education sciences

MDPI

Article

Tutorials for Integrating CAD/CAM in Engineering Curricula

AMM Sharif Ullah [1,*] and Khalifa H. Harib [2,*]

[1] Faculty of Engineering, Kitami Institute of Technology, Kitami 090-8507, Japan
[2] College of Engineering, United Arab Emirates University, Al Ain 15551, UAE
[*] Correspondence: ullah@mail.kitami-it.ac.jp (A.S.U.); k.harib@uaeu.ac.ae (K.H.H.); Tel.: +81-157-26-9207 (A.S.U.)

Received: 23 August 2018; Accepted: 14 September 2018; Published: 19 September 2018

Abstract: This article addresses the issue of educating engineering students with the knowledge and skills of Computer-Aided Design and Manufacturing (CAD/CAM). In particular, three carefully designed tutorials—cutting tool offsetting, tool-path generation for freeform surfaces, and the integration of advanced machine tools (e.g., hexapod-based machine tools) with solid modeling—are described. The tutorials help students gain an in-depth understanding of how the CAD/CAM-relevant hardware devices and software packages work in real-life settings. At the same time, the tutorials help students achieve the following educational outcomes: (1) an ability to apply the knowledge of mathematics, science, and engineering; (2) an ability to design a system, component, or process to meet the desired needs, (3) an ability to identify, formulate, and solve engineering problems; and (4) an ability to use the techniques, skills, and modern engineering tools that are necessary for engineering practice. The tutorials can be modified for incorporating other contemporary issues (e.g., additive manufacturing, reverse engineering, and sustainable manufacturing), which can be delved into as a natural extension of this study.

Keywords: engineering education; CAD/CAM; digital engineering; accreditation

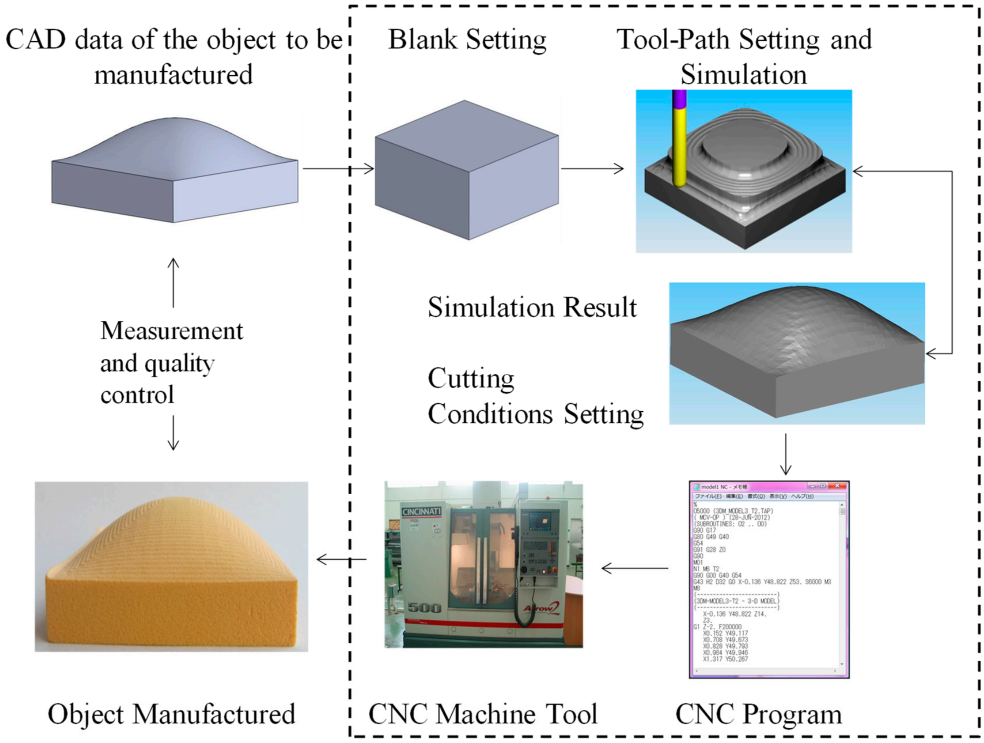
1. Introduction

Manufacturing is a wealth-generating activity, and, thereby helps ensure the economic well-being of a country. Nowadays, computer-aided hardware devices (e.g., Computer Numerically Controlled (CNC) machine tools, three-dimensional (3D) printers, robots, programmable logic controller, 3D scanners, Coordinate Measuring Machines, computerized assembly lines, and alike) and software packages for solid modeling (e.g., 3D Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM) packages) are extensively used to perform the activities related to product life cycle (conceptualize, design, manufacture, assemble, distribute, use, maintain, and recycle). Therefore, the individuals who can design, analyze, use, and maintain the hardware devices and software packages mentioned above are essential for the economic growth of a country (in this article, the above-mentioned devices and software packages are collectively referred to as CAD/CAM). As such, higher education institutes all over the world need to educate students with the knowledge and skills of CAD/CAM and other relevant issues for meeting industry needs [1–3]. Thus, engineering degree programs all over the world offer courses related to CAD/CAM [4]. Similar to the courses where the students learn engineering science (e.g., [5]), a technology-oriented course also needs customized methods and tools for the educational effectiveness and outcomes assessment (e.g., see the tools developed for the courses related to CAD [6] and courses related to the Internet of Things (IoT) [7]). This is true for the education of CAD/CAM in the academia. Some of the selected articles that deal with the educational aspects of CAD/CAM in academia are described below. Araki [8] and Zhao et al. [9]

described the contents to educate learners to use the CAD/CAM packages, which are made available through the multimedia network. deWeck et al. [10] described the students' activities and their performances while using commercially available CAD/CAM systems and other relevant software packages. The purpose is to provide students with the opportunity to materialize a conceptual design. Ishitsuka et al. [11] described the methodology to teach CAD/CAM using an exercise for creating the CNC program from a given solid model prepared by using the CAD system. Lin et al. [12] described the teaching methodology to integrate CAD with CAM for undergraduate students. Alemzadeh [13] described how to introduce CAD/CAM from the perspective of concurrent design for senior-level engineering students. Kaneko and Kashimoto [14] developed a customized computer-aided tool for educating students with the knowledge and skills of CAD/CAM. Abouelala et al. [15] described a methodology for selecting a CAM package for educating students with the knowledge and skills of CAD/CAM, considering the opinions of the stakeholders. Moschonissiotis and Vosniakos [16] developed a methodology to train individuals with the knowledge and skills of CAD/CAM where a set of structured rules and utilization of these rules in process planning are emphasized for achieving tangible results.

However, consider the issue of the assessment of students' performance. In this regard, Main and Staines [17] reported that the practical examinations are a more informative and efficacious method for assessing the students' performance in CAD/CAM. Hamade et al. [18] reported how to evaluate the skills of the senior-level students while building solid models using commercially available CAD packages. In both cases, the learners are the mere users of the available technology. Whether or not the students are aware of how the systems work and why the systems can produce the desired shape or product has not been the issue of the assessments mentioned above. This is perhaps not the goal of the outcomes-oriented education programs recommended by the accreditation bodies (e.g., ABET, http://www.abet.org). The reason is that the educational activities underlie three aspects: namely, idealism, realism, and pragmatism. The idealism and realism prepare an individual to be a "knower", whereas pragmatism prepares an individual to be "doer" only. Therefore, several outcomes are set so that the student becomes a "knower" and "doer" simultaneously. As far as the ABET-recommended outcomes are concerned, the content of a course (or a series of courses) regarding CAD/CAM should help students attain at least the following four outcomes: (1) an ability to apply the knowledge of mathematics, science, and engineering; (2) an ability to design a system, component, or process to meet the desired needs; (3) an ability to identify, formulate, and solve engineering problems; and (4) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice. At the same time, the contents must be derived from the fundamental aspects of CAD/CAM, as schematically illustrated in Figure 1. In particular, the region defined by the dotted box in Figure 1 illustrates the central issues underlying CAM. As shown in Figure 1, in CAD/CAM, the CAD data of the object to be manufactured becomes the main input. Based on this input, the user sets the blank and the cutting tool to be used. The user then selects the appropriate tool-paths (morphing, zigzag, and/or contour parallel), the cutting tool parameters (number of cutting edges, tool nose radius, tool length, and/or tool diameter), and cutting operation (rough-cut and or finish cut). Based on the selections that the user made, the CAD/CAM package simulates the tool-paths, showing their effects on the blank. When the simulation result satisfies the user (i.e., a simulation result produces the desired shape), the CAD/CAM package creates the CNC program, i.e., it formats (i.e., post-processes) the cutter locations given by the tool-paths and the cutting conditions (feed rate, cutting velocity, depth of cut, and alike) in a way so that the controller of the CNC machine tool can understand and execute the program. Once the user is satisfied with the program, it can be executed to produce the desired shape or product, as schematically illustrated in Figure 1. The digital measurement technology can be used to report the dimensional and form accuracies, as well as the surface roughness for the sake of quality control.

Based on the above-mentioned description of the educational aspects of CAD/CAM, this article presents three tutorials. Thus, the remainder of this article is organized as follows: Section 2 describes the first tutorial for calculating the tool-path for machining a 2D shape given by a parametric curve. Section 3 describes the second tutorial for determining 3D tool-path for machining a freeform surface, which is given by a parametric surface. Section 4 describes the last tutorial, where one integrates the advanced machine tools (e.g., hexapod-based machine tools) and geometric modeling of freeform surfaces. Section 5 provides the concluding remarks of this study.


CAD data of the object to be manufactured
Blank Setting
Tool-Path Setting and Simulation
Simulation Result
Cutting Conditions Setting
Measurement and quality control
Object Manufactured
CNC Machine Tool
CNC Program
model1 NC - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

```
%
O5000 (3DM_MODEL3_T2.TAP)
( MCV-OP ) (28-JUN-2012)
(SUBROUTINES: O2 .. O0)
G90 G17
G80 G49 G40
G54
G91 G28 Z0
G90
M01
N1 M6 T2
G90 G00 G40 G54
G43 H2 D32 G0 X-0.136 Y48.822 Z53. S6000 M3
M8
(--------------------------)
(3DM-MODEL3-T2 - 3-D MODEL)
(--------------------------)
   X-0.136 Y48.822 Z14.
   Z3.
G1 Z-2. F200000
   X0.152 Y49.117
   X0.708 Y49.673
   X0.828 Y49.793
   X0.984 Y49.946
   X1.317 Y50.267
```

Figure 1. The flow diagram of Computer-Aided Design and Manufacturing (CAD/CAM).

2. Tutorial for 2D Tool-Path Generation

As shown in Figure 1, one of the major tasks of CAD/CAM is to create the tool-paths. How do the CAD/CAM packages generate the tool-paths? The tutorial that helps a student understand the process of creating the tool-paths can also help achieve the outcomes (1)–(4) as mentioned above. According, this section describes a tutorial for generating a two-dimensional (2D) tool-path. For the sake of better understanding, the parametric curve-based tool offsetting method is used in the tutorial. The description is as follows:

Parametric curves are extensively used to represent a large variety of engineering shapes and parts [19,20]. The following equation defines a parametric curve denoted as $C(t)$.

$$C(t) = \sum_{i=1}^{n} P_i \times B_i(t) = [\boldsymbol{t}][\boldsymbol{M}][\boldsymbol{G}] \tag{1}$$

In Equation (1), P_i is the i-th control point, B_i (t) is the blending function associated with P_i, and t is the parameter so that $\forall\ t \in [0,1]$. As such, the matrix that is denoted as $[\boldsymbol{t}]$, which is a row matrix, is called the parameter matrix. The matrix that is denoted as $[\boldsymbol{M}]$, which is an $n \times n$ matrix, is called the shape matrix. The matrix that is denoted as $[\boldsymbol{G}]$, which is a column matrix, is called the control point, vertex, or geometric matrix.

When one models a shape using a parametric curve as defined in Equation (1), the user must decide the number of control points (n), the set of control points $\{P_i \mid i = 1,\ldots,n\}$, and the blending functions $\{B_i \mid i = 1,\ldots,n\}$. The blending functions depend on both the number of control points and the type of parametric curve (e.g., Bezier curve, B-spline, and alike). The number of control points and the type of the parametric curve decide [$\boldsymbol{M}$] and [$\boldsymbol{t}$]. For example, consider the case of a parametric curve called the quadratic B-spline ($C_{QB}(t)$). The coordinates of the three arbitrary control points (P_{p1}, P_{p2}, and P_{p3}) are needed to construct the [$\boldsymbol{G}$] for $C_{QB}(t)$. Here, the subscript "p" denotes one of the coordinates out of x, y, and z, i.e., $p \in \{x, y, z\}$. The equation of a coordinate of $C_{QBp}(t)$ is as follows:

$$C_{QBp}(t) = \begin{bmatrix} t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 0.5 & -1 & 0.5 \\ -1 & 1 & 0.5 \\ 0.5 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_{p1} \\ P_{p2} \\ P_{p3} \end{bmatrix} \tag{2}$$

As such, for a quadratic B-spline, [$\boldsymbol{t}$] = [t^2 t 1], [$\boldsymbol{G}$] = [P_{p1} P_{p2} P_{p3}]^T, and [$\boldsymbol{M}$] is a 3 × 3 matrix, as shown in Equation (2) between [$\boldsymbol{t}$] and [$\boldsymbol{G}$].

If one needs to find out the tangent of $C_{QBp}(t)$, denoted as $C'_{QBp}(t)$, [$\boldsymbol{t}$] is replaced by a matrix denoted as [$\boldsymbol{t}$]$'$. The elements of [$\boldsymbol{t}$]$'$ are the first derivative of the elements of [$\boldsymbol{t}$]. Thus, $C'_{QBp}(t)$ is given as follows:

$$C'_{QBp}(t) = \begin{bmatrix} 2t & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.5 & -1 & 0.5 \\ -1 & 1 & 0.5 \\ 0.5 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_{p1} \\ P_{p2} \\ P_{p3} \end{bmatrix} \tag{3}$$

An instructor can assign students to construct the equations of other B-spline curves using the concept of shape matrices as well as the tangent vectors. This way, the students learn more intensely how to apply the knowledge of mathematics and science to an engineering problem.

Sometimes, a single quadratic B-spline curve is not good enough to model the entire object (or any other parametric curves in this matter). In this case, the modeling is carried out by using several curves. Suppose that one models a shape using a finite set of B-spline curves that are denoted as $C_{j,QBp}(t)$, j = 1, 2, ... As such, all of the possible pairs of two consecutive curves denoted as $C_{j,QBp}(t)$ and $C_{j+1,QBp}(t)$ are connected in a piecewise manner, fulfilling the C^0 and C^1 continuity [19,20]. This means that the last point of a curve is equal to the first point of the next curve (C^0 continuity), and the slope at the last point of a curve equal to the slope at the last point of the next curve (C^1 continuity). Thus, the following proposition must be true in the modeling process:

$$(C_{j,QBp}(t=1) = C_{j+1,QBp}(t=0)) \wedge \left(C'_{j,QBp}(t=1) = C'_{j+1,QBp}(t=0)\right) \tag{4}$$

The assignments based on the formulation defined in Equation (4) help students achieve the outcomes (2) and (3), i.e., an ability to design a system, component, or process to meet the desired need, and an ability to identify, formulate, and solve engineering problems. It is worth mentioning that the students have the choices of using other parametric curves (e.g., Bezier curves) to model the given object or shape. However, basic concept remains the same, as described above.

In addition, given a 2D parametric curve, as defined by Equation (1), the students can be assigned to derive an offset curve denoted as $D(t)$, as schematically illustrated in Figure 2a. The offset curve becomes the tool-path for machining the shape (i.e., $C(t)$). The CNC program ensures that the center of the cutting tool follows $D(t)$. Thus, this information is taken as the cutter locations for constructing a CNC program (Figure 1).

Now, for determining $D(t)$, one first needs to determine the normal vector $N(t)$ for each set of two consecutive points on $C(t)$. The distance between $C(t)$ and $D(t)$ is equal to the radius of the cutting tool that is denoted as R. Thus, the expression of the offset curve $D(t)$ is as follows:

$$D(t) = C(t) + R\frac{N(t)}{\|N(t)\|} \tag{5}$$

Figure 2b schematically illustrates how one can calculate a point denoted as $D(t1)$ = ($D_x(t1)$, $D_y(t1)$)) on the offset curve $D(t)$ using two points denoted as $C(t1)$ = ($C_x(t1)$, $C_y(t1)$) and $C(t2)$ = ($C_x(t2)$, $C_y(t2)$) so that $t1$, $t2 \in$ [0,1] and $t2$ = $t1$ +Δt, where Δt is a very small positive number (e.g., 0.01).

Since the line passing through the points $D(t1)$ and $C(t1)$ and the line passing through the points $C(t1)$ and $C(t2)$ are orthogonal to each other, the coordinates of $D(t1)$ can be calculated as follows:

$$D_x(t1) = C_x(t1) \pm R\frac{C_y(t2) - C_y(t1)}{d(t1)} \quad D_y(t1) = C_y(t1) \mp R\frac{C_x(t2) - C_x(t1)}{d(t1)} \tag{6}$$

In Equation (6), $d(t1)$ denotes the distance between the points $C(t1)$ and $C(t2)$, which is calculated as follows:

$$d(t1) = \sqrt{(C_x(t2) - C_x(t1))^2 + (C_y(t2) - C_y(t1))^2} \tag{7}$$

It is worth mentioning that the students need to decide the combinations of the signs (plus–minus or minus–plus) to use the Equation (6). The combination depends on the orientations of the points denoted as $C(t1)$ and $C(t2)$ (i.e., clockwise or anticlockwise direction). The case shown in Figure 2b results in a clockwise direction. For this case, the combination of signs is minus and plus for calculating $D_x(t1)$ and $D_y(t1)$, respectively. At the same time, the student may recall the vector algebra while validating Equation (6), particularly the concepts of unit vector, vector dot product, direction cosine, and alike.

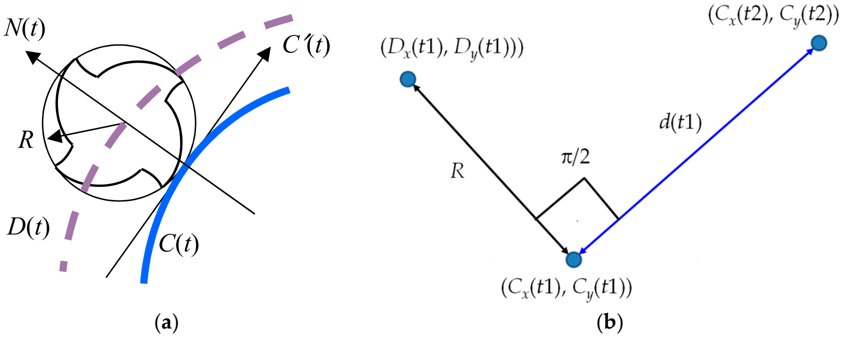


Figure 2. The concept of two-dimensional (2D) tool offsetting. (**a**) The concept of offset; (**b**) calculating a point on the offset curve.

As an exercise, the students can consider the airfoil shape shown in Figure 3. The instructor can introduce the shape to the students by plotting it on a graph paper, as shown in Figure 3. Later, the students can manufacture the part, as shown in Figure 4. In this case, the students first create the CNC program and use it for machining the part using a CNC machine tool (Figure 1). The students can be given two options to do this, as described below.

In the first option, the students reconstruct the shape shown in Figure 3 using a CAD software and input the CAD data to a CAD/CAM software for generating the CNC program. In this case, the performance of the students can be determined by an output, as shown in Figure 5. This is the screenprint of the tool-path generated by commercially available software (MasterCAM$^{\text{TM}}$) from the CAD data of the shape that is shown in Figure 3. This type of learning ensures that the students can use the techniques, skills, and modern engineering tools that are necessary for engineering practice. However, the students' ability to apply the mathematical and engineering science knowledge gained in other courses cannot be tested or enhanced by using such an exercise. Thus, the students should be given the other option to create the tool-path by applying Equations (1)–(7). The emphasis should be given to both how to model the shapes using some piecewise B-spline curves, and then to create the offset curve $D(t)$. The students should create the CNC program, too, using the information of offset curve $D(t)$. The screenprint shown in Figure 6 shows one of the expected achievements of the students. The CNC program can be observed in Figure 6. Comparing the results obtained from the commercial CAD/CAM package (Figure 5) and the spreadsheet-based analytical approach (Figure 6), the instructor can evaluate the performance of the students and explain other issues (e.g., how the tool should approach the blank for the first time, how the tool should exit once the countering is done, how to optimize the cutting conditions, how to select the optimal cutting tool, and alike).

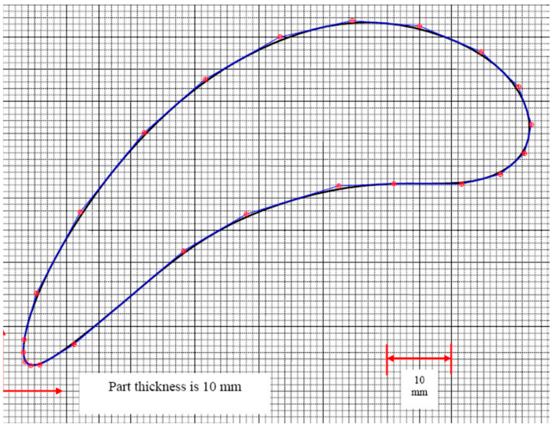


Figure 3. The schematic diagram of the shape of an airfoil.

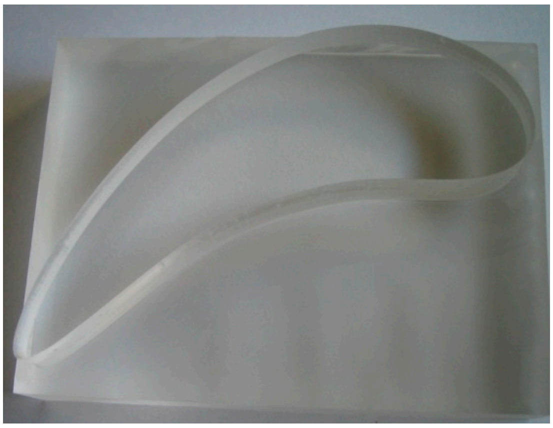


Figure 4. The shape of airfoil manufactured by Computer Numerically Controlled (CNC) machining.

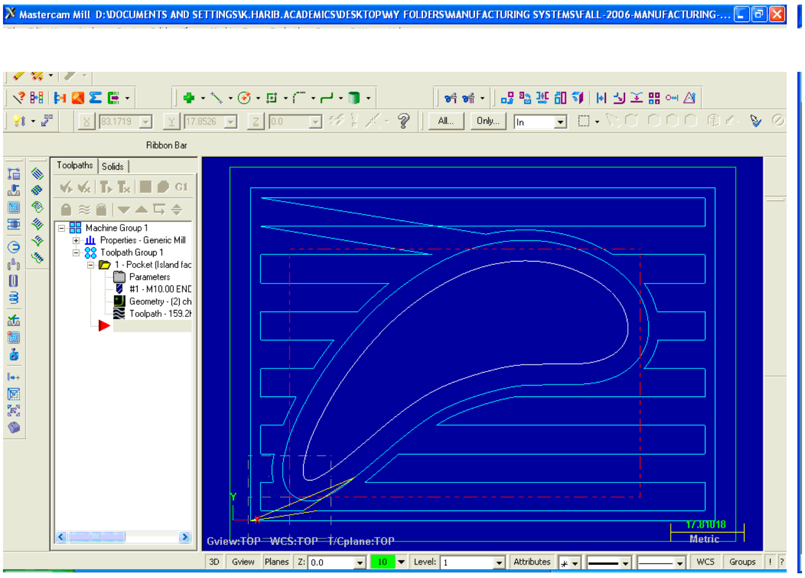
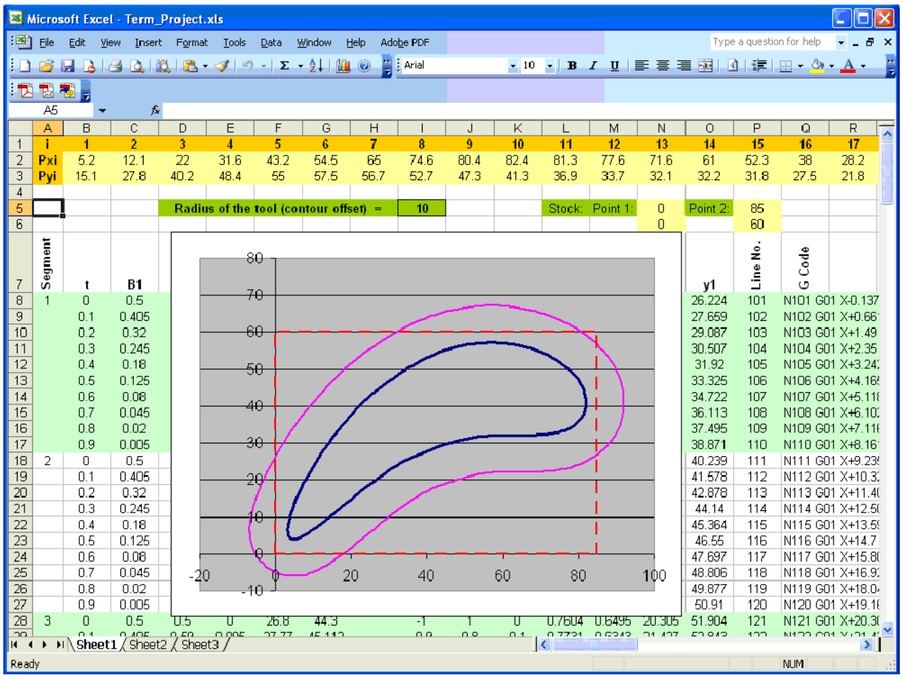


Figure 5. Tool-path generation by using a commercially available CAM software.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	**i**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
2	**Pxi**	5.2	12.1	22	31.6	43.2	54.5	65	74.6	80.4	82.4	81.3	77.6	71.6	61	52.3	38	28.2
3	**Pyi**	15.1	27.8	40.2	48.4	55	57.5	56.7	52.7	47.3	41.3	36.9	33.7	32.1	32.2	31.8	27.5	21.8
4																		
5				**Radius of the tool (contour offset) =**					**10**			Stock:	Point 1:	0	Point 2:	85		
6														0		60		
7	**Segment**	**t**	**B1**												**y1**	**Line No.**	**G Code**	
8	1	0	0.5												26.224	101	N101 G01 X-0.137	
9		0.1	0.405												27.659	102	N102 G01 X+0.66	
10		0.2	0.32												29.087	103	N103 G01 X+1.49	
11		0.3	0.245												30.507	104	N104 G01 X+2.35	
12		0.4	0.18												31.92	105	N105 G01 X+3.24	
13		0.5	0.125												33.325	106	N106 G01 X+4.16	
14		0.6	0.08												34.722	107	N107 G01 X+5.11	
15		0.7	0.045												36.113	108	N108 G01 X+6.10	
16		0.8	0.02												37.495	109	N109 G01 X+7.11	
17		0.9	0.005												38.871	110	N110 G01 X+8.16	
18	2	0	0.5												40.239	111	N111 G01 X+9.23	
19		0.1	0.405												41.578	112	N112 G01 X+10.3	
20		0.2	0.32												42.878	113	N113 G01 X+11.4	
21		0.3	0.245												44.14	114	N114 G01 X+12.5	
22		0.4	0.18												45.364	115	N115 G01 X+13.5	
23		0.5	0.125												46.55	116	N116 G01 X+14.7	
24		0.6	0.08												47.697	117	N117 G01 X+15.8	
25		0.7	0.045												48.806	118	N118 G01 X+16.9	
26		0.8	0.02												49.877	119	N119 G01 X+18.0	
27		0.9	0.005												50.91	120	N120 G01 X+19.1	
28	3	0	0.5	0.5	0	26.8	44.3		-1	1	0	0.7604	0.6495	20.305	51.904	121	N121 G01 X+20.3	

Figure 6. Tool radius offsetting using CAD/CAM package and spreadsheet application.

3. Tutorial for 3D Tool-Path Generation

This section describes a tutorial for 3D tool-path generation. This is an advanced topic compared to the previous one. The 3D tool-paths are an integral part of modern manufacturing, because freeform parametric surfaces are extensively used to model a large variety of shapes [19,20]. For constructing a freeform surface, a surface patch is generated from the product of two sets of parametric curves for two parameters, u and v. Therefore, the equation of a freeform parametric surface denoted as $C(u,v)$ is as follows:

$$C(u,v) = \sum_{i=1}^{n} \sum_{k=1}^{m} P_{ik}\ B_{ik}(u) B_{ik}(v) \quad u,v \in [0,1] \tag{8}$$

In Equation (8), P_{ik} are the control points or vertices, and $B_{ik}(u)$ and $B_{ik}(v)$ are the blending functions. The expression of $C(u,v)$ in matrix form is as follows:

$$C(u,v) = [\boldsymbol{u}][\boldsymbol{M}][\boldsymbol{P}][\boldsymbol{M}]^T[\boldsymbol{v}] \tag{9}$$

In Equation (9), $[\boldsymbol{u}]$ is a row vector having n elements, $[\boldsymbol{M}]$ is an $n \times m$ shape vector, $[\boldsymbol{P}]$ is an $n \times m$ vector of one of the coordinates of the control points, and $[\boldsymbol{v}]$ is a column vector having m elements.

For a cubic Bezier surface [19,20], the freeform surface denoted as $C_{CB}(u,v)$ is given by the following equation:

$$C_{CB}(u,v) = \left[u^3\ u^2\ u\ 1\right][\boldsymbol{M}]_{CB}[\boldsymbol{P}]_{CB}[\boldsymbol{M}]_{CB}^T\left[v^3\ v^2\ v\ 1\right]^T \tag{10}$$

In Equation (10), the shape matrix $[\boldsymbol{M}]_{CB}$ is given as follows:

$$[\boldsymbol{M}]_{CB} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{11}$$

Now, the question is: how to create the tool-path to machine a surface similar to the one defined in Equation (10)? The machining of the surfaces similar to the one defined in Equation (10) is often done by using a ball-nose end-mill cutter, as schematically illustrated in Figure 7. The center of the ball-nose moves along an offset surface denoted as $D(u,v)$ (Figure 7). As shown in Figure 7, generating the offset surface $D(u,v)$ means that each point on $C(u,v)$ is mapped into its corresponding point on $D(u,v)$ by shifting it along the normal direction of $N(u,v)$. This requires two tangential vectors that intersect at the point, as shown in Figure 7. Thus, the expressions of the tangent vectors in the u and v directions, respectively, are as follows:

$$C'_u(u,v) = \frac{\partial C(u,v)}{\partial u} \tag{12}$$

$$C'_v(u,v) = \frac{\partial C(u,v)}{\partial v} \tag{13}$$

The normal vector $N(u,v)$ is thus obtained from the cross-product of the two tangential vectors resulting in the following expression:

$$N(u,v) = C'_v(u,v) \times C'_u(u,v) \tag{14}$$

Once the normal direction is determined, the offset surface is determined similarly to the previous case, which is as follows:

$$D(u,v) = C(u,v) + R\,\frac{N(u,v)}{\|N(u,v)\|} \tag{15}$$

In Equation (15), R is the radius of the ball-nose of the end-mill cutter.

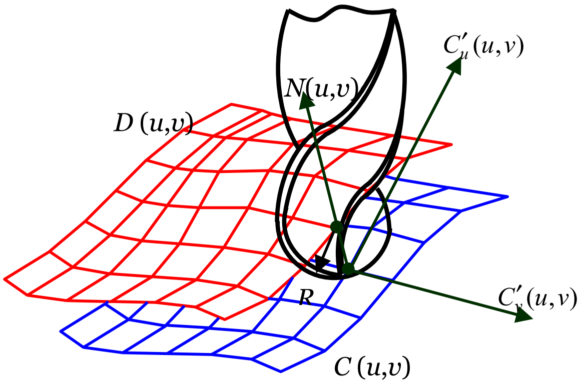


Figure 7. Tool offsetting for machining a freeform surface using a ball-nose end-mill cutter.

To exercise the above-mentioned offsetting technique for freeform surfaces, the students can use a programming tool to generate the codes required for the CNC program. Appendix A shows the required program written in MATLAB$^{\text{TM}}$ in terms of two functions. The function shown in Figure A1 creates the offset surface from the given Bezier surface in accordance with Equations (10)–(15). It also plots the Bezier and offset surfaces. On the other hand, the function shown in Figure A2 is used for generating the codes (G-codes) for the CNC programming. It writes the code to a text file, which can be edited based on the requirement of the given CNC controller. Needless to say, the function shown in Figure A2 takes the necessary outputs from the function shown in Figure A1.

Figures 8 and 9 show the expected outcomes of the tutorial. In particular, Figure 8 shows the given and offset surfaces, which is the output of the function shown in Figure A1. For this case, the coordinates of the control points are shown below in the form of matrices.

$$[P_x] = \begin{bmatrix} 0 & 20 & 40 & 60 \\ 0 & 20 & 40 & 60 \\ 0 & 20 & 40 & 60 \\ 0 & 20 & 40 & 60 \end{bmatrix} \text{mm}, \ [P_y] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 20 & 20 & 20 & 20 \\ 40 & 40 & 40 & 40 \\ 60 & 60 & 60 & 60 \end{bmatrix} \text{mm}, \ [P_z] = \begin{bmatrix} 0 & -15 & -15 & 0 \\ 0 & 0 & 0 & 0 \\ -10 & 0 & 0 & -10 \\ -10 & 10 & -10 & 0 \end{bmatrix} \text{mm}$$

Figure 9 shows the freeform surface manufactured by using a three-axis CNC machine tool (Figure 1). The sharp marks seen on the manufactured freeform surface are the tool-path marks. This is due to the coarseness of the increment of u and v (the increment was 0.05) and tool diameter (offsetting for a 4-mm radius of the ball-nose end-mill cutter). The students can be assigned to make necessary changes in the formulation to reduce the sharpness of the marks. Similar to the previous tutorial, the students can achieve all of the outcomes (the outcomes denoted as (1)–(4)) mentioned above by solving the problems underlying this assignment.

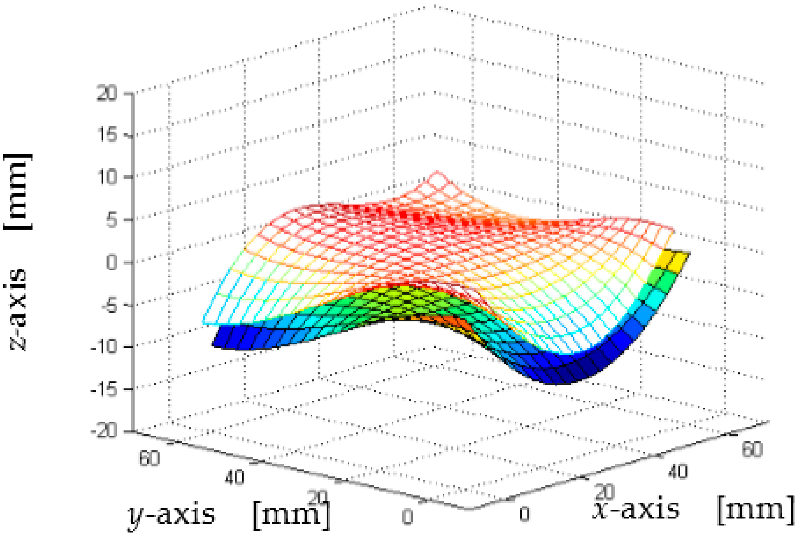


Figure 8. The given freeform surface it its offset.

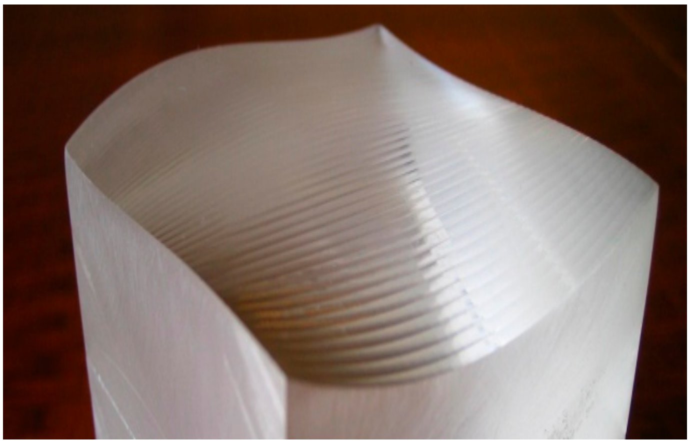


Figure 9. Manufactured freeform surface using CNC machining.

4. Programming Advanced Machine Tools

The previous two tutorials are meant for educating students with the knowledge and skills of CAD/CAM technology using conventional machine tools with a serial kinematics structure. This is not the end. The field is evolving very fast. From this perspective, the advances in the CNC machine tools can be considered. In particular, as a replacement of the conventional CNC machine tools (having a serial kinematics structure), researchers have been studying the CNC machine tools that have parallel and hybrid kinematics structures [21,22]. These machine tools are more economical, light, and functional compared to the conventional ones. Moreover, in the near future, these machine tools are expected to play their role in manufacturing.

The previous two tutorials are meant for educating students with the knowledge and skills of CAD/CAM technology that can be implemented using the conventional machine tools. Note that the conventional machine tools have a serial kinematics structure. These machine tools are by nature heavy and less economical. However, researchers have been studying a new type of CNC machine tools that have parallel or hybrid kinematics structures [21,22]. These machine tools are more economical, light (portable), and functional compared to the conventional ones, and thereby are expected to play their roles in the near future. Therefore, the students majoring in manufacturing, mechatronics, and robotics are likely to engage in scholarly activities where the parallel and hybrid kinematics' structure-driven machine tools are conceptualized, analyzed, designed, and implemented. As a result, the relevant senior-level undergraduate and graduate-level courses must prepare the students with the knowledge and skills of parallel and hybrid kinematics structures. Based on this contemplation, this section presents a tutorial that is meant for educating interested students with the knowledge of how to program a parallel kinematics machine tools for machining a 3D shape. The description of the tutorial is as follows:

Consider CNC machine tools [21] that have parallel kinematics structures, as shown in Figure 10. As seen in Figure 10, the cutting motion is controlled by six extendable linear actuators that carry the cutting tool platform. See Harib et al. [21] for more detail on the kinematics analysis and the trajectory planning. However, for the *i*-th actuator, the extension l_i is computed as follows:

$$l_i = \left\| \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} c^2A + cB\,s^2A & sA\,cA - cB\,sA\,cA & sB\,sA \\ cA\,sA - cB\,cA\,sA & s^2A + cB\,c^2A & -sB\,cA \\ -sA\,sB & cA\,sB & cB \end{bmatrix} \begin{bmatrix} x_{ai} \\ y_{ai} \\ z_{ai} \end{bmatrix} - \begin{bmatrix} X_{bi} \\ Y_{bi} \\ Z_{bi} \end{bmatrix} \right\| \tag{16}$$

In Equation (16), s and c denote the sine and cosine functions, respectively; $(X_{bi}, Y_{bi}, Z_{bi})^T$ is the constant position vector of the i-th attachment point b_i with respect to the world coordinate frame W; $(x_{ai}, y_{ai}, z_{ai})^T$ is the constant position vector of the i-th attachment point a_i with respect to the cutting tool coordinate frame C; $(X, Y, Z)^T$ is the position vector of the tip of the cutting tool with respect to W; and A and B are the angles that define the orientation of the cutting tool with respect to W; A is the horizontal rotation of the vertical plane that contains the cutting tool axis, and B is the tilting angle of the cutting tool inside that plane.

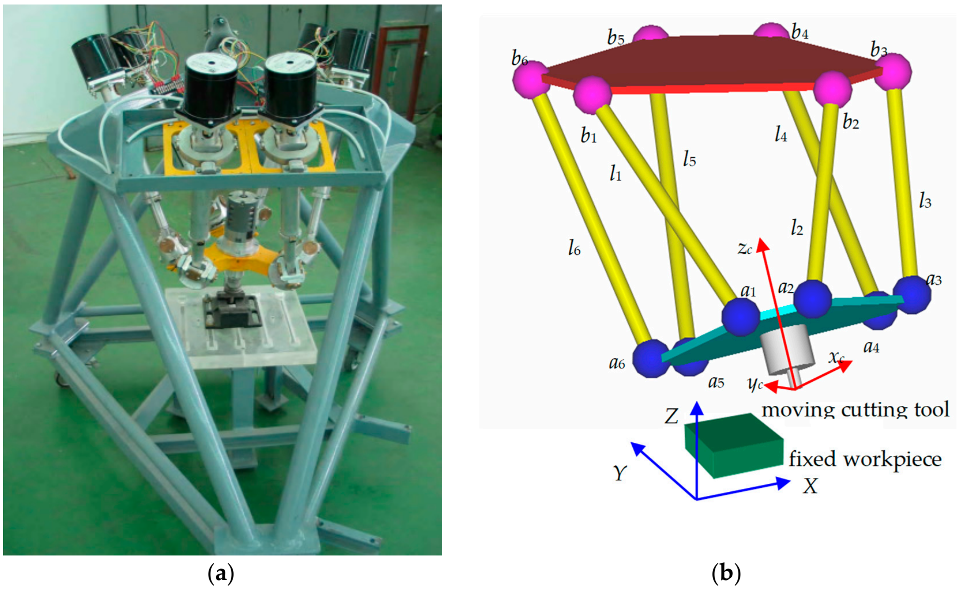


Figure 10. A parallel kinematics-driven machine tool structure [21]. (**a**) prototype; (**b**) operational parameter.

To machine a freeform surface similar to the one described in the previous section, the following procedure can be used.

Step 1: Calculate X, Y, and Z from the three coordinates of the parametric surface $C(u,v)$.

Step 2: Calculate the normal vector *N*(*u*,*v*) according to Equation (14).

Step 3: Calculate the unit vector along the cutting tool *n*(*u*,*v*) as follows:

$$n(u,v) = \frac{N(u,v)}{\|N(u,v)\|} \tag{17}$$

The third column of the rotation matrix in Equation (16) is the unit vector along the axis of the cutting tool. This means that:

$$n(u,v) = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}^T = \begin{bmatrix} sBsA & -sBcA & cB \end{bmatrix}^T \tag{18}$$

Step 4: Determine *A* and *B*, as follows:

$$A = \tan^{-1}\left(\frac{n_x}{\sqrt{n_x^2+n_y^2}}, \frac{-n_u}{\sqrt{n_x^2+n_y^2}}\right) \tag{29}$$

$$B = \tan^{-1}\left(\sqrt{n_x^2+n_y^2}, n_z\right) \tag{20}$$

It is worth mentioning that if the students use a package of five-axis milling, they will be able to determine (*X*,*Y*,*Z*,*A*,*B*) without any effort provided that a flat-end mill is used in the programming. However, if the students follow the above steps, they realize the whole process without any support of a commercial package. This way, the students having more advanced analytical abilities in CAD/CAM can be nurtured.

5. Concluding Remarks

The authors have been offering CAD/CAM and related courses for a long time for both undergraduate and graduate students of engineering degree programs. The authors have been observing that if the commercial packages are used for creating the tool-paths automatically, the students can gain the application ability, but they fail to gain the analytical ability, i.e., they become doers, but not knowers. On the contrary, the students can achieve the same using the presented tutorials where they do not need to use any dedicated CAD/CAM packages. This means that the presented tutorials contribute positively to the objective of strengthening students' ability to apply the knowledge of mathematics and engineering science, making students both doers and knowers together. Students also learn that designing a shape using CAD software is not everything. If the design is not converged to a CNC program, the materialization of the shape does not occur. This way, the students become aware of the activities of a product life cycle more intensely, which should be the case from the perspective of their future employers. In the presented tutorials, the students are introduced to the concept of parametric curves and surfaces. This improves their knowledge of dealing with the shapes having complex geometry. In addition, the students are exposed to hexapod-based machine tools. This makes the students aware of the CNC technology advancing rapidly. Therefore, it is essential for the students to understand which module of a system of CAM (or CAD) affects the operations of such machine tools from the perspective of the structure of a machine tool, as well as from the perspective of the shapes of the parts to be manufactured. Moreover, in the presented tutorials, the students get the opportunity to use standard computing and visualizing tools (e.g., Excel$^{\text{TM}}$ and MATLAB$^{\text{TM}}$). This makes them competent enough while using these tools in other courses, too. However, the instructors or the students have the freedom to use any computing platform and programming languages they prefer, although the descriptions here show the usages of spreadsheet and MATLAB$^{\text{TM}}$ coding. Having said that, it is not meant that the mathematical formulations presented in this article should not be obeyed. The first two tutorials can be offered to the senior-level undergraduate students enrolled in mechanical, manufacturing, and industrial engineering degree programs, whereas the other tutorial is recommended for the students who want to design mechatronics devices. While offering the tutorials, the authors recommend the approach called thematic integration or problem-based learning (PBL). In the case of thematic integration, the instructor may offer the tutorial in line with the contents of other courses (e.g., engineering design or mechanics course), so that the geometric shapes used in the other relevant course can be used in the proposed tutorials. While offering the tutorials as PBL, the students should be given a chance to form a team. The team then sets the objectives and outcomes based on the contents of the tutorials. Finally, the students present their teamwork for the evaluation.

Nevertheless, the presented tutorials can be modified by incorporating more outcomes (e.g., contemporary issues, teamwork, ethics, and alike) and the contents that are needed for mastering the digital manufacturing technology. Particularly, the issue of functioning effectively in a multidisciplinary team, and being aware of the professional and ethical responsibility, can also easily be accommodated more explicitly with the presented tutorials if the relevant degree program needs it. Similarly, the tutorials can easily be modified to accommodate the issues of sustainability and sustainable manufacturing [23], reverse engineering [24], design for additive manufacturing of complex shapes [25], and e-making [26]. At the same time, the issues of meaningful learning using a set of concept maps [27] can be considered. It is worth mentioning that concept maps [27] can represent the contents of the tutorials described in the article, and the students can even access them using their cell phones [4]. Since the concepts underlying any topic of manufacturing are ultimately linked to the universal concept maps, namely, the sustainability universe, the control universe, the material universe, the precision universe, and the process universe [4], one must be careful about this linkage while representing the tutorials using a set of concept maps. Otherwise, the desired learning objectives may not be materialized.

Author Contributions: Both authors, A.S.U. and K.H.H., equally contributed in all sections of this article.

Funding: This research received no external funding.

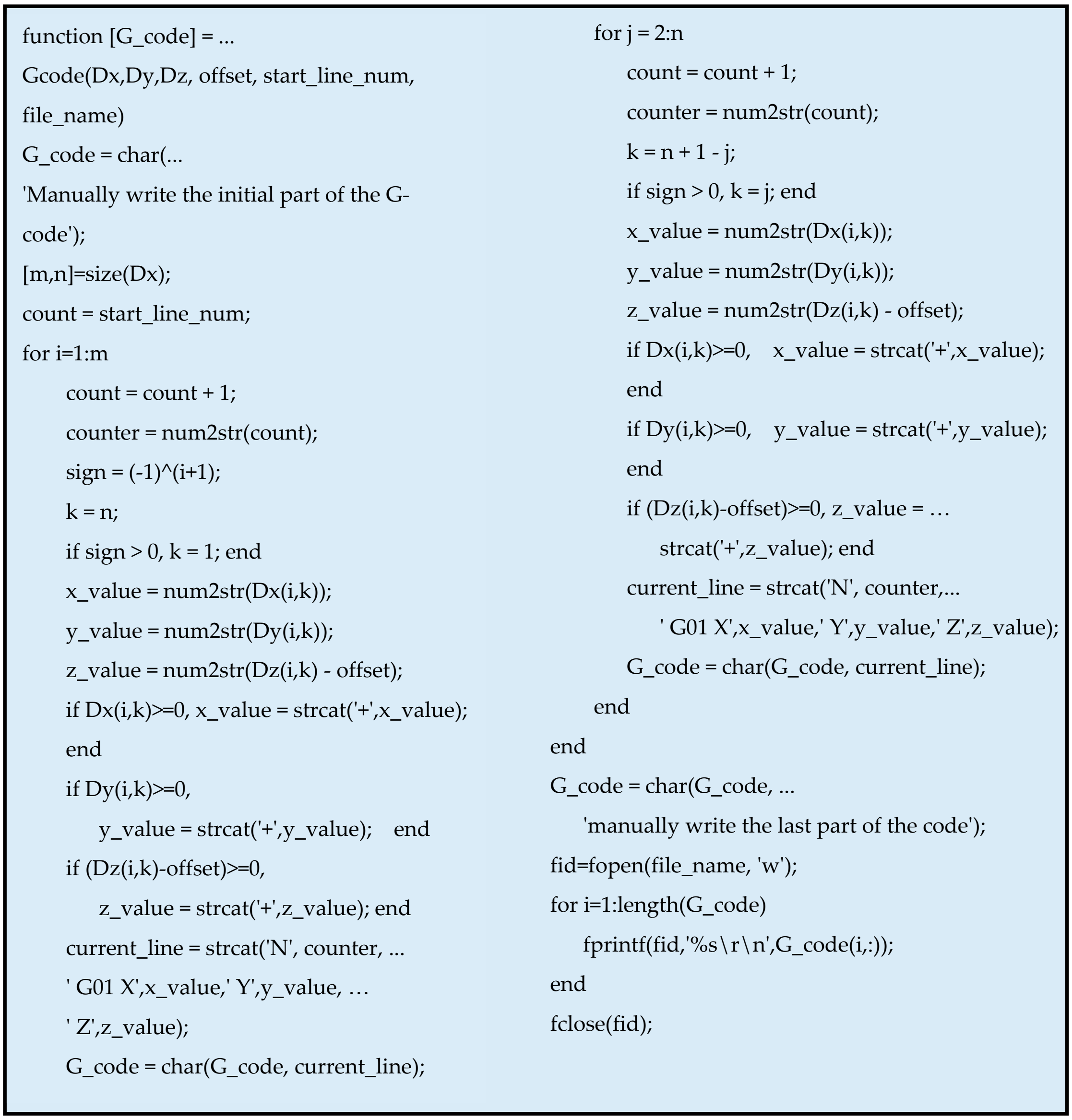
Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

This Appendix shows the MATLAB^{TM} codes needed for creating the offset surface from a cubic Bezier surface as defined in Section 3 and the relevant CNC programming. In particular, Figure A1 shows the MATLAB^{TM} codes for tool offsetting of a Bezier freeform surface, whereas Figure A2 shows the MATLAB^{TM} codes for CNC programming for machining the Bezier freeform surface.

```
function [Cx,Cy,Cz,Dx,Dy,Dz] = …
        surface(Px,Py,Pz,offset,step_size)
M = [-1 3 -3 1;3 -6 3 0;-3 3 0 0;1 0 0 0];
i = 0;
for v = 0:step_size:1
    i = i + 1;
    j = 0;
    V = [v^3, v^2 ,v, 1];
    Vp = [3*v^2, 2*v, 1, 0];
    for u = 0:step_size:1
        j = j + 1;
        U = [u^3, u^2, u, 1];
        Up = [3*u^2, 2*u, 1, 0];
        px = V*M*Px*M'*U';
        py = V*M*Py*M'*U';
        pz = V*M*Pz*M'*U';
        P=[px;py;pz];
        Cx(i,j) = px;
        Cy(i,j) = py;
        Cz(i,j) = pz;
        ppvx = Vp*M*Px*M'*U';
        ppvy = Vp*M*Py*M'*U';
        ppvz = Vp*M*Pz*M'*U';
        ppux = V*M*Px*M'*Up';
        ppuy = V*M*Py*M'*Up';
        ppuz = V*M*Pz*M'*Up';
        N = cross([ppux;ppuy;ppuz], …
        [ppvx;ppvy;ppvz]);
        PO = P + (N/norm(N)) * offset;
        Dx(i,j) = round(1000*PO(1))/1000;
        Dy(i,j) = round(1000*PO(2))/1000;
        Dz(i,j) = round(1000*PO(3))/1000;
    end
end
mesh(Dx,Dy,Dzz)
hold
surf(Cx,Cy,Cz)
hold
```

Figure A1. The MATLAB^{TM} codes for tool offsetting for a Bezier freeform surface.



```
function [G_code] = ...
Gcode(Dx,Dy,Dz, offset, start_line_num,
file_name)
G_code = char(...
'Manually write the initial part of the G-
code');
[m,n]=size(Dx);
count = start_line_num;
for i=1:m
    count = count + 1;
    counter = num2str(count);
    sign = (-1)^(i+1);
    k = n;
    if sign > 0, k = 1; end
    x_value = num2str(Dx(i,k));
    y_value = num2str(Dy(i,k));
    z_value = num2str(Dz(i,k) - offset);
    if Dx(i,k)>=0, x_value = strcat('+',x_value);
    end
    if Dy(i,k)>=0,
        y_value = strcat('+',y_value);    end
    if (Dz(i,k)-offset)>=0,
        z_value = strcat('+',z_value); end
    current_line = strcat('N', counter, ...
    ' G01 X',x_value,' Y',y_value, …
    ' Z',z_value);
    G_code = char(G_code, current_line);
    for j = 2:n
        count = count + 1;
        counter = num2str(count);
        k = n + 1 - j;
        if sign > 0, k = j; end
        x_value = num2str(Dx(i,k));
        y_value = num2str(Dy(i,k));
        z_value = num2str(Dz(i,k) - offset);
        if Dx(i,k)>=0,    x_value = strcat('+',x_value);
        end
        if Dy(i,k)>=0,    y_value = strcat('+',y_value);
        end
        if (Dz(i,k)-offset)>=0, z_value = …
            strcat('+',z_value); end
        current_line = strcat('N', counter,...
            ' G01 X',x_value,' Y',y_value,' Z',z_value);
        G_code = char(G_code, current_line);
    end
end
G_code = char(G_code, ...
    'manually write the last part of the code');
fid=fopen(file_name, 'w');
for i=1:length(G_code)
    fprintf(fid,'%s\r\n',G_code(i,:));
end
fclose(fid);
```

Figure A2. The MATLAB^{TM} codes for generating the CNC program for machining a Bezier freeform surface.

References

1. Melkanoff, M.A.; Puhl, F.; Langer, V.; Greenberg, D.P.; Shepard, M.S.; Voelcker, H. The challenge of CAD/CAM education. *SIGGRAPH Comput. Graph.* **1982**, *16*, 209–211. [CrossRef]
2. Lamancusa, J.S.; Zayas, J.L.; Soyster, A.L.; Morell, L.; Jorgensen, J. 2006 Bernard M. Gordon Prize Lecture*: The Learning Factory: Industry-Partnered Active Learning. *J. Eng. Educ.* **2008**, *97*, 5–11. [CrossRef]
3. Abele, E.; Chryssolouris, G.; Sihn, W.; Metternich, J.; ElMaraghy, H.; Seliger, G.; Sivard, G.; ElMaraghy, W.; Hummel, V.; Tisch, M.; et al. Learning factories for future oriented research and education in manufacturing. *CIRP Ann.* **2017**, *66*, 803–826. [CrossRef]
4. Ullah, A.M.M.S. On the interplay of manufacturing engineering education and e-learning. *Int. J. Mech. Eng. Educ.* **2016**, *44*, 233–254. [CrossRef]
5. Gómez-Galán, M.; Carreño-Ortega, Á.; López-Martínez, J.; Callejón-Ferre, Á.-J. Solving Power Balance Problems in Single-Traction Tractors Using PTractor Plus 1.1, a Possible Learning Aid for Students of Agricultural Engineering. *Educ. Sci.* **2018**, *8*, 68.
6. García, R.R.; Quirós, J.S.; Santos, R.G.; Peñín, P.I.Á. Teaching CAD at the university: Specifically written or commercial software? *Comput. Educ.* **2007**, *49*, 763–780. [CrossRef]

7. Tan, P.; Wu, H.; Li, P.; Xu, H. Teaching Management System with Applications of RFID and IoT Technology. *Educ. Sci.* **2018**, *8*, 26. [CrossRef]
8. Araki, T. Basic Education of CAD/CAM Through Multimedia and Network Aid. *J. Geom. Graph.* **1999**, *2*, 055–064.
9. Zhao, J.; Zhang, H.; Ma, L.; Chen, J.; Wen, Q.; Yin, S.; Wu, X. Basic CAD/CAM Engineering Education through Multimedia and Network Aid. In *Frontiers in Computer Education*; Sambath, S., Zhu, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 33–37.
10. De Weck, O.L.; Kim, I.Y.; Graff, C.; Nadir, W.; Bell, A. A Rewarding CAD/CAE/CAM and CDIO Experience for Undergraduates. In Proceedings of the First Annual Conceive-Design-Implement-Operate (CDIO) Conference, Kingston, ON, Canada, 7–8 June 2005.
11. Ishitsuka, K.; Arai, M.; Chida, K.; Koshimizu, M. 9-102 Practice of CAD/CAM Education using 3D-CAD Systems: Second Report: Development of Teaching-Materials for NC Machining by CAM Function. *JSEE Annu. Conf.* **2005**, *53*, 132–133.
12. Lin, T.; Ullah, A.M.M.S.; Harib, K.H. On the effective teaching of CAD/CAM at the undergraduate level. *Comput.-Aided Des. Appl.* **2006**, *3*, 331–339. [CrossRef]
13. Alemzadeh, K. A Team-Based CAM Project Utilising the Latest CAD/CAM and Web-Based Technologies in the Concurrent Engineering Environment. *Int. J. Mech. Eng. Educ.* **2006**, *34*, 48–70. [CrossRef]
14. Kaneko, T.; Kashimoto, H. Development of Assistant Teaching Materials in CAD/CAM Education (Part II): Preparing CAM learning environments by LinuxCNC. *Gunma-Kohsen Rev.* **2012**, *31*, 5–10.
15. Abouelala, M.; Janan, M.T.; Brandt-Pomares, P. Methodology of selecting CAM software package for education based on a questionnaire. *Int. J. Mech. Eng. Educ.* **2015**, *43*, 77–93. [CrossRef]
16. Moschonissiotis, S.; Vosniakos, G.-C. Analysis and structuring of process planning logic for CNC machining: An education and training perspective. *Int. J. Mech. Eng. Educ.* **2018**, *46*, 97–117. [CrossRef]
17. Main, D.; Staines, H.J. An empirical approach to the assessment of cadcam. *Comput. Educ.* **1995**, *25*, 53–57. [CrossRef]
18. Hamade, R.F.; Artail, H.A.; Jaber, M.Y. Evaluating the learning process of mechanical CAD students. *Comput. Educ.* **2007**, *49*, 640–661. [CrossRef]
19. Anand, V.B. *Computer Graphics and Geometric Modeling for Engineers*; Wiley: New York, NY, USA, 1993.
20. Piegl, L.; Tiller, W. *The NURBS Book*; Springer: Berlin/Heidelberg, Germany; New York, NY, USA, 1997.
21. Harib, K.H.; Sharif Ullah, A.M.; Hammami, A. A hexapod-based machine tool with hybrid structure: Kinematic analysis and trajectory planning. *Int. J. Mach. Tools Manuf.* **2007**, *47*, 1426–1432. [CrossRef]
22. Harib, K.H.; Kamal, A.F.M.; Ullah, A.M.M.S.; Salah, Z. Parallel, Serial and Hybrid Machine Tools and Robotics Structures: Comparative Study on Optimum Kinematic Designs. In *Serial and Parallel Robot Manipulators—Kinematics, Dynamics, Control and Optimization*; Kucuk, S., Ed.; INTECH Open Access Publisher: London, UK, 2012; pp. 109–124.
23. Ullah, A.M.M.S.; Takeshi, A.; Masahiro, F.; Chowdhury, M.A.K.; Akihiko, K. Strategies for Developing Milling Tools from the Viewpoint of Sustainable Manufacturing. *Int. J. Autom. Technol.* **2016**, *10*, 727–736. [CrossRef]
24. Ullah, A.M.M.; Watanabe, M.; Kubo, A. Analytical Point-Cloud Based Geometric Modeling for Additive Manufacturing and Its Application to Cultural Heritage Preservation. *Appl. Sci.* **2018**, *8*, 656.
25. Sharif Ullah, A.M.M. Design for additive manufacturing of porous structures using stochastic point-cloud: A pragmatic approach. *Comput.-Aided Des. Appl.* **2018**, *15*, 138–146. [CrossRef]
26. Loy, J. eLearning and eMaking: 3D printing blurring the digital and the physical. *Educ. Sci.* **2014**, *4*, 108–121. [CrossRef]
27. Novak, J.D. *Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations*; Routledge: New York, NY, USA, 2012.