MSEL
Modelling in Science Education and Learning

Modelling in Science Education and Learning
Volume 12 (1), 2019 doi: 10.4995/msel.2019.10976.
Instituto Universitario de Matemática Pura y Aplicada
Universitat Politècnica de València

Computer Animation to teach interpolation
Animación por ordenador para enseñar inperpolación

Marta Pla-Castells, Ignacio García-Fernández, Jesús Gimeno, Irene Ferrando
Universitat de València
marta.pla@uv.es, ignacio.garcia@uv.es, jesus.gimeno@uv.es, irene.ferrando@uv.es.

Abstract

While mathematics courses are a basic topic in engineering studies, they are often considered as a difficult subject by students. In this work we present a learning experience based on computer animation and using the perspective of mathematical modelling. Our goal is to provide the students with a context that motivates the study of function interpolation. We present a problem statement that is intended to be solved by means of the Modeling Cycle. The development of the activity and the strategies identified during the process are presented and discussed.

Aunque las asignaturas de matemáticas son un tema básico en los estudios de una ingeniería, a menudo son considerados por los estudiantes como una asignatura difícil. En este trabajo presentamos una experiencia de aprendizaje basada en la animación por ordenador mediante el uso de la modelización matemática. Nuestro objetivo es proporcionar a los estudiantes un contexto que motive el estudio de la interpolación de funciones. Presentamos un planteamiento del problema que se pretende resolver mediante el Ciclo de Modelización. Se presentan y discuten tanto desarrollo de la actividad como las estrategias identificadas durante el proceso.

Keywords: Modeling cycle, Interpolation, Computer animation, Computer graphics, Engineering education.
Palabras clave: Ciclo de modelización, Animación por ordenador, Gráficos por ordenador, educación en ingeniería.

1. Introduction

Mathematics courses are a core element during engineering undergraduate studies. Mathematics, however, remain to be considered as a difficult topic by Higher Education students; since mathematics is typically taught at early stages of engineering degrees, students do not have a perspective of their importance in their development as engineers (Gil et al., 1991, Font et al. 1995, Sánchez-Pérez et al., 1999).

Since part of engineering practice involve the description of the problems that are to be solved, it seems clear that mathematical modelling competencies should play a central role in engineering education. In agreement with this idea, the European Society of Engineering Education has included the mathematical modelling competences in the Curriculum Document of the Mathematics Working Group (Alpers et al., 2013).

In this work we present an approach to teaching interpolation of functions through a modelling perspective Frey, 1989, Ärlebäck, 2009, Borromeo-Ferri, 2017). This approach provides the students with an usage context that motivates the topic and encourages them to learn interpolation techniques not as part of a course but as the way to solve a real engineering problem the could face in their later professional activity.

Our purpose is to describe an experience that has been carried out with students of the third course of the Computer Science degree in a subject entitled Computer Animation. When the students followed the activity, they had already studied two basic mathematical content courses (algebra and calculus for engineers) and are supposed to know some elementary facts about numerical methods.

The rest of the paper is organized as follows. In what remains of this section, we describe the context of the experience and the theoretical background behind our proposal. In Section 2 we present the problem statement and describe the expected resolution process through the Modelling Cycle (Blum y Leiß, 2007, Borromeo-Ferri, 2017). In Section 3 we review some interpolation methods that are relevant to our activity. Section 4 describes the development of the activity in the classroom, presenting some of the productions generated by the students. In Section 5 the results of the learning experience are discussed. Finally, Section 6 addresses the conclusions and future research lines.

1.1. Context

For decades, during the animation process artists have used the technique of key-frames (Kerlow, 2004). In traditional (drawn) animation an artist will draw first a reduced set of frames that clearly define the action that is performed by a character. These are called the *key-frames*. Once these frames have been completed, the remaining frames between each pair of consecutive key-frames are drawn to ensure continuity in the action, during what is called the *inbetweening* process.

This technique has a direct translation in the context of computer animation. During a typical computer animation production artists use some animation software in which the scene and the characters are represented as three-dimensional geometric objects, mostly triangle meshes subject to transformations (translations, rotations, scalings,. . .). Animation and movement is defined by making these transformations to be functions of time.

To create the movement and the action of a scene, animators define the pose of the characters and the position of the involved objects. This process is again done using the key-frames technique; the animator defines the state of the scene at certain frames that properly determine the action to be performed. In this case, however, the remaining frames between two consecutive key-frames are automatically generated by the software. During this process, the functions that define the animation are generated by interpolating the state of the scene at the key-frames.

The numerical methods used to perform this interpolation are present in the animation software that is used by the animation studios. Nevertheless, it is very common that each studio personalizes different aspects of the animation software. Moreover, in high quality productions, specially in feature films, it is also common that the functionality of the software needs to be extended to meet the requirements of the production.

These tasks are preformed by software engineers or programmers, using the extension capabilities of the animation software, which typically consist on the development of a series of scripts on a programming language, most commonly python. This activity is known in the animation industry as *tool development*. We propose a task to learn interpolation using mathematical modelling based on the role of a tool developer in an animation studio. The students will have to develop a small software tool integrated in the animation software Blender (Blender, 2018) to generate trajectories from a set of keyframes.

The problem is proposed in a Computer Science degree with a major orientation towards multimedia content generation and management. Students have programming skills and have been introduced to the basic rules that govern movement in animation –the 12 principles of animation– (Kerlow, 2004).

1.2. Theoretical framework

Some researches show that, also in higher education, the introduction of complex mathematical concepts through contextualized tasks helps students to give sense to mathematical contents in a more significant way (Langereis, 2013).

Modeling tasks, understood as those that pose a complex, open and contextualized problem, are appropriate for engineering students to use mathematical tools on problems close to their study context. According to Borromeo-Ferri (Borromeo-Ferri, 2017), a problem that is appropriate for modeling tasks must have the following properties: it should be open, complex, realistic and authentic. These characteristics imply that the task would pose a problem to the students and should be solvable through modelling.

Different works describe the process of solving a modeling task through a cycle. In our case, the *Modeling Cycle*, based on Borromeo-Ferri (Borromeo-Ferri, 2017) and on Blum and Leiß (Blum y Leiß, 2007), will be the key to guide the students' work during the development of the described experience. Following the cycle, the resolution process involves several steps which can be summarized as: identifying the real situation, building a situation model, building a real model, building a mathematical model, producing a mathematical result and, finally, translating it into a real result. Indeed, the *Modeling Cycle* is described in terms of actions of the solver and, therefore, it is easy to use it as a lens to identify the work of the students –also in terms of competencies involved in the resolution process, as presented in (Kaiser, 2015)–. In the following section, we will consider these properties in the design of the problem statement and also in the analysis of the resolution process of the problem.

2. Problem statement and resolution process

In the problem that guides the proposed activity, the students have to develop a piece of software that helps to build a computer animation scene fitting a set of requirements. These requirements are described in terms of the motion of some objects in the scene. However, the information is not given in a precise, numerical, way. Instead, a movement is described with a series of informal properties, such as *the object accelerates during a few seconds*, or *after remaining still for a while...* An example statement could be:

> A girl is holding a balloon. At a certain moment, the thread escapes from her hand, and the balloon starts raising. After a few seconds, a bird punctures the balloon, which falls to the ground.

The students are expected to build a piece of software, integrated in Blender, that allows them to reproduce a scene from a similar description. More precisely, their software will have to use a reduced set of times and locations and generate a trajectory for an object based on these data.

2.1. Characteristics of the problem statement

In the sequel, we identify those properties of the problem that characterize it as a modelling task following Borromeo-Ferri (Borromeo-Ferri, 2017).

Open: The question is open, in the sense that the students do not have a unique way to solve the problem.

Complex: The students need to make decisions about the numerical data they are going to use, taking into account their knowledge about how movement must be animated.

Realistic: The proposed problem is based on a situation that actually can take place in an animation or a video-game studio.

Authentic: The problem is not crafted from scratch for the course. Instead it is taken from actual animation scenes.

Poses an actual problem: It cannot be solved trivially from a mathematical point of view. It requires the elaboration of a mathematical formulation and a critical decision about how to build the solution.

Solvable through modeling: Students must model the concept of movement through geometric concepts and numerical mathematics. The information about the mathematical representation is not given in the statement.

The movement is subject to some restrictions that are not included in the statement. In particular, the different animated movements must follow the Principles of Animation. This information has been the subject of previous lessons in the course, and can be considered what Borromeo-Ferri calls an Extra Mathematical Knowledge (Borromeo-Ferri, 2017).

Resolution process

The task is expected to be solved following the Modelling Cycle (Blum y Leiß, 2007), next we will describe these steps of the resolution process in the framework of the proposed task.

Real Situation. The real situation involved in our problem is task of providing a computer animation studio or software company with a module that is able to reproduce a movement as is can be described by an animator.

Situation Model. A mental representation of the situation should describe the context of the problem, including the tools that are available to solve the problem. As a result of the Situation Model, the students have to clearly understand what is the problem to be solved and what is the form the solution is expected to have. In the more general case, the students are expected to build (using a programming language) a module for a computer animation software that is able to reproduce a movement. A restricted version of the problem could be the construction of a set of mathematical expressions that describes the position of each object as a function of time.

Real model. From the statement of the problem, the students will need to build a more precise description of the movement. As it has been discussed, the problem will have very little information about the actual locations and times of the movement. Instead, it will be described in a very informal and imprecise way. The students need to decide that data they need and estimate or set the missing values. In this phase, the students must use the standards in computer animation for generating movement; the Principles of animation (Extra-Mathematical Knowledge).

Mathematical model. The students have to build a mathematical description of movement. The standard modeling solution to describe motion use differentiable curves in space as the main mathematical object. Moreover, they need to understand that the data will come as a series of times and associated locations. In this case, the classical technique of keyframe animation (Kerlow, 2004) will be the required EMK.

Mathematical result. The mathematical solution will be some mathematical description of the movement based on the mathematical model. Typically some form of interpolation or approximation. The former is preferred since it meets the standard requirements in a typical animation workflow.

Real results. The construction of a piece of software that is able to generate a video with the prescribed motion is the result. The result will be validated taking into account the requirements described in the Real Model.

It is important to note that the resolution process usually involves several iterations of the cycle. Moreover, it is common that the students do not go through it linearly (Borromeo-Ferri, 2017).

3. Interpolation of functions

Before giving the details about the development of the activity, in this section we present the mathematical contents that the students are expected work on during the activity proposed in Section 4. The main concept that we want to introduce is interpolation of functions. We will formalize the construction of movements in keyframe-based animation as a tool to introduce this subject.

In computer animation, the location of any object in a scene is given by its Cartesian coordinates, $\mathbf{p} = (x, y, z)$. Thus, generating the movement of an object requires building a series of locations as the animation time advances,

$$\mathbf{p}(t) = (x(t), y(t), z(t)).$$

The use of keyframe-based animation, implies that an artist will define a series of locations for a set of given times, $\mathbf{p}_0 = \mathbf{p}(t_0), \mathbf{p}_1 = \mathbf{p}(t_1), \ldots, \mathbf{p}_n = \mathbf{p}(t_n)$. The animation software must, then, generate a trajectory that interpolates these data, component-wise. Next we present some interpolation methods that are within the scope of the course. We restrict to methods based on polynomial functions, since these are the most commonly used interpolants in computer animation and computer graphics.

Their properties are discussed according to the suitability of each method for an animation system. We will not go through an exhaustive description of the theory. The interested reader can refer to any undergraduate level monograph on numerical analysis, such as (Burden y Faires, 2010). Since interpolation will be done for each component separately, in this section the discussion will be done for interpolation of real functions unless stated otherwise.

3.1. Langrange polynomial

The problem of finding a polynomial function that interpolates the values y_i at times t_i, $i = 0, \dots, n$ is equivalent to finding its coefficients. If we consider a linear function, we need two points (t_0, y_0) and (t_1, y_1), with $t_0 \neq t_1$ to determine, in a unique way, the polynomial

$$f(t) = \frac{t - t_1}{t_0 - t_1} y_0 + \frac{t - t_0}{t_1 - t_0} y_1.$$

In general, a unique polynomial of degree at most n can be found with $n + 1$ interpolation points, which is called the Lagrange interpolation polynomial (Burden y Faires, 2010).

This strategy, however, has the important limitation that it provides a global interpolation; i.e. the change in any of the points to be interpolated, (t_i, y_i) can lead to a new polynomial that has a different value in all the points of the interval $t \in [t_0, t_n]$ for which $t \neq t_i$. If this behaviour is translated to the context of key-frame based computer animation, it means that the alteration of a single key-frame would affects the whole movement sequence, which is highly undesirable. For his reason, in our context Lagrange interpolation is rarely used and will not be further discussed.

3.2. Cubic splines

Instead of using a polynomial to interpolate a set of points for arbitrary n, usually piecewise interpolation is the choice in computer graphics and, particularly, in computer animation. The most common approach is to use a set of n polynomials of degree three, in which polynomial $f_i(t)$ is defined in the interval $[t_{i-1}, t_i]$, for $i = 1, \dots, n$ (Lengyel, 2012).

Among the set of cubic piecewise functions, cubic splines are defined to guarantee continuity up to the second derivative. That is, a cubic spline is a piecewise cubic polynomial that guarantees

$$f_i(t_i) = f_{i+1}(t_i) = y_i; \quad f_1(t_0) = y_0; \quad f_n(t_n) = y_n,$$

and, in addition,

$$f_i'(t_i) = f_{i+1}'(t_i); \quad f_i''(t_i) = f_{i+1}''(t_i).$$

This leads to a system of $4n-2$ equations to determine $4n$ coefficients. Hence, two additional equations are necessary, which typically are defined as $f_1''(t_0) = f_n''(t_n) = 0$, leading to so-called *natural splines*. However, if the second derivative $f''(t)$ is assumed to vary linearly within each interval

$$f_i''(t) = \frac{f_{i-1}''}{t_i - t_{i-1}}(t_i - t) + \frac{f_i''}{t_i - t_{i-1}}(t - t_{i-1}), \; t \in [t_{i-1}, t_i],$$

where $f_i'' = f''(t_i)$, then the system of equations can be reduced to a system with only $n - 1$ equations with the second derivatives f_i'', $i = 1, \dots, n-1$ as unknowns (Burden, Faires, 2010, Lengyel, 2012).

Second order continuity of cubic splines have the property of being a possible movement of an object under the action of forces that are continuous along time, which could make them interesting from the point of view of computer animation. However, as it happened with the Lagrange polynomial, a cubic spline provides a global interpolation, where a change in a single point can affect the whole function.

3.3. Hermite interpolation

If the second order continuity is relaxed to continuity in the first derivative, then an independent set of four equations can be written to determine $f_i(t)$ in each segment $[t_{i-1}, t_i]$. This system of four equations is given by

$$f_i(t_{i-1}) = y_{i-1}; \quad f_i(t_i) = y_i; \quad f_i'(t_{i-1}) = y_{i-1}'; \quad f_i'(t_i) = y_i',$$

where $y_i' = f_i'$ are imposed values of the derivatives of f_i. The solution to this system is the cubic Hermite polynomial which, for the segment $[t_{i-1}, t_i]$, is given by

$$H_i(u) = (1 - 3u^2 + 2u^2)y_{i-1} + u^2(3 - 2u)y_i + u(u-1)y_{i-1}' + u^2(u-1)y_i'$$

where

$$u = \frac{t - t_{i-1}}{t_i - t_{i-1}}.$$

From this formulation, we can see that f_i is determined using only the data in the interval where it is defined. As a consequence, Hermite interpolation is a local interpolation method in the sense that the coefficients of each polynomial f_i will not change whenever f_j or f_j' are modified for $j \notin \{i-1, i\}$.

Hermite interpolation provides a higher control over the behaviour of the function, given that both the location and the velocity are defined by the user. This apparent benefit, however, can be a drawback in computer animation, since animation software does not usually include the option to explicitly define velocity at a given location in the way required by Hermite interpolation.

In the context of our exercise, the students will be asked to modify the user interface of the program to add the velocity as one of the parameters that can be defined for a trajectory. After this addition, Hermite interpolation is one of the methods that are expected to be implemented in the final result by the students.

3.4. Catmull-Rom splines

While the option of modifying the user interface to consider velocity enables the use of Hermite interpolation, this is not usually the best option; users of computer animation software, with an artistic background, often do not feel comfortable with the idea of tweaking numerical parameters to control movement.

In the mid seventies an interpolation method was roposed that allows certain control over the velocity modifying only the location values of a trajectory and which is one of the most common interpolation methods in the computer graphics industry (Catmull y Rom, 1974, Barry y Goldman, 1988).

Catmull-Rom splines are defined so that the derivative at interpolation point (t_i, y_i) is proportional to the difference $y_{i+1} - y_{t-1}$ with a coefficient of proportionality $\tau > 0$ that can be controlled by the user. A Catmull-Rom spline is a piecewise function that, in the interval $[t_{i-1}, t_i]$, is a polynomial of degree three and

$$f_i(t_{i-1}) = y_{i-1}, \quad f_i(t_i) = y_i, \quad f_i'(t_{i-1}) = \tau(y_i - y_{i-2}), \quad f_i'(t_i) = \tau(y_{i-1} - y_{i+1}).$$

This family of splines can be considered as a particular case of Hermite interpolation in which the velocities are automatically obtained as $y_i' = \tau(y_{i-1} - y_{i+1})$. The parameter τ is called *tension* and controls the shape of the the resulting curves. Figure 1 shows two examples of two-dimensional Catmull-Rom curves with different values of τ.

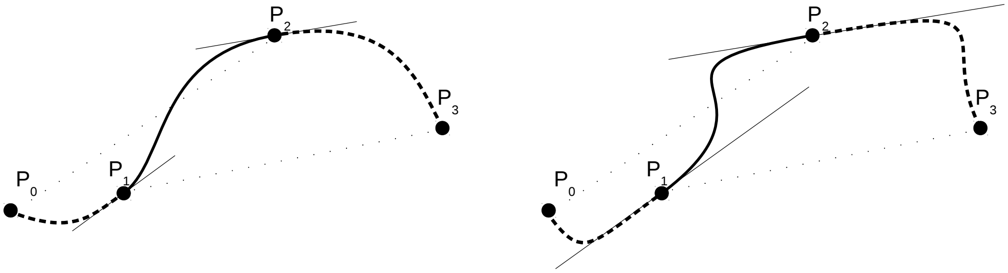


Figure 1: A 2-dimensional curve using Catmull-Rom splines. The interpolation is shown for the segment between $\mathbf{p}_1$ and $\mathbf{p}_2$. The tangent at point $\mathbf{p}_1$ is defined to be parallel to the line passing through $\mathbf{p}_0$ and $\mathbf{p}_1$. The curve on the left hand side uses $\tau = 0.5$ while the curve on the right uses $\tau = 1.2$. Solid segments indicate the tangent direction at a point, which are parallel to dotted lines, passing through adjacent points.

As a historical remark, it is noteworthy that Catmull-Rom splines are not only a technique used in computer graphics, but also are a method that was proposed in the context of computer graphics research. E. Catmull, who is one of the authors that proposed the method, was one of the founders of Pixar Animation Studios and, by the date this paper is written, is the president of the company.

3.5. Bézier curves

For the reasons discussed so far, among the four interpolation methods discussed, Catmull-Rom splines are the most common interpolation technique in computer graphics. However, there are many other techniques that appear frequently in computer animation and graphics applications. A thorough discussion of interpolation methods is out of the scope of this work. But we consider that at least Bézier curves should be mentioned (Burden y Faires, 2010, Lengyel, 2012).

Bézier curves were defined by designer Pierre Bézier when he was working in the French automotive industry to describe a parametrized curve. The main difference of Bézier curves compared to the other methods in this section is the fact that they use a set of points larger than just the points to be interpolated. An order three Bézier curve in the interval $[0, 1]$ will use four *control points*, $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ and $\mathbf{p}_3$. However, the curve only passes through $\mathbf{p}_0$, when $t = 0$, and $\mathbf{p}_3$, when $t = 1$, while points $\mathbf{p}_1$ and $\mathbf{p}_2$ are used to define the tangent vectors at $\mathbf{p}_0$ and $\mathbf{p}_3$, as seen in Figure 2.

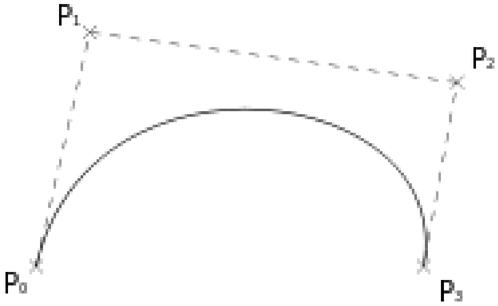


Figure 2: A cubic Bézier curve uses four *control points*. It interpolates, the first and last points, while the other two are used to define the tangent direction.

Bézier curves are a standard choice when developing graphical user interfaces for computer graphics, CAD and animation software. Indeed, the control points $\mathbf{p}_1$ and $\mathbf{p}_2$, which are not interpolated, are the way artists control tangent direction and module, having an intuitive vision of velocity.

In the task we propose, students are not expected to use Bézier curves. The reason is that, while the management of the points to be interpolated is very straightforward in Blender, by means of the insertion of key-frames, the insertion and access to additional control points would require more sophisticated programming skills. This complication in the development would distract them from the main goal, which is the development of interpolation methods.

4. Development of the activity

The activity was done in groups of three or four students during three sessions of two hours each. During the first two sessions, the tasks were structured in three phases, and the last session was left for autonomous work by the students. The sequence of tasks was done as follows. First the statement of the problem was presented, followed by some discussion. In a second phase a simpler, less open problem was proposed with the intention of helping the students to develop a solution strategy. Finally the original problem was addressed again to find a solution. Next we describe these steps in more detail, paying special attention to the second phase of the activity where the mathematical modelling activity was developed.

4.1. Exposition of the problem and discussion

First, the students were presented an statement similar to the one described in Section 2, and they were given thirty minutes to discuss possible solutions to the activity. During this discussion, some of the questions that arose were related to the missing information in the statement of the problem. Aspects such as the times at which the different movements start, or the way velocity changes were common questions at this point.

This kind of questions are commonly observed when the students face open problems, specially if they have little experience with this type of projects (Borromeo-Ferri, 2017). Moreover, since the solution also includes programming activities, there is a risk that the students pay too much attention on the implementation details. To prevent this, at this point we propose an alternative problem before the original task is addressed entirely.

4.2. Proposal of a reduced problem: mathematical modelling of movement

In the second step of the activity, we presented the students a less open problem to help them start the discussion and that involves very basic programming techniques. In this problem, the students could see a video with the movement of a cube along the x axis and they were asked to reproduce the movement in the video, following the same idea as in the previous exercise: they have to develop some code that generates the movement when executed inside Blender. In this case, however, they only had to develop a simple script of a few lines, without any integration with the Blender interface; the solution would consist, essentially, of a *for* loop that would go through the frames of the scene, calling a function that computes the location of the cube for each instant of time.

In the video provided to the students, there were marks that helped them know the distance travelled and a chronometer. Figure 3 shows an image of the video. In this new exercise, the students could play the video back as many times as they wanted, to know the location of the cube at different instants during the movement. Since the movement is along the x axis, and the

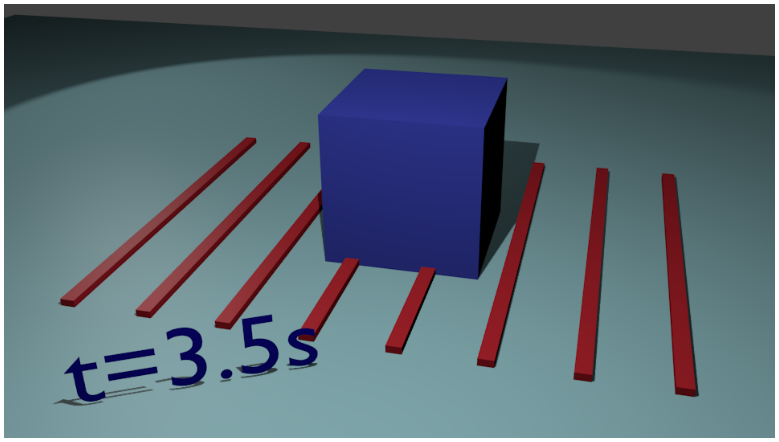
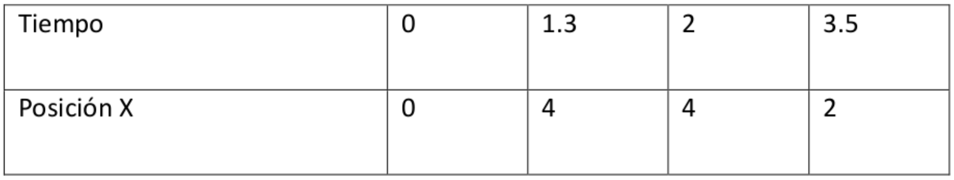


Figure 3: A still image of the video that has to be reproduced by the students. A mark on the ground appears every meter and a clock shows the time from the beginning of the movement.



Tiempo	0	1.3	2	3.5
Posición X	0	4	4	2

Figure 4: A table with pairs of time and distance values produced by one of the groups based on the movement observed in the video.

y and z coordinates remain constant, the location is equivalent to distance to the origin. Along with the video, the students were given notes on how movement can be represented by means of a function of position versus time. The notes included also examples of piecewise-defined functions.

The students were asked to propose a solution that could be implemented on a simple script of a few lines of code. In this session, however, the should not program their solution, but only elaborate the proposal. During forty-five minutes, they had to discuss their proposals in their group and they had the option to ask questions to the professor. However, the professor did not guide the students towards a predefined, supposedly right strategy. Instead, any strategy that could lead to a solution was accepted and the answers were directed to help them clarify and analyze its details.

In this phase, the students first decided to build a table with pairs of time and distance values. This table would be used later to generate positions between the recorded moments. We observed differences regarding the selection of the instants to be registered in the table. While some students decided to use frames located at regular distances, other students identified relevant instants of the movement and used those for their table. In the later case, the selected instants coincided, in most the groups, with the moments in which the object had zero velocity. Figure 4 shows the table produced by of one of the groups, in which the instants when the object stops or starts a movement are identified.

After registering the selected times and locations, the students addressed the task of generating the position of the object for the times missing in the table. It has to be taken into account that the students had been taught an introduction to interpolation during the prior university year. However, when presenting this task to the students the professor made no reference to the concept of interpolation, in order to let them search their own strategy. In this phase there was some discussion within the groups to decide the best procedure. In part of the groups the

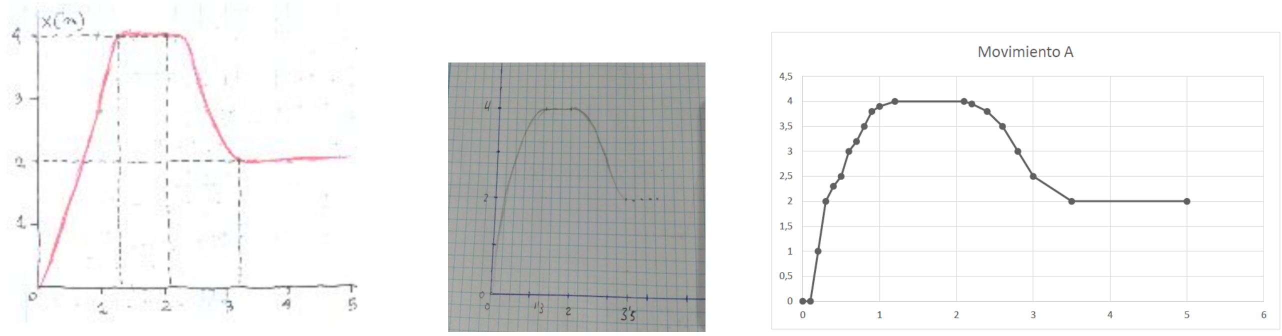


Figure 5: Three sketches of the function that represents location versus time, produced by three groups during the activity.

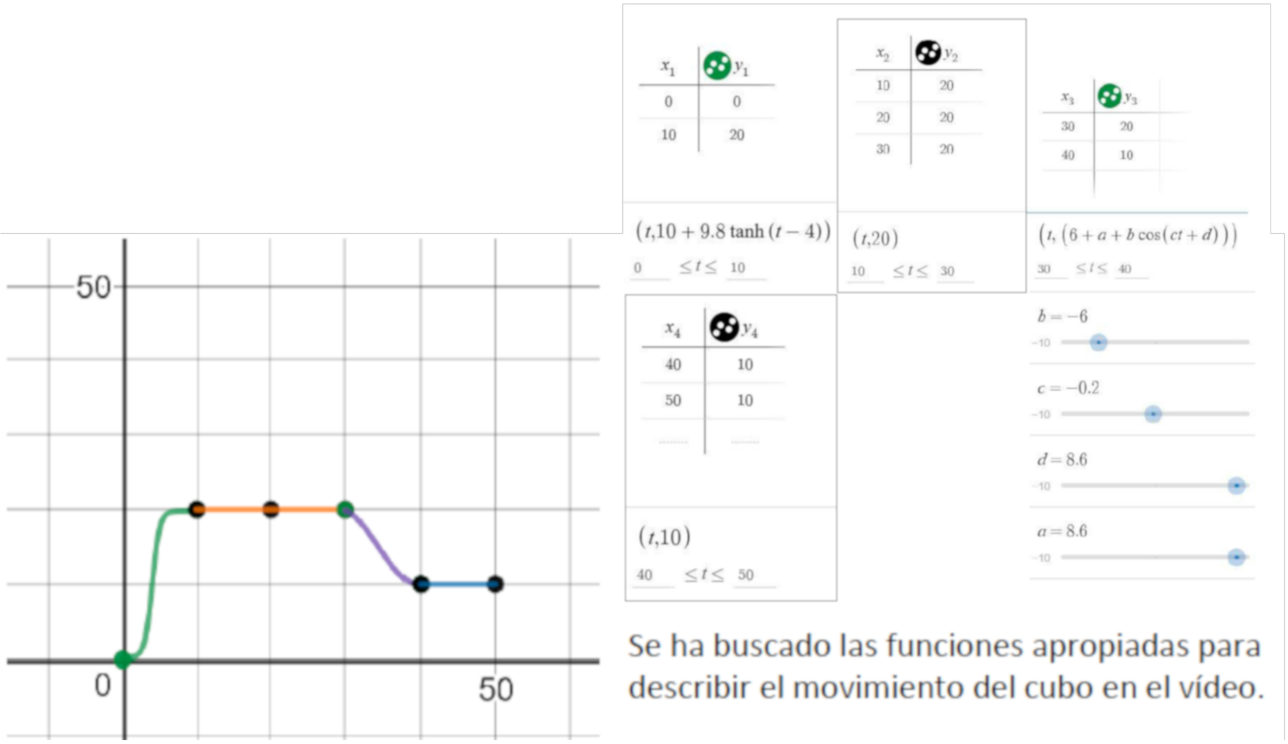
possibility of using linear interpolation was identified. This strategy however, come in different forms, mainly the use of the equation of a segment joining two points and the application of proportionality. In other groups, the concept of interpolation was proposed from the beginning, and the discussion focused on whether the methods they knew (namely linear interpolation and Lagrange's polynomial) were appropriate or not. In general, these students made a reduced use of graphical aid and based their discussion on the numerical data. A few groups, on the contrary, drew the points registered in the table and sketched a graph with the possible function relating position with time. Figure 5 presents the sketches produced by three groups to build the graph of the position of the cube as a function of time. In the figure, it can be observed that the two leftmost sketches are based on the points in which the object starts or ends its movements. The sketch in the middle was produced by the same group that presented the table of Figure 4. On the contrary, the third sketch uses additional points that cover the movements with higher resolution, although they are not evenly spaced.

Some of the groups that drew an sketch of the function of positions based their strategy on the identification of sections of their figure with the graph of known families of functions, such as trigonometric functions or polynomials of order two and three. E.g., the use of the cosinus function in the interval $[0, \pi]$ was proposed to describe an acceleration-deceleration process. Figure 6 presents an sketch of the function, and the interface of an online function plotting system used to generate it. It can be observed that $\tanh(x)$ function was used for the first section, while $\cos(x)$ function was used for the third one. This sketch was produced by the same group that generated the rightmost draft in Figure 5. The equations that are shown in Figure 7 correspond to the system of equations proposed by one of the groups to interpolate the values that had been written down in the table.

4.3. Sharing of the proposals

To conclude the first session, the proposals were shared and there was some discussion to analyze their strengths and weaknesses. This phase was initially driven by the professor, who started by asking some of the groups to briefly explain their solution strategy. The decision was to start by the strategies that had been identified by the professor as the less general or robust ones. However, this information was not given to the students, and they were presented as perfectly valid solutions. After a few questions, some debate was established as different groups started to explain their strategies in order to support their opinions and comments about those of the other students. Next we briefly present the arguments that we consider as most relevant from this discussion.

- Some of the groups proposed to use linear interpolation to generate the locations between



x_1	y_1
0	0
10	20

$(t, 10 + 9.8\tanh(t-4))$ $0 \le t \le 10$

x_2	y_2
10	20
20	20
30	20

$(t, 20)$ $10 \le t \le 30$

x_3	y_3
30	20
40	10

$(t, (6 + a + b\cos(ct + d)))$ $30 \le t \le 40$

$b = -6$

$c = -0.2$

$d = 8.6$

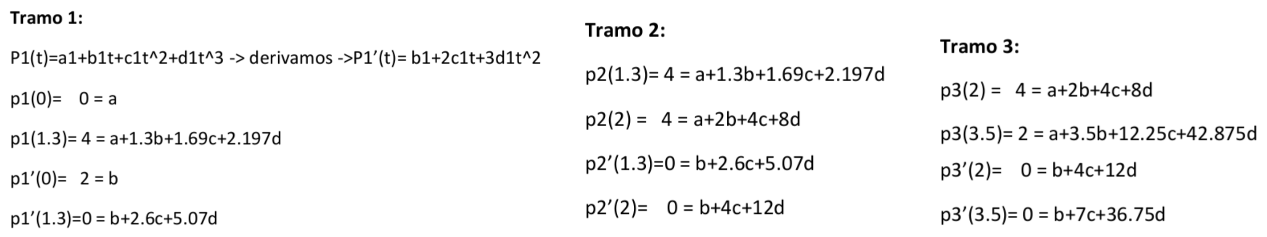
$a = 8.6$

x_4	y_4
40	10
50	10

$(t, 10)$ $40 \le t \le 50$

Se ha buscado las funciones apropiadas para describir el movimiento del cubo en el vídeo.

Figure 6: Some groups built a piecewise function by fitting the graph of known functions, such as trigonometric or hyperbolic functions, to the shape of their sketch. The panels that appear on the right hand side show the parameters of the functions used.



Tramo 1:

P1(t)=a1+b1t+c1t^2+d1t^3 -> derivamos ->P1'(t)= b1+2c1t+3d1t^2

p1(0)= 0 = a

p1(1.3)= 4 = a+1.3b+1.69c+2.197d

p1'(0)= 2 = b

p1'(1.3)=0 = b+2.6c+5.07d

Tramo 2:

p2(1.3)= 4 = a+1.3b+1.69c+2.197d

p2(2) = 4 = a+2b+4c+8d

p2'(1.3)=0 = b+2.6c+5.07d

p2'(2)= 0 = b+4c+12d

Tramo 3:

p3(2) = 4 = a+2b+4c+8d

p3(3.5)= 2 = a+3.5b+12.25c+42.875d

p3'(2)= 0 = b+4c+12d

p3'(3.5)= 0 = b+7c+36.75d

Figure 7: One of the groups proposed a system of equations to obtain the coefficients of several polynomials of degree three that interpolated the values in the table.

key-frames. This was identified by some of the students as a problem since it gives no control on the velocity. This discussion revealed that most of the groups had not considered velocity as something relevant when building the set of new positions and none of them had built an equivalent table or list with velocities.

- A group explained that they had considered using Lagrange's polynomial. However, after consulting the properties of the algorithm on the Internet and discussing it with the professor, they had decided to disregard the method to prevent unwanted oscillations. They decided to use piecewise polynomials instead.

- Two groups proposed as a solution to build ad-hoc, piecewise functions, based on the shape of their graph (see Section 4.2 and Figure 6). While some students found this approach very intuitive, others had difficulties to understand how to make a general cosinus function of the form $f(x) = a + b\cos(cx + d)$ interpolate two points and two velocity values, by defining a system of equations on the four parameters.

- The previous strategy also was identified as less general. Some students argued that, if an ad-hoc function was used for each type of movement, then the solution would have to be completely rewritten if the movement in the video was replaced by a different one.

From this discussion, several conclusions were drawn. Some of them were directly verbalized by the students, while others required some elaboration by the professor.

- The students identified the need to have a general solution that was useful regardless the particular movement to be generated.
- As a result of the previous statement, there was a consensus that a good strategy should directly use the values stored on a table in a general way. To support this statement the professor recalled the fact that, in Blender, this is the way the students would find the data.
- It was also agreed the need to study interpolation techniques as the way to generate the locations between two key-frames.
- The need to control the velocity in some way was stressed by the professor.

At the end of the session, the students were asked to review the original problem in their study time, outside of the classroom, and to prepare a proposal to the original open problem, based on the discussion and the conclusions of the debate.

4.4. Solution of the open problem

The second session was devoted to the solution of the original problem, proposed at the beginning of the activity. The session was started with a one hour lecture on interpolation, in which cubic splines, Hermite interpolation and Catmul-Rom splines were discussed. Then, the students had an additional hour to work on their implementation of a solution to the problem. During this hour the students could also ask questions to the professor, both about the interpolation techniques and about the implementation details. A week after the end of the second session, the students had an additional session of two hours, supervised by the professor, to finish their solution to the proposed problem.

5. Discussion

By means of the activity, the students work on the problem of building a function that interpolates a series of values. The problem is presented as a task in the context of a computer animation studio, which results a motivating scenario to them. By means of the modelling cycle, the students are encouraged to build their own solution and are forced to confront it to the requirements of the task and the context (Borromeo-Ferri, 2017).

In our opinion, one of the most relevant results of the task is the fact that the students identify, by themselves, the need to use some mathematical technique to solve the problem. The necessity to learn interpolation is not presented by the professor, but is proposed by the students as a very convenient solution which overcomes the limitations they have identified on other strategies.

Moreover, the students seem to show higher interest in the exposition of interpolation techniques when they have worked on a problem where they can be applied, compared to a methodology in which they are first presented the theory and then are asked to apply it. In this way, during the exposition they can refer to their own experience to understand the motivation and the properties of the different methods. Without this prior experience, they have to rely on application scenarios described by the professor, for which they lack most of the detail or, directly, they find difficult to understand by themselves.

In previous editions of the course, the same contents were introduced using a more traditional approach. During two lectures, interpolation techniques were introduced using computer animation as a motivation. Then, a exercise solution was proposed to be done during one or more practice sessions. Compared to this organization of the activity, based on a classic lecture followed by the practical exercise, it could seem that the modelling based activity can require more classroom time. However, in previous editions of the course, in which the proposed methodology was not used, the time devoted to interpolation was essentially the same. According to our experience, the students seem to required more time to work their solution if they have not developed their own solution to the preliminary problem. In particular, many students take some time to get used to the data model and programming architecture proposed by the professors. On the contrary, if they base the final solution on the code they have developed by themselves, this part of the exercise takes less time in our proposal.

During the development of the experience we observed two issues that have to be taken into account. On the one hand, the students are given complete liberty to look for their solution and this can lead to some groups focusing on accessory aspects of the solution, while they pay very little attention to the mathematical part of the problem. E.g., at the beginning of the experience some of the groups started discussing only the aspects related to the implementation and integration in Blender of their solution, reducing the discussion about the movement to generalities such as “we have to generate the locations of the object”. On the other hand, we also observed that the students that had not a solid background in programming skills tended to feel frustrated when they reached the moment to implement the solution. Students with this profile seem to feel more comfortable if detailed guidelines are provided to develop the solution. Surprisingly, we did not observe the same situation regarding mathematical background; in the first part of the experience the students could overcome their limitations since they were not forced to use mathematics explicitly; in the second part, when they were requested to develop a more mathematical solution, we have no evidence that the methodology helped them in any special way but we neither observed specific difficulties.

This exploratory experience has served as an initial step towards the application of the modeling cycle in the course and has helped us to identify some weaknesses need to be revised in the future. In particular we will focus on those students that show difficulties with the implementation of the model during the use of the modeling cycle. On the other hand, in order to assess whether this methodology effectively helps to overcome difficulties linked to the use of mathematical tools and fosters the understanding of applied mathematical concepts, the experience should be repeated including a control group and a final test including only mathematical content.

6. Conclusions

In this work we have presented a modelling activity for undergraduate Computer Science students aimed to the study of function interpolation. This activity is currently being used in the Multimedia Engineering Degree at University of Valencia. We have presented the context of the activity and described the task to be carried out by the students.

The learning experience that has been presented is an starting point for the inclusion of the modeling methodology in the context of Computer Science studies at our university. The task that is developed by the students contains many elements of real practice, and requires the construction of a mathematical model to reach the solution. Upon this proposal, and taking into account previous experiences in engineering education (Diefes et al., 2004, Hamilton et al., 2008), our intention is to extend this experience to an even more realistic scenario, by proposing Model Eliciting Activities (Lesh y Doerr, 2003) based on a production of an animation or visual effects sequence which require the use of the contents of the whole course.

In our future research we expect to identify and analyze in more detail the strategies that the students follow to find the way into their solution. We also intent to study whether the modeling methodology provides a different point of view over the solution of technical problems to the students.

Referencias

Alpers, B. A., Demlova, M., Fant, C., Gustafsson, T., Lawson, D.,
Mustoe, L., Olsen-Lehtonen, B., Robinson, C. L., Velichova, D. (2013).
*A framework for mathematics curricula in engineering education:
a report of the mathematics working group.*
Technical report, European Society for Engineering Education (SEFI).

Ärlebäck, J. B. (2009).
*On the use of realistic fermi problems for introducing mathematical
modelling in school.*
The Mathematics Enthusiast, 6(3), 331– 364.

Barry, P., Goldman, R. (1988).
Recursive evaluation algorithm for a class of catmull-rom splines.
In International Conference on Computer Graphics and Interactive Techniques,
volume 22, pages 199–204.

Blender Online Community,
Blender - a 3D modelling and rendering package (2018).
Blender Foundation, Blender Institute, Amsterdam.

Blum, W., Leiß, D. (2007).
How do students and teachers deal with modelling problems?
In Mathematical Modelling: Education, Engineering and Economics.
ICTMA 12, pages 222–231. Elsevier.

Borromeo-Ferri, R. (2017).
Learning How to Teach Mathematical Modeling in School and Teacher Education.
Springer.

Burden, R. L. , Faires, J. D. (2010).
Numerical analysis.
Cengage Learning.

Catmull, E., Rom, R. (1974).
A class of local interpolating splines.
In Computer aided geometric design, pages 317–326. Elsevier.

Diefes-Dux, H. A., Moore, T., Zawojewski, J., Imbrie, P. K., Follman, D. (2004).
A framework for posing open-ended engineering problems: Model-eliciting activities.
In 34th Frontiers in Education.

Frey, K. (1989).
Integrated science education: 20 years on.
International Journal of Science Education, 11(1), 3–17.

Gil, D., Carrascosa, J., Furió, C., Martínez-Torregrosa, J. (1991).
La enseñanza de las ciencias en la educación secundaria.
Institut de Ciències de l'Educació. Universidad de Barcelona.

Hamilton, E., Lesh, R., Lester, F., Brilleslyper, M. (2008)
Model-eliciting activities (MEAs) as a bridge between engineering education research and mathematics education research.
Advances in Engineering Education, 1(2), n2.

Kaiser, G., Brand, S. (2015).
Modelling competencies: Past development and further perspectives.
In Mathematical modelling in education research and practice, pages 129–149. Springer.

Kerlow, I. (2009).
The art of 3D: computer animation and effects.
Wiley Publishing.

Langereis, G., Hu, J., Feijs, L. (2013).
How to introduce mathematical modelling in industrial design education?
In Teaching Mathematical Modelling: Connecting to Research and Practice, pages 551–561. Springer.

Lengyel, E. (2012).
Mathematics for 3D game programming and computer graphics.
Cengage Learning.

Lesh, R., Doerr, H. (2003).
Foundations of a model and modeling perspective on mathematics teaching, learning, and problem solving.
Lawrence Erlbaum Associates.

Núñez Espallargas, J. M., Font Moll, V. (1995).
Aspectos ideológicos en la contextualización de las matemáticas: una aproximación histórica.
Revista de educación, 306, 293–314.

Sánchez Pérez, E. A., Garcia-Raffi, L. M., Sánchez Pérez, J. V. (1999).
Introducción de las técnicas de modelización para el estudio de la física y de las matemáticas en los primeros cursos de las carreras técnicas.
Enseñanza de las ciencias: revista de investigación y experiencias didácticas, 17(1):119–129.