An Interactive MATLAB Program for CAGD

Andrés Iglesias, Akemi Gálvez

Department of Applied Mathematics and Computational Sciences, University of Cantabria

Santander, Spain

Abstract

This paper introduces a new user-friendly MATLAB based-on program for dealing with the most important curves appearing in CAGD, as Bézier and B-splines curves. Since its user interface has been carefully designed, the program is specially recommended for educational purposes and it has been successfully applied for teaching both ungraduate and graduate students. The paper also discusses the main features of the program and its performance is shown by means of some interesting and illustrative examples.

Keywords: CAGD, Curves, Education, MATLAB.

1. INTRODUCTION

In the last years, teachers and educators are concerned about the extraordinary changes in the way of teaching and understanding the education. Many of these changes are given by the new scientific programs, which allow students and teachers to deal with complex real situations by using more realistic simulations by computer. Among them, the general-purpose symbolic or numerical computation programs become more and more popular and they offer an interesting alternative to the traditional programming languages, as Fortran, Pascal, C, etc. These programs incorporate many commands, avoiding the user to implement them, and a great number of graphical and programming facilities. Today, teachers can take advantage of these graphical capabilities when introducing students to computer graphics and/or visualizing the geometry of the real-world objects to be analyzed.

With this idea in mind, we introduced MATLAB [1] for learning CAGD to both ungraduate and graduate students. A fundamental reason for this choice was the powerful graphical MATLAB capabilities, including the interactive use of the screen (for example, rotations of three dimensional figures through the `rotate3d` command) and the great variety of possibilities to display a curve and/or a surface. Moreover, MATLAB numerical capabilities are very remarkable and improve the run speed we appreciated when using symbolic computation programs, as Mathematica or Maple, which are not able to deal with the large series of inputs that appears in many real problems.

Our experience was so positive that we decided to create a software in MATLAB for introducing students to CAGD. This paper presents an user-friendly program for dealing with the most important curves in CAGD, as Bézier and B-splines curves.

In Section 2 we briefly describe the main features of the program, whereas Section 3 illustrates its perfomance by means of some interesting and illustrative examples. Finally, Section 4 closes with the main conclusions of the paper.

2. DESCRIPCION OF THE PROGRAM

The program introduced in this paper, based on MATLAB, deals with both two and three dimensional Bézier and B-splines curves. Its user-friendly interface has been carefully designed by extensively applying the object-oriented MATLAB programming.
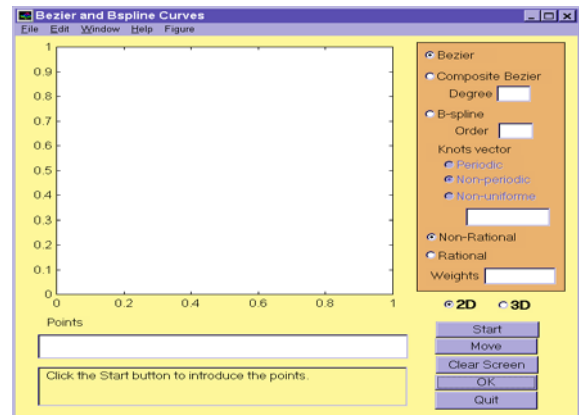
Figure 1: Initial window of the application.

After opening the application, the user finds an initial window with a set of buttons, radio buttons and some dialog and input boxes, as shown in Figure 1. The buttons (the rectangles in blue), indicate the basic operations supported by the program:

- **Start**. It allows the user to introduce the points to control the curve. They are usually called *control points* in CAGD [2].

- **Move**. Indicates that the user is going to move one of the control points to another position on the screen.

- **ClearScreen**. The obvious task of clearing the screen.

- **OK**. Runs the application for obtaining the required curve.

- **Quit**. After pressing it, the user quits the application.

The radio buttons allow the user to indicate the system what kind of curve he/she is going to deal with. The admissible values are **Bezier**, **Composite Bezier** and **B-splines**. Bézier curves do not require anything else, as its

degree is completely determined by the control points. On the contrary, composite Bézier curves are built by parts, so the degree of each part to be drawn is required by means of a input box called **Degree**. Finally, B-splines curves require to know their order and knots vector, which can take the values **Periodic**, **Non**-**periodic** or **Non-uniform**. In the first two cases, the knots vector is automatically calculated by the program, while the last case corresponds to an user's choice and therefore it must be introduced into the corresponding input box. System is also able to work with both non-rational and rational curves. Rational curves mean that non unity weights are considered, so user is required to introduce their new values into a input box. Finally, a dialog box informs the user about the correct input to be introduced at each step.

3. SOME EXAMPLES OF APPLICATION

In this section we describe a typical application work session, which is given by the following steps:

1. Choose the dimension of the curve to be drawn.

2. Press the button **Start** to introduce the control points, either by clicking with the mouse on the screen for the two dimensional case or introducing them into the input box in the 3D case.

3. Specify the kind of curve and, when required, its degree or order, according to the cases described in the previous section.

4. Press the button **OK**. The desired picture is obtained. Figure 2 shows a typical 3D B-spline curve, its control polygonal and points. This picture can be interactively rotated by using the mouse.
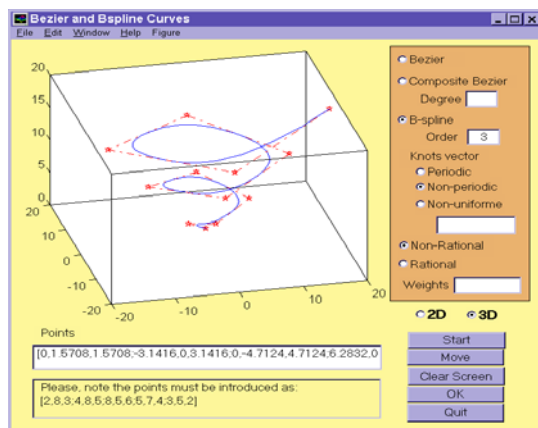
Figure 2: A display of a 3D B-spline curve.

At this point, you have several options: redraw another 3D curve (belonging to the same or a different kind) on the current one, with the same control points, quit the application or start again and obtaining a new curve, for example, a 2D B-spline curve. Let us suppose this is your case. Then, you should press the **ClearScreen** button and repeat the steps 1 to 4. Pressing now the **Move** button, you

can select a point, apply drag and drop and recalculate the new curve associated with the new points. Figure 3(a) shows two different B-splines obtained by moving a selected control point. Finally, you can display several curves on the same screen: Figure 3(b) shows some B-spline curves corresponding to different orders but keeping fixed their control points.
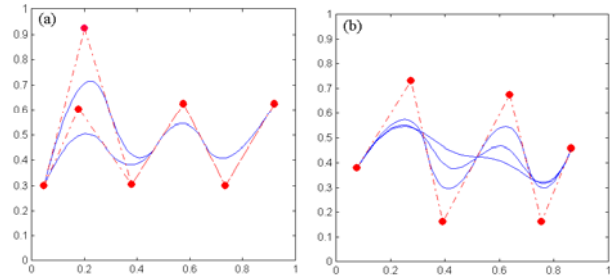
Figure 3: Effects on the shape curve when: (a) changing a control point of a B-spline, showing its "local control" (b) changing the order of the curve from 3 to 5.

4. CONCLUSIONS AND REMARKS

In this paper we have introduced a program based on MATLAB for dealing with the most important curves in CAGD. The program is useful for introducing students to fundamentals of CAGD and it is specially recommended for educational purposes. As illustration of this, Figures 3(a) and (b) show how the shape curve is deformed when changing a selected control point or the order of the curve, respectively. The concepts of "local control" and "B-spline order" are, by this way, easier to introduce. This means that visualization and interaction capabilities of this program can be advantageous with respect to other more traditional educational methods. Finally, we remark that all the numerical routines have been implemented throughout by the authors from [2]. Finnancial support from CICYT (project TAP98-0640) is acknowledged.

5. REFERENCES

[1]   MathWorks. *MATLAB User's Guide* (1996).

[2]   V. B. Anand. *Computer Graphics and Geometric Modeling for Engineers*. John Wiley and Sons (1993).

Author(s):

Andrés Iglesias holds a Ph.D. in Mathematics and currently is associate profesor at the Department of Applied Mathematics, E.T.S.I. de Caminos, s/n 39005, University of Cantabria, Spain. Tel: 34 (942) 20-17-23 / Fax: 34 (942) 20-17-03. E-mail: iglesias@ccaix3.unican.es

Akemi Gálvez is a Ph.D. candidate at the same Department. She holds a B.Sc. in Chemical Engineering at U.N.T. (Perú) and a M.Sc. in Computation at Cantabria University. E-mail: uc8031@cclx1.unican.es