CREATION OF CUTTING PATHS FOR MACHINING 2D FEATURES

Rovilson Mafalda, rovilson.mafalda@ufabc.edu.br
Eduardo Issamu Koike, eduardo.koike@gmail.com
Waldemir Martins Lira, waldemir.lira@ufabc.edu.br
Universidade Federal do ABC - Center for Engineering and Applied Social Sciences, Rua Santa Adélia, 166 - CEP 09090-400 - Santo André-SP, Brazil

Abstract. *The operation of machines with computer numerical control (CNC) is generally easier once that almost all of its functions can be programmed. In this paper we propose a semi-automated creation of the cut path for milling and turning operations of 2D pockets. For this, we build a program in C language and tested with several examples. In this proposal, the user enters the coordinates of the vertices that form a given polygon, which may have straight or circular shape. This paper presents examples of the cut creation for milling simulation results of the cavities. The main conclusion here is that this type of computer program may be useful in small business where resources for acquisition of commercial software are scarce.*

Keywords: *cad/cam systems, computer numerical control, machining features.*

1. INTRODUCTION

Milling and turning represent a significant percentage of the machining operations performed in mechanical production industry. According to the literature, there is not a systematic machining solution. In a previous studying, we compared two strategies of machining: The first uses the contour parallel machining and the second makes use of a combination of a contour parallel and zigzag machining. In the first case, we obtained a cutting path that has 627.83 mm verses 643.07 mm in the second case. As a result of this, 2.5% difference in the cutting path length, we obtained a 7% difference in the machining pocket time (Mafalda, 1999). We show an example of the paths in Fig. 1.

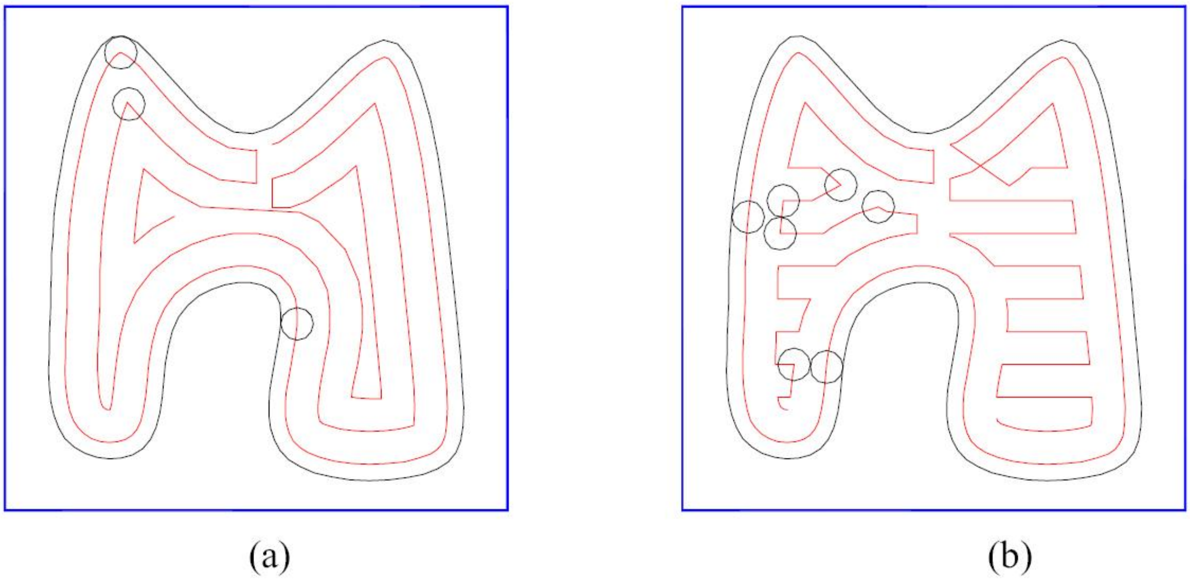


Figure 1. Studied cases - a) contour parallel machining b) hybrid approach machining

Surprisingly, there are geometric features that are repeated frequently, as the holes and pockets, in many types of mechanical parts. To machine these characteristics, computer programs can be prepared to automatically generate the numeric control code for the machines. The numerical control is a technique that involves sending instructions to a machine via a microprocessor embedded on the machine. These instructions are in the shape of numbers, letters and other symbols which can vary the position of the tool, control its speed and direction, select tools or turn the machine on or off. The instructions are received by the machine in a logically organized block of information.

Currently, the focus of the studies on the cut path creation has been their automatic generation from solid model in a computer aided design system. Although this is the model for studies on the subject, especially in Brazil and in the Great ABCD Region, it is still significant the using of technical drawings as the main element of preparing data for CNC machine tools, mainly in small and medium enterprises. On the other hand, it called attention the fact that a large amount of work that is made with aid of these machine tools are cavities whose shape can be described by straight lines or even arcs.

This paper presents a computer program built in C Language through which anyone can describe a wide variety of geometric features and can automatically obtain a corresponding two-dimensional cutting path. We use a mixed approach in the program which combines contour parallel and zig-zag machining. Parameters related to tool path generation such as the diameter of the toll used, size of the steps, cutting speed machining are informed at the beginning of the creation of each one of the program. A CNC simulator software packge guarantees the validity of the codes generated by the system.

2. CUTTING PATH CREATION

In machining processes, as we show in Figure 2, the metal or non metal parts are machined by removal of material, by the action of tools. In these processes, they are considered operational parameters such as advancement in milling and rotation of the part in turning operations (Ferraresi, 1985). In this study they are considered only the geometric aspects of the creation of the trajectory of the cutting tool, in particular, to obtain a method that describes the volume of material to be removed by machining process. For a detailed classification of these cases see, for example, the reference (Dragomatz and Mann, 1997).

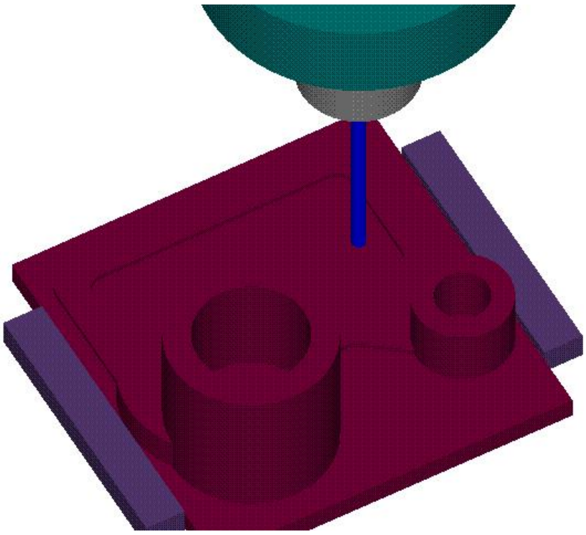


Figure 2. Machining $2^1/2D$ part

The known techniques for pockets machining can be classified according to the machining strategy and the way that the cutting path is defined. Despite the variations in the cutting path, there are only two types of strategies machining. Contour parallel machining and direction parallel machining (zig-zag), Figure 3. In the second case there is a variation called machining "zig" that occurs only in one direction to prevents the opposite direction of the CNC machine tree (spindle).

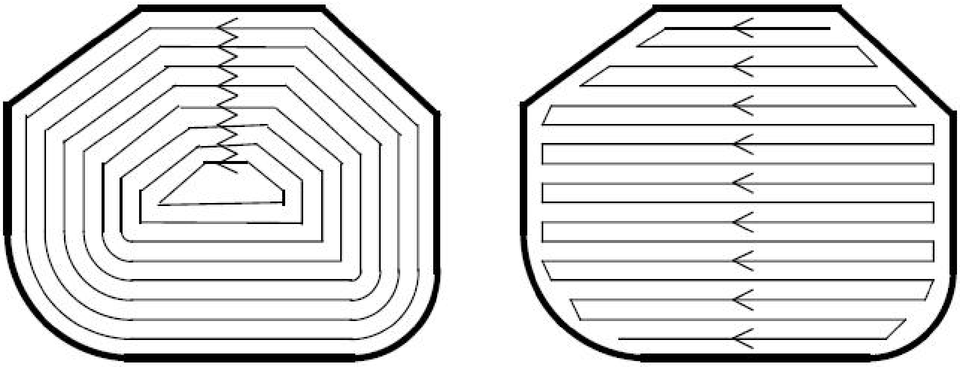


Figure 3. Contour parallel and direction parallel machinning - (Tsuzuki and Moscato, 1996)

The contour parallel machining strategy uses the concept of moving the contour elements parallel to define the cutting path. The basic idea of machining parallel to a direction is deceptively simple: after selecting an initial reference, the machining occurs parallel to this reference. The main disadvantage of this strategy is the need for a machining cycle (Tsuzuki and Moscato, 1996). We can divide the machining parallel contours strategies in two main techniques:

1. calculating the offset of the contour pocket,
2. calculating the offset of the contour pocket using Voronoi diagram.

The algorithms that implement the first technique focuses the definition of offsets succession from the original reference. This process is implemented in accordance with the following steps:

1. for each element of a contour pocket an offset element is built;
2. gaps between consecutive elements resulting from offsets are filled by connecting arcs. This can result in closed curves and in some cases, self-intersecting curves;
3. the self-intersections of the curves and parts of the then are removed. Thus, the final offset curve is determined.

The main problem in this technique is the need to determine all self-intersections. There are various proposals for this technique but theuy do not eliminate this need, in addition, often result in complex algorithms (Suh and Lee, 1990).On this issue, several articles have been published. See for example Ferreira (1993) that proposed a technique based on graph theory and, Cota and Queiroz (1993) that proposed a technique based on sub contours that are displaced parallel to the original element contour.

The technique characterized by the use of Voronoi diagram has the work of Person (1978) as initial reference. Your study is based on two steps. Initially, the pocket is divided into independent sub-areas, and then the cutting path is created directly from the sub-areas. The main idea is to select a point **P** of intersection of two consecutive contour segments of the cutting path. This point has the following properties: it has the same minimum distance of two boundary elements and is a greater distance from any other element boundary. Otherwise, it is over a locus of equidistance of the contour geometric elements considered. In the Figure 4 we show a polygon and the bisectors of the angles.

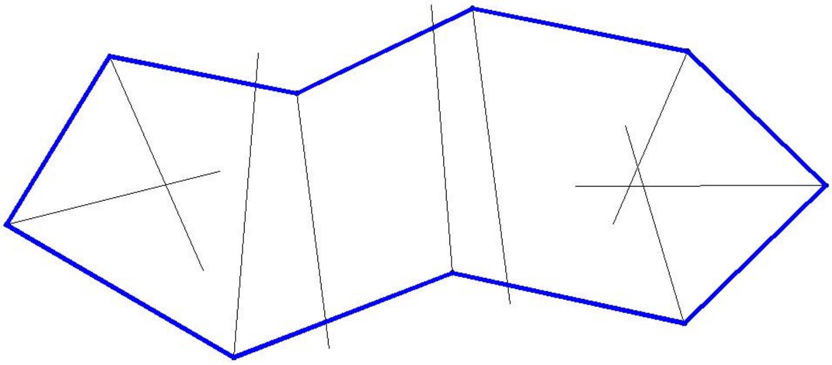


Figure 4. Angles bisectors of a polygon

Whereas these properties extend to the entire set of points, we get a graph known in computational geometry as the Voronoi diagram Fig. 5.

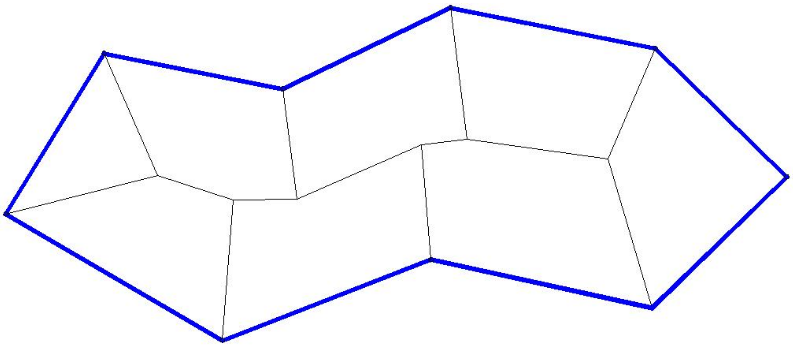


Figure 5. Voronoi diagram of a polygon

The creation of an element offset from Voronoi diagrams can be made as follows:

1. build the offset of a contour element, and
2. make the intersection of this element with the bisectors that delimit the area associated with the Voronoi segment obtained by the offset.

The idea proposed by Person is that the bisectors must be expressed by functions whose parameters represent the minimum distance from the bisector to the contour of the pocket, for example, line or arc, or simply distance of offset.Thus, the determination of the intersections of the second step of the algorithm is reduced to simple evaluation of parametrized formulas of the bisectors. The result of this process is exemplified in Fig. 6.

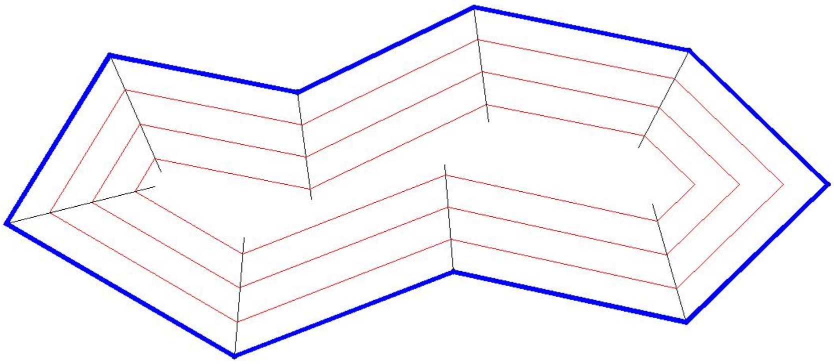


Figure 6. Parallel contours

To calculate parallel boundaries using Voronoi diagrams, there are two main methods. The first was developed by Lee (1982) in 1982 and the second, developed by Yao and Rokne (1990) in 1988. Both have focused on the representation of polygons without islands. According to Held (1991), they were the first to publish and implement the idea originally proposed by Person, according to a more fully vision. Today, many researchers agree that the best way that we know to build cutting path parallel to the contour of a pocket is by using Voronoi diagrams (Lai, 2000).

In this section we showed an overview of strategies for generating cutting path. These strategies can be applied for machining pockets with islands or without islands as are the cases treated in this study. The strategy adopted here for pocket machining is to use a mixed approach which combines parallel contour and parallel direction machining. The cutting path is constructed using offsets and calculations of intersections between straight lines, straight lines and arcs. In general, these procedures are similar to methods described above and showed in Fig. 1. With this strategy, we believe that it can be possible to obtain cutting paths with shorter extention than those generated in the traditional way. Additionally, we believe that it is possible to achieve the same levels of surface finish.

3. METODOLOGY

In this study, we first identify commonly features found in mechanical parts and subject to machining processes. The identification of these characteristics was performed using bibliographic materials and other documentary sources such as technical manuals. Then, these characteristics were grouped into a sequence of increasing of complexity as follows:

1. Polygonal pockets
2. Polygonal pockets with steps
3. Polygonal pockets with circular contours
4. Polygonal pockets with circular contours and steps

The features described above are found in different parts as feature of a finished product. After identifying these features starts the modeling phase of then in a program. In the program the input information given by the user are:

1. Type of feature to be machined
2. Dimensions: width, height and depth of the workpiece
3. Machine settings: cutting speed, diameter, placement and advancement of the tool

The results of the program is a code for the machining feature chosen, generated according to the syntax of the G code. The next sections describes examples of programs created for machining from each of the characteristics listed above.

3.1 Polygonal pockets

The first case study were the pockets with contour formed by straight edges. The strategy adopted to accomplish this routine was to move in a zig-zag along with contour parallel movement. In the Figure 7 are show an example of generating

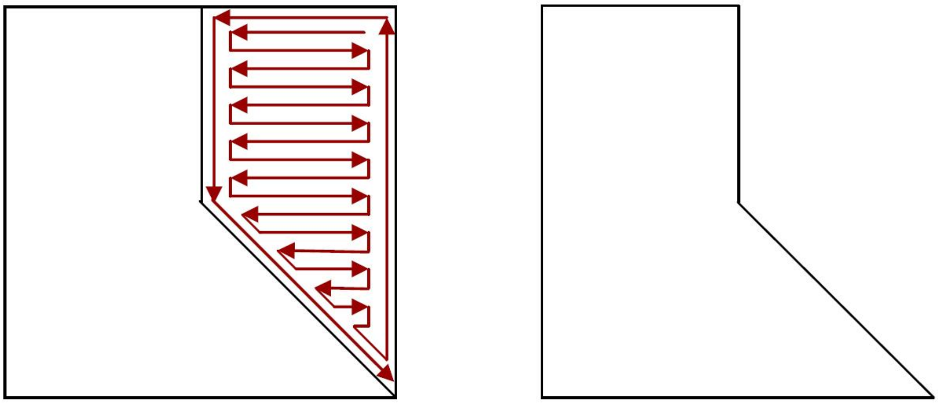
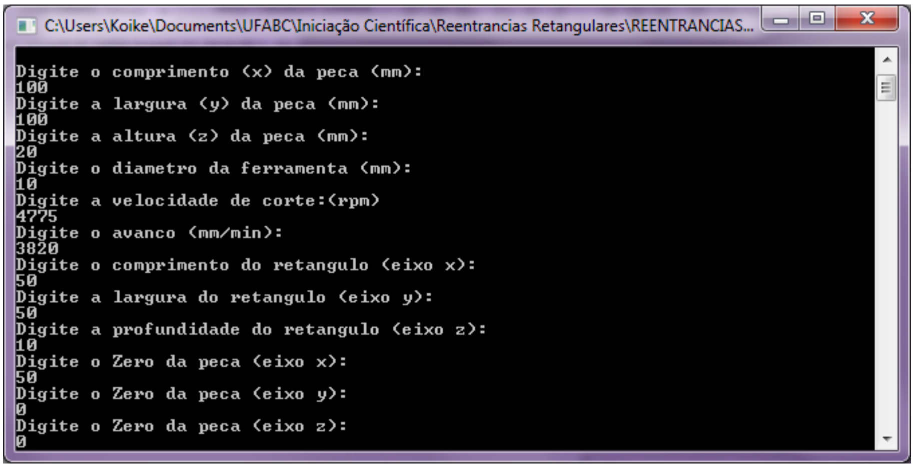


Figure 7. Polygonal pocket.

cutting path for this type of pocket. On the left side of the figure show the cutting path. On the right, the top view of the pocket.

In this example the workpiece to be machined has dimensions of 100x100x20 mm. The pockect to be produced has the shape showed in Fig. 7. The user must select the type of pocket and press the ENTER key to confirm Fig. 8. The user enters data about the operation of the machine and geometric data of the pocket to be machined, the G-code is created.



```
Digite o comprimento (x) da peca (mm):
100
Digite a largura (y) da peca (mm):
100
Digite a altura (z) da peca (mm):
20
Digite o diametro da ferramenta (mm):
10
Digite a velocidade de corte:(rpm)
4775
Digite o avanco (mm/min):
3820
Digite o comprimento do retangulo (eixo x):
50
Digite a largura do retangulo (eixo y):
50
Digite a profundidade do retangulo (eixo z):
10
Digite o Zero da peca (eixo x):
50
Digite o Zero da peca (eixo y):
0
Digite o Zero da peca (eixo z):
0
```

Figure 8. Program interface: data entry

The user must select the type of pocket and press the ENTER key to confirm. After the user enters data about the operation of the machine and geometric data of the pocket to be machined, the G-code is created by following the flowchart shown below Fig. 9.

The simulation result of the code created is show in Fig. 10. These simulation was created with the software CncSimulator [1]. It has full 2D and 3D simulation of both turning and milling machines.

3.2 Polygonal pockets with steps

The second case study were the pockets with contour formed by straight edges and steps.The difference for the first case is that in this case the user enters the two data sets. One for the first cavity and the other for the second. After the user enters data about the operation of the machine and geometric data of the pocket to be machined, the G-code is created by following the flowchart shown below Fig. 11.

We can see that the two data sets to perform the same machining operations performed in the first case.The simulation result of the code created is shown below Fig. 12.

3.3 Polygonal pocket with circular contours

The third case study were the pockets with contour formed by straight edges and arcs.The difference for the first and second cases is that in this case the user enters the G02 and G03 codes to describe the poligon. After the user enters data

---

[1] http://www.cncsimulator.com/

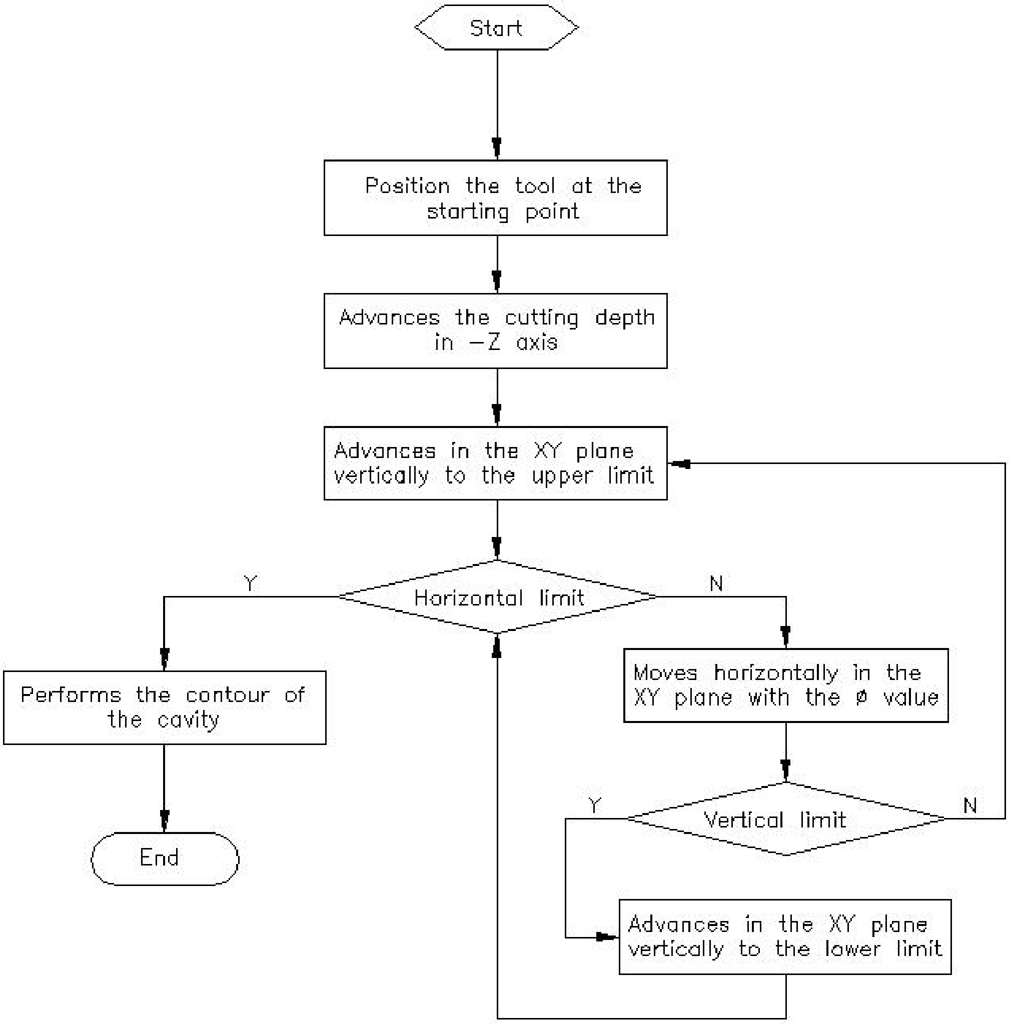


Figure 9. Flowchart of the program to construct polygonal pocket

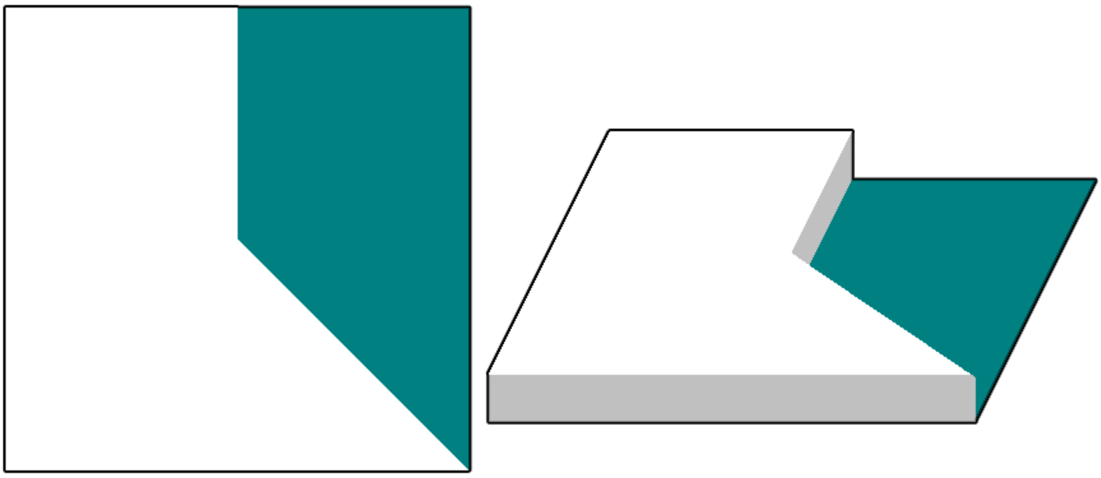


Figure 10. Polygonal pocket: milling simulation

about the operation of the machine and geometric data of the pocket to be machined, the G-code is created by following the flowchart shown below Fig. 13.

We can see that the two data sets to perform the same machining operations performed in the first case.The simulation result of the code created is shown below Fig. 14.

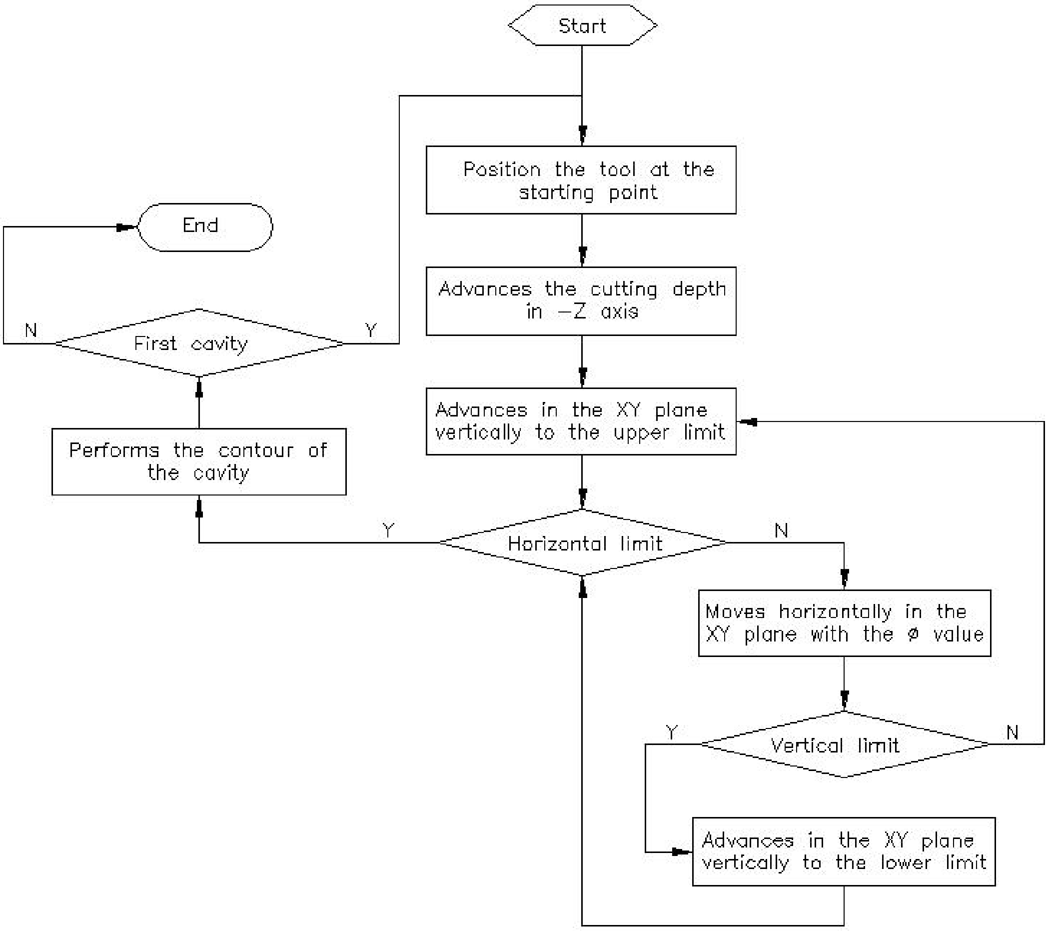


Figure 11. Flowchart of the program to construct polygonal pocket with steps

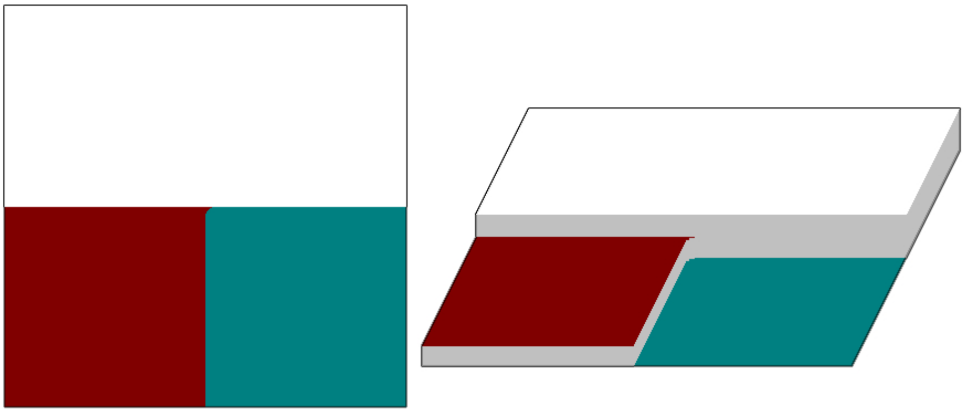


Figure 12. Polygonal pocket with steps simulation

3.4 Polygonal pocket with circular contours and steps

The fourth case study were the pockets with contour formed by straight edges, arcs and steps. Similarly in the second case the user enters the two data sets. In Figure 15 are show the graphical interface of the program. One for each depth of cut. Additionally, for each of the two sets the user apply codes G02 and G03 to describe the circular contours. After the user enters data about the operation of the machine and geometric data of the pocket to be machined, the G-code is created by following the flowchart shown in figure Fig. 17.

We can see that the two data sets to perform the same machining operations performed in the first case, second and in third cases. Another way to think about this is to imagine that the user can choose in the main menu of program the number of characteristics that want describe and either one of them bearing in mind what the outcome of the process of the conjunction of all these features. The simulation result of the code created is shown below Fig. 16.

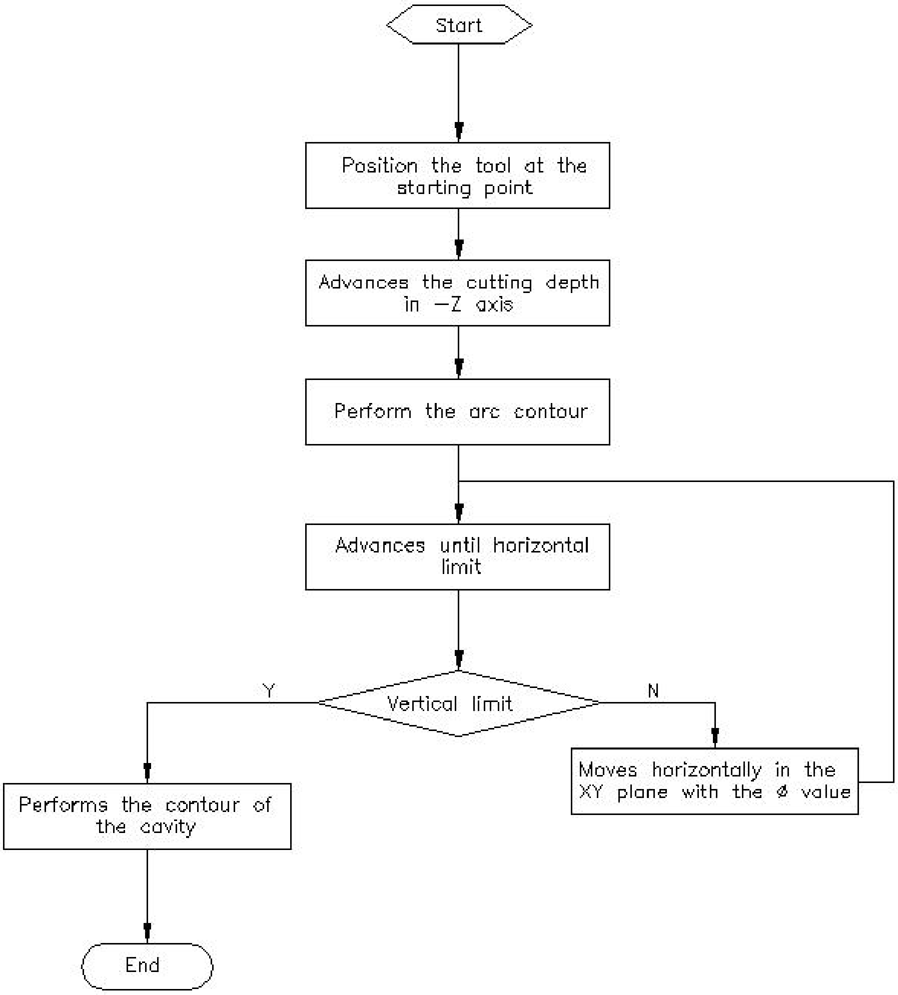


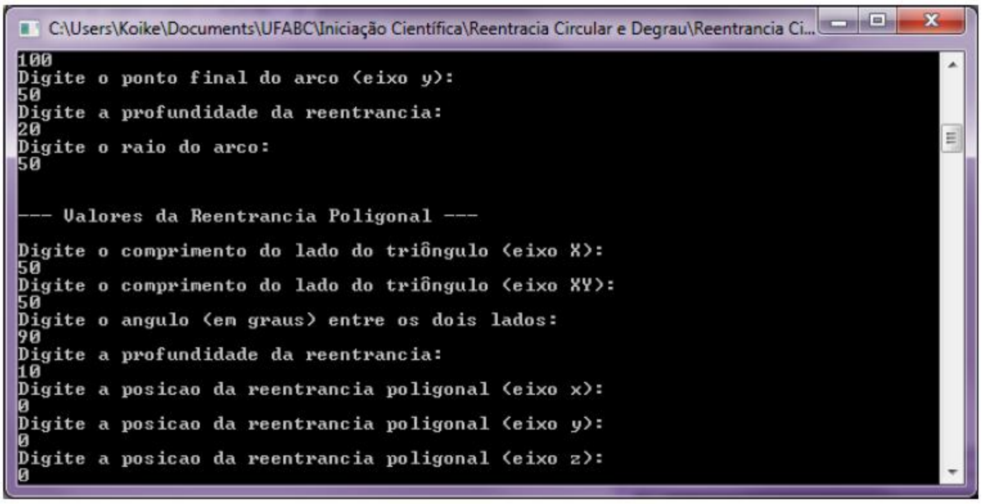
Figure 13. Flowchart of the program to construct polygonal pocket with circular contours

Figure 14. Polygonal pocket with circular contours simulation

4. FINAL REMARKS

In this study we propose a method to describe features $2^1/2D$ that occur frequently in mechanical parts and create a basic program to generate corresponding codes. We can say that this process was completed successfully once they have obtained results through simulations of machining operations created with the aid of the program. We're still working on the final definition of the interface of the program and the provision of some facilities such as libraries of raw parts in different formats. We are also working on the study of new possibilities for describing polygonal shapes.

Currently, the primary application of the program is in the activity of teaching in undergraduate courses, because of the various possibilities to be explored. We hope that in a few more years he will be able to be made available for applications in companies in which are used CNC machine tool.



```
100
Digite o ponto final do arco (eixo y):
50
Digite a profundidade da reentrancia:
20
Digite o raio do arco:
50

--- Valores da Reentrancia Poligonal ---

Digite o comprimento do lado do triôngulo (eixo X):
50
Digite o comprimento do lado do triôngulo (eixo XY):
50
Digite o angulo (em graus) entre os dois lados:
90
Digite a profundidade da reentrancia:
10
Digite a posicao da reentrancia poligonal (eixo x):
0
Digite a posicao da reentrancia poligonal (eixo y):
0
Digite a posicao da reentrancia poligonal (eixo z):
0
```

Figure 15. Program interface:pocket with circular contours and steps

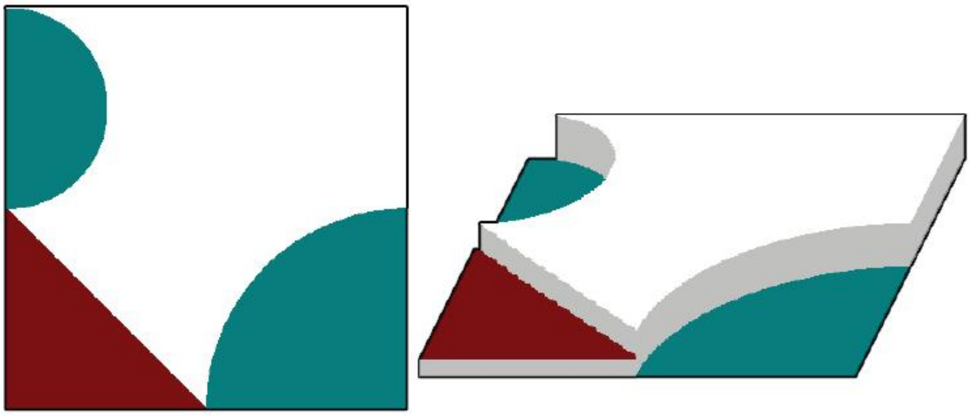


Figure 16. Pocket with circular contours and steps

We hope to help increase competitiveness through improved quality, increased flexibility (customization) and reduced cycle times. The program developed can also be deployed in enterprises to benchmark their cost benefit with commercial programs that perform these same functions.

5. ACKNOWLEDGEMENTS

FAPESP Fundação de Apoio a Pesquisa do Estado de São Paulo for supporting the first author at the beginning of the study of this subject.

6. REFERENCES

Cota, F.E. and Queiroz, A.A., 1993. “Usinagens de cavidades em formas arbitrárias em máquinas ferramentas de comando numérico”. In *Anais do XII Congresso Brasileiro de Engenharia Mecânica*. Brasília, Brasil, Vol. III, pp. 1599–1602.

Dragomatz, D. and Mann, A., 1997. “A classified bibliography of literature on nc milling path generation”. *Computer-Aided Design*, Vol. 29, No. 3, pp. 239–247.

Ferraresi, D., 1985. *Usinagem dos Metais*. Ed. Mc Graw Hill, São Paulo, Brazil.

Ferreira, J.C.E., 1993. “An algorithm for generating the tool paths for machining complex 2¡d components”. In *Anais do XII Congresso Brasileiro de Engenharia Mecânica*. Brasília, Brasil, Vol. III, pp. 1587–1590.

Held, M., 1991. *On the computational geometry of pocket machining*. Springer, Berlim, Germany.

Lai, M., 2000. “Incremental algorithms for finding the offset distance and minimum passage width in a pocket machining toolpath using the voronoi technique”. *Journal of Materials Processing Technology*, , No. 100, pp. 30–35.

Lee, D.T., 1982. *Medial axis transformation of planar shape*. IEEE Transactions Pattern Analysis Machine Intelligence.PAMI-4(4), New York.

Mafalda, R., 1999. *Comando Numérico Aplicado ao Ensino de Desenho para Engenharia*. Relatório FAPESP - proc.97/09825-8, São Paulo.

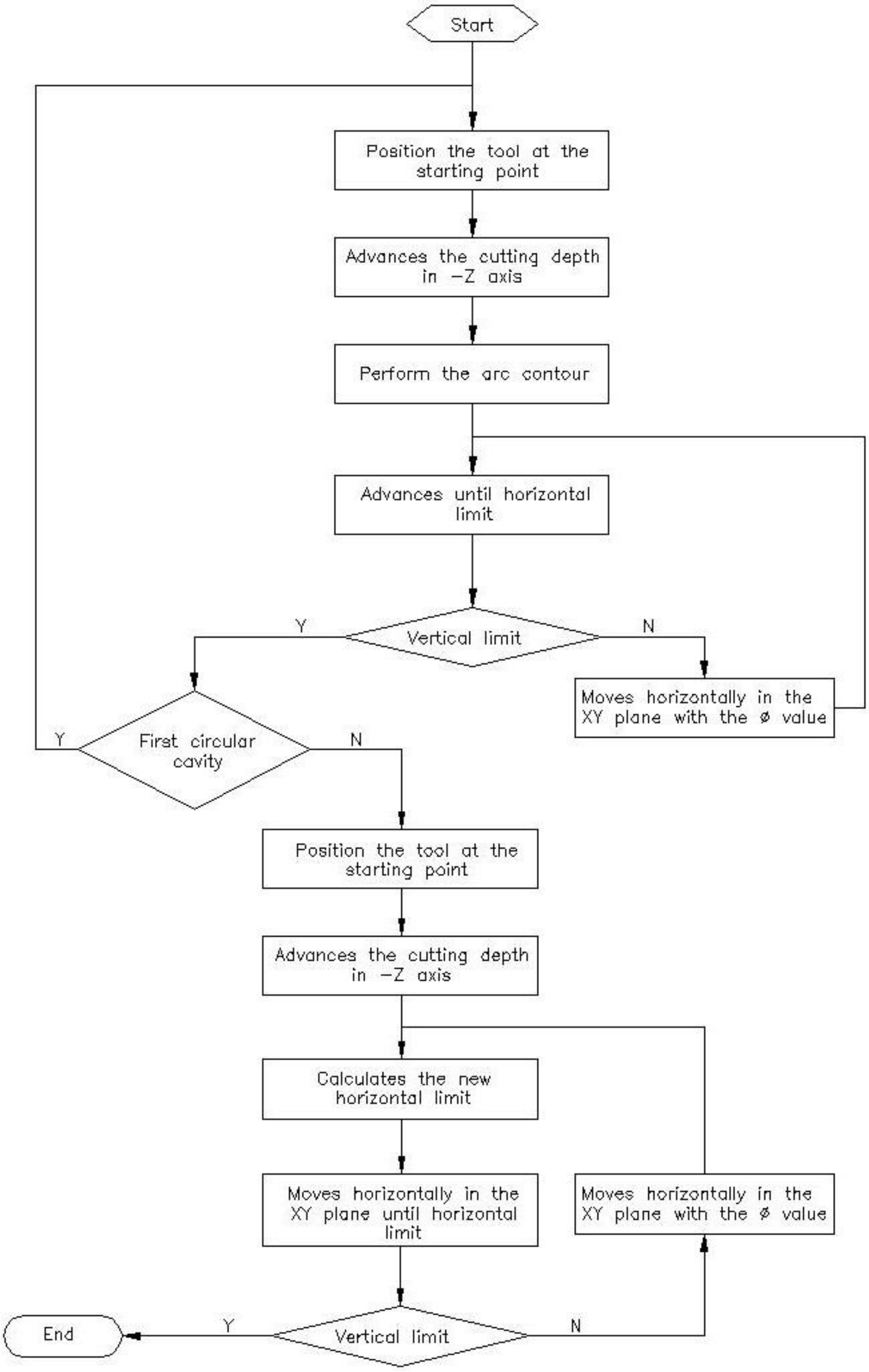


Figure 17. Flowchart of the program to construct polygonal pocket with circular contours and steps

Person, H., 1978. “Nc machining of arbitrarily shaped pockets”. *Computer-Aided Design*, Vol. 10, No. 3, pp. 169–174.

Suh, Y.S. and Lee, K., 1990. “Nc milling tool path generation for arbitrary pockets defined by sculptured surfaces”. *Computer-Aided Design*, Vol. 22, No. 5, pp. 273–284.

Tsuzuki, M.S.G. and Moscato, L.A., 1996. “Gerando o caminho de corte de reentrâncias utilizando diagramas de voronoi”. *RBCM – Journal of the Brazilian Society of Mechanical Sciences*, Vol. 18, No. 4, pp. 383–394.

Yao, C. and Rokne, J., 1990. “A straightforward algorithm for computing the medial axis of a simple polygon”. *International Journal Computing Mathematics*, Vol. 10, No. 39, pp. 51–60.

7. Responsibility notice

The author(s) is (are) the only responsible for the printed material included in this paper