( m \) quantities of interpolation isotherms between them. In order to generate spiral trajectory in parametric domain, we need to standardize management rearrange control points for each isotherm. The line segment \( p_k p_{k+1} \) crosses each isotherm at \( p^k_r = \{P_{(u,v)} \} \); these intersections are added as the first and end control points of each corresponding isotherm and rearrange the points \( p^k_{(u,v)} \) for each isotherm in the same direction; let a set of closed standard NURBS curves be \( C(s_k) = \{C(s_k_1) \ldots C(s_k_m)\} \), the parameter range of each standard curve is \([0, 1]\); In standard parametric domain, they are corresponding to \( m \) quantities of line segment of unit length in \((s, t)\) unit \( \{t \in \{t_k, t_{k+1}\}, s \in [0, 1]\} \).

The quantities of the points on the two adjacent isothermal lines \( r_i(t), r_{i+1}(t) \) may differ. Through the approximate handling of each of these isothermal lines into \((m + 2)\) quantities of parameter sections, \( S(k) = \{s(1) = 0, s(2) \ldots s(m + 1), s(m + 2) = 1\} \) [see Fig. 12(b)]. By this way, every machining point in the parametric domain can get a corresponding point in the standardization parametric domain.

It should be noted that for the area which is perfectly round, we can simplify the complexity of calculation. As mentioned above, symmetric solutions of circle is a set of approximate concentric circles (see Fig. 13(a)). As illustrated in Fig. 13(b), the point \( p^k_{(u,v)} \) is calculated by equi-chord deviation method on each initial isothermal line \( r_i(t) \) will be are very evenly distributed. Let point \( C \) be the center of the round and the extension lines of line segment \( p_k p_{k+1} \) pass through it.

According to quantity of \( m, s \) unit is divided into \((m + 2)\) aliquots, the \( s \)-coordinates are \( s = \{s(1) = 0, s(2) = \frac{1}{m+1}, \ldots, s(m + 1) = \frac{m}{m+1}, s(m + 2) = 1\} \); the corresponding \( t \)-coordinates are \( t = \{t_k, t_k + \frac{\alpha+1-\alpha}{m+1}, t_k + \frac{\alpha+1-\alpha}{m+1}, \ldots, t_k + \frac{m-1-\alpha}{m+1}, t_{k+1}\} \), so each coordinate point can be represented as \((s, t) = (i \frac{1}{m+1}, j \frac{k+1-\alpha}{m+1}) \), \( i, j \in \{0, 1, 2, \ldots, m + 1\} \).
5.2.3. Planning the spiral trajectory in parametric domain

In the standard parametric domain, the points are connected one by one in order of \( s \) value increase, which should be kept synchronously increasing with \( t \) value (see Fig. 14(a), black filled circles are representatives of these points). The \( t \) value of the end of current trajectory is just equal to the start of the next trajectory (see Fig. 14(b)); the spiral trajectory will be accordingly achieved in the parametric domain (see Fig. 14(c)).

Then they are mapped onto the parametric surfaces respectively. Finally, we will obtain the spiral tool-path that satisfies the machining parameters. In this paper, we just call the spiral which is generated by this method as the PDE spiral.

In the standard parametric domain, the variation of the parameter value \( s \) from 0 to 1 is taken as one independent trajectory (a machining unit). It is easy to understand that the planning of spiral trajectory shall meet the following demands:

1. Each of the isothermal lines in the parametric domain is an entire closed curve, so it can be represented as a straight line segment with \( s \) value varies from 0 to 1 in the corresponding standard parametric domain. Therefore, the planning of the two adjacent trajectories in the parametric domain shall satisfy this condition: The \( t \) value of the end of the current trajectory is just equal to the start one of the next trajectory;

2. The adjacent trajectories are connected from the start trajectory to the end one. The incremental ratios of the \( s \) values between every two adjacent trajectories are generally not constant except for the rotational parts;

3. The increment ratios of \( t \) values between every two adjacent trajectories are constant.

6. Double spiral tool-path generation and linking method

The above double spiral tool-path has a self-complementary structure, which means the starting and ending points are both located on the boundary of the workpiece. If this kind of tool-path is applied in actual machining, both high processing efficiency and high processing quality would be obtained.

Among the double spiral trajectories, the simplest one is the Archimedes double spiral trajectory. It is named after the 3rd century BC Greek mathematician Archimedes. It is the locus of points corresponding to the locations over time of a point moving away from a fixed point with a constant speed along a line which rotates with constant angular velocity. The polar coordinates equation of Archimedes spiral is \( \rho = \rho_0 + a \theta \), also known as uniform speed spiral. In a polar coordinate system, the polar radius \( \rho \) has a linear relationship with polar angle \( \theta \).

As illustrated in Fig. 15, spiral A and B are both Archimedes spirals. The double spiral AB with self-complementary pattern is obtained by linking the centers of the interior endpoints of spiral A and B.

6.1. Double spiral generation method in rectangle domain

We can refer Archimedes double spiral trajectory generation method for the obtainment of the double PDE spiral trajectory in
rectangle domain, as illustrated in Fig. 16. The generation method can be based on standard parametric domain or general rectangle domain. We will take, for example, the former case to illustrate the generation method.

The generation method for the corresponding points on PDE spiral $A$ is illustrated in Fig. 17: the standard parametric domain is divided into $m \times n$ equal parts in the $u$ and $v$ direction, respectively. First, the PDE spiral $A$ can be generated in the sub-domain which contains the $(0, 0)$ point. The center coordinate of this sub-domain is $(1/2m, 1/2n)$. Now, we consider the following transformation from the original coordinate $(u, v)$ (an arbitrary point $a'$ on the PDE spiral $A$) into a new set of coordinate $(u', v')$ (a corresponding point $b'$ on the PDE spiral $B$). The PDE spiral $B$ can be obtained by rotating the PDE spiral $A$ to $180^\circ$. Point $b'$ can be obtained by the following equation:

$$b' := R(a')^T = \begin{pmatrix} 1/m - u & 1/n - v & 1 \end{pmatrix}^T$$

where $R := \begin{pmatrix} -1 & 0 & 1/m \\ 0 & -1 & 1/n \end{pmatrix}$; $(a')^T := (u, v, 1)^T$.

The coordinates which are located in the rest of sub-domains can be calculated through analogous matrix as previously mentioned. Now, more generally, we just generate the PDE spiral $A$ in the sub-domain $(m \times n)$ columns). The center coordinate of this sub-domain is $C_{p,q} := (1/2m) \times (1/2n) \times (1/2n)$. In this case the original coordinate $(u_{p,q}, v_{p,q})$ (an arbitrary point $a_{p,q}$ on the PDE spiral $A_{p,q}$) is transformed into a new set of coordinate $(u_{p,q}, v_{p,q})$ (a corresponding point $b_{p,q}$ on the PDE spiral $B_{p,q}$). Point $b_{p,q}$ can be obtained by the following equation:

$$b_{p,q} := R_{p,q}(a_{p,q})^T = \begin{pmatrix} p/m - u_{p,q} & q/n - v_{p,q} & 1 \end{pmatrix}^T$$

where $R_{p,q} := \begin{pmatrix} -1 & 0 & p/m \\ 0 & -1 & q/n \end{pmatrix}$; $(a_{p,q})^T := (u_{p,q}, v_{p,q}, 1)^T$.

Finally, all the points are mapped on the parametric surface, connecting these corresponding mapping points for the obtainment of double spiral tool-path.

6.2. Double spiral generation method in arbitrary domain

As for a non-rectangular domain, we can obtain the spiral $A$ by the preceding algorithm. Based on the spiral $A$, for the obtainment of spiral $B$, taking the processing path intervals which are between $A$ and $B$ into account, the processing path intervals need to be as even as possible. Therefore, the ideal spiral $B$ can be obtained by connecting the equidistant spacing points between every two adjacent loops of spiral $A$.

The spiral trajectory $B$ can be obtained by connecting the points in the standard parametric domain as illustrated in Fig. 18. For the three (triple) adjacent curves (i.e., $a$, $b$, and $c$), which are illustrated in Fig. 18(a), we first connect the equidistant spacing points on the adjacent curves successively. Then we obtain the curves $a'$ and $b'$ as the mapping trajectories. Finally, the spiral $B$ can be generated in the parametric domain, as illustrated in Fig. 18(b).

The path interval between every two adjacent loops of double spiral $AB$ can be guaranteed to be just approximate equidistance. As illustrated in Fig. 19, the double spiral $AB$ with self-complementary spiral pattern is obtained by linking the centers of interior endpoints of spiral $A$ and $B$.

It should be noted that, the double spiral generated in rectangular domain in Chapter 6.1 can also be obtained by the method in this chapter, which can be regarded as a special case.

6.3. Linking method in rectangle domain

For large-size workpieces machining, the parametric surface can be divided into several machining sub-domains. For explaining the tool-path linking method, we construct the following PDE spiral models, as illustrated in Fig. 20.

The standard parametric domain will be divided into various methods according to different machining requirements. Then different linking methods will be adopted by corresponding combinations of models $A$ and $B$, as illustrated in Fig. 21.

The linking rules can be generalized as follows:

1. If model $A$ and $B$ are just located on the same row but located on two adjacent columns, connecting the input of model $A$ with the output of model $B$, or vice versa.

2. If model $A$ and $B$ are just located on the same column but located on two adjacent rows, connecting the input (output) of model $A$ with the output of model $B$, or vice versa.
When machining the next row, the trajectories in current row should be inverted.

It should be noted that, the linking method can be applied to treat two cases: (1) sequentially machining a large amount of rectangular workpieces; (2) a single large pocket can be divided into multiple rectangular sub-domains. We will give examples in the following text.

6.4. Linking method for complex domain

For complex surface machining, the determination of manufacturing procedure has to take account of many factors such as geometrical characteristics of the surface and requirements of machining parameters. There are many segmentation methods, including: equi-slope method [38], curvature method [39], contours method, concave-convex domain method, and artificial-definition method which based on the characteristics of complex surface and machining properties [40]. Surface segmentation is aimed at improving the machining quality and efficiency. Our segmentation method is based on the combination of the mathematical computation and artificial segmentation. First, we compute the mean curvature partitioning. Then, we can partition surface according to curvature contours: based on the obtainment of intersections on the surface by manually drawing contours with surface and combine with geometry characteristics of this trimmed surface, we can partition this surface into sub-domains.

By defining abstract points as the representatives of sub-domains (centroid point or center of gravity, etc.), we can prove the following question: given an arbitrary number of points (2D), draw a path that consists of straight lines between points, visit each point exactly once and does not intersect with itself. Take convenience of feed and retreat process into account. It is still possible if start and end point are fixed. So it is always possible to link all the double spiral tool paths into one path, based on this, we give the generic way (be simply classified the as “zig-zag” type or “spiral” type) for tool paths linking. The former is described in Chapter 6.3 and the latter is illustrated in Fig. 22(c). Indubitably, a good linking method cannot only shorten machining time but also improve machining quality.
Take a complex domain illustrated in Fig. 22 for example: firstly, the isotherm lines are generated in every sub-domain, respectively (see Fig. 22(a)); secondly, the initial isotherm lines are determined according to the processing path interval in every sub-domain, respectively (see Fig. 22(b)); thirdly, double spiral trajectories are generated in every sub-domain, respectively (see Fig. 22(c)); finally, the linking sequence are defined as 1IN → 1OUT → 2IN → 2OUT → ... → 5IN → 5OUT (counterclockwise), or vice versa (clockwise), as illustrated in Fig. 22(c).

In the foregoing example, shown in Fig. 22, the linking method must satisfy the following steps:

(1) the obtainment of start and end point: The prime principle is to ensure the path intervals so as to make material removal as even as possible and reduce the processing repeat as much as possible to avoid leaving unprocessed domains residue simultaneously.

Based on the above requirement of path interval, for every current sub-domain to be processed, the ideal start point should be adjacent to the end point of the previous tool-path. The end point should be adjacent to the start point of the next tool-path. For interpolating Bezier curves as linking trajectories, the following step (2) must also be satisfied.

(2) the obtainment of intermediate point: To generate quadratic Bezier curves, we need to set three constraint points. The generation method of quadratic Bezier curve is as follows:

The linking trajectories should pass through the central sub-domain (No. 5 sub-domain), therefore offsetting the outer boundaries of No. 5 sub-domain with an offset distance $\varepsilon$ which is obtained by scallop-height method, as illustrated in Fig. 23. Let us take, for example, the domain I to describe the generation method.

$L_i$ and $L_{i+1}$ are two adjacent boundaries of No. 5 sub-domain, the corresponding offset curve is $L'_i$ and the original curve is $L_i$, $P'_1$ is the endpoint of $L'_i$, $T1$ and $T2$ are the tangential directions at point $P_{2IN}$ and $P'_1$, respectively. $P_i$ is the intersection of $T1$ and $T2$. So $P_i$ is the intermediate point. $P_{2IN}$ is the start point of the next tool-path.

The quadratic Bezier curves can be determined by three control points $P'_1$, $P_i$, $P_{2IN}$, as shown in Fig. 23(b). The parametric quadratic Bezier curves are given by

$$B(t) = (1 - t)^2 P'_1 + 2t(1 - t)P_i + t^2 P_{2IN}, \quad t \in [0, 1].$$

The two-terminal tangential directions of Bezier curve are equal to $T1$ and $T2$, respectively. The same method will be applied to link all the double spiral tool-paths one by one sequentially.

7. Simulation and verification

This method has been implemented and verified by simulation on a Windows XP + CPU 1.84 GHz + RAM 2 GB hardware platform with Microsoft Visual C++ 2005 + ACIS + Hoops. A machining example for a complex pocket is given to illustrate the actual machining quality.

Example 1 (Pocket Machining). For the given test pocket model in Fig. 24, the thermal conductivity model is constructed and PDEs are solved for the obtainment of isothermal lines. The temperature value $T$ of external boundaries is set to be 1000 °C. The presented method can generate the double spiral tool-path automatically according to the given parameters. The value of scallop height $h_f$ is set to be 0.2 mm and the chord approximation error $\varepsilon$ along the feed direction is set to be 0.12 mm; the selected tool radius is 12 mm, and the cutter taper $\theta$ is 4°.

As can be seen from the figures, the generated double spiral tool-path is continuous and smooth.
Example 2 (Rotational Parts Machining). For the given test pocket model in Fig. 25, the thermal conductivity model is constructed and the circle symmetric solution method are applied for the obtainment of isothermal lines. Table 1 shows the cutting parameters and tool geometries.

As indicated in Fig. 25(a), the rotational part consists of disc cylinder, spherical cap and disc. The tool-path can be obtained by circle symmetric solution method in the previous Chapter 4.2. The machining path interval is obtained by scallop height $h$. The generatrix of the rotational part consisted of three parts which can be taken as a whole guide line. The intersection points of the concentric circles and generatrix are the temperature points which can be used for the obtainment of isothermal lines. The double spiral tool-path of another rotational part can be generated by the same method (see Fig. 25(b)).

Special attention should be paid to the domain where tool-paths present as several parallel lines with $s$ value changing from 0 to 1 in the standard parametric domain corresponding to the cylinder domain in rotational part. That is, the domain to be machined which corresponds to stationary temperature zone stands for cylinder. Otherwise, in case the circle radius of rotational part changes, the temperature varies. As the temperature increases or decreases, the corresponding radius of the domain to be machined becomes larger or smaller.

Example 3 (Pocket Partitioning Machining). For the given shoe last model in Fig. 26, according to the topology geometric shape, the pocket can be divided into two sub-domains. For each sub-domain, the thermal conductivity model is constructed and PDEs are solved for the obtainment of isothermal lines, respectively. Take the example of left sub-domain, the process parameters are shown in Table 2. The same method can be applied for planning the right sub-domain. The cutting parameters and tool geometries are set to be the same as the left sub-domain.

The generated double spiral tool-paths are continuous and smooth. The tool-paths linking method can be referred to relevant parts of the previous Chapter 6.3. No more details are illuminated herein.

For a single large pocket in Fig. 27, the parametric surface can be divided into $3 \times 3$ processing domains according to the topology geometric shape. The tool-path linking method can be referred to Fig. 21(a). The temperature value of outer boundaries of each sub-domain are respectively set to be 1000 °C. Then the double spiral tool-paths and the linking tool-paths will be obtained, as illustrated in Fig. 27. Table 3 shows the cutting parameters and tool geometries.
Example 4 (Complex Pocket Partitioning Machining). For the fish-shaped complex pocket in Fig. 28, it can be divided into several sub-domains according to the topology geometric shape. So first, we divide this domain into five simple sub-domains and calculate the initial isothermal lines, respectively (see Fig. 28(a)). Then the trajectories are planned out in the standard parametric sub-domains, respectively, five double spiral trajectories can be generated and linked by the proposed algorithm in the corresponding parametric domains, let the machining sequence be from No. 1 to No. 5 (as illustrated in Fig. 28(b)). Finally, these double spiral trajectories are mapped onto the parametric surface, then five double spiral tool-paths are obtained (see Fig. 28(c)). The corresponding cutting tool is a ball-bottomed cutter mill of 3 mm in diameter, with a shank length l of 50 mm. Table 4 shows the cutting parameters and tool geometries used in rough machining.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$h_f$ (mm)</th>
<th>$e$ (mm)</th>
<th>$\theta$ (°)</th>
<th>$r$ (mm)</th>
<th>$l$ (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values</td>
<td>0.3</td>
<td>0.12</td>
<td>4</td>
<td>3</td>
<td>50</td>
</tr>
</tbody>
</table>

Example 5 (Actual Machining Fish-Shaped Pocket). For the parametric surface in Example 4, the size of original parametric model is miniﬁed by half first. Fig. 29 shows the actual machining processes of half-size parametric model. The rough and finish machining are carried out in the HNC-210B 3-axis machining center which is equipped with an embedded industrial PC CNC unit. During the experiment, the 1050 aluminum sheet is used and speciﬁc oil is also applied to minimizing the friction between tool and aluminum sheet.

The cutting parameters and tool geometries of finish machining process is shown in Table 5.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$h_f$ (mm)</th>
<th>$e$ (mm)</th>
<th>$\theta$ (°)</th>
<th>$r$ (mm)</th>
<th>$l$ (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values</td>
<td>0.01</td>
<td>0.02</td>
<td>4</td>
<td>2</td>
<td>50</td>
</tr>
</tbody>
</table>

First, we repeat the rough machining tool-paths which are same to Example 4 for several times until the whole sub-domain of fish-body is machined (see Fig. 29(b)). In contradistinction to other machining parameters, rough and finish cutting parameters are shown in Table 6.

The machining results are illustrated in Fig. 29. The total cutting time of finish machining is 2 h 23 min 18 s. For comparison purposes, the corresponding results of the chosen commercial CAM software, in which the zig-zag and contour-parallel strategies were provided here and the machining parameters are just the same as our proposed method (see Fig. 30). The total cutting time of finish
Table 6
Cutting parameters—contradistinction between rough machining and finish machining.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Rough machining</th>
<th>Finish machining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth of cut (mm)</td>
<td>1.2</td>
<td>0.6</td>
</tr>
<tr>
<td>Spindle speed (rpm)</td>
<td>5000</td>
<td>7000</td>
</tr>
<tr>
<td>Feed rate (mm/min)</td>
<td>700</td>
<td>1300</td>
</tr>
</tbody>
</table>

Fig. 29. Actual machining complex pocket. (a) Rough-cut tool-path simulation; (b) rough machining result; (c) finish-cut tool-path Simulation; (d) finish machining result.

Fig. 30. Tool paths generated by commercial CAM software. (a) zig-zag tool path; (b) contour-parallel tool path.

machining are 3 h 13 min 14 s and 2 h 51 min 26 s, respectively. In summary, our method, which indicates that efficiency, smoothness and no retraction are superior to existing methods.

In order to clearly show tool paths, the error criterion (i.e., scallop height) is set to be much larger than those in real cases. As the figures show, for zig-zag tool-path the number of cutter lifting times is much greater than contour-parallel tool-path. For contour-parallel tool-path the removing of intersections between offsetting paths is needed. Another problem of conventional tool-path generation methods is that they preserve sharp corners, which limits the feed-rate when approaching these corners. Thus, machining efficiency is reduced and tool wear is intensified. However, the method proposed rounds these corners automatically and gradually.

The finish machining result is evaluated by the surface roughness measurement. The actual values (experiment results) of Ra are observed on the above-mentioned selected cutting conditions. The amount of standard surface roughness parameter (Arithmetic average deviation from the mean line Ra) is carried out using surface roughness tester (SRT-6200, produced by Starmeter Instruments Co., Ltd., China). Results of surface roughness are analyzed with the use of repeated measurements analysis of variance. Finally, the experiment result of Ra is 0.933 µm.

As seen from the finish machining result (see Fig. 29(d)), there are no retractions, cracks, wrinkles, distortion, or other phenomena of instability, and an excellent surface quality has gained by adopting the new method.

Simulation analysis
Curvature values

The curvature values of the corresponding discrete finish CL points can be evaluated [25]. The simulation results in Example 5 are analyzed as follows:

The curvature values of each machining point are illustrated in Fig. 31. The maximum curvature of each sub-domain is located at the center of that sub-domain. The maximum curvature occurred at the fishtail (No. 3 sub-domain) with a value of only 5.74122.

The relatively high curvature values are mainly concentrated upon the fish pectoral fin and fish tail. These two parts are
illustrated for doing independent research, respectively (see Fig. 32). Since the tool-paths generated by our algorithm will result in uneven distribution of tool-path spacing. The problem will be amplified when a pocket has some narrow cross-sections (or bottlenecks), like these two sub-domains.

The mean curvature of each tool-path is calculated, as illustrated in Fig. 33.

The mean curvature values are small, meaning that the double spiral tool-paths will be very smooth. As seen from simulation result (see Fig. 29(a)), there are no retractions or other signs of instability and an excellent double spiral tool-path has been gained by adopting the new method.

Errors analysis

Fig. 34(a) shows a colormap of scallop height errors $h_f$ of fish-body. Actually, the comparatively large distortion only appears at the tool-paths which are very close to boundary corners. According to the experiments we conducted, the number of these tool-paths is 3–5. In order to avoid leaving unprocessed domains residue, the maximum scallop height error should never exceed the limits of preset value $h_f = 0.01$ mm. If the scallop height errors exceed limitation of preset value, by appropriately adopting smaller scallop height value at these 3–5 tool-paths close to corner areas, we can effectively control local machining accuracy of these corner areas and guarantee the scallop height value for remaining areas be relatively large.

Furthermore, we define a concept called bisection error $\varepsilon_b$. It is used to evaluate the bisection level of spiral Tool path $A$ divided by spiral Tool path $B$. After calculating path intervals $L_1$ and $L_2$, we compute scallop height $h_1$ and $h_2$ by above mentioned equation, respectively. $h_f$ can be evaluated as average value of $h_1$ and $h_2$ ($h_f = \frac{h_1 + h_2}{2}$). The bisection error $\varepsilon_b$ is evaluated by $\varepsilon_b = \frac{|h_1 - h_2|}{h_1 + h_2}$. Fig. 34(b) shows a colormap of bisection errors. Therefore, the scallop height error can be restricted within 0.01 mm and the bisection error is around 0.8%.

8. Conclusions

This paper proposes a new spiral tool-path generation algorithm for high speed machining to ensure high efficiency and high quality. This method possesses the following advantages:

(1) This method reasonably constructs the thermal conductivity model and determines boundary conditions to obtain the machining tool-path which satisfy the machining parameters for the complicated surface. The machining path interval is determined by the scallop height method. The feed step length is determined by equi-chord height difference method.

(2) To further improving smoothness of contours, let the definition of “C Key Loop” be the contour which is determined by curvature threshold value, while let the definition of “L Key Loop” be the contour which is determined by length threshold value. Furthermore, a smoothness optimization method is proposed to improve the smoothness of the isothermal lines and avoid taking up too much memory during solving PDEs.

(3) This method constructs custom mapping rules between standard parametric domain and parametric for generating double spiral tool-path. The double spiral tool-path is generated by means of the selected contours without generating any self-intersection loops. The presented double spiral tool-paths
have a self-complementary structure with characteristics of low curvature, continuous and smooth.

(4) The linking method is flexible for machining pocket with complex geometric shape. The Bezier curve is adopted as the linking trajectory. It can guarantee the linking trajectory which is between two adjacent machining domains to be smooth and no retraction.

Acknowledgments

The authors are grateful for the support provided by National Key Basic Research Project of China (grant # 2011CB302400) and National Natural Science Foundation of China (grants # 50975274, 50975495).

References


Bo Zhou is currently a Ph.D. student at Chinese Academy of Sciences. His research interests include computational geometry for CAD/CAM, computer graphics.
Jibin Zhao is a professor at Shenyang Institute of Automation, Chinese Academy of Sciences. He received his Ph.D. from Chinese Academy of Sciences in 2004. His research focuses on geometric processing and solid modeling, computer graphics and NC machining. He can be reached at jbzha@sia.cn (the best way to reach him).

Lun Li is an associate professor at Shenyang Institute of Automation, Chinese Academy of Sciences. He received his Ph.D. from Harbin Institute Of Technology in 2004. His research focuses on computational geometry for CAD/CAM, computer graphics and NC machining.