Norbert Annuš
Mgr., Master, PhD student, Professor's Assistant
J. Selye University, Department of Informatics, Komárno, Slovak Republic
ORCID ID 0000-0002-8535-2042
annusn@ujs.sk

Ondrej Takáč
PhD, Doctor, Head of Department
J. Selye University, Department of Informatics, Komárno, Slovak Republic
takaco@ujs.sk

TEACHING OF INTERPOLATION AND APPROXIMATION CURVES THROUGH COMPUTER SIMULATION

Abstract. The use of digital tools in the education process has become a matter of course. This is particularly true in computer science lessons. Our study assesses the performance of students in higher education within each subject area of computer graphics. Computer graphics and its education are very versatile. A better understanding of the algorithms that perform the graphical adjustments is much easier if the student is not only theoretically familiar with the field and its procedures but also has the opportunity to try them out. Our aim was to develop a teaching tool that would allow students attending a graphics class to learn about and test the possibilities of interpolation and approximation curves. In our study, we focused on three specific types of curves: the Hermite, Bézier and Coons curves. For universal use, students are able to simulate the curves concerned in nth-degree cases. In the development, we focused on modelling the mathematical formulas describing the curves trying to design an easy-to-use teaching tool. In this case, the teaching tool is ~~a~~ computer software designed for this purpose. In our research, we conducted qualitative surveys on small groups of students. Our study was conducted on control groups. We also conducted interviews with students using the software to assess the strengths and weaknesses of our software. Based on this feedback, we optimized our software and set the direction for future improvements. The practical application suggests that the educational software has a positive impact on students' learning outcomes. This paper describes the mathematical models describing the processed curves. It then presents the application's options and graphical results. Finally, we would like to present the results of the student testing and their feedback on the educational software.

Keywords: educational software; modelling; simulation; interpolation; approximation; computer graphics.

1. INTRODUCTION

Our primary focus revolves around aiding students who aspire to complete our university's computer graphics course. The computer graphics course encompasses more than the utilization of various CAD (Computer-aided design) systems. The aim is for students to understand the mathematical logic used by these systems, the algorithms running in the background, deepening their understanding of the subject and making them better professionals themselves. The basic element needed for the effective teaching of computer science can be achieved through a combination of theory, abstraction and implementation [1]. It would of course be impossible to cover all computer graphics within the scope of this paper. The goal of our study is to use a self-developed educational software to help teaching interpolation and approximation curves. Students are introduced to computer graphics in primary schools and are already familiar with graphical primitives that require the use of curves from the very first tasks [2]. An example of this is the drawing of a house with the simplest graphical editors. For students who wish to acquire knowledge in the field of computer science in higher education, knowledge of computer graphics is essential.

© Norbert Annuš, Ondrej Takáč, 2023.

The authors themselves have completed a university course in computer graphics and teach in the field. Hence, based on our experience so far, we recognize the necessity for our thesis in fulfilling students' aspirations to gain hands-on experience through experimentation. Teaching computer graphics requires an understanding of it supported by models and visualizations [3]. However, a recurring problem is the lack of variety. By this we refer to the absence of student engagement in existing resources, such as tutorial videos and pre-made simulations. Currently, students receive the course material digitally, as if they were reading an electronic book. This greatly limits them in experimentation. In an effort to address this issue, we have introduced self-developed computer simulations, the final output of which is that students have the opportunity to influence. Previous experiments have compared static and interactively animated and simulated learning materials. The results show that students prefer interactive animated learning materials because they are more conducive to comprehension [4]. For modern education, we believe it is important to take advantage of existing technological opportunities. With the rapid development of information technology, education (with a focus on university education in this article) is undergoing a profound transformation. This means using new methodologies, involving students more in the teaching and learning process and giving greater importance to information and communication technologies (ICT) [5]. According to the current profile of the modern European teachers, they should not only be skilled users of modern digital tools, ICT and other educational technologies, but also be able to create their own didactic applications [6]. Furthermore, studies show that the use of appropriate platforms has a positive impact on the educational process [7]. Computer-assisted instruction is a great help for students with an audio-visual learning style. The visual learning material is often presented using some form of model. By model we mean a mathematical construct that describes the observed phenomenon, supplemented by some verbal interpretation. The use of models in education has many positive benefits. Models can convey a better understanding of nature and its phenomena, giving scope for experimental observation. The teaching material is clearer and more memorable than a written text, so students' practical skills are reinforced.

From the results of our research, we can conclude that the visualization software is a useful tool for the students, also the influence of the location of the given points is much more tangible when represented in software, rather than simply shown as coordinate points on the board or presented as algorithms describing mathematical formulas. This allows students to participate in the learning process in an interactive way. In the case of simulation, it is possible to intervene in the changes of the phenomenon and observe its behavior. By using experiential learning and trial-and-error learning methods, they can acquire more lasting knowledge.

This paper is divided into the following sections: the second section describes the results of other researchers from an educational perspective. The third section describes the interpolation and approximation curves we have worked on. The fourth section describes our software and its capabilities. The fifth section illustrates the graphical results of our software. The sixth section presents our results in pedagogical practice. Finally, the seventh section discusses plans for future improvements.

2. RELATED WORK

Already in the late nineties, it was clear that there was a strong need for an integrative approach combining lectures, examples, programming exercises and documentation, supported by a common, sophisticated interface [8]. From an educational point of view, however, the field is still not fully covered.

At the same time, Marchese proposed a spreadsheet approach [9]. The author used a spreadsheet editor to help students understand computer graphics concepts such as affine

transformations, splines and Bezier curves. The author used the spreadsheet as an interactive visualisation tool. Students changed some values in the spreadsheet and the shape of the curve changed automatically as a result. Interactive teaching with spreadsheets in modelling and simulation classes are also recommended by Paksi et al. [10].

Pla-Castells et al. also see the importance of a proper approach and understanding of computer graphics education from a modelling perspective. Their work presents a modelling activity, mainly for computer science students, aimed at studying function interpolation. They used Blender (modeling software) to carry out the exercises [11].

An earlier approach to demonstrating the points that make up Bézier curves in education was experimented with by calculating on pocket calculators [12]. Given today's technology, we can see that these approaches are now outdated. Educational software has also been developed to process the curves used in graphics [13], [14]. The software presented in the studies, however, does not provide an opportunity to represent these three specific topics. This software is restricted to Windows operating systems only, and no access to these applications was found during the study.

James Miller's paper discusses the potential of *freeform* educational software, which mainly covers aspects related to NURBS (Non-uniform rational B-spline) curves, but also supports Bezier and Hermite curves, as well as Bezier and B-spline surfaces. The author also emphasizes the importance of interactivity in his work. The author describes how users can use *freeform* to observe in side-by-side windows the changes in the shape of curves as a result of interactive knot and weight adjustments. The results of his survey suggest that students value these pedagogical software tools, have a better understanding of the underlying mathematics of the different graphic design software and are becoming better users of these tools [15]. In 2010, the software was only available on Macintosh and Linux operating systems. The authors set a goal to implement it on the Windows platform. At the time of this article, however, the *freeform* software was no longer available for testing. The various browser search engines only found the digital whiteboard application published by Apple. The success of further development is therefore unknown.

Peng and Bo's study presents a massive open online course platform that also offers the opportunity to teach computer graphics. Their paper deals with the visualization of parametric curves. The platform they describe gives the possibility to visualize two-dimensional curves, namely Hermite, Bézier and B-splines [16].

Throughout history, of course, there have been attempts not only at 2D but also at 3D approaches. Eyad uses Autodesk 3ds Max to teach mathematical concepts such as linear interpolation, Bézier curves and the mechanics of motion through interactive 3D animation. Results show that although students found this module difficult due to the amount of theory taught, they were able to understand the topics well and 91% of students were able to complete the course [4]. Peternier et al's study, in addition to demonstrating the use of VR (Virtual Reality) technology, sought to deepen the understanding of computer graphics and geometrics among students. According to them, these tools allow students to have direct access to parameters and algorithms, offering a dynamic causal explanation that is not available through static methods [3].

Among the most recent papers on the topic is the work of Zakharov and Zakharova, which discusses an interactive software that allows students to gain hands-on skills in working with splines and visually analyze the impact of input data on the final output. For example, students have the opportunity to work with Hermite or natural cubic splines, irrational or rational Bezier curves, and B-splines. The software allows students to control a number of parameters, such as the position and number of control points, surface and spline parameters, as well as additional visualisation settings. Their research aims to give students practical skills in working with spline

representations. However, the software that was the focus of their study was not named and its availability was not indicated either [17].

Looking at the treatment of the topic, we can see that there are areas that are recurrent examples in almost all the research work. From an educational point of view, however, there are also some topics that have only been marginally implemented. Furthermore, the teacher-student relationship, i.e. feedback, is lacking in several software applications. For the topics on curves and surfaces of the computer graphics course we offer, we could not find a software package that covers the whole curriculum and satisfies the task of project control. The lack of a unique, common and shared platform can lead to confusion and can reduce or even hinder collaboration between students and teachers [8]. For this reason, we wanted to avoid having students learning on different platforms and with different software. This was the main motivation for the authors to create the software and to implement it in pedagogical practice.

3. CONCEPTUAL FRAMEWORK

Curves and surfaces are used in a variety of applications in computer graphics and related applications. They can be found in two- and three-dimensional modelling, in computer animation, in defining the path of some moving objects [18]. This area, like many others in information technology, is quite broad.

There are basically three ways to describe curves (Salomon 2006):

– Explicit mode: e.g. $x = x(t)$
– Implicit mode: e.g. $f(x, y) = 0$
– Parametric mode: e.g. $r(t), t \in [a, b]$

Of the enumerations, the explicit and implicit modes can only describe plane curves, while the parametric mode can describe both plane and space curves [19]. The previous methods were done in analytical form. In addition, we have the option of specifying the curves empirically and using a sequence of points. For these cases, their representation is given, but their mathematical expression is unknown. If we want to model something using curves, we can rely exclusively on so-called control points. If we use this procedure, we can talk about free-form curves [20].

In practice, curves are specified by the user in terms of points and are constructed in an interactive process [21]. The use of control points is relatively simple, both in terms of operation and design. A point can also be easily selected using interactive methods such as a cursor or touch screen - depending on the device. Thus, when used in class, students can easily manage such software. Once the control points have been entered, the task of the application is to fit a curve to the given series of points. Accordingly, two types of curves can be distinguished:

– Interpolation curves
– Approximation curves

In this section, we will describe the types of curves we have worked on and the mathematical models describing them, which have been used as a basis for the programming.

3.1. Interpolation Curves

Definition 1. Interpolation: all points of the basic figure are located on the created figure called interpolation curve segment.

We can talk about interpolation curves when we are looking for an image of a curve that fits given points. One of these curves is the Hermite curve, often used, but referred to in some literature as the Fergus curve [18]. During our study, we dealt with some cases of this interpolation curve. Two principal points denoted P_0 and P_1, and two vectors, denoted V_0 and V_1, associated with these points, play an important role in the representation of the simplest

first-degree Hermite curves. The points P_0 and P_1 define the position of the curve. By this we mean that it starts at P_0 and ends at P_1. The direction and magnitude of the vectors determine the degree of curvature of the arc. In computer graphics, Hermite curves are usually trained with polynomials of degree three [21], [22], where $t \in [0,1]$. The curve starts at point P_0 and arrives at point P_n, and does so exactly between $t = 0$ and $t = 1$.

$$\begin{aligned} F_0(t) &= (1-t)^2(1+2t) = 2t^3 - 3t^2 + 1 \\ F_1(t) &= t(1-t)^2 = t^3 - 2t^3 + t \\ F_2(t) &= -t^2(1-t) = t^3 - t^2 \\ F_3(t) &= (3-2t)t^2 = 3t^2 - 2t^3 \end{aligned} \quad (1)$$

Hermite curves can also be expressed in the following form:

$$H(t) = P_0 F_0(t) + P_1 F_1(t) + V_0 F_2(t) + V_1 F_3(t) \quad (2)$$

where F_0, F_1, F_2 and F_3 are the Hermite or Fergus polynomials in equation (1).

3.2. Approximation Curves

Definition 2. Approximation: all points of the basic figure need not be located on the created figure called approximation curve segment.

We can talk about approximation curves when the curve only approximates the points. In this case, it is not guaranteed that the curve passes all the control points, but rather just follows their paths. Each point influences the direction of the curve by pulling it toward itself, and that influence is strongest when the curve gets nearest the point [21]. The Bézier curve is one of the most widely used approximation curves. Bezier curves do not depend on the underlying coordinate system as do the polynomials obtained from interpolation [12]. Traditional Bézier curves, which is formed by the classical Bernstein basis functions and control points, have many excellent properties like symmetry, terminal properties, non-negativity, linear precision, integral property or convex hull property [23]. Their application appears in two-dimensional modelling, in the definition of font styles. Of course, they are not only applicable to two-dimensional modelling. The results of Ameer's research [24] showed that Bézier-like symmetric rotation surfaces facilitate the modelling of three-dimensional surfaces. Furthermore, a Bézier curve can parameterize a smooth function with a minimum number of parameters. Given a sufficiently large number of appropriately chosen control points, any smooth function can be approximated with arbitrary accuracy by a Bézier curve. Kmeť [25] proposed Bézier parametrisation and echo state networks-based algorithms to solve optimal control problems for systems governed by differential equations. These examples also illustrate the versatility of Bézier curves.

In teaching computer graphics, we focused on two-dimensional curves, where we defined the curve using polynomials. Using two points, we can only draw a straight line. However, as we increase the number of control points, we can get any curve shape [23]. As with Hermite curves, the use of third-degree polynomials is most common for Bézier curves. To plot a third-degree Bézier curve, start by defining four control points - denoted P_0, P_1, P_2 and P_3,, where P_0 is the starting point of the curve and P_3 is the arrival point. Third-degree Bézier polynomials, also called Breinstein polynomials, can be written as [22]:

$$\begin{aligned} B_0(t) &= (1-t)^3 \\ B_1(t) &= 3t\,(1-t)^3 \\ B_2(t) &= 3t^2(1-t) \\ B_3(t) &= t^3 \end{aligned} \tag{3}$$

The parameter t also plays a role in the Bézier curve, where $0 \leq t \leq 1$, and where $t = 0$ at the starting point P_0 and $t = 1$ at the arrival point P_3. An important factor in plotting these curves is how many control points we want to define the curve by. This is why, in two, three, four, and n-point bounded modes, we compute different equations with different degrees of polynomials. In computer graphics, the use of a polynomial of degree three is most common, but n-degree cases can also be expressed. A graphical representation of such cases is presented in the next sections of this article.

Coons curves also belong partly to the class of approximation curves. A Coons arc is generally defined by four control points P. These control points affect the starting and arrival positions of the curve. If we use four control points, then our starting point $R(0)$ is located at $1/3$ of the centroid of the triangle of points P_0, P_1 and P_2, while our arrival point $R(1)$ is located $1/3$ of the centroid of the triangle of points P_1, P_2 and P_3. Thus, the Coons curve can be said to start at $R(0)$ and proceed to $R(1)$, where the points of the curve can be described as [22]:

$$C(t) = K\sum_{i=0}^{3} P_i c_i(t), \qquad t \in [0,1] \tag{4}$$

where $K = 1/6$ due to its geometric properties,
$c_i(t)$ denote the Coons polynomials, which can be written as:

$$\begin{aligned} c_0(t) &= (1-t)^3 \\ c_1(t) &= 4 - 6t + 3t^3 \\ c_2(t) &= 1 + 3t + 3t^2 - 3t^3 \\ c_3(t) &= t^3 \end{aligned} \tag{5}$$

t represents the time between 0 and 1.

Coons curves are defined by control points that are not affected by the curve. However, these control points define the points at which the Coons curve interpolates. In practice, in addition to Coons curves, so-called Coons patches or Coons surfaces are often used. These play an important role in modelling interpolation surfaces and curves. The traditional Coons patch is a powerful tool for surface representation and design, but it is not useful in case of for quadratic surface representation. Chen and Su's study defines and presents quasi-Hermite curves and quasi-Coons surfaces [26], which are shown to be more accurate for expressing special curves and surfaces than either general Hermite curves or Coons surfaces. In many engineering situations, the designer is given a set of often many data points and is tasked with tracing a "smooth" curve/surface through these points. In computer-aided design (CAD), Hermite curves and Coons patches are the most widely used for interpolation on a given data set [27].

Although the computational processes and applications of interpolation and approximation curves are different, it is possible to investigate the relationships between them and to carry out experiments that influence each other. Xiuzi Ye's study [28] deals with the conversion between different curve and surface shapes. Their paper presents methods for the direct generation of Bézier points: explicit relations between the derivatives of Hermite curves

and the corresponding Bézier points, and relations between Bézier points and the partial derivatives of the corner points of Coons-Hermite Cartesian sum patches. Their paper explicitly expresses the control points that form the Bézier curve using Hermite curves.

This paper presents the classical two-dimensional graphical processing of the curve types considered above and focuses on their education. The algorithms modelling the curves are constructed using equations (1 – 5) with the corresponding degree numbers. The next two sections focus on the operation of the software and the graphical results obtained with it.

4. SOFTWARE DEVELOPMENT

Our aim was to develop a versatile software that serves as a feature-rich application, aiding in both visualizing and practicing the curriculum.. Additionally, it offers project creation and archiving capabilities for students while expediting assignment evaluations for teachers. We have tried sought to create a simple interface that is easy to navigate for the user. The software options shown in Figure 1 are described below. Students have two main types of curves to choose from: interpolation and approximation. Within these, the user can choose between the Hermite, Bézier and Coons curves discussed in the previous section.. In particular, separate plotting of the most frequently utilized first-order, second-order, and third-order curves in the university course. The same is true for the simplest Coons curve, which is controlled by four control points. In addition, if students wish to experiment, it is possible to plot curves of n points in all three cases. Students can select the type of curve from a list (marked A). Depending on the type and on the number of degrees of the selected curve, a mathematical formula describing the points of the curves which is displayed on a canvas surface (marked F). If we want to plot n-degree curves, we need to specify the number of points we want to plot, using an input field (marked H) and a button (marked I). The software takes into account the chosen curve type and limits the minimum number of control points accordingly. For Hermite and Bézier curves, the number is two. For Coons curves, it is four, since that is the minimum number of control points needed to calculate $R(0)$ and $R(1)$, as described in the previous section. Next, the user must use the cursor to specify the points P_i and, in the case of a Hermite curve, their corresponding vectors V_i (where $i = 0,1,\dots,n$) on the surface of a canvas (marked K). The procedures are executed by watching the mouse clicks. Left-clicking allows the user to place P-points on the canvas, while right-clicking allows the user to place V-vectors on the canvas. For Hermite curves, after placing a point, the corresponding vector must be given immediately. In order to avoid errors, the program will not allow any further points to be placed until the vector associated with the point has been specified, and vice versa. The exact location of points is made easier by the fact that the current coordinate is continuously plotted as the mouse moves. It is also helpful if multiple control points are used, and the indexes of the points are plotted under the points that have been placed. This function can be toggled on and off using a checkbox (marked G). After the last point or vector is placed, the graphical simulation starts immediately, where the program plots the curves as points. The user can manipulate the simulation in several ways. The first one is the speed of the simulation, which can be changed during the simulation by means of a slider (marked M). Furthermore, since the curves are made up of points that are computed on the basis of time (t), the frequency of the points that make up the curve can be manipulated by controlling the rate of change of the parameter t, which can also be done with a slider (marked N). In both cases, it is possible to monitor the extent of the changes. As the simulation progresses, it is possible to track the value of t in the bottom right corner (marked O). When the simulation is finished, the resulting image can be saved as an image file by clicking on a button (marked E).

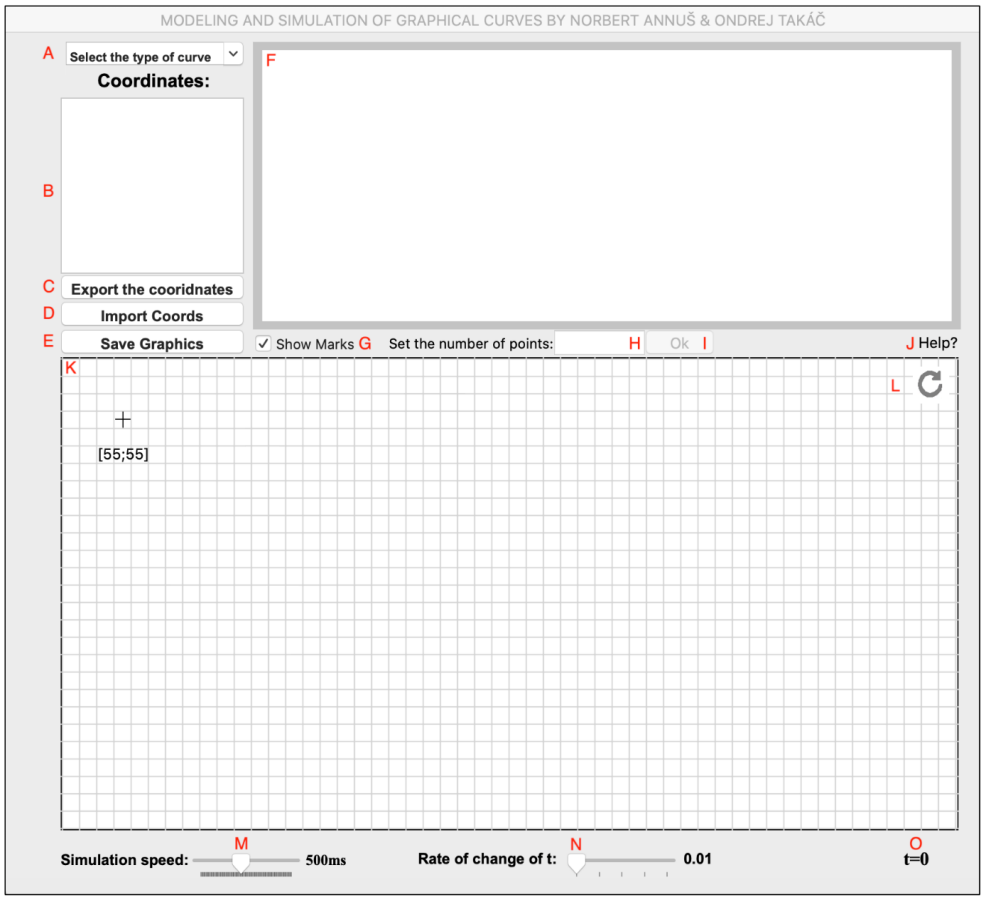

Figure 1. User interface of the software

The points and vectors, as well as the points that make up the curve plotted during the simulation, are listed. The contents of the list (marked B) can be exported by the user to an external file, also using a pushbutton (marked C) with a .csv extension. Students can upload the file with the saved points to the university (moodle) system. After downloading these files, the teacher has the option to open them using the import function (marked D). The loading of points is followed by an instant simulation, which makes it easy and simple to check assignments that students have to submit. Another option is to update the canvas (marked L). This resets the program, which enables and disables certain functions. This procedure can be performed even during the simulation. Although, our aim is to provide the user with a clear overview of how to use each function, we have also added a comprehensive help guide (marked J), which provides the user with a 12-step summary of the software's instructions.

The software can display a total of 10 types of curves. For Hermite curves: first degree, second degree, third degree and nth degree. Similarly, for Bézier curves, these four options are available. Coons curves can be represented by four control points or n control points. The equations presented in the next section were used to calculate the points that make up the curves.

5. GRAPHICAL RESULTS OF THE SOFTWARE

In the following pages, we present the graphical results of our software, which are illustrated separately for the first degree and the options of higher degree numbers with an example.

5.1. First-degree Hermite Curve

The positions of the points on the curve at time t can be given as:

$$H_1(t) = (2t^3 - 3t^2 + 1)\,P_0 + (-2t^3 + 3t^2)\,P_1 + (t^3 - 2t^2 + t)\,V_0 + (t^3 - t^2)\,V_1 \quad (6)$$

where H_1 takes the value of the points on the Hermite curve at time t. P_0 is the first point and V_0 is its direction vector, and P_1 is the second point and V_1 is its direction vector.

Figure 2 demonstrates the simulation result of our software - a graphical representation of the first-degree Hermite curve.

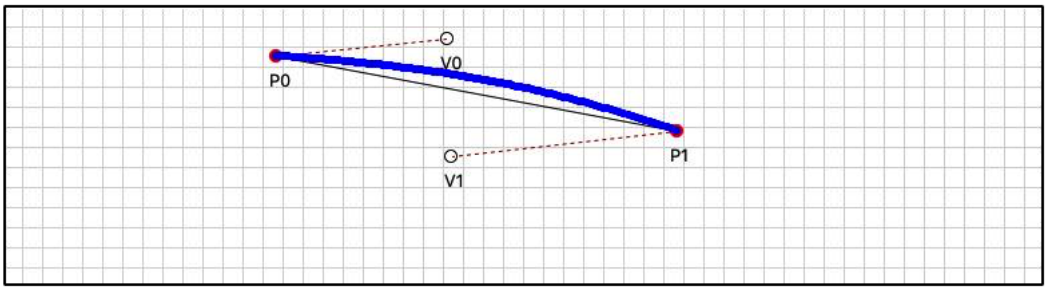

Figure 2. First-degree Hermite Curve with a frequency of 0.01

5.2. *n*-degree Hermite Curve

The positions of the points on the curve at time t can be given as:

$$H_i(t) = (2t^3 - 3t^2 + 1)P_j + (-2t^3 + 3t^2)P_{j+1} + (t^3 - 2t^2 + t)V_j + (t^3 - t^2)V_{j+1};\ i = 1 \ldots n;\ j = 0 \ldots n \quad (7)$$

where H_i takes the values of the points of the i-th Hermite curve at t time, separately for each H. P_j is the starting point and V_j is its vector and P_{j+1} is the arrival point and V_{j+1} is its vector.

Figure 3 demonstrates the simulation result of our software - a graphical representation of the n-degree Hermite curve, where we have worked with our software, in this case with 10 points.

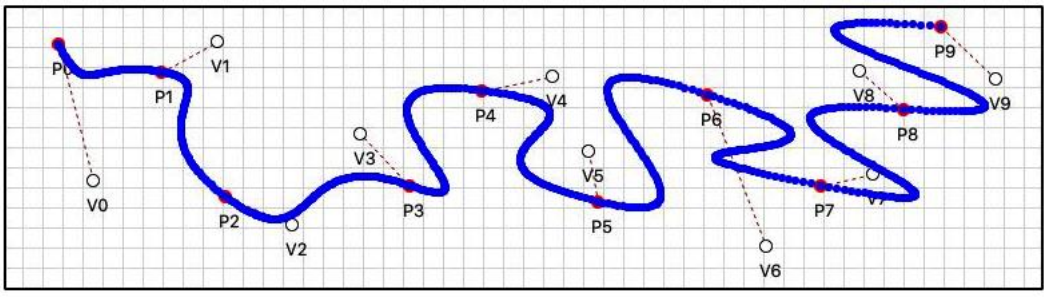

Figure 3. Hermite Curve operating with 10 points with a frequency of 0.01

5.3. First-degree Bézier Curve

The positions of the points on the curve at time t can be given as:

$$B(t) = (1 - t)P_0 + tP_1 \tag{8}$$

where B takes the value of the points that make up the curve. Of the points P_0 and P_1, P_0 is the starting point and P_1 is the arrival point, and t is the time.

Figure 4 demonstrates the simulation result of our software - a graphical representation of the first-degree Bézier curve.

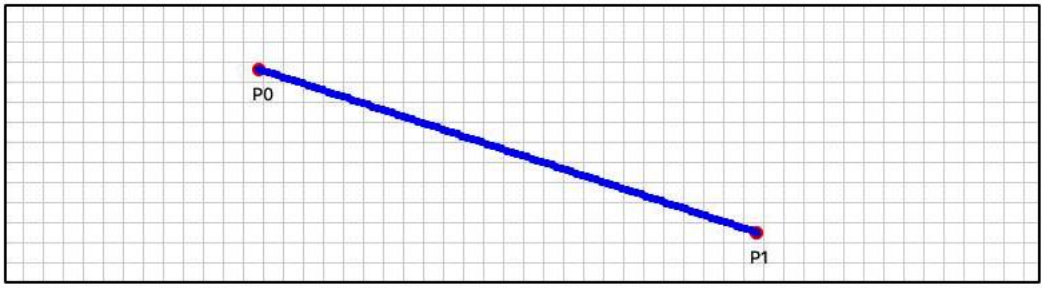

Figure 4. First-degree Bézier Curve with a frequency of 0.01

5.4. *n*-degree Bézier Curve

The positions of the points on the curve at time t can be given as:

$$B(t) = (n\ i\ )(1-t)^{n-i}t^{i}P_i; \qquad i = 0, \dots, n \tag{9}$$

where B takes the value of the points that make up the curve. P_i marks the i-th control point. For the nth-degree Bézier curve, P_0 is the arrival point and P_n is the arrival point. The curve does not pass through the points $P_1, P_2, \dots, P_{n-1}$ but follows their direction as it approaches them.

Figure 5 demonstrates the simulation result of our software - a graphical representation of the n-degree Bézier curve, where in this case 10 control point was used.

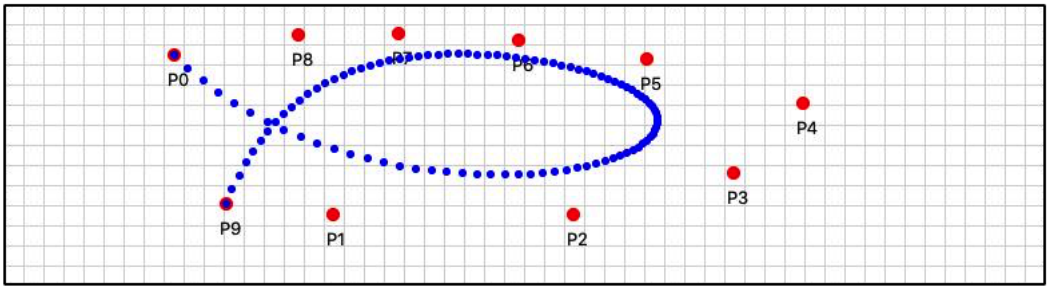

Figure 5. Bézier Curve operating with 10 points with a frequency of 0.01

5.5. Coons Curve with four control points

The positions of the points on the curve at time t can be given as:

$$R(0) = \frac{(P_0 + 4P_1 + P_2)}{6} \tag{10}$$

$$R(1) = \frac{(P_1 + 4P_2 + P_3)}{6} \tag{11}$$

$$C(t) = (((1-t)^3 P_0) + ((4 - 6t^2 + 3t^3)P_1) + ((1 + 3t + 3t^2 - 3t^3)P_2) + (t^3 P_3))K \tag{12}$$

where C takes the value of the points that make up the curve. Points P_0, P_1, P_2 and P_3 are the control points, which influence the position of $R(0)$, the starting- and $R(1)$ the arrival points.

Figure 6 demonstrates the simulation results of our software.

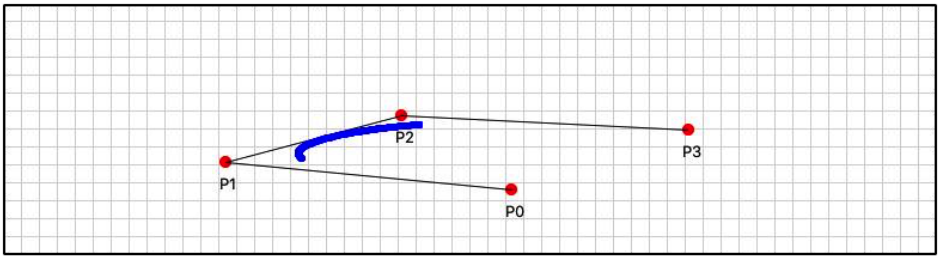

Figure 6. Coons Curve with 4 control points with a frequency of 0.01

5.6. Coons Curve with *n* control points

The positions of the points on the curve at time t can be given as:

$$R(j) = \frac{(P_{i-2} + 4P_{i-1} + P_i)}{6}; \quad i = 2, \dots, n; j = 0 \dots n-2 \tag{13}$$

$$C(t) = \Big(((1-t)^3 P_{i-3}) + \big((4 - 6t^2 + 3t^3)P_{i-2}\big) + \big((1 + 3t + 3t^2 - 3t^3)P_{i-1}\big) + (t^3 P_i)\Big)K; \quad i = 3, \dots, n \tag{14}$$

where C takes the value of the points forming the curve between $R(j)$ and $R(j+1)$. P marks the control points, which influence the position of the points R through which the curve passes.

Figure 7 demonstrates the results of our software, where the simulation was run with 20 points by selecting the n-point function.

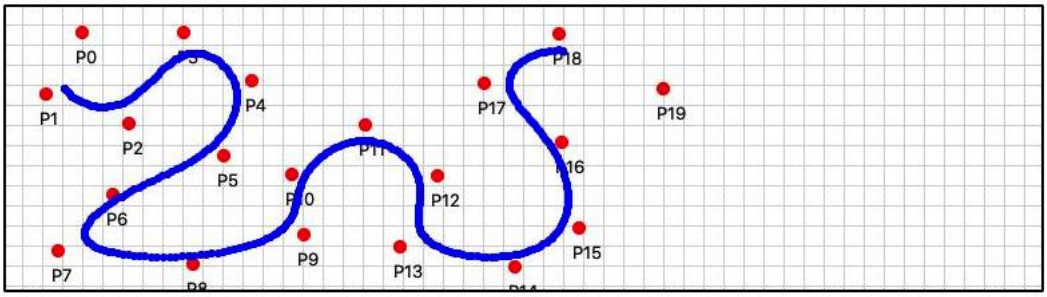

Figure 7. Coons Curve with 20 control points with a frequency of 0.01

5.7. Results of changing the parameter *t*

As mentioned above, the frequency of the points that make up the curve can also be customized by the user. This adjustment affects the rate of change of t (denote by t_r). In Figure 8, this is illustrated by a fifth-degree Hermite curve, where 6 points are used to create 5 Hermite curves. In the simulation, t_r was modified as follows:

- between points P_0 and P_1: $t_r = 0.01$
- between points P_1 and P_2: $t_r = 0.02$
- between points P_2 and P_3: $t_r = 0.03$
- between points P_3 and P_4: $t_r = 0.04$
- between points P_4 and P_5: $t_r = 0.05$

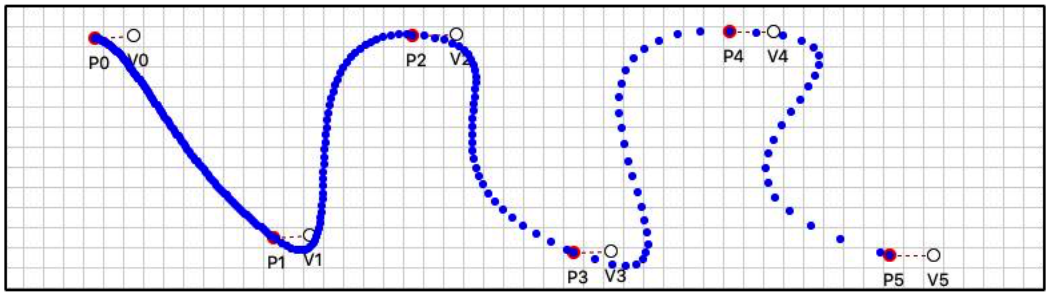

Figure 8. Hermite curve with 6 points, with different t_r values per arc

6. EDUCATIONAL RESULTS

The potential of the software presented in the previous sections was tested on control groups in education. We wanted to verify the following hypothesis (marked H0 and H1).

H0: the pedagogical software produced is an effective tool for teaching the subjects concerned.

H1: the pedagogical software produced does not help students to understand the material.

The testing was carried out on groups of students participating in the courses "Computer Graphics" and "Graphic Editors" in the academic year 2022/23, where half of the groups used the software. The knowledge assessment consisted of four practical exercises. During the exam, the students were no longer allowed to use the software, they could only rely on their theoretical knowledge and the experience gained from experiments with the software. First, we will describe the types of tasks and then the proportions of students' results.

6.1. Task 1: Specify the vectors associated with the Hermite arc

The students' task was to draw the direction vectors that affect the path of the Hermite arc shown on the left in Figure 9. V_0 and V_1 are the factors affecting the direction of the curve, so the vectors had to be aligned to the curves accordingly. The solution is also shown on the right in Figure 9, using the software.

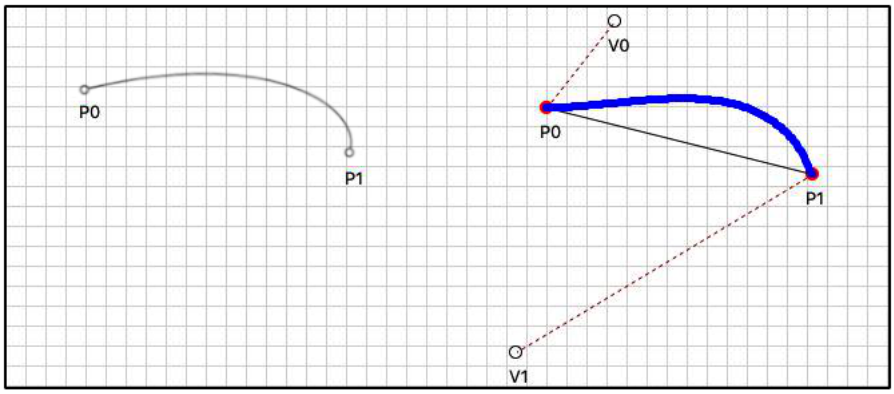

Figure 9. On the left: Hermite curve without V vectors; On the right: solve the problem

6.2. Task 2: Draw the Hermite arc starting at point P_0 and arriving at point P_1 if vectors V_0 and V_1 are given

The students' task was to draw the expected Hermite curve given the points P_0, P_1 and the vectors V_0, V_1 on the left side of Figure 10. From the interpolation property, the curve must start from P_0 and end in P_1. V_0 and V_1 are the factors affecting the direction of the curve. The solution is also shown on the right in Figure 10, using the software.

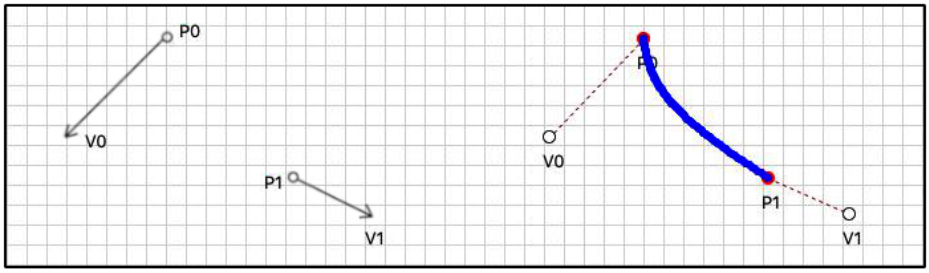

Figure 10. On the left: P points and V vectors of Hermite curve; On the right: solve the problem

6.3. Task 3: Draw the control points P_0, …, P_4 for the Bézier curve

The students were asked to give the positions of control points P_0, P_1, P_2, P_3 and P_4 for the curve on the left in Figure 11. Since it is a Bézier curve, we can use its approximating property, P_0 is the starting point of the curve, P_4 is the arrival point and the other points are points that influence the direction of the curve and are only approximated. The solution is also shown on the right in Figure 11, using the software.

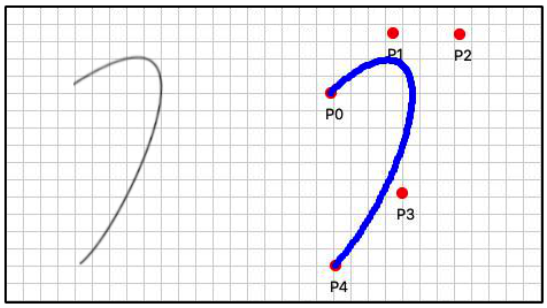

Figure 11. On the left: Bézier curve without P points; On the right: solve the problem

6.4. Task 4: Add the control points P_0, …, P_3 of the Coons curve

The students were asked to give the positions of control points P_0, P_1, P_2 and P_3 for the curve on the left in Figure 12. Since the starting point $R(0)$ of the Coons curve is influenced by control points P_0, P_1 and P_2, and the arrival point $R(1)$ is influenced by points P_1, $P_2$and P_3, students had to give the positions of the points accordingly. The solution is also shown on the right in Figure 12, using the software.

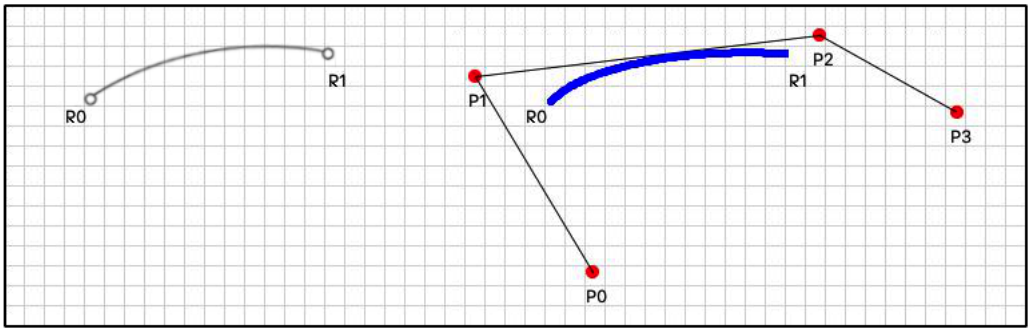

Figure 12. On the left: Coons curve with $R(0)$ starting and $R(1)$ ending points; On the right: solve the problem

6.5. The survey results

The students when tested for frequency, students who did not use the software answered correctly 42% of the time, while students experimenting with the software answered correctly 79% of the time. Separately, the frequency of correct answers for the four tasks is shown in Figure 13 (students not using the software in red and students using the software in green).

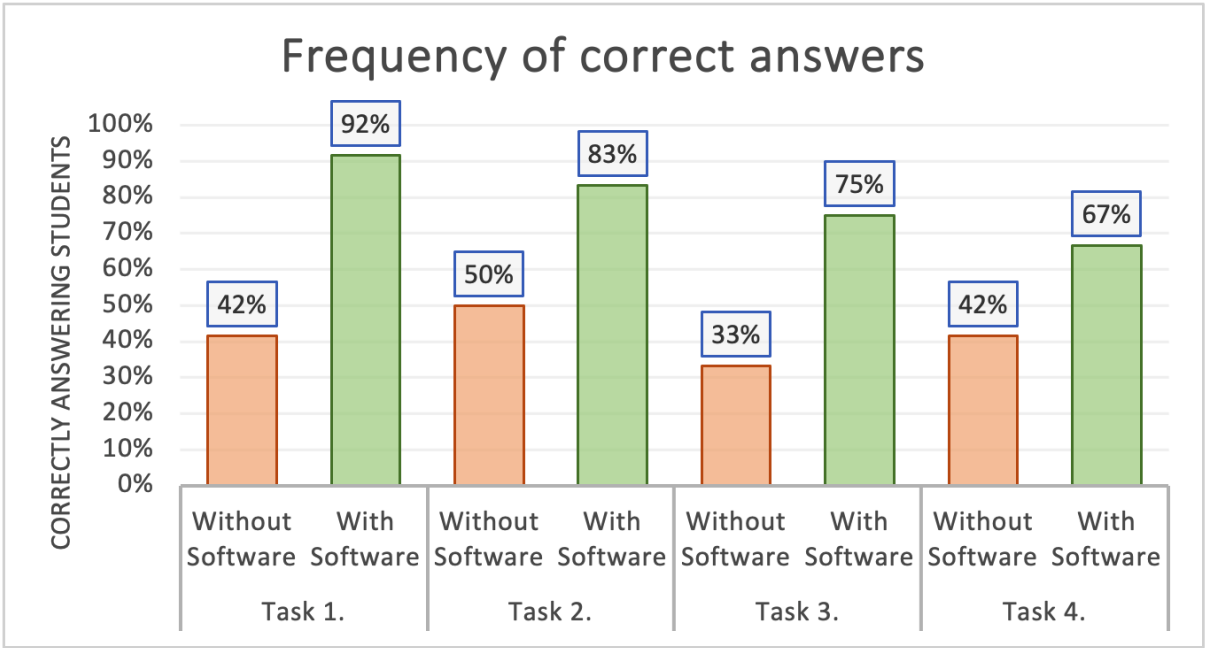

Figure 13. Results

6.6. Student Feedback

Following the testing, we called the students together for a discussion, where we asked for their opinions on how the software helped them during the exercises and in preparing for the exam.

- *"I like that the many pop-up information windows clearly guide the user on how to use it. This makes it easy to use the software even if you don't have prior information on how to use it."*
- *"I liked the simple interface. Also, the fact that the program itself didn't require a lot of storage space and didn't need to be installed. I just downloaded it and used it."*
- *"My other subject is not maths, so it's a bit difficult for me to learn just from the formulae. But this program makes it much easier to understand the algorithms that describe curves."*
- *"Putting the points listed in time t back into the formula helped me understand the mathematical formulas that determine the position of the curves."*
- *"The controllable simulation is definitely the biggest positive of the program."*
- *"What I liked about the experiments was that, in addition to the speed of the simulation, you could also control the frequency of the points that make up the curve. That way, when I experimented, I could quickly check that the final curve was correct for the task."*
- *"I tried to do the third-degree Bézier curve in Geogebra. It worked, but it does not have the simulation settings that your program has."*
- *"I like that I had the opportunity to try n-degree curves. I got much more interesting shapes with a higher number of degrees."*
- *"I didn't just use the canvas to set the control points. It proved to be a very quick procedure to enter my own coordinate points in the .csv file and have the program read them in."*
- *"I think the archiving function is very good. I like the fact that I don't have to take a screenshot to save the graphical output, I can simply save the picture. Saving the points also helps, so I can easily save the coordinates of my project and the output."*
- *"The exported coordinates can be easily refined within the .csv file, so I don't need to re-enter all the control points to refine the curve, I just overwrite the ones I need."*
- *"An interesting feature is that I can use the control points of one type of curve to plot the other. This makes properties such as the position of the last vector that influences a Hermite curve clear, for example, the last point of the Bézier curve where the curve ends. Conversely, the last control point affecting a Bézier curve will be the vector corresponding to the last point of the Hermite curve, where the curve no longer interpolates."*
- *"I find the import function interesting. It allows me to see what shape Hermite, Bézier and Coons curves are drawn by the software with the same points. With well-chosen points I got interesting shapes."*

7. CONCLUSIONS AND PROSPECTS FOR FURTHER RESEARCH

The use of our software has received positive feedback. In classes where students had the opportunity to experiment with the software, exam results were significantly better compared to classes without access to the software. Students' feedback has also reflected that the quality of our software has improved. During the semester, in addition to the positive feedback of students, there were several comments that improved the quality of the application. Here are a few improvements that have emerged from students' feedback. While using the application,

students noticed, in the students' own words, "ghost points", which were caused by the error that, when trying to place two points consecutively while plotting Hermite curves, the direction vectors would start from an invisible point (a position where they could not place another point). This resulted in graphical output that was difficult to interpret. An additional aesthetic flaw was that the coordinates describing the position of the cursor fell outside the canvas when we were very close to the edge of the canvas. Aside from the identified bugs, additional recommendations have been proposed. These suggestions don't pertain to bugs, but rather involve adding content to enhance comprehension. One example was the addition of predefined patterns. By these patterns, it is meant that, in addition to the freely definable points, it is possible to plot predefined shapes, thus illustrating the operation of the curves with concrete built-in examples. Taking into account the constructive criticism of the students, we have corrected the errors concerned and added to the software content according to the suggestions.

As we can see, the use of ICT tools is essential in the teaching of computer graphics. Teachers need to find the right tools to help students explore and better understand the basic concepts of the subject. Our goal was to create a self-built application that includes all the curves within an educational platform and satisfies all the functions that facilitate the teaching of the course: displaying mathematical models, applying interactive simulation, archiving and monitoring projects. Our software can run on both Windows and MacOS operating systems. We have not yet achieved full platform independence, but in the future, we aim to create a web application that can be run online.

REFERENCES (TRANSLATED AND TRANSLITERATED)

[1] M. V. Shvetsky, "Methodological system for the fundamental training of future teachers of computer science in higher education institutions undergoing a two-stage education", D.Sci. thesis, St. Petersburg, 1994. pp. 480

[2] V. Grinshkun, E. Y. Bidaibekov, S. Koneva, G. Baidrakhmanova, "An Essential Change to the Training of Computer Science Teachers: The Need to Learn Graphics," *European Journal of Contemporary Education*. vol. 8, 2019. pp. 25-42. doi:10.13187/ejced.2019.1.25.

[3] A. Peternier, D. Thalmann, F. Vexo, "Mental Vision: A Computer Graphics Teaching Platform," in *Pan, Z., Aylett, R., Diener, H., Jin, X., Göbel, S., Li, L. (eds) Technologies for E-Learning and Digital Entertainment*. Edutainment 2006. Lecture Notes in Computer Science, vol 3942. Springer, Berlin, Heidelberg, 2006. doi: https://doi.org/10.1007/11736639_31.

[4] E. Eyad, "Enhanced interactivity and engagement: Learning by doing to simplify mathematical concepts in computer graphics and animation", IEEE Global Engineering Education Conference, EDUCON, 2012. pp. 1-8. doi:10.1109/EDUCON.2012.6201073.

[5] M. R. A., García-Valcárcel, T. T. Javier, "The incorporation if ICT in higher education," The contribution of ROC curves in the graphic visualization of differences in the analysis of the variables. *British Journal of Educational Technology*, 2012. pp. 1-19.

[6] K. Czakóová, "Mathematical Model Based Iteractive Simulations in Education", presented at the 12th annual International Conference of Education, Research and Innovation, Seville, Spain, 2019. pp. 10120-10125. doi: 10.21125/iceri.2019.2479.

[7] M. Csóka, D. Paksi, K. Czakóová, (2022) "Bolstering Deep Learning with Methods and Platforms for Teaching Programming," AD ALTA: *Journal of Interdisciplinary Research*. vol. 12, 2022. pp. 308-313. doi:10.33543/1202308313.

[8] R. Klein, F. Hanisch, W. Straßer, "Web-based teaching of computer graphics: concepts and realization of an interactive online course" in ACM SIGGRAPH 98 Conference abstracts and applications (SIGGRAPH '98). Association for Computing Machinery, New York, NY, USA, 1998. pp. 88–93. doi:https://doi.org/10.1145/280953.281013.

[9] F. T. Marchese, "Teaching computer graphics with spreadsheets," in ACM SIGGRAPH 98 Conference abstracts and applications, ser. SIGGRAPH '98. New York, NY, USA: ACM, 1998, pp. 84– 87.

[10] D. Paksi, M. Csóka, Sz. Svitek, "Spreadsheets Feasibility for Modeling in Education", in 14th International Conference on Education and New Learning Technologies, Palma, Spain, 2022. pp. 5855-5861 doi:10.21125/edulearn.2022.1371.

[11] M. Pla-Castells, I. García-Fernández, J. Gimeno, I. Ferrando, I, "Computer Animation to teach interpolation". *Model. Sci. Educ. Learn*. 2019. pp. 31-46.
[12] B. D. Wong, "Bezier Curves and Surfaces in the Classroom," *Mathematical Modelling in Education and Culture*, 2003. pp. 173-180.
[13] D. R. Gutierez, F. J. Jimenez-Hornero, J. V. Giraldez, "A Computer Application for Teaching and Learning Approximation and Interpolation Algorithms of Curves," *Computer Applications in Engineering Education*, vol. 19(1), 2011. pp. 40-47.
[14] N. Guid, S. Kolmanic, D. Strnad, "SURFMOD: teaching tool for parametric curve and surface methods in CAGD based on comparison and analysis," *IEEE Transactions on Education*, vol. 49, 2006. pp. 292-301.
[15] J. R. Miller "*freeform*: A Tool for Teaching the Mathematics of Curves and Surfaces," *Computer-Aided Design and Applications*, vol. 7 no. 2, 2010. pp. 257-267, doi:10.3722/cadaps.2010.257-267.
[16] P. Bo, X. Zhang, "Design of Programming Exercises for a MOOC Course of Computer Graphics", *DEStech Transactions on Social Science, Education and Human Science*. 2018. pp. 51-56, doi:10.12783/dtssehs/icems2018/20096.
[17] A. A. Zakharov, Y. V. Zakharova, (2022). "The use of modern computer technologies in the study of the Geometric Modeling course," *J. Phys.: Conf. Ser*., 2022. pp. 1-8. doi:10.1088/1742-6596/2308/1/012007
[18] J. Žára, B. Beneš, "Křivky a plochy," in Moderní počítačová grafika, 2nd ed., Brno: Computer press, Brno, 2004. pp. 177–233.
[19] I. Juhász, S. Lajos, "Görbék leírása és megjelenítése and Felületek leírása és szemléltetése," in *Számítógépi grafika*, 1st ed., Miskolc, 2007. pp. 65-83.
[20] L. Szirmay-Kalos, "A geometriai modellezés," in *Számitógépes grafika*, 1st ed., Budapest: ComputerBooks, Budapest, 2001. pp. 47-62.
[21] D. Salomon, "*Curves and Surfaces for Computer Graphics*", New York. Springer New York, NY, 2006.
[22] O. Takáč, "Görbék és felületek," in *A számítógépes grafika*, 1st ed., Komárno: Univerzita J. Selyeho, Komárno, 2016. pp. 256-277.
[23] S. BiBi, M. Abbas, K. R. Miura, M. Y. Misro, "Geometric Modeling of Novel Generalized Hybrid Trigonometric Bézier-Like Curve with Shape Parameters and Its Applications," *Mathematics*, vol. 8, 967, 2020. doi:https://doi.org/10.3390/math8060967.
[24] M. Ameer, M. Abbas, K. T. Miura, A. Majeed, T. Nazir, "Curve and Surface Geometric Modeling via Generalized Bézier-like Model," *Mathematics*, vol. 10, 1045, 2022. https://doi.org/10.3390/math10071045
[25] T. Kmeť, M. Kmeťova, "Bézier curve parametrisation and echo state network methods for solving optimal control problems of SIR model," *Biosystems*, vol. 186, 104029, 2019. doi:https://doi.org/10.1016/j.biosystems.2019.104029.
[26] S. Chen, B. Su, "Geometric modeling with quasi-Hermite curves and surfaces," presented at the 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, Huangshan, China, 2009. pp. 300-305. doi: 10.1109/CADCG.2009.5246886.
[27] L. Piegl, "Hermite-and Coons-like interpolants using rational Bézier approximation form with infinite control points," *Computer-Aided Design*, vol. 20(1), 1988. pp. 2-10. doi:https://doi.org/10.1016/0010-4485(88)90135-2.
[28] X. Ye, "Generating Bézier points for curves and surfaces from boundary information," *Computer-Aided Design*, vol. 27(12), 1992. pp. 875-885. doi:https://doi.org/10.1016/0010-4485(95)00001-1.

Text of the article was accepted by Editorial Team 04.07.2023

НАВЧАННЯ ІНТЕРПОЛЯЦІЇ ТА АПРОКСИМАЦІЇ КРИВИХ ЗА ДОПОМОГОЮ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ

Норберт Аннуш
магістр, магістр, аспірант, асистент професора
Університет ім. Яноша Шеє, факультет інформатики, м.Комарно, Словацька Республіка
ORCID ID 0000-0002-8535-2042
annusn@ujs.sk

Ондрей Такач
доктор філософії, доктор, завідувач кафедри
Університет ім. Яноша Шеє, факультет інформатики, м.Комарно, Словацька Республіка
takaco@ujs.sk

Анотація. Використання цифрових інструментів в освітньому процесі стало вже звичною справою. Особливо це стосується уроків інформатики. Наше дослідження спрямоване на оцінювання успішності з комп'ютерної графіки студентів вищих навчальних закладів будь-якої предметної галузі. Комп'ютерна графіка та її вивчення є дуже багатогранними. Краще зрозуміти алгоритми, за якими виконуються графічні коригування, набагато легше, якщо студент не тільки теоретично знайомий з цією галуззю та її процедурами, але й має можливість випробувати їх на практиці. Нашою метою було розробити навчальний інструмент, який би дозволив студентам, що відвідують заняття з графіки, дізнатись про можливості інтерполяції та апроксимації кривих і протестувати їх. У нашому дослідженні ми зосередились на трьох конкретних типах кривих: кривих Ерміта, Безьє та Куна. Для універсального використання студенти можуть моделювати відповідні криві у випадках n-го степеня. При розробці ми зосередились на моделюванні математичних формул, що описують криві, намагаючись створити простий у використанні навчальний інструмент. З цією метою було розроблено комп'ютерне програмне забезпечення. Під час дослідження проводились якісні опитування в експеритментальних групах студентів. Ми також провели інтерв'ю зі студентами, які використовували програмне забезпечення, щоб оцінити його сильні та слабкі сторони. На основі цього зворотного зв'язку ми оптимізували програмне забезпечення та визначили напрямок його майбутнього вдосконалення. Практичне застосування свідчить про те, що навчальне програмне забезпечення позитивно впливає на результати навчання студентів. У цій статті представлено математичні моделі, які описують оброблені криві, а також можливості програми і графічні результати. Крім цього представлено результати тестування студентів та їх відгуки про навчальне програмне забезпечення.

Ключові слова: навчальне програмне забезпечення; моделювання; імітація; інтерполяція; апроксимація; комп'ютерна графіка.

(cc) BY-NC-SA

This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.