

Article

A Non-Uniform Offset Algorithm for Milling Toolpath Generation Based on Boolean Operations

Giuseppe Venturini, Niccolò Grossi , Lorenzo Morelli and Antonio Scippa * 

Department of Industrial Engineering, University of Firenze, Via di Santa Marta 3, 50139 Firenze, Italy

* Correspondence: antonio.scippa@unifi.it; Tel.: +39-05527-58726

Featured Application: The algorithm presented in this work could find useful application in advanced toolpath definition for milling operations when non-uniform (i.e., variable) machining allowance is desired.

Abstract: In milling, the advancement of CAM strategies has increased the need for tailored algorithms for semi-finished phase computation. In some cases (e.g., thin-wall milling), variable radial engagement of the tool during the toolpath is desired, leading to the need of non-uniform machining allowance on the component that could be achieved only with a non-uniform offset algorithm, i.e., offset where the distance to the initial contour varies along that input. This work presents a general algorithm for non-uniform offset of polyline curves. The approach is based on 2D polygons and Boolean union operation, following these steps: (i) projection segments are generated, (ii) polygons (trapezoids and circular sectors) are created, (iii) Boolean union of all the polygons is performed, (iv) boundary of interest is extracted. The proposed algorithm is able to handle both internal and external offset and is robust for complexity of both the polyline and variable offset magnitude along that line, as proven by several examples and two applications to thin-wall milling provided.

Keywords: milling; toolpath; non-uniform offset



Citation: Venturini, G.; Grossi, N.; Morelli, L.; Scippa, A. A Non-Uniform Offset Algorithm for Milling Toolpath Generation Based on Boolean Operations. *Appl. Sci.* **2023**, *13*, 208. <https://doi.org/10.3390/app13010208>

Academic Editor: Qiang Cheng

Received: 2 December 2022

Revised: 20 December 2022

Accepted: 22 December 2022

Published: 24 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Milling is one of the most used technologies to achieve the final shape of mechanical components due to its flexibility and accuracy. Process productivity has been increased during the last decades by exploiting the advancements in tooling system, machine structure and numerical control (NC). The latter, in synergy with modern CAD/CAM software, allows for the implementation of advanced toolpath strategies [1]. Traditionally, in 2.5-axis milling, two main strategies are used: direction-parallel and contour-parallel [2]. In pocket milling, other approaches can be pursued, such as zig-zag milling, or component-offset pattern [3]. All these strategies require algorithms to offset the final geometry to both compute the position of the center of the tool and, in case of multiple passes, to update the stock geometry. Several algorithms have been developed with the aim of improving efficiency and solve specific issues [4–8]. Liu et al. [4] obtained the offset curve by trimming and jointing the raw curve, resulting from just projecting the points of the polyline. A similar approach was proposed by Seong et al. [5] that exploits distance maps to trim local and global self-intersections, which are the main sources of errors in offset computation. Omirou [6] proposed to use a locus tracing concept for cutting radius offset and provides some applications to CNC machining. Choi and Park [7] developed a method based on intersections of sculptured surfaces for offsetting point-sequence curves (PS-curves), while Wicklin computed offset regions by using circles at each vertex and rectangles along each edge [8]. Contour offset can also be obtained by using a straight skeleton, but this could lead to sharp corners [9]. Linear axis is an interesting alternative as a means to compute such offsets, reducing the sharp corners issue, as investigated by Palfrader and Held [9].

However, all the presented techniques are dedicated to uniform offset of a polyline curve, characterized by a constant distance of each offset element (i.e., segments) from the initial contour elements.

Uniform offset is fundamental for most of the traditional toolpath approaches based on a geometric basis [10]. Nevertheless, in the case of advanced CAM strategies increasingly adopted, the limitation of constant value offset, i.e., uniform machining/stock allowance, could be critical. Indeed, in some cases variable radial engagement of the tool during the toolpath could be preferred, leading to the need of a non-uniform stock allowance on the component that could be achieved only with a non-uniform offset algorithm, i.e., offset where the distance to the initial contour varies along that input. Generally, such an approach is required when the toolpaths are not only calculated on a geometric basis (i.e., considering only the removal of the material), but also involving technological aspects. For example, in thin-wall milling, if stiffness of the part is considered in selecting the optimal engagement, non-uniform machining allowance on the components could be desired. This is the case of the approach proposed by Wang et al. [11] that developed a cutting sequence algorithm to reduce workpiece deformation, suggesting the use of different depths of cut during the cutting cycle, depending on the changing stiffness of the part. Starting from such a concept, Grossi et al. [12] developed a method to compute the toolpath for milling thin-wall components. In their work, in each pass, the radial depth of cut changes continuously to achieve the desired quality without compromising productivity. To achieve this variable radial depth of cut, variable machining allowance on the component is required, and to implement such machining allowance on the final component a non-uniform offset algorithm is needed. Few works were dedicated to non-uniform offset, known also as variable-radius offset. Yan et al. [13] proposed the so called “uneven offset” method, but the method is very specific and focuses on the elimination of materials remaining left on the corners. Qun and Rokne’s [14] and Zhuo and Rossignac’s [15] works are dedicated to the geometrical definition of the different offsets; however, they do not provide efficient and robust algorithms. Variable offset can be constructed based on sampling, using rendering-based methods [16]; however, computational efficiency and accuracy in extracting the actual offset from the rendered images are still open issues. Recently, Held et al. [17,18] computed a variable offset exploiting the new variable-radius Voronoi diagrams. Although interesting, their works require several steps, increased effort in the implementation and are specific, since they do not apply to external offset.

This limitation is common to most of the proposed approaches for non-uniform offset that are designed on specific cases and they lack generality. This work presents a general algorithm for non-uniform offset of polyline curves, inspired by the uniform offset algorithm proposed by Wicklin [8]. The method, based on 2D polygons and Boolean union operation, is robust for complexity of both the polyline and variable offset magnitude along that line. Indeed, the main contribution of the proposed approach is the generality: the algorithm can be applied to both external and internal offset and it is effective for any type of contour and variable offset desired, not returning ambiguities or intersections in the resulting contour. In addition, another novel aspect characterizing the proposed approach is its implementation: the method is not based on specific complex algorithms (e.g., rendering-based, straight skeleton or Voronoi diagrams), which generally not easy to implement, but only on simple basic steps, exploiting traditional operations (i.e., projections, Boolean operations). Such operations are already adopted in CAM software, but in the proposed approach they are adapted to be used to compute non-uniform offset and achieve a desired variable machining allowance on the part.

In Section 2, the scope of the algorithm and the working principle is proposed, then in Section 3 results on several examples and applications to thin-wall milling cases are shown. Finally, in Section 4 some conclusions are drawn.

2. Materials and Methods

2.1. Scope

The proposed technique is designed to compute the non-uniform offset of a polyline curve. There are two main inputs of the method:

- The initial contour: a closed polyline curve, composed by several sequential points in 2D space;
- The offset array, including the offset values for each point; since the offset is non-uniform, the offset value could be different for each point (if the offset array is composed by all the same value a uniform offset is achieved).

The output is a new polyline curve generated by offsetting the initial contour points, internally or externally, by the values specified in the offset array.

Such offsets should be directed normally to the initial contour (when this operation is possible), and the sign depends on the selection of internal/external direction. In the 2.5 axis milling application, this offset procedure applied to contour, internal (pocket) or external, results in adding a non-uniform machining allowance to the geometry. An example of the desired new polyline is provided in Figure 1, where two different offsets of the initial contour are presented, i.e., small and high magnitude (Figure 1a) applied to external (Figure 1b) and internal (Figure 1c) directions. As is clear from Figure 1, the result of a non-uniform offset on a contour is a new polyline that can be seen as the initial contour with the addition of a machining allowance. The start point is highlighted as a red dot in the figure and the same is provided in the offset evolution plot (Figure 1a), where offset values in the offset array are shown. As a convention, the points move clockwise in the case of an internal offset and counterclockwise in the case of an external offset. The offset result is similar to the one of a classical (i.e., uniform) offset approach, except from the fact that the machining allowance is non-uniform. Such desired polylines cannot be achieved by just projecting the initial contour points along the normal to the contour by the required allowance, especially in the internal scenarios. Indeed, in the external scenario, the resulting points are part of the new contour and a strategy to connect them as in traditional offset can be pursued. In concave scenarios such an approach fails since intersections appear, as shown in Figure 1e.

A uniform offset example is provided in Figure 2, where the same initial contour is offset by a constant value (two different values are presented, Figure 2a). In the uniform allowance scenario, the intersections are regular (Figure 2) and approaches to remove them can be pursued, as it is described the uniform offset methods based on trimming [5]. Differently in non-uniform offset, such intersections could be very complex since the offset points could be found in several different configurations depending on the arbitrary different values of allowance (i.e., distance along the normal). Such degrees of freedom could generate complex offset polylines, even with a simple initial contour. An example is provided in Figure 3. In this example, the same initial contour should be offset internally with a variable sinusoidal machining allowance (Figure 3a), the idea is to generate thin walls in two zones, while thick walls are required in the other regions. The desired offset polyline is provided in Figure 3b, while the points projection is in Figure 3c. In this case, the non-uniform offset generates ambiguities. First, the intersections at the center lead to a polyline with three zones, but the center one should be removed, and two polylines must be considered. The second aspect is even more complex, indeed, the fillets between lines generate projected points that affect the actual offset of other points. It is hence clear the need of a general method for an arbitrary non-uniform offset of a contour. The scope of the presented study is detailed in the next section.

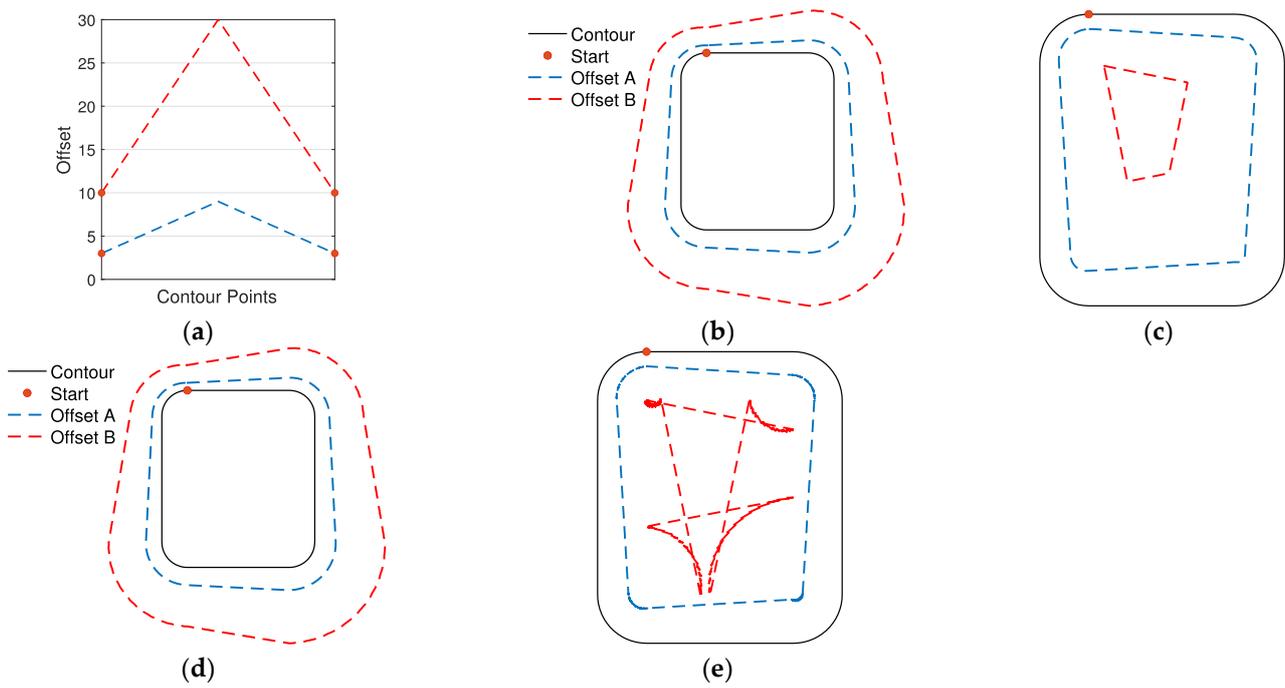


Figure 1. Non-uniform offset: (a) offset value array (b) desired external (convex) (c) desired internal (concave) (d) points projections in external (convex) (e) points projections in internal (concave).

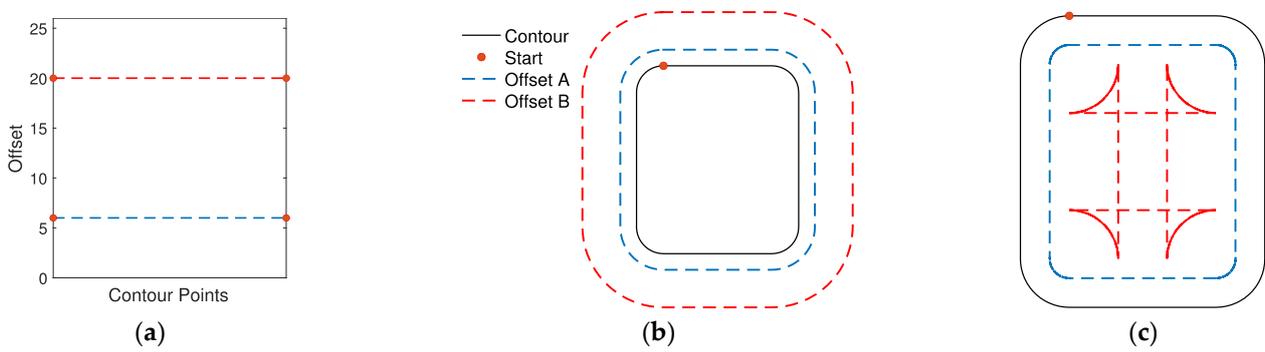


Figure 2. Points projection in uniform offset: (a) offset value (b) external (convex) (c) internal (concave).

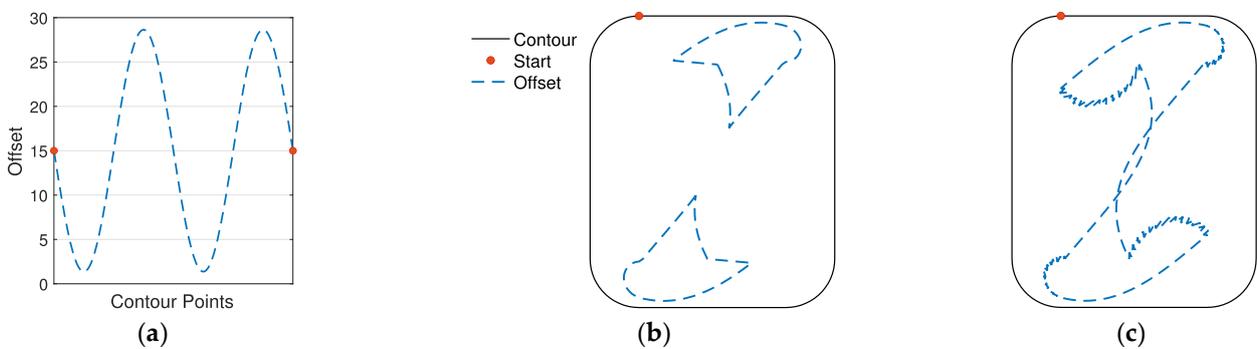


Figure 3. Non-uniform offset: (a) offset value array (b) desired (c) points projection.

2.2. Proposed Approach

The proposed approach aims at overcoming the previous presented issues and, starting from a generic polyline curve, generates a new polyline using a non-uniform offset array. The approach, based on 2D polygons and Boolean union operation, can be summarized in five steps:

1. **Projection:** each point of the initial contour is projected (internally or externally as defined) at the assigned offset value along the two normal vectors of the two adjacent segments it belongs to, generating two projecting points.
2. **Trapezoid:** a trapezoid is created for each segment by connecting the initial contour segment and the offset segment composed by the two projecting points. Since the same normal is considered (i.e., same starting segment) a trapezoid is always created (two right angles).
3. **Circular sector:** a circle arc is created connecting the two projected points generated by the same initial contour point, the center of the arc. This circle arc along with the projecting lines creates the circular sector. In the case of a multi-point straight line, since projected points of the initial contour point overlap, circular sectors are not computed.
4. **Union:** Boolean union operation is performed on all the polygons (i.e., trapezoids and circular sectors), and a single polygon is hence created.
5. **Offset contour:** internal or external boundaries of the Boolean union polygon are then extracted, thus achieving the offset polyline.

To highlight the working principle of the method, two examples of three points of initial contour are provided for internal (Figure 4) and external (Figure 5) cases. Figure 4a shows the first step of projection for an internal offset: each segment of the initial contour generates two additional projected points, the normal is the same while the magnitude could be different.

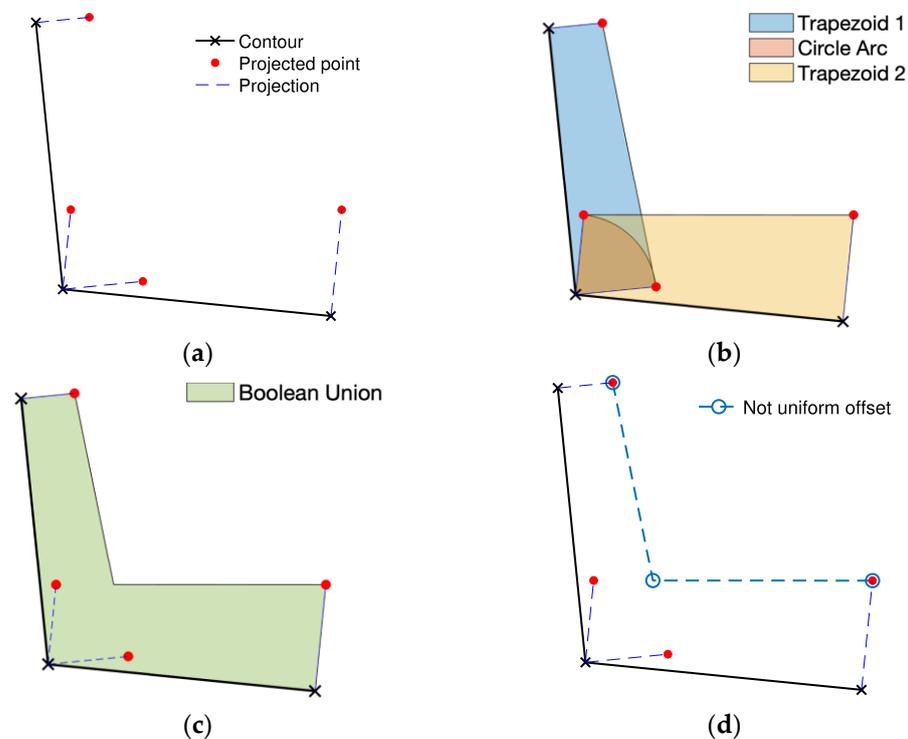


Figure 4. Proposed approach for concave example: (a) points (b) trapezoid and circle arc (c) Boolean union (d) results.

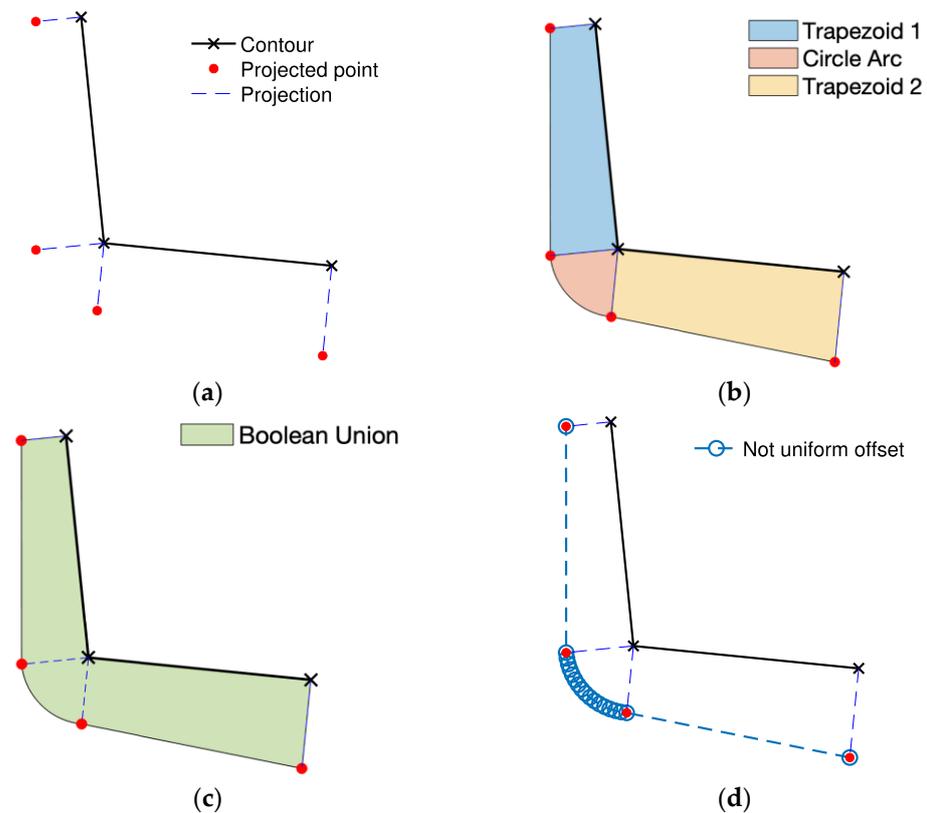


Figure 5. Proposed approach for convex example: (a) points (b) trapezoid and circle arc (c) Boolean union (d) results.

Therefore, starting from an internal point (in this case only the middle one) two points are generated. Figure 4b shows the polygon's generations at step 2 and 3. Trapezoids are created directly connecting the projected points with the original points, while circular sectors are the results of the rotation of one projection segment on the other originating from the same point of the initial contour. Such polygons are merged in the union polygon of Figure 4c using a Boolean union operation. Finally, this last united polygon is used to extract the non-uniform offset as its internal boundary. Such an approach is robust, returns continuous contours and allows for consideration of the generation of multiple contours starting from the initial one, as shown in the examples (Section 3.1). The same procedure is described for the external offset in Figure 5. In this case, at the end of the procedure, after the union polygon is computed (Figure 5c), the external boundary is extracted (Figure 5d).

The proposed approach has been developed in MathWorks MATLAB using polyshape objects. In variable offset, several points could be needed even for straight lines to achieve the desired non-uniform offset (i.e., each point could require to be offset by a different value). To improve efficiency, in such cases (i.e., same normal points), a single trapezoid-like polygon (i.e., curvilinear trapezoid) is created to be merged subsequently. It is worth pointing out that the proposed method does not aim at computing the machining toolpath, but its goal is to generate the contour of the semi-finished workpiece required to achieve a non-uniform allowance on the part, crucial in case a variable depth of cut is desired (e.g., thin-wall). The machining toolpath to achieve such semi-finished workpiece can be obtained by offsetting the resulting contour (e.g., using a general uniform offset) of the tool radius.

3. Results

In this section, the proposed algorithm is applied to different scenarios. First, different examples are provided to show how the algorithm could deal with complexity both in terms of initial contour and offset array. Then, two case studies are provided to show the potential application of the proposed approach.

3.1. Examples

In Table 1 six examples are shown, in the first column the initial contour and the algorithm results are provided, while in the other two columns the polygon phase and the offset values applied are shown.

As is clear from the results, the proposed algorithm is able to handle both internal and external offset of complex initial contours and offset arrays. The first example shows how the algorithm works in an internal simple geometry with a triangle-shape offset array (same example in Figure 1) and how self-intersection is intrinsically managed by the proposed approach. In the second, a sinusoidal offset array with high frequency is imposed to the same initial contour. The fluctuations generate a sinusoidal allowance on straight lines and asymmetrical intersections at the vertices, accurately represented by using the proposed algorithm. A sinusoidal offset array is provided to a more complex shape featuring convex and concave angles. Applying this offset externally (example 3) leads to variable thicknesses of the straight lines and edge rotation with different radii on the four convex vertices, while the intersection is computed in the concave angle. Example 4 shows the same complex shape with an internal sinusoidal offset and similar considerations can be drawn. The algorithm correctly computes both circular edge rotation at the convex vertex and the four intersections at the concave angles. In example 5, an offset array causing the creation of two different polylines is applied to the same initial contour of the previous two examples, while in example 6 an external offset to a curve (i.e., airfoil-shape) using a complex offset array is shown.

3.2. Application to Thin-Wall Components

In addition to the previous examples, which represent more theoretical applications rather than real case scenarios, more practical case studies are proposed.

Indeed, the proposed algorithm is dedicated to contouring operations in 2.5 axis milling considering a flat endmill as the cutter. In this context, the proposed algorithm starting from the polyline, which represents the finished part of the contour, produces the shape of the machining allowance to be left on the part based on the offset values assigned to each point of the starting polyline (i.e., semi-finished contour). These offset values identify the amount of allowance to be left on each point of the finished contour. Then, the actual toolpath could be obtained by applying an offset algorithm to the semi-finished contour previously obtained using the same offset value (i.e., endmill radius) for each point. Therefore, in 2.5 axis milling contouring operations, the proposed algorithm creates a toolpath able to achieve a variable radial depth of cut along the part contour. These conditions are particularly interesting in contouring operations in which both the tool and the workpiece are characterized by low stiffness. This type of operation is referred to as thin-wall milling, and it is a challenging topic in the manufacturing field. In detail, thin-wall components are increasingly adopted in modern industries for their favorable stiffness to weight ratio, but precision milling on these components is still a complicated task because unstable vibrations and large surface errors are inevitable during the process, limiting productivity. To ensure the required surface quality without compromising productivity, it is important to identify the potential of each portion of the thin-wall component in terms of stiffness so that the most profitable cutting parameters, which each portion can handle, are selected. In this regard, two applications for the definition of the semi-finished part in 2.5 milling are provided. It must be noted that, in the context of milling operations, the proposed method does not create the actual toolpath, but the desired semi-finished contour, which must be offset uniformly by the tool radius to obtain the effective toolpath.

As already mentioned above as well as in the introduction, one potential application of the proposed algorithm is the definition of a variable machining allowance on the finished thin wall part to achieve a variable radial depth of cut during the finishing operation. Indeed, the possibility to adopt a different radial depth of cut during the finishing cycle allows for removal of more material on the stiffer zones, reducing the actual material removal only in the critical portions of the part. As shown in [11], this allows an increase in productivity in comparison to the traditional constant engagement approach, in which the most conservative (i.e., minimum) radial depth of cut during the entire machining cycle must be used. Firstly, the proposed algorithm was applied to the case study proposed in [12]: 2.5 milling of a NACA 0005 airfoil profile 60 mm long and 20 mm high (Figure 6).

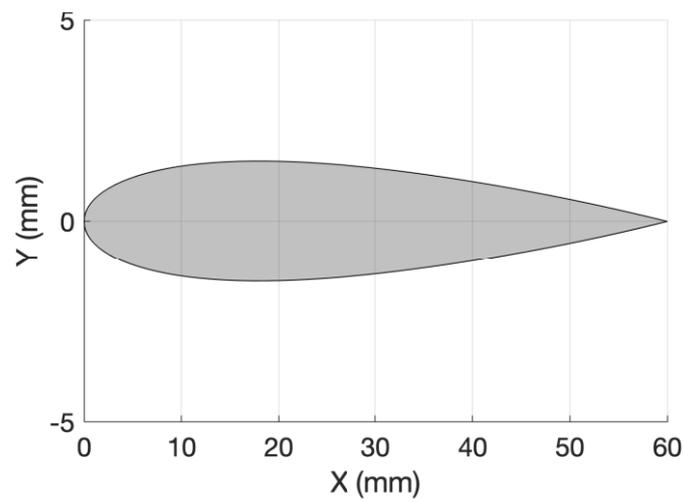


Figure 6. Blade 2D profile.

In the cited work, the authors aimed at identifying, from a static point of view, the highest value of the radial depth of cut in each portion of the airfoil so that the form error caused by the static deflection of the tool-workpiece couple does not fail the commanded tolerance. Workpiece deflection is predicted, including the effect of the material removal through finite elements methods. In detail, the method proposed by the authors starts by finding the stiffest portion of the airfoil. It evaluates the highest radial depth of cut possible within the imposed constraints, it updates accordingly the airfoil stiffness in the analyzed portion, and, following this practice, it continues analyzing the rest of the airfoil profile in reverse order with respect to the classic milling cycle. In this way the output of the method is the “desired” machining allowance to be left on the workpiece stock to obtain an airfoil which respects the commanded tolerance in each portion.

In the mentioned work, radial depth of cut was optimized to achieve a commanded tolerance (± 0.02 mm) leading to a variable depth of cut. Indeed, since the blade presents significant differences in terms of local stiffness, in the thicker zones of the profile a higher depth of cut can be achieved, while in the critical points (e.g., trailing edge) a more conservative depth should be used. The resulting radial depth of cut is presented in Figure 7. As expected, the figure shows the minimum radial depth of cut (i.e., offset of the contour) at the trailing edge, the most flexible portion of the blade. The starting point was set on the stiffest portion and discontinuity on this point is found between the beginning and the end of the cycle. This result is caused by the fact that during the operation, the thickness of the blade changes, affecting the stiffness and hence the allowable radial depth of cut. Please refer to [12] for additional details on radial depth of cut computation.

Table 1. Examples (measures in mm).

Id	Offset Value Array	Initial Contour (Solid) and Offset Results (Dashed)	Polygons
1			
2			
3			
4			
5			
6			

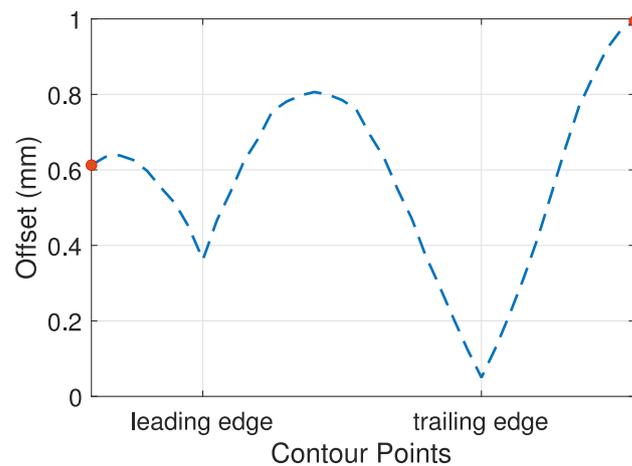


Figure 7. Desired radial depth of cut for the blade, i.e., offset values to be applied.

To achieve the desired radial depth of cut during the finishing operation, the semi-finished stock should be computed based on the initial contour adding the defined radial depth of cut as a machining allowance. This procedure was carried out using the proposed algorithm, and results are shown in Figure 8. Results show that the proposed algorithm is able to perform the non-uniform offset of the blade contour, dealing with the discontinuity and returning the desired semi-finished profile.

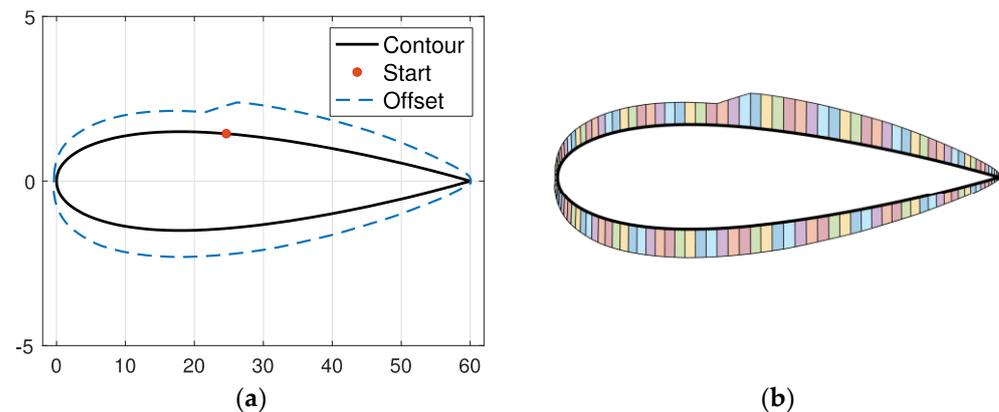


Figure 8. Blade results: (a) Initial contour and proposed algorithm results (b) polygons of the proposed approach.

Since the blade application was dedicated to an external offset, generally easier to manage, an application to an internal offset in pocket 2.5 milling operations is also provided. The case study is a Pylon Bracket for aerospace application, such components often entail thin-wall structures and pocket milling. In our case, the pocket analyzed is the one in the middle, as highlighted in the 2D representation of Figure 9. The 2.5 milling of the investigated pocket is not affected by the flexibility of the workpiece (thin-wall structure) for most of the operation, except for one of the sides that is characterized by a very thin wall (3 mm). Therefore, in this portion a variable radial depth of cut could be profitable: constant and standard radial depth of cut should be used in the thick portions, while drastically reduced radial depth of cut must be adopted in the thin-wall portion. An example of such possible depth of cut variation is provided in Figure 10.

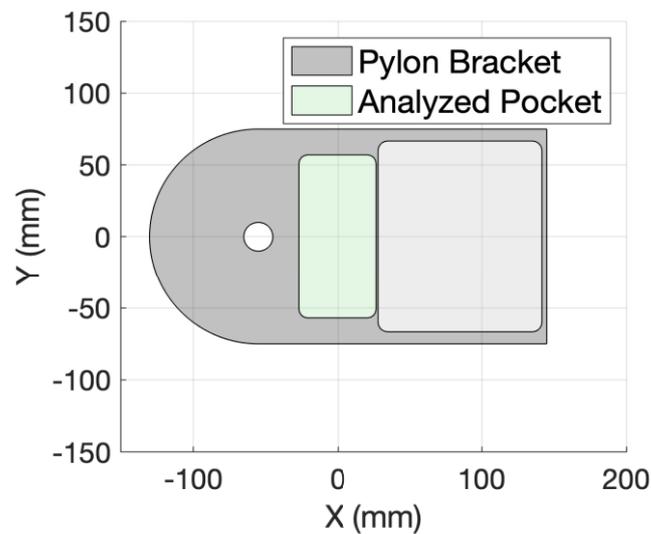


Figure 9. Component and analyzed pocket for the second application case study.

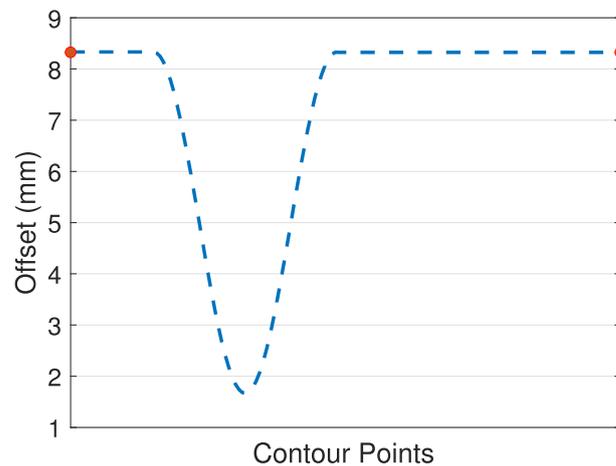


Figure 10. Desired radial depth of cut for the pocket, i.e., offset values to be applied.

The desired radial depth of cut is applied through the developed offset algorithm to the initial contour to obtain the semi-finished pocket and results are shown in Figure 11. The algorithm produces what is required: a constant offset in the thick portions and a reduced offset in the critical one. Milling the resulting offset contour to obtain the initial contour will entail a high radial depth of cut in all the regions, except for the thin zone where such depth of cut will be more conservative, as desired. As already mentioned, it is worth pointing out that since the offset polyline should be obtained by a milling operation, sharp edges cannot be achieved, and the real machining allowance will depend on the tool geometry (i.e., diameter) used for the roughing operation. This consideration is valid in general, the offset algorithm could result in sharp edge contours if such a contour represents the semi-finished component. As in the applications provided in this section, the diameter of the tool will affect the actual machined contour.

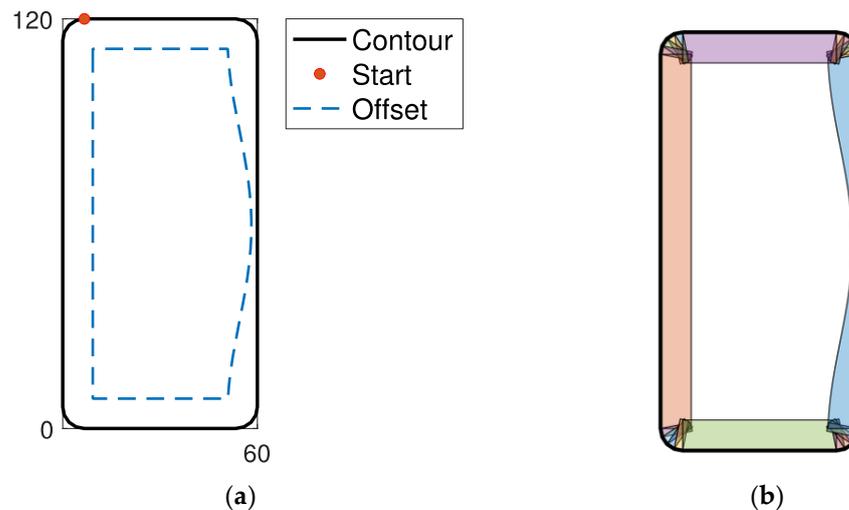


Figure 11. Pocket results (a) Initial contour and proposed algorithm results (b) polygons of the proposed approach.

4. Discussion

In this work, a non-uniform offset (i.e., variable-radius offset) algorithm for CNC milling toolpath generation is proposed. This algorithm computes the non-uniform offset contour of a polyline both internally and externally. Basically, the proposed approach requires two inputs: (i) the initial contour, a closed 2D polyline curve and (ii) an offset array, including the offset values for each point. The method is based on Boolean union of 2D polygons and follows five simple steps: (i) projection segments of the initial contour are generated, (ii) a trapezoid for each initial segment is created, (iii) a circular sector is generated by rotating one projection segment on the other originating from the same point of the initial contour, (iv) Boolean union of all the polygons is performed, (v) boundary of interest (i.e., internal or external) is extracted. Results show the effectiveness of the proposed approach in several examples. In addition, the potential application to thin-wall 2.5 milling toolpath definition is shown using two different case studies: the external milling of a blade profile and the internal milling of a pylon bracket pocket. The efficiency of this algorithm needs further investigations requiring the implementation in a more suitable programming language (e.g., C++). Moreover, improvements to the implementation could be considered to increase the robustness of the approach. In addition, as a future development the algorithm applicability could be extended to be used in path planning, which is achieving a growing interest in the world of automation. This involves the generation of a route in presence of obstacles that must be avoided to achieve a collision-free operation of robots or entities moving in a certain scene. A promising tool widely used in this field is the Minkowski sum algorithm, which is the point-wise sum of two sets [19]. In motion planning, the first of the two sets could be the obstacle, while the second the end effector or, in general, moving entity. A similar result could be achieved with the algorithm proposed in the present paper where, point by point along the perimeter of the obstacle, the shape of the moving entity could be added to the one of the obstacles taking into account also the orientation of the moving entity itself, thus allowing for rotation. To achieve such a goal, further investigation and implementations of the proposed concept are needed.

Author Contributions: Conceptualization, G.V. and N.G.; methodology, G.V.; software, N.G.; validation, N.G.; formal analysis, G.V.; investigation, N.G. and L.M.; writing—original draft preparation, N.G.; writing—review and editing, G.V., L.M. and A.S.; supervision, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are available upon request.

Acknowledgments: The authors would like to thank Machine Tool Technology Research Foundation (MTTRF) and its supporters.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Narooei, K.D.; Ramli, R. New approaches in tool path optimization of CNC machining: A review. *Appl. Mech. Mater.* **2014**, *663*, 657–661. [[CrossRef](#)]
2. Held, M.; de Lorenzo, S. On the generation of spiral-like paths within planar shapes. *J. Comput. Des. Eng.* **2018**, *5*, 348–357. [[CrossRef](#)]
3. Hatna, A. Automatic CNC milling of pockets: Geometric and technological issues. *Comput. Integr. Manuf. Syst.* **1998**, *11*, 309–330. [[CrossRef](#)]
4. Liu, X.Z.; Yong, J.H.; Zheng, G.Q.; Sun, J.G. An offset algorithm for polyline curves. *Comput. Ind.* **2007**, *58*, 240–254. [[CrossRef](#)]
5. Seong, J.K.; Elber, G.; Kim, M.S. Trimming local and global self-intersections in offset curves/surfaces using distance maps. *CAD Comput. Aided Des.* **2006**, *38*, 183–193. [[CrossRef](#)]
6. Omirou, S.L. A locus tracing algorithm for cutter offsetting in CNC machining. *Robot. Comput. Integr. Manuf.* **2004**, *20*, 49–55. [[CrossRef](#)]
7. Choi, B.K.; Park, S.C. A pair-wise offset algorithm for 2D point-sequence curve—PDF Free Download. *Comput. Des.* **1999**, *31*, 735–745.
8. Wicklin, R. Offset Regions: Find All Points within a Specified Distance from a Polygon. Available online: <https://blogs.sas.com/content/iml/2018/07/16/points-distance-from-polygon.html> (accessed on 30 November 2022).
9. Palfrader, P.; Held, M. Computing Mitered Offset Curves Based on Straight Skeletons. *Comput. Aided Des. Appl.* **2015**, *12*, 414–424. [[CrossRef](#)]
10. Jacso, A.; Szalay, T. Optimizing the numerical algorithm in Fast Constant Engagement Offsetting Method for generating 2.5D milling tool paths. *Int. J. Adv. Manuf. Technol.* **2020**, *108*, 2285–2300. [[CrossRef](#)]
11. Wang, J.; Ibaraki, S.; Matsubara, A. A cutting sequence optimization algorithm to reduce the workpiece deformation in thin-wall machining. *Precis. Eng.* **2017**, *50*, 506–514. [[CrossRef](#)]
12. Grossi, N.; Scippa, A.; Croppi, L.; Morelli, L.; Campatelli, G. Adaptive toolpath for 3-axis milling of thin walled parts. *MM Sci. J.* **2019**, *2019*, 3378–3385. [[CrossRef](#)]
13. Yan, S.; Shuilai, W.; Shuiguang, T. Uneven offset method of NC tool path generation for free-form pocket machining. *Comput. Ind.* **2000**, *43*, 97–103. [[CrossRef](#)]
14. Lin, Q.; Rokne, J.G. Variable-radius offset curves and surfaces. *Math. Comput. Model.* **1997**, *26*, 97–108. [[CrossRef](#)]
15. Zhuo, W.; Rossignac, J. Curvature-based offset distance: Implementations and applications. *Comput. Graph.* **2012**, *36*, 445–454. [[CrossRef](#)]
16. Li, C.L.; Gang, Z.; Chan, C.W. A graphical approach to approximate offset computation. In Proceedings of the 2009 Sixth International Conference on Computer Graphics, Imaging and Visualization, CGIV2009, Tianjin, China, 11–14 August 2009; pp. 217–221. [[CrossRef](#)]
17. Held, M.; Huber, S.; Palfrader, P. Generalized offsetting of planar structures using skeletons. *Comput. Aided Des. Appl.* **2016**, *13*, 712–721. [[CrossRef](#)]
18. Held, M.; De Lorenzo, S. Weighted skeletal structures for computing variable-radius offsets. *Comput. Aided Des. Appl.* **2021**, *18*, 875–889. [[CrossRef](#)]
19. Liu, L.; Zlatanova, S. An approach for indoor path computation among obstacles that considers user dimension. *ISPRS Int. J. Geo-Inf.* **2015**, *4*, 2821–2841. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.