# Fidelity in the Use of Apps for Solving Differential Equations

## *Fidelidad en el uso de app para la resolución de ecuaciones diferenciales*

http://dx.doi.org/10.32870/Ap.v11n1.1463

Alberto Camacho Ríos*
Marisela Ivette Caldera Franco**
Verónica Valenzuela González***

**ABSTRACT**

En este trabajo planteamos los resultados del uso de sistemas algebraicos computacionales conocidos como aplicaciones para programas de cómputo (app), de contenido matemático, que fueron incorporadas a dispositivos móviles de estudiantes de ingeniería en un curso de ecuaciones diferenciales ordinarias. El objetivo fue que los estudiantes adquirieran capacidades en el empleo de estas tecnologías para que resolvieran diferentes ecuaciones de las que se estudian en el curso, así como obtener la gráfica de su solución. Desde el concepto de fidelidad del software, se analizaron y manejaron aplicaciones como Differential Equations, Wolfram, Desmos, Photomath, entre otras, cuyos resultados destacan las de mayor utilidad. Con ello, elaboramos una situación didáctica mediante la cual interactuaron estudiantes con las app en la resolución de ecuaciones y en la graficación. Los hallazgos muestran deficiencias en la evolución de la interfaz de las app utilizadas, cuyo uso provoca fenómenos didácticos importantes.

**Keywords**
TIC, educative technology, educational phenomenon, mathematics, m-learning

***RESUMEN***

*This report shows the results of using Computational Algebraic Applications (CAA) for mobile devices, intended as aid in college courses of Ordinary Differential Equations (ODE). The main purpose was for students to gain skills in such software so they could proficiently solve ODEs as well as to provide graphics from the solutions obtained. From the "fidelity" concept of the software, several applications were analyzed and evaluated such as: Differential Equations, Wolfram, Desmos, Photomath, among others. The given results proved than those applications were the most useful. With those results, a Didactic Situation was created in which students interacted with the app to solve equations and to graph results. The results shows some deficiencies in the interface evolution of the app that was used, this cause significant didactic phenomena.*

***Palabras clave***
*TIC, tecnología educativa, fenómeno didáctico, matemáticas, aplicación móvil*

Received: September 26, 2018
Accepted: February 11, 2018
Online Published: March 30, 2019

* PhD in Educational Mathematics
National Technological System of Mexico, Chihuahua Technological Institute II. ORCID: https://orcid.org/0000-0002-0685-4723; camachoalberto@hotmail.com
** PhD in Education. National Technological System of Mexico, Chihuahua Technological Institute II. ORCID: https://orcid.org/0000-0001-5574-5817; marisela.caldera@itchihuahuaii.edu.mx; mcaldera.tec2@gmail.com
*** PhD in Education. National Technological System of Mexico, Chihuahua Technological Institute II. ORCID: https://orcid.org/0000-0002-4363-4930; vvalenzuelamx@yahoo.mx

## INTRODUCTION

This paper shows the results of the use of computer algebra systems (CAS), through mobile devices in ordinary differential equations (ODEs) courses to help students acquire solution abilities and skills. What interests us in these activities is the "fidelity" of the CAS language regarding the ODEs solutions and graphs that students compute by hand in their notebooks. We also registered didactic phenomena that arose unexpectedly and which highlight CASs.

The ODE program at the National Technological Institute of Mexico (TecNM, [Spanish acronym]) developed in the engineering studies, suggests the use of information and communication technologies (ICTs) to acquire competences that allow the symbolic and graphic solution of general cases of equations. The program is identified under code ACF-0905 (TecNM, 2016) which seeks to consolidate the students' mathematical formation. It is articulated in five topics or main units, which are: first order differential equations; n-order differential equations; Laplace transformation; differential equations systems, and Fourier series.

The program mentioned above has a background of vector calculus and linear algebra. In these five units, the specific didactic competences establish model dynamic processes through differential equations that describe them. This implies that the students possess generic competences such as the capacity of abstraction, analysis and synthesis that, among other activities, help them "solve first and n-order differential equations and interpret the solutions graphically by using ICTs, and model engineering situations by using differential equations" (TecNM, 2016, p. 8).

The ODEs constitute the central axis on which engineering and physics survive, as well as portions of life sciences related to mathematical models. The main problem in this course is to develop models. Activities are divided into theoretical aspects, mathematical techniques selected to solve equations and the practice of algorithms necessary to the solution. Both stages, together with the graphication of the solution and its interpretation, organize the complete ODE traditional course and it intersperses in each one of the topics.

Once the second stage is overtaken by the learning of equation solution methods through exercises, it is possible to step into solution calculus and graphs developed through a mathematical software that provides real answers related to the nature of the modeling problem. These answers are described through a dependent variable, commonly written as $y(x)$, which represents the solution to the equation. The use of software is, in this sense, an important synthesis of the operative and cognitive efforts students developed in their notebooks to determine the solution and its graph.

This program suggests a software containing CASs such as Mathematica, Maple, Derive, Mathcad and Matlab (TecNM, 2016, p. 10). In teaching this course, the software plays different roles. The most important aspect is establishing technological environments to learn this type of equations (Cortés Zabala, Guerrero Magaña, Morales Ontiveros and Pedroza Ceras, 2014). Another aspect is to use the academic mathematical software with a license to solve problems subject to the modeling developed in problems associated to mathematical physics.

Artificial intelligence has also evolved and streamlined the commercial mathematical software interface and language to be incorporated in mobile applications that are useful in a classroom, a process known as *mobile-learning*. The interface is a space of communication between the user and the electronic content that processes the requested representations. For teaching mathematics, this software allows visualizing knowledge representations.

We have conducted experiments with this latest type of software which, hereafter, will be referred to as Android (app.m)[1] mathematical applications, and we have obtained encouraging results that lead us to reconsider including the proposed ICTs in the program. Its visualization and handling development involves abstract mathematic entities such as: algebra, vectors, geometry objects, differential calculus, differential equations, among others, that open new original teaching perspectives. Besides, all the students at the engineering level possess a mobile devise (smartphone or tablet) with homogeneous usability specifications that make possible to use it in the classroom. These are great connectivity tools that facilitate downloading and executing app.m via Internet.

These architectures mark an important break with the use of commercial software. This type of tools possess many aspects related to their operability which turn out to be an obstacle for using them in the classroom: their use requires a computer desktop while their license is extremely costly for both students and academic institutions. Moreover, in general, the didactic schedules to address the course topics do not suffice to incorporate these tools, thus leave a gap in the students' acquisition of the skills to use them. It is also common that most of the time the teachers of ODE courses are unaware of the existence of the software suggested in the program and its usefulness.

This paper proposes experimenting in the course with type app.m CAS course that includes packages and libraries such as those contained in the commercial software suggested in the program. To do so, we designed and applied a didactic situation (DS). The fundamental characteristic of these app.m is their facility in downloading Android mobile devices, and even other different operative system devices.

Hence, our concern was that the app.m interface would have a certain degree of fidelity regarding the mathematical language the students use when solving ODEs and, mainly, the graph they submit. Secondly, we focused on identifying and analyzing the didactic phenomena that arose during the experimentation, which plays an important role during the immersion of the app.m in the classroom by supporting the representation and mathematical practices of the students.

The above is justified since the app.m allow solving the most common types of differential equations seen in the course; these help solving the problems modeled in the latter; they offer good graphic accuracy of the solution; in using them, they arouse computer tool management abilities and skills, as well as conditions to construct "a thought process involved in formulating problems and their solutions through computer agents", that Wing (2006) has coined *computational thinking*.

## BIBLIOGRAPHIC REVIEW

The trend in using CAS is not new and it seems recurrent in the specialized bibliography recommended for the ODE course. Since the 1960s, in works such as those of Rainville (1969) and Rainville, Bedient and Bedient (1998), numerical methods to solve equations are suggested in the first work, while the second includes the use of CAS at the end of each chapter.

The textbooks most recommended for the course, as is the case for Zill and Cullen (2018), who suggest resorting to commercial software numerical commands such as *DSolve* (in *Mathematica*), which, given its capacity, gives symbolic solutions of differential eq*uatio*ns; e.g., the homogeneous equation and its solution:

DSolve [y´´[x]+2y´[x]+2y[x] = 0, y[x], x]

Kreyszig (2011) used *Mathematica* to develop solution graphs of differential equations of the examples contained in his book, including a Guide for using *Maple* and *Mathematica* at the end of the book. On the other hand, Edwards and Penney (2001) propose to their readers the so-called "calculus projects" whose development is feasible in *Maple*, *Mathematica* and *Matlab*. One of those examples is shown on page 28 of the book and reads as follows:

> Research A. Trace (in *Maple*) a field of direction and solution curves typical of the differential equation $\frac{dy}{dx}$ = sen ($x$-$y$), with a window of -10 <$x$ <10, -10 <$y$ <10, several solution curves that are straight lines must be visible.

These examples of school textbooks show the evolution of the ODE solution programming instructions in commercial software such a *Maple* and *Mathematica*. The software evolution finds a parallel in different applications with the use of symbolic expressions such as *DSolve* as well as *Plot* and *Plot3D* for solution graphing.

The advantage of these instructions is that they produce symbolic expressions for the solution that resembles the solutions the students obtain algorithmically in their notebooks. Moreover, this evolution is characterized by the same computer language and interface that app.m have inherited. It is also characterized by the tendency of trying to create, from texts, abilities in the students to use specialized software in the study and solution of ODEs through the teaching instructions given; the example of Edwards and Penney (2001) illustrates well this tendency.

We reviewed some of the latest research related to the use of mathematical content software to be applied in the ODE higher educational level. Rackauckas' research (2018) was one of them; he made a comparison between different languages commonly used in solving ODEs based on programming languages. Among the software reviewed are those of *Matlab*, *R*, *Julia*, *Phyton*, *C*, *Mathematica*, *Maple* and *Fortran*. Rackauckas made comparisons between the weaknesses and strengths of each language; the results of his research allowed verifying that *Julia* "objectively had a set of greater characteristics, exceeding most of the others describing them as common solvers" (Rackauckas, 2018, p. 10). Julia is a high level programming language that also has solvers for non-ordinary differential equations.

On the other hand, Rodríguez and Quiroz (2016) show the role of technology in the ODE engineering course in its transit through the different stages of mathematical modeling. They report the design of a specific situation in the context of RC circuits which use different technological resources in developing activities. The resources highlighted are the use of a TI Nspire CX CAS calculator, IT voltage sensors, IT browser, capacitor, resistance, batteries and connectors.

Meanwhile, Mosquera and Vivas (2017) selected more current app.m resources to conduct a comparative analysis of their functions to develop learning mathematical competences for the course in differential calculus. As a result of the assessment, they obtained three app.m that met different standards imposed; these were *MalMath*, *Symbolab* and *Grapher*. The standards assigned that stand out are: the software portability, the requirements of an Android operative system for mobiles, the advantage of being a free software, the facility of installation and operation and the graphic interface that does not require the development of a programming code.

The quality criterion for the use of the best app.m in the classroom is that most of them have been designed and constructed with quality standards similar to those imposed by researchers such as Mosquera and Vivas (2017). Currently, some of these criteria have been overstepped by the growing evolution of cellular technology; e.g., it is widespread that students have a mobile with an Android operative system. Most of the free app.m are easy to upload and, given their own limitations, the graphic interface does not allow developing the programming code. Hence, we believe that the researchers above mentioned, and others not mentioned in this paper, have omitted “fidelity”, one of the app.m most important elements that provide solutions to differential equations and their graphs.

## TRANSPOSITION BY COMPUTER

Balacheff (1994) coined this term when expanding the phenomenon of *didactic transposition* defined by Chevallard (1985), which should be understood as the reproduction of a situation involving knowledge of academic mathematics in contexts different from those that have been produced and also with different forms of expression. Meanwhile the transposition by computer refers to the introduction of software in the teaching of mathematics whose consequences complicate the simple didactic transposition of knowledge since its inclusion in math courses determines restrictions and obstacles regarding the representation and the computer internal processing as well as the representation and interface processing.

### *The emergence of knowledge external to the classroom*

The approaches of computer sciences toward the teaching of mathematics have entailed the emergence of knowledge and concepts as well as tools and techniques external to the academic knowledge; hence the importance of studying this irruption.

The object of this analysis is to assume the control of the association that occurs between knowledge and the techniques involved, in addition to the didactic phenomena triggered by this emergence. As for the incorporation of computer tools in the classroom, Artigue (2015) concludes that the use of this type of technologies is not without conflicts.

Regarding the immersion of knowledge external to the teaching of mathematics, Camacho and Romo-Vázquez (2015) deconstructed the gradient mathematical concept into topography, a non-mathematical concept. As a result of this deconstruction process, numerous techniques allowed them to establish a mixed definition of the concept to be used in classroom teaching.

On the other hand, Artigue (1997) experimented with the immersion and the use of *DERIVE* in operations with arithmetic fractions with high school students (Liceo francés, [French High School]) and analyzed both didactic phenomena that arose with the use of parentheses. *DERIVE*’s interface eliminates from the screen a number of useless parentheses which differ from those included in the calculus of the students’ notebook. These software’s actions caused semiotic confusions in the students since the display of the results on the screen does not correspond to those determined in their notebook.

### *Didactic phenomena*

While Artigue's article (1997) does not clarify the term *didactic phenomena*, we understand them as those teaching problems caused by the transposition by computer of external agents when interacting with the academic mathematical knowledge. They arise as disturbances that alter the order of the didactic activity and they differ from the semiotic, cognitive and epistemological disturbances (the case experimented by Artigue is a semiotic type disturbance). In all three cases, these disturbances lead to errors in the algorithmic decision making or in determining the solution to the problems of academic mathematics.

### *Fidelity*

According to Balacheff's estimate (1994), these situations must be analyzed in terms of the degree of fidelity regarding the phenomena they are confronted with; e.g., in computer environments simulating physics-mathematics phenomena, when we question ourselves about the closeness of the simulated environment to the real world; this closeness is recognized as fidelity. A high fidelity simulation is almost indistinguishable from reality. Wenger (1987, p. 313, quoted by Balacheff, 1994, p. 14) introduced the concept of *epistemic fidelity* with which it is possible to qualify the difference between the physical representation and the knowledge of reference established at an epistemic level.

In this sense, the research, through the processes occurring during the transposition by computer, should measure the fidelity that "designates the work on knowledge that allows a symbolic representation and the implementation of said representation through a computer device, either to show the knowledge or manipulate it" [1] (Balacheff, 1994, p. 11). In the context of this approach, the transposition takes on a particular importance; it means a knowledge contextualization that can have important consequences in learning outcomes.

The concept of fidelity is a rule that aims at minimizing the epistemic distortions and disturbances caused by the association of mathematical knowledge and the software. Hence, we ask ourselves the following questions: what is the relation between the app.m interface and its compared use in a didactic situation?What consequences does this relation have on learning mathematics? What will be the result of this interaction with this software?

In computer and classroom domains, fidelity should be viewed through the distance that separates the academic environment from the software, and takes two routes that complement one another. The first has to do with the work of the software on academic knowledge, whose manipulation refers to an association of contexts installed in an epistemological environment. The second refers to the software interface and the symbolic parenthesis of the elements that integrate it with those who communicate in the classroom.

### High fidelity software. *Mathematica*

A high fidelity (HF) mathematical software can be distinguished by its user interface's functional and semiotic characteristics that make up the "validity domain of its representations" (Balacheff, 1994). If we limit ourselves to the part of this domain that interests us, on the one hand, we have the input and output of the numerical and symbolic information and, on the other hand, the resolution of the graphs it provides. A HF software is *Mathematica*. Its most advanced versions use a numerical and precise resolution to solve initial value ODEs. It offers a high level interface for all standard databases. *Mathematica* is identified because the symbols that appear on its information keyboard are almost identical to those the students use in their notebook, while the resolution of the graphs provided is of high quality and completely interactive. This is the case of the image on the left of Figure 1 (the interface and its representation "look alike").

In the graph above mentioned, it was necessary to include the sign "×" [by] for the multiplication operations to work in the expression requested to be charted. This does not occur in the same software most recent versions, e.g., the 11.3 version downplays the HF since this symbol is not necessary in the students' notebooks. We can estimate that the *Mathematica* 10.4 software is HF at 99% given the need to incorporate the × symbol in its domain. By reviewing the graph provided, we note a good resolution; it is even possible to use another type of coloring if desired; however, the arrows, the *x*and the *y* do not appear, although it is possible to include them. Every software interface defect or error can demerit one point from the percentage that determines its fidelity.

### Good fidelity mathematic software

As for the app.m, the discrimination criterion is applied equally when taking the HF software as benchmark. We should check if the latter is functional for the desktop and if the way the app.m are used in the classroom is done directly on the students' mobile. This difference represents a strong discriminatory criterion that helps characterize app.m as good fidelity (the interface and its representation "look almost alike") compared to those HF softwares, mainly because of factors regarding the resolution of the graphs provided.

In the image on the right in Figure 1, we have put the same graph of the function developed in *Mathematica*. The graph on the right is denser or more "granulated" than the left one, i.e., its pixels are not distributed homogeneously on the image as in the first one, or, the pixel distribution model is different from the HF software. Regarding its resolution, we can say that the image on the left has an approximate resolution of 100 × 100 pixels, while the one on the right, is of 50 × 50 pixels.

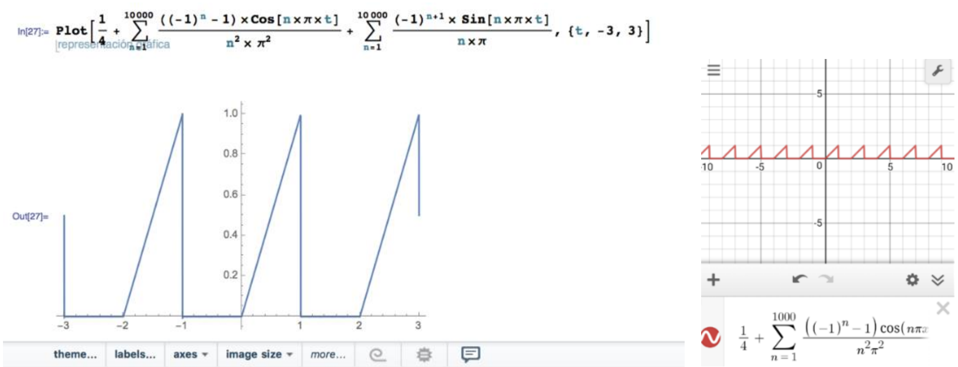


**Figure 1.** On the left, a graph developed through *Mathematica* 10.4, HF software. On the right, the same graph developed in *Desmos*, a good quality software.

Even when the same graph has been developed with different software, it only shows the granular difference between both, which can be caused by the type of function; in other situations, this difference is more accentuated in app.m in the *Desmos* domain. The latter confuses the user's perception when a "cognitive disturbance phenomenon" is presented at the moment students are interpreting the graph.

## METHODOLOGY

The research design responds to an internal methodology based on the analysis of didactic situations (DS) in which app.m computer tools are involved. In this case, we designed and applied a DS –more exactly an exam– and we accompanied this intervention with a research process aiming at verifying the results of the participants.

To develop the acquisition and verification of the abilities the students acquire with the use of app.m, we formulated activities developed through an inductive process that concludes with the application of the DS and the analysis of the results. The activities were:

- Resolution and graphication of the types of ODEs suggested in a traditional program, i.e., without the use of the software.
- Review of the different app.m considered portable in the Android operative system of the students' mobiles; easy to load, free software; the graphic interface does not allow developing a programming code and, most importantly, that the computer language, graphic and equation editor displayed on the screen be as faithful to the natural academic mathematical language (NAML) that results from solving ODEs in the students' notebook.

- We will not discuss the description of the students' mobiles and the app.m downloading since we have taken into account that most mobiles possess an Android operative system and that downloading the app.m to the device did not represent any problem.

- Experimental use of app.m in solving ODEs and their graphs, which leads to acquiring abilities and skills.

- Choosing problems in the course to design a DS whose resolution verifies the abilities acquired.

- Group application of the DS.

- Analyzing the results of the app.m use and the didactic phenomena that arise.

Given the characteristics of the research, we are not interested in obtaining quantitative results since we do not deliberately handle variables because our intention is only to describe and interpret the results all at once, so as to reproduce them in subsequent courses. We are more interested in assessing the cognitive decisions that can be verified in the students' results when addressing the problems we proposed.

### *Mobile devices and app.m*

Throughout the 2018 first semester, we reviewed the app.m software that the students of the ODE course had downloaded in their mobiles, with the purpose of use already mentioned; among others and by their common name, *Differential Equations*, *Geogebra, Calculadora de Integrales* [Integral Calculator], *WolframAlpha* and graphers such as *Desmos* and *Photomath*. Most of these are license free app.m and are easily downloaded to most of the students' Android mobile devices. In general, they satisfy the attributes previously mentioned. Each application was downloaded as necessary for every topic of the course, under the instruction that the students would display a NAML on the screen as soon as possible.

The first of these was the *Photomath* grapher that was used to produce the graphs for solving first and n-order ODEs. *WolframAlpha* followed and was also used in some cases to graph families of solutions of the different types of ODEs. As the course progressed, we noticed that some of the graphers such as *Mathematics*, *Maths Differential Equation*, *Mathway*, *Malmath*, did not meet the fundamental characteristic of NAML fidelity regarding the language imposed in the interface domain, hence they were gradually discarded.

Arriving at the stage of the Laplace transformation topics (third and fourth course), we noticed that there were hardly no evolved app.m to solve the

ODEs involved, and that there was only a desktop software. Hence, the course continued with these topics in the traditional order, i.e., solving initial condition ODEs using Laplace transformation whose solution was interpreted in some app.m grapher. As for the fifth topic of the course, Fournier series, our interest was set on graphers that would provide a graph solving differential equations representing mass-spring physical systems whose order function was periodical at a given interval. In this case, and given the conditions imposed, *Desmos* was the only application to produce graphs in the required order.

### *Description of the app.m selected*

*Photmath* and *Desmos* were the two app.m graphers we saved from the course experience, and *WolframAlpha* and *Differential Equations* [2] were the ODE solvers. Since this paper aims at showing a didactic experience in which *Desmos* graphing is involved, we will describe its main attributes without taking into account the other app.m mentioned.

*Desmos* is an app.m tool developed in the city of San Francisco in the United States. One of its characteristics, it that it can easily be accessed online at: https://www.desmos.com/, from a mobile device (tablet or smartphone). It does require a user, it is multi-language and it is collaborative. Its interface domain possesses an equation editor similar to that of *Mathematica* in which expressions are typed in the same order as those seen in the classroom. The graphs displayed concentrate in a grid that starts from two central axes *x, y* (See Figure 2), which can also be distinguished using different colors, if desired. The control of the equation editor is located below the grid, with a pallet in which the different functions and symbols most often used are concentrated.

### *Didactic Situation*

A DS refers to the development of articulated and task-oriented activities so the students involved can develop specific competences. The activities are ordered in didactic sequences to solve cognitive conflicts that arise in the activities. This articulation allows carrying out a careful control of the computer resources and implicit knowledge. DSs may include tasks, class exercises, exams, software use, modeling, projects, as well as small practices that ensure a response accepted by those involved.

The intention of putting a student or group of students into a situation means a) to experiment with them new creations that are not yet explicitly accepted in the classroom, or, b) to seek deciphering something hidden in the development of the DS, through mixing c) different knowledge and knowledge areas to achieve a solution. In our case, we seek that the students interact with the app.m and recognize in the interface domain the objects of academic mathematics that, to some extent, we will assume as the standard that guarantees fidelity between both environments.

### *DS design and problematization of the item chosen*

Our objective in designing a DS is to provide the following series of S sequences:

S.1) Provide a group of 20 students of the ODE course of the fourth and fifth semester of the Computer Systems Engineering studies [3] with a mathematical object that represents a Fourier series solution to a differential equation

S.2) Our interest is that the students type this solution in the app.m downloaded in their mobile, in this case, the *Desmos* grapher,

S.3) Then convert the mathematical object into a computer object placed in its interface

S.4) The next sequence consists of obtaining the graph of the expression thus typed in app.m

S.5) Finally, students may identify, within the graph, the mathematical object to which the given Fourier series solution corresponds.

### Problematization and item selection

The students presented an ordinary exam in the fifth unit of the course at the end of the first semester of 2018; it lasted one hour and twenty minutes approximately and the item chosen was the second of the three that had been proposed. The exam was developed based on the inclusion of one of the items in the content of a textbook commonly used by professors and students at the institution where we conducted our research. The item was chosen among others that are found in Edwards and Penney (2001, problem 16, section 9.2, page 595) and which is described as shown below.

The function shown in (1) expressed in Fourier series corresponds to a development of the form:

The item was problematized using the software and knowledge involved in the opposite sense of the form proposed in the textbook above mentioned. Its approach is as follows:

(a) Let's consider that $f$ is a function of period 2 in such a way that $f(t) = 0$, if -1 < $t$ < 0 and $f(t) = t$ if 0 < $t$ < 1. Proves that:

$$f(t) = \frac{1}{4} - \frac{2}{\pi^2}\sum_{n=1}^{\infty}\frac{\cos(n\pi t)}{n^2} + \frac{1}{\pi}\sum_{n=1}^{\infty}\frac{(-1)^{n+1}\text{sen}(n\pi t)}{n} \ldots(1),$$

with period $f(t) = f(t+2)$.

(b) Now trace $f$ graph indicating the value of every discontinuity.

Moreover, the professor of the course uses equation (1) to ask the students to perform the following activities:

1) To type it in *Desmos*, 2) request the graph of the same in the interface and 3) determine the $f(t)$ function corresponding to the given Fourier series.

In the classroom, throughout the work on the techniques related to the topic, the students graphed in *Desmos* a good number of functions expressed in Fourier series and ODE solutions of initial conditions, whose ordering function is periodical as shown in (1). Unlike these activities, the item presented in the DS is in opposite direction, i.e., it exits the development of the Fourier series functions and is asked to "return" and recognize the mathematical function of origin; in this interaction, the app.m is used as a bridge. This last activity is not common to a traditional course. From our standpoint, the item in itself represents a challenge for the students; however, with a minimum of use of the app.m, a solution can be achieved. This leads the students to consider this item as the easiest to solve, hence most of them opted to address it.

### Cognitive Conflict

A cognitive conflict arises when students are asked to verify the mathematical object displayed in the *Desmos* graph of the Fourier series. In this contrast, the students must recognize that the object represents two straight lines joined at the common starting point, "sawtooth" shape, according to the authors of the book.

$$f(t) = \begin{cases} 0, -1 < t < 0 \\ t, 0 < t < 1 \end{cases} \ldots(2)$$

of period $f(t) = f(t+2)$.

However, by assuming the authorship of the text, the professor did not realize that the expression (1) did not correspond to the expected solution (2), since it is an error of the authors of the book. The real expression for the expected solution is:

$$f(t) = \frac{1}{4} + \frac{1}{\pi^2}\sum_{n=1}^{\infty}\frac{((-1)^n-1)\cos(n\pi t)}{n^2} + \frac{1}{\pi}\sum_{n=1}^{\infty}\frac{(-1)^{n+1}\operatorname{sen}(n\pi t)}{n} \ldots(3)$$

Under the same conditions of the period. The graph expected for the equation (1) is shown in the app.m interface in Figure 2. As well as the expression typed in the software equation editor, while the graph determining the function (3) is shown in Figure 1.

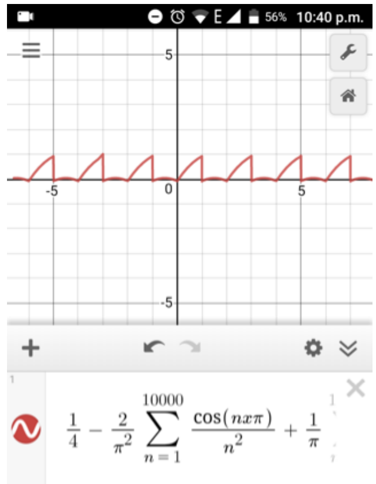


**Figure 2.** Below, the expression typed in *Desmos* for $n$ = 10,000 values. Above, graph of the displayed expression.

By not realizing the mistake, the professor approached the item as suggested in function (1), and assumed that the expected answer was that shown in (2). A student that has achieved managing *Desmos* adequately throughout the semester will take two to three minutes approximately to type and display a graph. It takes him/her another five minutes to transcribe the corresponding graph in his/her notebook. This interaction leads him/her to make a cognitive decision for the mathematical expression displayed.

## RESULTS

Nine of the 20 students that applied for the exam achieved "satisfactory" results in the solution the professor expected for the problem situated in DS. Seven of them solved the problem erroneously, and four did not choose it as part of their evaluation. The eleven students that failed solving the problem enrolled in regularization session of the fifth unit for which three new problems were developed including one similar to the one in DS.

Next we will show the most outstanding regularities of the results obtained with the use of app.m during the development of the DS, with the error mentioned.

The graph displayed by *Desmos* for the problem approached in DS (See Figure 3) shows one of the student's result that corresponds to the result the professor expected, i.e., $f(t) = \begin{cases} 0, -1 < t < 0 \\ t, 0 < t < 1 \end{cases}$, whose regularity has been verified in five of the nine students.

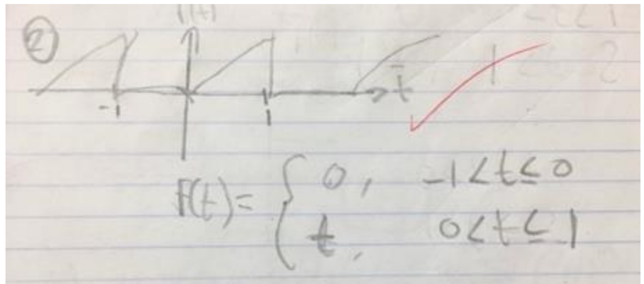

**Figure 3.** Development in the notebook of the graph displayed by Desmos.

In this execution, the student gave a maximum value to the sum of $n$=100 terms, which shows that the “straights” expected are seen as “curved”, which is perceived in the graph transcribed into the notebook. However, this was not an obstacle for the five students to choose the expected expression. Three of them even transcribed the graph in their notebook avoiding the curve and left the “straights” as the professor expected, unaware of the error they were making.

In Figure 4 graph, another student exaggerated the curves even more and the sum casted a value for $n$=10,000 terms, which convinced him/her that the curves being displayed were a graph corresponding to Fourier development situated in DS.

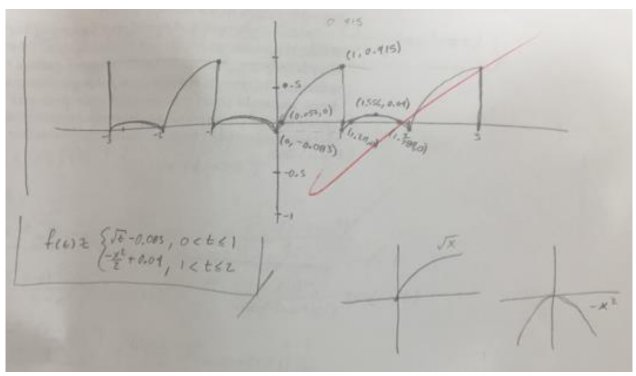

**Figure 4.** Development in the notebook of the graph displayed by Desmos.
In this case, the student interpreted the expected graph with two curves.

In this scenario, the student amplified the resulting image in the app.m interface and noted that the curve accommodated on the $x$ axis cuts it in two points whose coordinates were obtained with the software. He also estimated those in which both folds join, as well as those corresponding to the maximum value of the first. At that time, the student’s interest was to determine the analytical expressions of every curve and to do so, he used the rectangular coordinates above described as well as the vision he had of the tendential behavior of each. He/she supposed that the first, located between $1 < t < 2$, was an inverted parabola and the other, located between

0 < $t$ < 1, was a radical function $\sqrt{t}$ transferred to the $y$ axis; hence, expressing the expected function as:

$$f(t) = \begin{cases} \sqrt{t} - 0.083, 0 < t < 1 \\ \dfrac{-t^2 + 0.04}{2}, 1 < t < 2 \end{cases}$$

Another student used the same criterion and concluded that the expected function was a "straight" located between 0 < $t$ < 1, and an inverted parabola transferred to the $x$ axis, located at the interval 1 < $t$ < 2. He/she wrote the expression as:

$$f(t) = \begin{cases} t - 0.083, \quad 0 < t < 1 \\ -t^2 + 2t - 0.917, 1 < t < 2 \end{cases}$$

The other students carried out similar operations.

DISCUSSION

The granular effect of the graphs displayed led the first five students to make the wrong decision which was manifested as a cognitive disturbance phenomenon given the mental representation they made of the expected function. Figure 2 shows that the ripples that determine both curves blend into the environment of the app.m interface. The one located on the $t$ axis, is folding on the latter; and the other, is similar to the "straight" *f (t)= (t),* which causes the phenomenon.

In light of those results, the graph shown in Figure 3 represents a "model of students' behavior" that derives from the graph displayed in the app.m interface and the representation made by the students through their diagnosis level. These decisions seem important in light of a possible design of a "modeling" situation involving the students. This is currently known as "tutorials".

However, the application of the DS shows that the app.m can be used by the students as support in technological academic environments in solving problems without the need to resort to a specialized or commercial software. In these software, the students' actions and responses are dynamic, and determine that app.m seem to be essential in ODEs solution and the graphing of their solutions.

## CONCLUSIONS

The immersion of the app.m software in the ODE course provides significant epistemological differences between the symbolic representation of the graphs provided by their interface and the knowledge reference that allow the students to develop it. The symbolic representation is determined by the limited resolution of the graphs displayed, which causes a series of didactic phenomena during their interaction with the academic mathematical knowledge.

The phenomena have two facets: the first highlights the progress of the software technology showing that mobile devices in which the app.m are downloaded promote the latter given the reduced size of their interface, or, for technological reasons of their programming structure. Therefore, the identification of this type of phenomena shows the limitations of the app.m used.

The second facet is more complicated and it determines the students' erroneous decisions regarding the expected functions in light of the problem. Nevertheless, those decisions can be corrected through previously agreed conventions that would lead to question the DS design.

[1] Cuando Balacheff habla de software o dispositivos informáticos, se refiere a las versiones de aquellos que se encontraban en uso a mediados de los años noventa del siglo pasado, como Derive y Cabri Géomèter. En su artículo no concibe aún el uso de dispositivos móviles en la enseñanza, y se preocupa, principalmente, por el software citado, así como por grandes proyectos de ambientes informáticos, tutoriales como Geometry-Tutor y micromundos; es el caso de Logo.

[2] Ver la opinión sobre Differential Equations de Rackauckas (2018) en las conclusiones.

[3] Instituto Tecnológico de Chihuahua II, TecNM. Al ser alumnos de la carrera de Sistemas Computacionales, los estudiantes se encuentran en el ambiente requerido del uso de app.m. Reconocen fácilmente términos usuales de esa disciplina, como *código de programación*, *interfaz*, entre otros.

## REFERENCES


Artigue, Michèle. (1997). Le logiciel 'Derive' comme révélateur de phénomènes didactiques lies à l'utilisation d'environnements informatiques pour l'apprentissage. *Educational Studies in Mathematics*, (33), pp. 133-169. https://doi.org/10.1023/A:1002996128978

Artigue, Michèle. (2015). Tecnologías de la información y de la comunicación y aprendizaje basado en la investigación: ¿qué sinergias? En Consejería de Educación de la Junta de Castilla y León (ed.). *Las nuevas metodologías en la enseñanza y el aprendizaje de las matemáticas* (pp. 17-27). España.

Balacheff, Nicolas. (1994). Didactique et intelligence artificielle. *Recherches en Didactique des Mathématiques*, (14), pp. 9-42. Recuperado de: https://telearn.archives-ouvertes.fr/hal-00190648/

Camacho, Alberto & Romo-Vázquez, Avenilde. (2015). Déconstruction-construction d'un concept mathématique. En L. Theis (ed.). *Pluralités culturelles et universalité des mathématiques: enjeux et perspectives pour leur enseignement et leur apprentissage* –Actes du colloque EMF2015– GT5 (pp. 443-453).

Chevallard, Yves. (1985). *La transposition didactique*. Grenoble, Francia: Editions La Pensée Sauvage. Recuperado de: https://www.persee.fr/docAsPDF/rfp_0556 7807_1986_num_76_1_2401_t1_0089_0000_1.pdf

Cortés Zabala, José Carlos; Guerrero Magaña, Lourdes; Morales Ontiveros, Christian y Pedroza Ceras, Lourdes. (2014). Tecnologías de la información y la comunicación (TIC): aplicaciones tecnológicas para el aprendizaje de las matemáticas. *Unión. Revista Iberoamericana de Educación Matemática*, (39), pp. 141-161. https://dialnet.unirioja.es/servlet/articulo?codigo=4870037

Edwards, Henry y Penney, David. (2001). *Ecuaciones diferenciales* (4ª. ed.). Prentice-Hall.

Kreyszig, Erwin. (2011). *Advanced engineering mathematics*. EUA: John Wiley and Sons Inc.

Mosquera, Mauricio y Vivas, Sandra. (2017). Análisis comparativo de software matemático para la formación de competencias de aprendizaje en cálculo diferencial. *Revista Plumilla Educativa*, pp. 98-113. Recuperado de: http://revistasum.umanizales.edu.co/ojs/index.php/plumillaeducativa/article/view/2476/2802

Rackauckas, Christopher. (2018). A Comparison Between Differential Equation Solver Suites Matlab, R, Julia, Phyton, C, Mathematica, Maple, and Fortran. *The Winnower*. Recuperado de: https://thewinnower.com/papers/9318-a-comparison-between-differential-equation-solver-suites-in-matlab-r-julia-python-c-mathematica-maple-and-fortran

Rainville, Earl. (1969). *Ecuaciones diferenciales elementales*. Ciudad de México: Trillas.

Rainville, Earl; Bedient Philip & Bedient, Richard. (1998). *Ecuaciones diferenciales* (8ª. ed.). Prentice Hall.

Rodríguez, Ruth y Quiroz, Samantha. (2016). El papel de la tecnología en el proceso de modelación matemática para la enseñanza de las ecuaciones diferenciales. *Revista Latinoamericana de Investigación en Matemática Educativa*, *19*(1), pp. 99-124. https://dx.doi.org/10.12802/relime.13.1914

TecNM. (2016). Ecuaciones diferenciales ordinarias. Plan de estudios de la carrera de Ingeniería en Sistemas Computacionales. México.

Wing, Jeannette. (2006). Computational thinking. View point. *Communication of ACM*, *49*(3), p. 35. https://doi.org/10.1145/1118178.1118215

Zill, David y Cullen, Michael. (2018). *Matemáticas avanzadas para ingeniería. Ecuaciones diferenciales* (vol. 1). McGraw-Hill