INVESTIGATION FOR THE AUTOMATED GENERATION OF TOOLPATHS ON A 3AXIS CNC MACHINE

Said ZERGANE[1,2], Salah AMROUNE[1,2,*], Mohamed SLAMANI[1,2], Khalissa SAADA[1,2],Moussa ZAOUI[1,2], Chouki FARSI[1,2], ,Abdenasser BAKHTI[1]

[1] *Mechanical Department, Faculty of Technology, University of Msila, Msila Algeria.*
[2] *Materials and Structural Mechanics Laboratory (LMMS). University of M'sila. Algeria*

E-mail: salah.amroune@univ-msila.dz

***ABSTRACT**: The field of automation in tool path generation for a 3-axis CNC machine is experiencing significant growth in the computer-aided manufacturing sector. Current research efforts are focused on improving the efficiency and precision of this process. To achieve this, new technologies are being explored to enable a more advanced and automated generation of tool paths. In this article, we will examine the current state of research concerning automated tool path generation on a 3-axis CNC machine using Matlab programming tools. As an example, we will consider a complex butterfly shape derived from a mathematical function that allows drawing the 2D geometric form. At the end of the process, a G-code is automatically generated for future use in the CNC machine. The obtained results are highly encouraging, which provides further motivation for continuing research in this direction.*

***KEYWORDS:** Tool path generation; CNC machine; 3-axis; Computer-aided manufacturing; G-code*

1 INTRODUCTION

Automating toolpath generation on a 3-axis CNC machine is a crucial challenge in the manufacturing industry. Since the introduction of computer numerical control (CNC) machines in the 1950s [1], computer-aided manufacturing has experienced exponential growth. CNC machines are now utilized in a wide variety of sectors, from automotive part production to the fabrication of medical prosthetics [2]. The ability to rapidly and precisely produce complex parts is essential to remain competitive in an ever-evolving global market. Nonetheless, generating toolpaths for a 3-axis CNC machine remains a complex task that demands significant expertise. Operators must choose the appropriate tools, determine the order of operations, and plan the paths for each tool. Additionally, they must consider factors such as cutting speed, cutting depth, the geometry of the workpiece, and machine limitations [3, 4]. The complexity of this task often leads to human errors, which can result in high costs in terms of time and materials. Moreover, manual toolpath generation can limit productivity, as it requires significant time for each project. To address these issues, efforts are being made to automate the toolpath generation process [5]. In the 1960s, the first attempt to automate toolpath generation began with the introduction of computer-aided manufacturing (CAM) programs for creating toolpaths on CNC machines. These programs enabled users to generate G-code files, which controlled the machine's movements. However, these early CAM programs were rudimentary and limited in their functionality, primarily handling basic tasks [6]. Since that time, significant advancements have been made in automating the toolpath generation process. CAD/CAM (Computer-Aided Design/Computer-Aided Manufacturing) systems can now generate sophisticated toolpaths for various CNC machines, including those with 3-axis capabilities. These systems leverage advanced algorithms to optimize toolpaths based on factors such as the workpiece's characteristics, the machine's capabilities, and the available tools [7]. The automation of toolpath generation provides a key advantage, allowing for faster and more efficient production of parts. With CAD/CAM systems, toolpaths can be generated in just a few minutes, significantly reducing project preparation time. Additionally, these systems can optimize toolpaths to minimize machining time, enabling the production of a greater number of parts in a shorter timeframe [8].

In summary, the automation of toolpath generation offers significant advantages, including increased production speed and efficiency. CAD/CAM systems can now generate toolpaths rapidly, reducing project preparation time considerably. Moreover, these systems optimize toolpaths to minimize machining time, resulting in higher productivity and faster production of parts. Additionally, automating toolpath generation leads to improved accuracy and higher quality of produced parts. By considering various factors, CAD/CAM algorithms enhance the precision of the toolpaths. This advancement contributes to better overall manufacturing outcomes. The research in this field aims to further advance the automation and sophistication of toolpath generation for 3-axis CNC machines. Utilizing Matlab programming tools, the goal is to develop an automated process for creating toolpaths without extensive manual intervention. The research explores new technologies to enhance the toolpath generation process, leveraging the latest advances in computer-aided manufacturing. Moreover, the focus on complex geometric shapes, exemplified by a butterfly, demonstrates the capability to generate toolpaths for intricate designs that would be challenging to achieve manually. A key aspect of the work lies in the automation of G-code generation, which controls the CNC machine's movements. This innovative approach allows for seamless integration between the design process and actual production, streamlining the manufacturing workflow. The research's highly promising results serve as motivation for future developments in this area, encouraging further exploration and advancements in automating toolpaths for 3-axis CNC machines.

2 LITERATURE REVIEW

David Culler and William Burd [9], embarked on this research project with the aim of tackling the difficulties encountered by smaller manufacturers when integrating business functions and product manufacturing. They recognized the limited success of Computer-Aided Process Planning (CAPP) systems due to the inaccessibility and incompatibility of information residing in proprietary software. To overcome this, their primary aim was to demonstrate an architecture that seamlessly integrates customer service, CAPP, and activity-based costing (ABC) into a unified system. By doing so, they sought to enable companies to monitor expenditures and resource usage for each job, facilitating better decision-making and cost control. The authors developed a software application tailored to the needs of FEMA Industrial S.A., a local machining and fabrication shop, which allowed for practical implementation and validation of their approach. This research not only provided significant contributions to the local industry and students' education but also played a vital role in advancing the field of Computer-Aided Process Planning (CAPP). The proposed reconfigurable and reprogrammable system emphasized the potential for growth and efficiency in manufacturing processes, offering an innovative and practical solution for companies seeking to optimize their operations and improve overall productivity. Wang et al [10] propose an intelligent decision-making method for computer-aided manufacturing (CAM) numerical control (NC) programming parameters. They address the issues of low efficiency and limited intelligence caused by manual work in parameter setting. To achieve their goal, they utilize model-based definition (MBD) technology to create a hierarchical process model, expressing the process flow and managing the process information. To enhance the tightness between the process and NC machining, the authors automatically sort out NC machining steps and extract their process information. They then use a back propagation (BP) neural network to mine the experience from historical NC programming cases. Techniques like the Sparrow search algorithm (SSA) and others are employed to optimize the initial weight and threshold, learning rate, and feedback process of the BP neural network. The result is the construction of a CAM parameter decision model based on the BP neural network. The authors conduct experiments using the connecting rod of a specific type of diesel engine as the experimental object. Their method increases programming efficiency by 47% and achieves an accuracy rate of 80%, demonstrating the practicality and feasibility of their intelligent decision-making approach for CAM NC programming parameters based on MBD and neural network techniques. Barbosa et al. [11] employed CNC parametric programming to enhance manufacturing process flexibility by simulating the operation using Siemens NX7R. The simulation involved hemispherical milling and was validated using a virtual machine tool model in Siemens NX7R. Following this, the real machining process was conducted in the workshop to machine the test surface. The outcomes revealed that virtual machine tools serve as an efficient resource for simulating and verifying the performance of CNC-controlled machining processes. Accurate CNC program simulations contribute to reducing the implementation risk. Ficko et al. [12] conducted a study on the automatic programming of CNC machines. They investigated different programming methods in CNC machines and distinguished their variations. The importance and advantages of automated programming for creating complex shapes were extensively discussed. The roles of CAM and CNC programmers were specified to maximize their effectiveness.

Azahri et al [13] investigate the machining analysis of a cylindrical product transformed into a pin chuck design using Mastercam software. They aimed to understand the relationship between spindle speed and machining time and explore the influence of three key parameters: spindle speed, feed rate, and plunge rate on the machining process. To achieve their goal, the authors used Mastercam software to design the pin chuck, simulate the toolpath process, and evaluate the machining time. Carbide was selected as the cutting tool for the cylindrical material. They performed various steps, including design, lathe machine operation, toolpath processes, and more, to create the pin chuck product. After completing the toolpath processes, backplot, and verify operations were carried out to ensure accuracy. Through simulation and analysis, the authors found that spindle speed has the greatest impact on differences in machining time. They discovered an inverse relationship between spindle speed and machining time, meaning higher spindle speeds resulted in shorter machining times. The study also revealed that the finishing process had the shortest machining time compared to other toolpath types processes. Overall, the authors aimed to gain insights into the Mastercam software's capabilities for machining analysis and to optimize the machining efficiency of the pin chuck design.

3 OUR APPROCH

The objective of this article is to fully automate the generation of paths for any 2D model without requiring any human intervention. The model is described by a mathematical function, as illustrated in Figure 1. We will specifically examine a complex butterfly shape derived from a mathematical function that enables the drawing of this 2D geometric form.

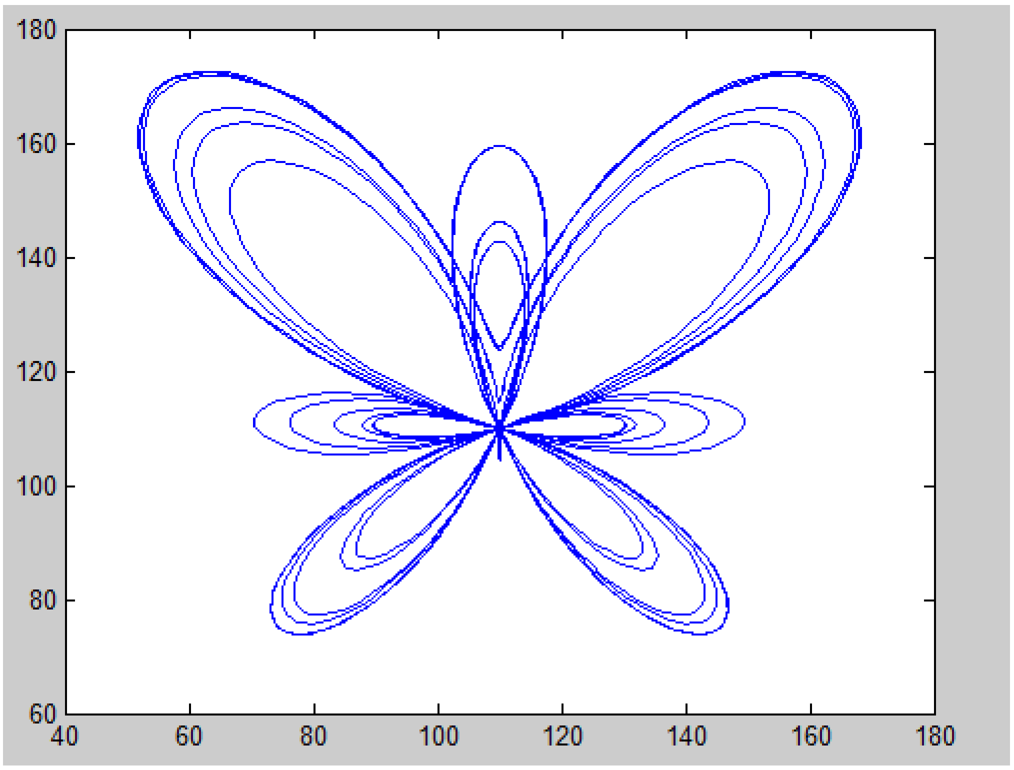


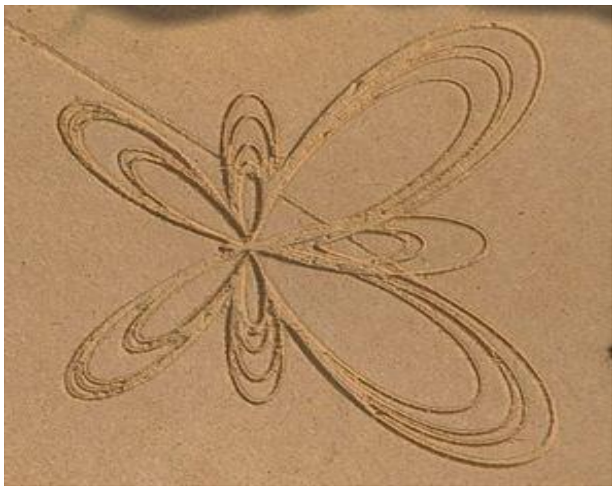
Fig.1 Drawing shapes in Matlab using a mathematical function

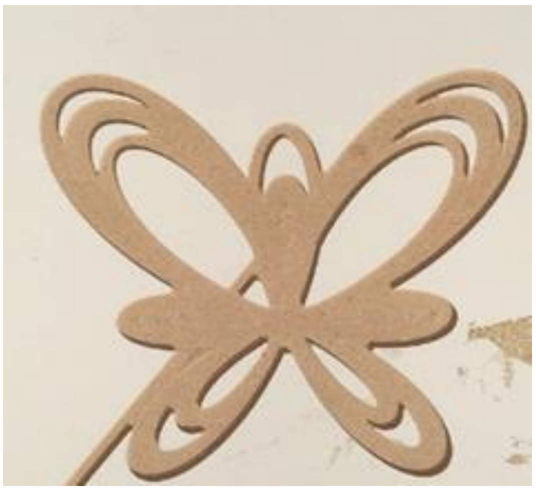
.

The process of generating 2D tool paths can be divided into several phases. Here is a general description of the process:

1. Defining the 2D model: Firstly, the 2D model is defined either using a mathematical function or specified geometric data. In the context of this article, a complex butterfly shape derived from a mathematical function is considered.
2. Discretization of the model: The 2D model is discretized into a set of points representing the geometric form. This discretization transforms the model into a series of coordinates that can be used in the tool path generation process.
3. Path planning: Once the model is discretized, the tool path is planned to traverse these points optimally. This planning aims to minimize unnecessary tool movements and ensure a smooth and coherent trajectory.
4. Tool and machining parameter selection: Depending on the material to be machined and the model's requirements, the appropriate tool is selected, and machining parameters such as cutting speed, depth of cut, etc., are defined.
5. Generation of G-code: After planning the tool path and setting the parameters, G-code is generated. G-code is a specific programming language used to control the CNC machine. This G-code contains instructions for the tool on how to move and machine the model.
6. Execution of the tool path: The generated G-code is then loaded into the CNC machine, which automatically executes the tool path, following the specified coordinates and machining parameters.
7. Verification and adjustments: After execution, the machining quality is verified, and adjustments may be made to optimize the final result.

By following this process, the automation of 2D tool path generation is achieved, enabling the creation of complex geometric shapes with high precision and efficiency.





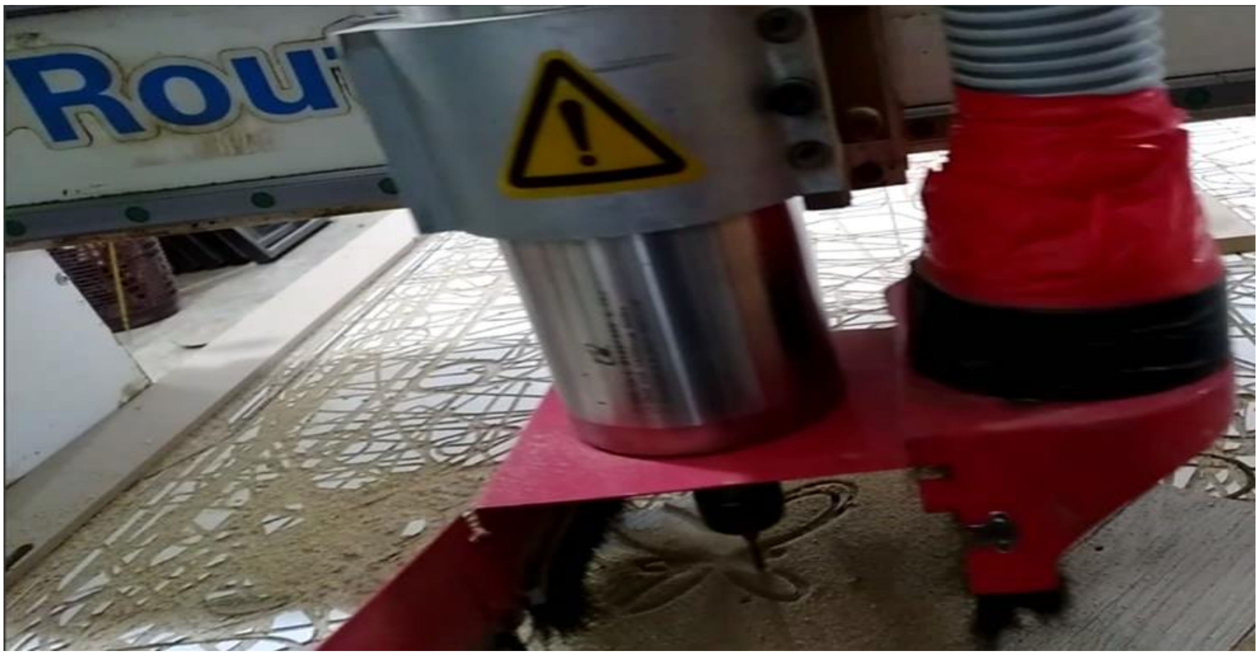


Fig. 2 Realization of the butterfly shape on a CNC machine

4 PROGRAM MATLAB

In this section, we present the source code developed in Matlab to make this offer accessible to as many users as possible. This Matlab program generates a G-code file to control a CNC machine during the machining of a 2D trajectory. Here is an explanation of how the program works:

1. The parameters of the CNC machine, such as cutting speed, feed rate, tool diameter, and depth of cut, are defined.
2. An empty G-code file named "programme_cnc.gcode" is created to record the CNC machine's instructions.
3. The 2D trajectory coordinates are imported from an Excel file named "trajectoire.xlsx". The x and y coordinates are stored in the variables x and y, respectively.
4. The program then generates the corresponding G-code commands based on the parameters and trajectory coordinates. The G-code commands control the CNC machine to move the tool along the specified trajectory.
5. The G-code file is closed after all the commands have been written.

In summary, this Matlab program automatically generates a G-code file from the coordinates of a 2D trajectory, allowing the
CNC machine to machine this trajectory with the specified parameters.

Programme Matlab

```
% Définir les paramètres de la machine CNC
cutting_speed = 200; % vitesse de coupe en mm/min
feed_rate = 100; % vitesse d'avance en mm/min
tool_diameter = 6; % diamètre de l'outil en mm
depth_of_cut = 1; % profondeur de coupe en mm
```

```
% Créer un fichier G-code vide
fileID = fopen('programme_cnc.gcode','w');

% Importer les coordonnées de trajectoire 2D
data = xlsread('trajectoire.xlsx');
x = data(:,1); % coordonnées x de la trajectoire
y = data(:,2); % coordonnées y de la trajectoire

% Générer le code G correspondant
fprintf(fileID,'G21\n'); % unités métriques
fprintf(fileID,'G90\n'); % mode de positionnement absolu
fprintf(fileID,'S%d\n', cutting_speed); % vitesse de coupe
fprintf(fileID,'T1 M6\n'); % sélection de l'outil
fprintf(fileID,'G1 F%d\n', feed_rate); % vitesse d'avance
fprintf(fileID,'G1 Z%.2f\n', -depth_of_cut); % positionnement de l'outil
for i = 1:length(x)
   fprintf(fileID,'G1 X%.2f Y%.2f\n', x(i), y(i)); % déplacer l'outil
end
fprintf(fileID,'G1 Z0\n'); % remonter l'outil
fprintf(fileID,'M30\n'); % fin du programme

% Fermer le fichier G-code
fclose(fileID);
```

5 AUTOMATED GENERATION OF TOOL PATHS

Automated generation of tool paths on a 3-axis CNC machine involves several steps:

1- - Modeling the workpiece using computer-aided design (CAD) software such as Solid Works, CATIA, or AutoCAD. This creates a 3D digital representation of the workpiece.

2- - Creating a CNC machine model in computer-aided manufacturing (CAM) software such as Mastercam, CAM Works, or Edgecam. This model contains information about the machine's dimensions, capabilities, and available tools.

3- - Defining machining strategies based on the workpiece requirements and machine constraints. This includes specifying the order of machining operations, cutting depth, feed rate, and tool rotation speed.

4- -Generating the tool path using the tool path generation algorithm. This algorithm calculates precise tool coordinates for each machining operation based on the workpiece geometry and defined machining strategies.

5- - Simulating the tool path to verify its feasibility and detect potential collisions between the tool, workpiece, or machine. This step allows for error correction before starting the machining process.

6- - Exporting the generated machining program to the CNC machine to initiate the machining process.

There are several tool path generation algorithms for 3-axis CNC machines, each with its advantages and disadvantages. Some common algorithms include:

1- - Spiral milling algorithm: Uses a spiral tool path for machining the workpiece, avoiding abrupt changes in direction and reducing tool vibrations.

2- - Zigzag milling algorithm: Utilizes a zigzag tool path for machining, reducing machining time and achieving a smoother surface.

3- - Contour milling algorithm: Follows the workpiece's contour closely for precise machining and achieving accurate finishes.

4- - Adaptive milling algorithm: Adjusts the tool path in real-time based on machining conditions, reducing machining time and extending tool life.

The choice of algorithm depends on the workpiece requirements, machine constraints, and programmer preferences.

6 GENERATING G-CODE FROM A MATHEMATICLE EQUATION

To generate G-code from a mathematical equation, you can follow the following steps using MATLAB:

1. Define the mathematical equation in MATLAB using a function. For example, if you want to generate a circle, you can use the equation x^2+y^2=r^2, where r is the radius of the circle.
2. Define the parameters of the CNC machine, such as feed rate, tool rotation speed, and the coordinates of the workpiece.
3. Use the MATLAB plot function to generate a graphical representation of the mathematical equation.
4. Use the gcode function to generate the corresponding G-code for the mathematical equation. This function takes the workpiece coordinates and CNC machine parameters as input and produces a text file containing the G-code.
5. Save the generated text file from step 4 on your computer.

Figure 3 represent the flowchart of generate G-code, the flowchart of generating G-code would show the sequence of actions and decisions required to translate a mathematical equation into the corresponding G-code. Each step would be represented by a specific symbol, such as a rectangle for a process, a diamond for a decision point, or an oval for the start and end points. The flowchart would help visualize and understand the logic and flow of operations involved in the G-code generation process.

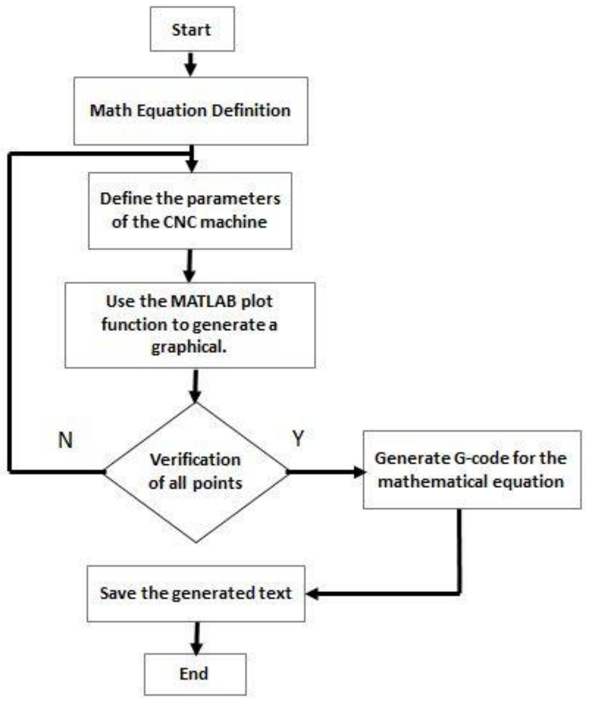
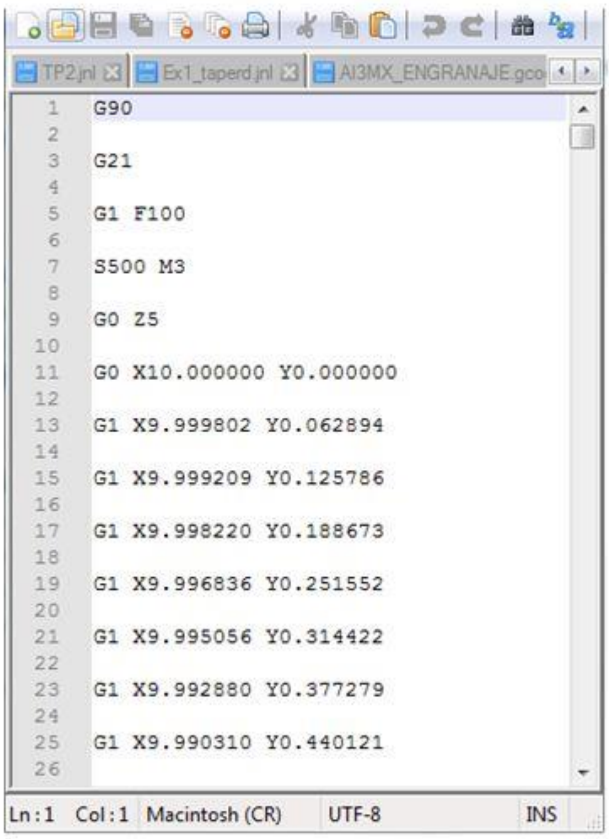
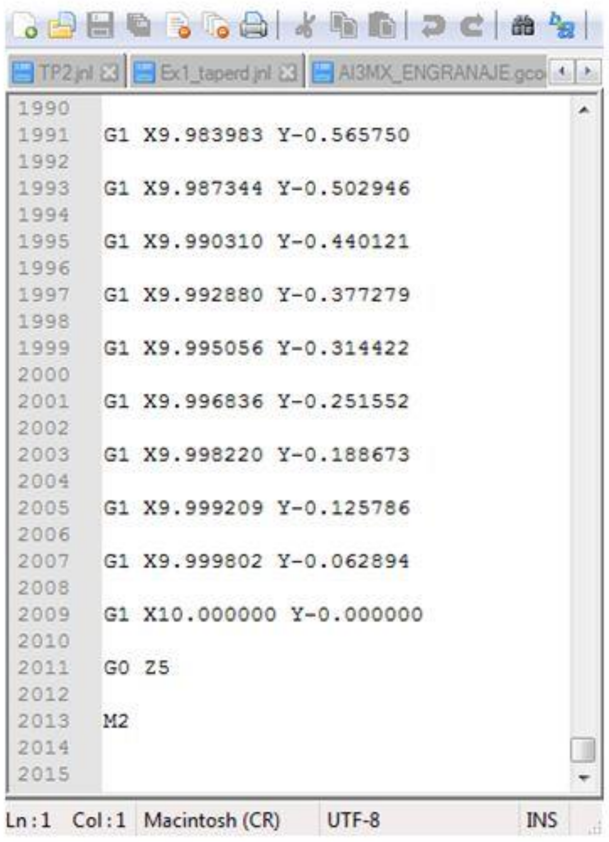

Start
Math Equation Definition
Define the parameters of the CNC machine
Use the MATLAB plot function to generate a graphical.
N
Verification of all points
Y
Generate G-code for the mathematical equation
Save the generated text
End

Fig. 3 The flowchart of generate G-code



```
G90

G21

G1 F100

S500 M3

G0 Z5

G0 X10.000000 Y0.000000

G1 X9.999802 Y0.062894

G1 X9.999209 Y0.125786

G1 X9.998220 Y0.188673

G1 X9.996836 Y0.251552

G1 X9.995056 Y0.314422

G1 X9.992880 Y0.377279

G1 X9.990310 Y0.440121
```



```
G1 X9.983983 Y-0.565750

G1 X9.987344 Y-0.502946

G1 X9.990310 Y-0.440121

G1 X9.992880 Y-0.377279

G1 X9.995056 Y-0.314422

G1 X9.996836 Y-0.251552

G1 X9.998220 Y-0.188673

G1 X9.999209 Y-0.125786

G1 X9.999802 Y-0.062894

G1 X10.000000 Y-0.000000

G0 Z5

M2
```

Fig .4 G-code program automatically generated from a Matlab program.

The automatic generation of G-code from a Matlab program represents a significant advancement in CNC manufacturing. By harnessing the power of Matlab's mathematical capabilities and automation, this approach streamlines the process of converting mathematical equations and geometrical shapes into machine-readable instructions. The automation of G-code generation offers numerous benefits, including improved efficiency, reduced human error, and enhanced precision in machining operations. It empowers manufacturers to create complex tool paths and intricate designs with ease, leading to higher productivity and superior quality in the final products.

As mentioned above, a series of codes can be compiled to generate geometric patterns using computer graphics technology. These patterns are designed to meet people's aesthetic preferences, as shown in figure 5

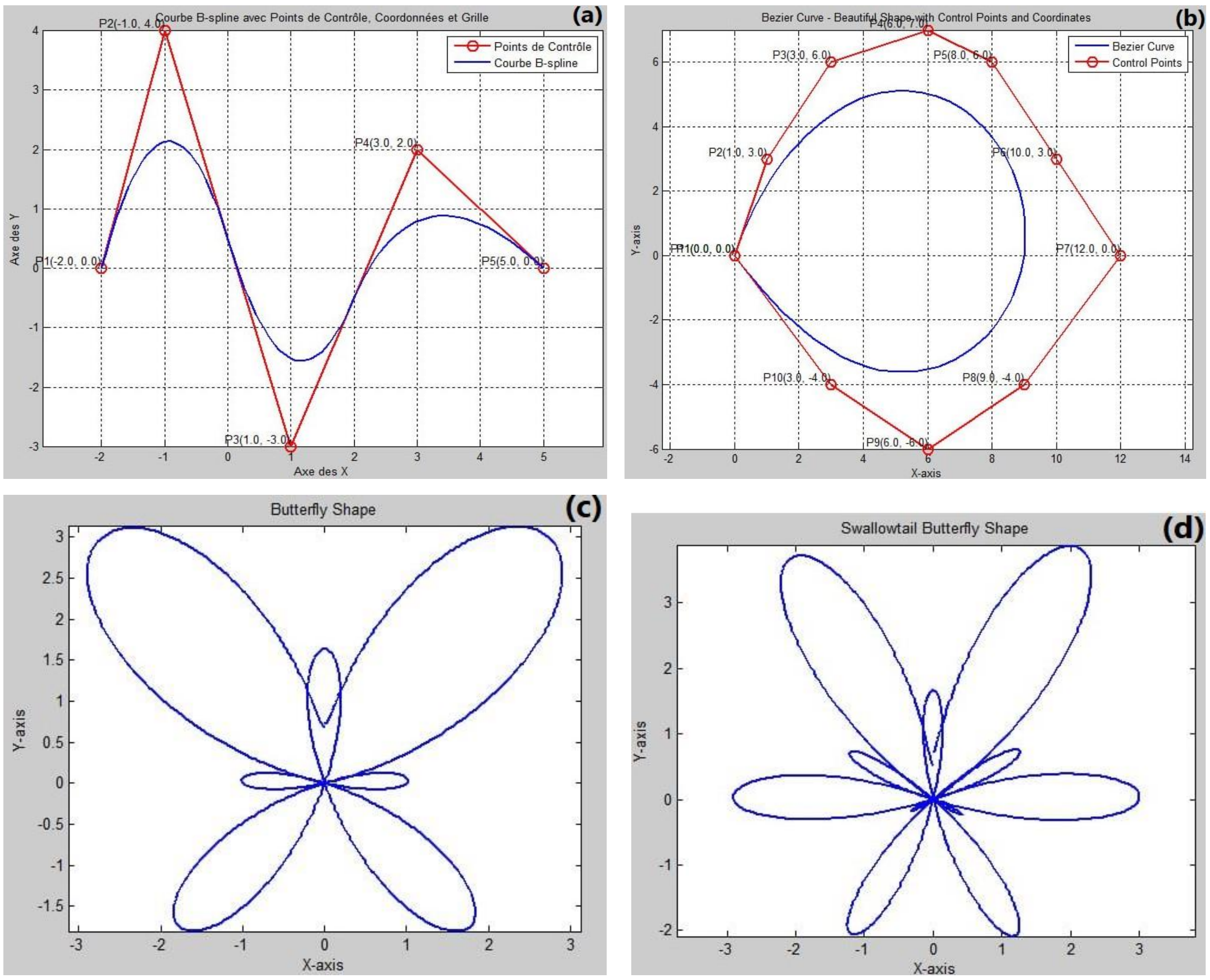


Fig. 5 The two-dimensional pattern with Bezier polynomial and Spline

Furthermore, this integration of Matlab and G-code generation simplifies the programming process, making it accessible to a broader range of users, even those without extensive programming knowledge. As a result, manufacturers can now explore a wider array of creative possibilities and optimize machining processes more effectively. As the fields of Matlab and CNC machining continue to advance, we can anticipate even more sophisticated and efficient G-code generation techniques in the future. This ongoing research promises to revolutionize CNC manufacturing, further streamlining the production process, and enabling the realization of intricate designs and complex geometries with unparalleled precision and ease. In summary, the automatic G-code generation from a Matlab program represents a major breakthrough in CNC automation, opening doors for exciting advancements in the manufacturing industry. As an illustration, the University of M'sila in Algeria utilized a CNC machine to create their acronym represented in figure 6.

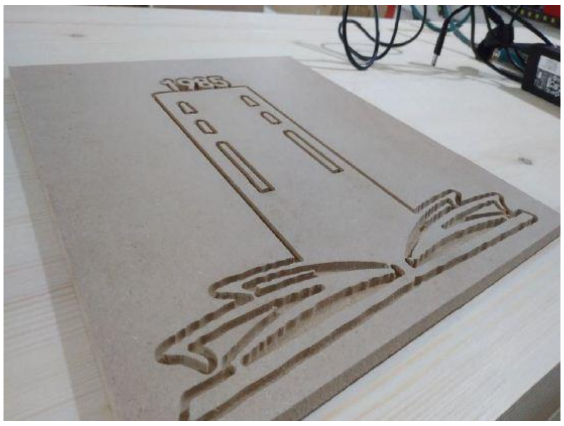


Fig. 6 Realization acronym of the University of M'sila, Algeria.

7 CONCLUSIONS

The automation of tool path generation on a 3-axis CNC machine is a crucial research area in the manufacturing industry. Advancements in CAD/CAM technologies have enabled the generation of sophisticated and optimized tool paths to minimize machining time and enhance the accuracy of produced parts. This automation offers significant benefits in terms of productivity and the quality of finished products. However, there are still challenges to overcome to make tool path generation even more efficient and precise. For instance, current CAD/CAM systems may be limited by the complexity of the part or the capabilities of the CNC machine. Additionally, investment and training costs can be high, making tool path generation automation unaffordable for some companies. Despite these challenges, automated tool path generation remains a promising research area that offers substantial advantages for manufacturers. Optimizing tool paths to minimize machining time and maximize part quality is essential for staying competitive in an ever-evolving market. By continuing to explore new technologies and improving existing systems, significant advancements in the automation of tool path generation for 3-axis CNC machines can be achieved. We have presented a comprehensive system for generating tool paths for a 3-axis CNC machine without human intervention, aiming to automate the generation of tool paths for complex 3D objects. We have demonstrated that the tool paths generated by experts and our sequence generator are similar. Overall, this system can be used for non-critical tasks by manufacturers and can assist professionals in tool path generation. We hope that our work will facilitate CNC manufacturing and make it as straightforward as printing.

8 REFERENCES

1. Minhat, M., V. Vyatkin, X. Xu, S. Wong, and Z. Al-Bayaa, *A novel open CNC architecture based on STEP-NC data model and IEC 61499 function blocks.* ***Robotics and Computer-Integrated Manufacturing***, ***2009***. **25**(3): p. 560-569**.**
2. Prashar, G., H. Vasudev, and D. Bhuddhi, *Additive manufacturing: expanding 3D printing horizon in industry 4.0.* ***International Journal on Interactive Design and Manufacturing (IJIDeM)***, ***2022***: p. 1-15**.**
3. Sheen, B.-T. and C.-F. You, *Machining feature recognition and tool-path generation for 3-axis CNC milling.* ***Computer-Aided Design***, ***2006***. **38**(6): p. 553-562**.**
4. Racz, G., R. Breaz, V. Oleksik, O. Bologa, and P. Brîndașu. *Simulated 3-axis versus 5-axis processing toolpaths for single point incremental forming*. in *IOP Conference Series: Materials Science and Engineering*. 2019. IOP Publishing.
5. Jacso, A., B.S. Sikarwar, R.K. Phanden, R.K. Singh, J. Ramkumar, and G.N. Sahu, *Optimisation of tool path shape in trochoidal milling using B-spline curves.* ***The International Journal of Advanced Manufacturing Technology***, ***2022***. **121**(5-6): p. 3801-3816**.**
6. Gan, K.C. and M.S. Abu Mansor. *High-Speed Machining for CNC Milling Simulation Using CAM Software*. in *Advances in Manufacturing Engineering: Selected articles from ICMMPE 2019*. 2020. Springer.
7. Li, X., X. Wang, J. Li, M. Zhang, M.S. Al Ansari, and B. Goyal, *Development of NC Program Simulation Software Based on AutoCAD.* ***2023*.**
8. Nassehi, A., W. Essink, and J. Barclay, *Evolutionary algorithms for generation and*

optimization of tool paths. ***CIRP Annals***, ***2015***. **64**(1): p. 455-458**.**

9. Culler, D.E. and W. Burd, *A framework for extending computer aided process planning to include business activities and computer aided design and manufacturing (CAD/CAM) data retrieval.* ***Robotics and Computer-Integrated Manufacturing***, ***2007***. **23**(3): p. 339-350**.**
10. Wang, N., S. Zhang, Z. Wang, J. Xu, and D. Liu, *Research on intelligent decision method of computer-aided manufacturing numerical control parameters based on model-based definition and back propagation neural networks.* ***Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture***, ***2023***. **237**(10): p. 1596-1607**.**
11. García Barbosa, J.A., J.M. Arroyo Osorio, and E. Córdoba Nieto, *Simulation and verification of parametric numerical control programs using a virtual machine tool.* ***Production Engineering***, ***2014***. **8**: p. 407-413**.**
12. Ficko, M., J. Balic, I. Pahole, J. Senveter, S. Brezovnik, and S. Klancnik, *Expectations of automatic programming of CNC machine tools.* ***Advances in production engineering & management***, ***2010***. **5**(3): p. 193-199**.**
13. Azahri, M.I., H. Hehsan, and M.R. Jaafar, *Machining Analysis on a Pin Chuck by Using Mastercam Software.* ***Progress in Engineering Application and Technology***, ***2022***. **3**(1): p. 729-741**.**