

2-7-2010

Using a Spreadsheet to Study the Oscillatory Movement of a Mass-Spring System

Margarida Cristina Oliveira

School Piscinas Lisbon, guidacoliveira@gmail.com

Suzana Nápoles

Mathematics Department of Science Faculty of Lisbon University, napoles@ptmat.fc.ul.pt

Follow this and additional works at: <http://epublications.bond.edu.au/ejsie>

Recommended Citation

Oliveira, Margarida Cristina and Nápoles, Suzana (2010) "Using a Spreadsheet to Study the Oscillatory Movement of a Mass-Spring System," *Spreadsheets in Education (eJSiE)*: Vol. 3: Iss. 3, Article 2.

Available at: <http://epublications.bond.edu.au/ejsie/vol3/iss3/2>

This Regular Article is brought to you by the Faculty of Business at [ePublications@bond](mailto:epublications@bond). It has been accepted for inclusion in Spreadsheets in Education (eJSiE) by an authorized administrator of [ePublications@bond](mailto:epublications@bond). For more information, please contact [Bond University's Repository Coordinator](#).

Using a Spreadsheet to Study the Oscillatory Movement of a Mass-Spring System

Abstract

In this paper we develop the study of the oscillatory movement of a mass-spring system to illustrate the use of differential equations in the development of mathematical models for the simulation of physical situations and the interest on developing computational modules using a spreadsheet to be used as structured tools to support the modelling and the analysis of the results.

We first present the physics and mathematics foundations that lead to the differential equation that describes the oscillatory movement and its general solution. We study several particular solutions based on different initial conditions. Then we use a rightly structured spreadsheet to interpret the different solutions we obtained, using sensibility response studies to the several parameters supported by adequate graphic representations. Our goal is to show how the development of computational interactive modules, with good graphic capacities and animations, can improve the study, in an integrated way, of mathematical concepts in different areas, developing the capacity to establish connections between several mathematical fields.

We present some essential procedures to construct, using Excel, interactive applications to support the modelling problem we are dealing with.

Keywords

Spreadsheet, mass-spring system, mathematical models, differential equations, oscillatory movement, stiffness, damping

Using a spreadsheet to study the oscillatory movement of a mass-spring system

Margarida Cristina Oliveira

School Piscinas, Lisbon
guidacoliveira@gmail.com

Suzana Nápoles

Mathematics Department, Faculty of Science, Lisbon University
napoles@ptmat.fc.ul.pt

Abstract

In this paper we develop the study of the oscillatory movement of a mass-spring system to illustrate the use of differential equations in the development of mathematical models for the simulation of physical situations and the interest on developing computational modules using a spreadsheet to be used as structured tools to support the modeling and the analysis of the results. We first present the physics and mathematics foundations that lead to the differential equation that describes the oscillatory movement and its general solution. We study several particular solutions based on different initial conditions. Then we use a rightly structured spreadsheet to interpret the different solutions we obtained, using sensibility response studies to the several parameters supported by adequate graphic representations. Our goal is to show how the development of computational interactive modules, with good graphic capacities and animations, can improve the study, in an integrated way, of mathematical concepts in different areas, developing the capacity to establish connections between several mathematical fields. We present some essential procedures to construct, using Excel, interactive applications to support the modeling problem we are dealing with.

Key words: mass-spring system, differential equations, spreadsheet

1. Introduction

Mathematical modelling is a subject of particular importance in the process of teaching/learning mathematics: it provides the students a valuable experience that shows how the mathematical concepts are useful to help us in the search for a better understanding of the physical world. We all know that mathematics is deeply related to several areas of knowledge - so, in this sense, mathematical modelling can be intended as a "unifying force" [5].

While choosing and preparing the examples to illustrate how useful mathematical models can be, it should be taken into account that the students should understand, from the beginning, the real behaviour of the phenomena under analysis. If they don't, all further developments will seem inaccessible and abstract. After the

observation and formulation of the main hypothesis the mathematical language (equations) is used in order to represent the idealized model. At this analytical stage the best way to overcome the main difficulties of mathematical abstraction is to extend the field of expertise and build models using numerical methods and computational tools [3].

Problems dealing with mathematical modelling often appeared in textbooks. However, once they are presented statically they fail to be fully understood by students. For instance, it is not possible, just by consulting a textbook to change the parameters of an algebraic expression and observe the effects on graphics rapidly and efficiently. This mathematical experimentation task is only possible if we use the computational tools we have at our disposal [8].

The software which is developed to complement the teaching of mathematics is, in general, conceived by teams with informatics experts who use sophisticated programming languages and achieve to create very appealing programs but which are true *black boxes* to the users.

The mathematical experience obtained from the use of that *black boxes* may seem to the students and teachers as something that they are able to understand but that they can not develop on their own [9]. The best way to get around this problem is to choose an adequate computational environment for the development of their own applications in order to acquire their own "mathematical experience" what, for instance, leads us to the spreadsheets group.

Spreadsheets like Excel, for example, due to its structure and tools, allow teachers and students to easily build their own computational applications based on an approach which main focus is the search for the solution of mathematical problems/modelling through the development of the applications [1].

Nowadays, Excel is one of the most used spreadsheets. It is so popular that many of the existing open source spreadsheets copy its main features.

In these spreadsheets we can find tools which allow the development of automated procedures (for example, to obtain animation effects) by using very simple programming commands.

The few automatic procedures of Excel for doing graphics of functions and geometric constructions, which is considered by many as a drawback, is an advantage to mathematics teachers because it requires that the students use simultaneously Algebra, Calculus and Geometry to achieve the desired graphical representations [2]

The presented computational applications were developed using Excel. However, it should be noticed that there are free software like Open Office.org that is a package equivalent to that of Microsoft Office which contains a spreadsheet with similar features to those of Excel, what makes it relatively easy to adapt the applications.

In this paper is presented a mathematical modeling approach based on the use of computational tools for numerical simulations. It goes from the problem formulation

and equations establishment to the computational implementation and the analysis of the results.

The example that was chosen was that of the oscillatory movement of a mass-spring system without applied forces, subjected to an initial disturbance [10]. This example is part of a wider set of activities performed with great success, since 2007, by students of civil engineering at Engineering Institute in Lisbon, in the subject named Applied Mathematics to Civil Engineering (<http://maecisel.googlepages.com>) as a complement to the study of Differential Equations. Weekly activities involving experiments, mathematical modelling and computer simulation are carried out using Excel. The engineering students prepared reports describing the work that was done and, generally, were enthusiastic about the proposed activities - for them the mass-spring system is replaced by a one-floor building oscillating horizontally (in the classroom they used a physical model of the building which is 20 cm tall, built with an iron floor and four aluminium columns).

Our aim when presenting this paper is that it might work as a motivation for future developments to be used in other classrooms in technical/scientific institutes all over the world. That's why we will present in detail the formulation and all the procedures needed to implement the computational simulation.

The main interest of this application is based on the direct participation of the students in their construction following the next stages:

1. Observation and analysis of the situation from a physical point of view;
2. Formulating the problem;
3. Mathematical modelling: definition of the dependent and independent variables, parameters, domain of the modelling functions and domain for visualisation;
4. Conception of the problem on the spreadsheet (a very important step that helps students to settle down the mathematical concepts introduced on the previous step);
5. Spreadsheet programming according to the structure previously defined. That includes programming the general expression for the function corresponding to the solution of the problem and the drawing of graphics;
6. Analysis of the results studying the influence of the changes on the physical parameters involved and the effect this change produces on the corresponding graphics. With Excel the change of parameters can interactively be done by means of a scrollbar easily programmed using VBA – Visual Basic for Applications.

2. Establishing the differential equation and its exact solution

The oscillatory movement of a mass suspended on a spring can be described by a differential equation that is obtained through the balance of forces that happens at each instant t , between the applied outer force (that can vanish or be constant with

time, for example the gravity force or even, in the general case, a variable force with time $F=F(t)$ and the inertial force F_i , the damping force F_d and the elastic force F_e as shown in figure 1.

$$F_i + F_d + F_e + F = 0 \quad (1)$$

F_e is proportional to the displacement and always acts in the opposite direction of the displacement: $F_e = -k u(t)$

F_d is proportional to the velocity and always acts in the opposite direction of the velocity: $F_d = -c \dot{u}(t)$

F_i is proportional to the acceleration and always acts in the opposite direction of the acceleration: $F_i = -m \ddot{u}(t)$ (it should be noticed that for negative values of u the acceleration is always positive whichever is the sign of the velocity, and for positive values of u the acceleration is always negative)

The equation of motion of the mass is

$$\underbrace{-m\ddot{u}(t)}_{\text{inertial force}} - \underbrace{c\dot{u}(t)}_{\text{damping force}} - \underbrace{ku(t)}_{\text{elastic force}} + \underbrace{F}_{\text{external force}} = 0 \quad (2)$$

or

$$m\ddot{u}(t) + c\dot{u}(t) + ku(t) = F \quad (3)$$

where m represents the mass (measured in Kg), c represents the damping coefficient ($\text{N}/(\text{ms}^{-1})$) and k represents the stiffness coefficient ($\text{N}/(\text{ms}^{-2})$).

$u(t)$ represents the displacement of the mass measured positive upward from its equilibrium position at time $t = 0$.

This equilibrium equation (3) is an ordinary second-order, complete and non-homogenous differential equation $u=u(t)$ that describes the movement of the mass

with time, with $\dot{u} = \frac{du}{dt}$ and $\ddot{u} = \frac{d^2u}{dt^2}$.

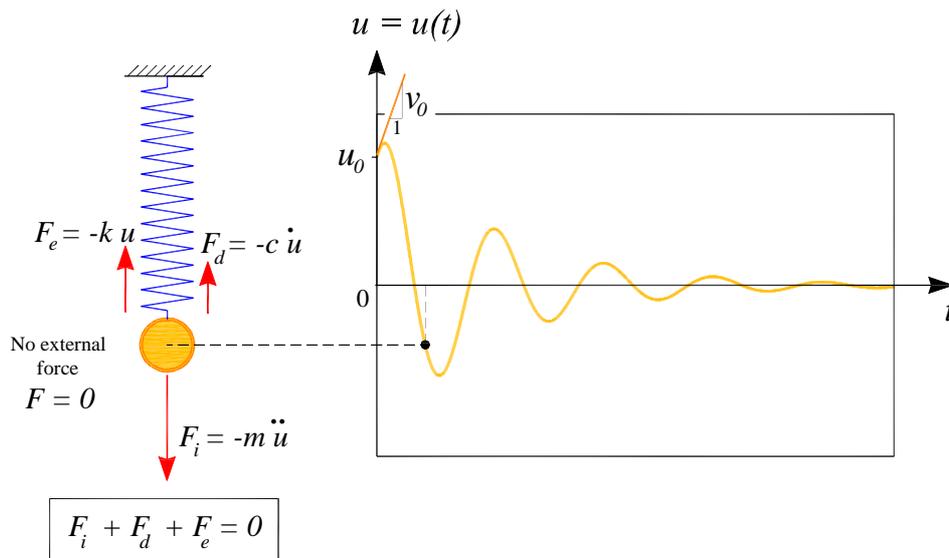


Figure 1: Mass-spring system. Establishing the differential equation through the equilibrium between inertial, damping and elastic forces.

The complete formulation of the vibration problem requires that we specify two initial conditions, namely, the initial position u_0 and the initial velocity v_0 of the mass.

We can have an oscillatory movement even when the external force F is null (only possible with null gravity). Thus we have the following initial value problem (IVP).

$$\begin{cases} m\ddot{u} + c\dot{u} + ku = 0 \\ u(0) = u_0 \quad \dot{u}(0) = v_0 \end{cases} \quad (4)$$

Mathematically speaking, this problem has a unique solution which is the unknown function $u = u(t)$. This matches our physical intuition: if an object is set in movement from a certain initial position $u(0) = u_0$ and with a given initial velocity $\dot{u}(0) = v_0$ then its position $u(t)$ can be determined in all instants.

The position of the object in each instant is determined by the above initial value problem.

It is often not so easy to find solutions of differential equations. If we find a function that we think may be a solution for a given equation, it is relatively easy to verify if that function effectively is a solution or not [6]. Now the problem is to answer the next question: Is there any λ that makes $u(t) = e^{\lambda t}$ a solution of our differential equation?

In this case we can try the function $u(t) = e^{\lambda t}$.

Replacing $u(t) = e^{\lambda t}$, $\dot{u}(t) = \lambda e^{\lambda t}$ and $\ddot{u}(t) = \lambda^2 e^{\lambda t}$ in the former equation we obtain

$$\underbrace{(m\lambda^2 + c\lambda + k)}_{\text{characteristic polynomial}} e^{\lambda t} = 0 \quad (5)$$

leading to the second order algebraic equation

$$m\lambda^2 + c\lambda + k = 0 \quad (6)$$

whose roots $\lambda = -\frac{c}{2m} + \frac{\sqrt{c^2 - 4mk}}{2m}$ and $\bar{\lambda} = -\frac{c}{2m} - \frac{\sqrt{c^2 - 4mk}}{2m}$ determine the general solution of the differential equation

$$u(t) = ae^{\lambda t} + be^{\bar{\lambda} t} \quad a, b \in \mathbb{C} \quad (7)$$

We have three different cases whether the binomial $(c^2 - 4mk)$ is positive, negative or null. To these three cases correspond three types of solutions $u(t)$ determined in physics by the values of the damping value c . For high values of c there is no oscillation. This can be observed for a mass and spring immersed in a medium, such as oil, which impedes the motion [7].

We are just going to study situations where there is oscillation, that is, the cases corresponding to low damping values c , such that $c^2 < 4mk$. In this case we can write

$$\lambda = -\alpha + \omega i \quad \text{and} \quad \bar{\lambda} = -\alpha - \omega i$$

$$\text{with } \alpha = \frac{c}{2m} \quad \text{and} \quad \omega = \frac{\sqrt{4mk - c^2}}{2m} \quad (4mk - c^2 > 0) \quad (8)$$

To determine the particular solution of the equation (7) corresponding to the initial conditions of the IVP (4) $u(0) = u_0$ and $\dot{u}(0) = v_0$ we should write the next equivalence

$$\begin{cases} u(0) = u_0 \\ \dot{u}(0) = v_0 \end{cases} \Leftrightarrow \begin{cases} a + b = u_0 \\ a\lambda + b\bar{\lambda} = v_0 \end{cases} \quad (9)$$

Solving the system in order to a and b we get:

$$\begin{cases} a = \frac{A - Bi}{2} \\ b = \frac{A + Bi}{2} \end{cases}, \quad A = u_0, \quad B = \frac{u_0\alpha + v_0}{\omega} \quad (10)$$

Replacing a and b in eq.(7) and taking into account that $e^{\lambda t} = e^{-\alpha t} e^{i\omega t}$

$$u(t) = \left[A \left(\frac{e^{i\omega t} + e^{-i\omega t}}{2} \right) + B i \left(\frac{e^{-i\omega t} - e^{i\omega t}}{2} \right) \right] e^{-\alpha t} \quad (11)$$

noting that $i = -\frac{1}{i}$ and using the Euler formula for complex numbers

$$u(t) = \left[A \underbrace{\left(\frac{e^{i\omega t} + e^{-i\omega t}}{2} \right)}_{\cos(\omega t)} + B \underbrace{\left(\frac{e^{i\omega t} - e^{-i\omega t}}{2i} \right)}_{\sin(\omega t)} \right] e^{-\alpha t} \quad (12)$$

so, the solution of the IVP (4) is

$$u(t) = \left[u_0 \cos(\omega t) + \frac{u_0 \alpha + v_0}{\omega} \sin(\omega t) \right] e^{-\alpha t} \quad (13)$$

The parameter ω is usually denoted by the damped frequency (when the damping coefficient c doesn't vanish): $\omega = \frac{\sqrt{4mk - c^2}}{2m}$.

As the damping c approaches zero, the parameter ω approaches $\sqrt{\frac{k}{m}}$, usually known as the undamped natural frequency of the system.

3. Using EXCEL to obtain graphical representations of the solution

Choosing the spreadsheet (Excel) to simulate the oscillatory movement of a mass-spring system along time, is due to the fact that it allows doing the numeric representation (in a table) using symbolic expressions as well as the visual representation using tables built for that purpose. The ability to switch on and explore the numeric, symbolic and visual representations as well as the possibility to do so in a dynamic way, make the spreadsheet an essential tool to promote abstraction and algebraic thinking [11].

Excel is a highly programmable product, and it is easily the best choice for developing spreadsheet-based applications. Excel uses the VBA language, which is now in widespread use. This language allows the user to create structured programs directly in Excel. On the other hand, with Excel it's easy to access to controls such as scrollbars because their implementation requires little or no macro programming [12].

Using a spreadsheet we can obtain the graphic of the previous solution $u(t)$ that expresses the relation between time t , from the initial instant, and the displacement of the mass $u(t)$. At the same time it is also possible to visualize the so called state curve that represents parametrically the graphic position versus velocity at each instant of the movement.

As the particular solution of our problem is a function depending upon the parameters m (mass), c (damping) and k (stiffness), and also from the initial position of the mass u_0 as well as from initial velocity v_0 , we will see how to build graphics that can be interactively modified using scrollbars. Conceiving the spreadsheet itself is fundamental for the development of the computational module. There should be devoted a special attention to this early stage. This is a moment of structuring mathematical thinking on this subject. For this reason it is important to project the structure of the computational module, not only in terms of the mathematics involved, but also in terms of the visual aspect desired. It is important to define the cells for the parameters m , c and k and for the initial conditions u_0 and v_0 . We will also include two cells for the values of α and ω .

We are going to begin with the organization of the spreadsheet for a simultaneous visualisation of the parameters m , c and k and the initial conditions.

	A	B	C	D	E	F	G	H	I	J	K	
1	Spring-mass system. Free-vibration with damping											
2												
3												
4		Parameters	Mass	m=	5.1					Initial conditions	u₀=	1.9
5	Damping		c=	2.3					v₀=		1.6	
6	Stiffness		k=	250					Time discretization			
7			w=	7.00		α =	0.22549				Δt=	0.04
8												

Figure 2: Excel display of our spreadsheet construction.

In figure 2 it is important to notice that the values in cells C4, C5 and C6 correspond respectively to the values of the mass m , the damping coefficient c and the spring stiffness k which will be controlled through scrollbars. The same is true in relation to the initial u_0 and v_0 .

The values appearing in cells C7 and E7, ω and α respectively are calculated using the expressions in (8).

With all these parameters previously defined it is possible to build a graphic to illustrate the displacements of the spring as a function of time, beginning with the definition of the table for the values of t and $u(t)$.

In figure 3, column B generates successive values of x , starting with 0, from cell B13 to cell B513 and incremented in steps of size Δt where we enter the value of Δt in the indicated cell J7. In addition, starting in cell C13 and going to cell C513, the corresponding values $u(t)$ will be calculated using the expression shown in (13).

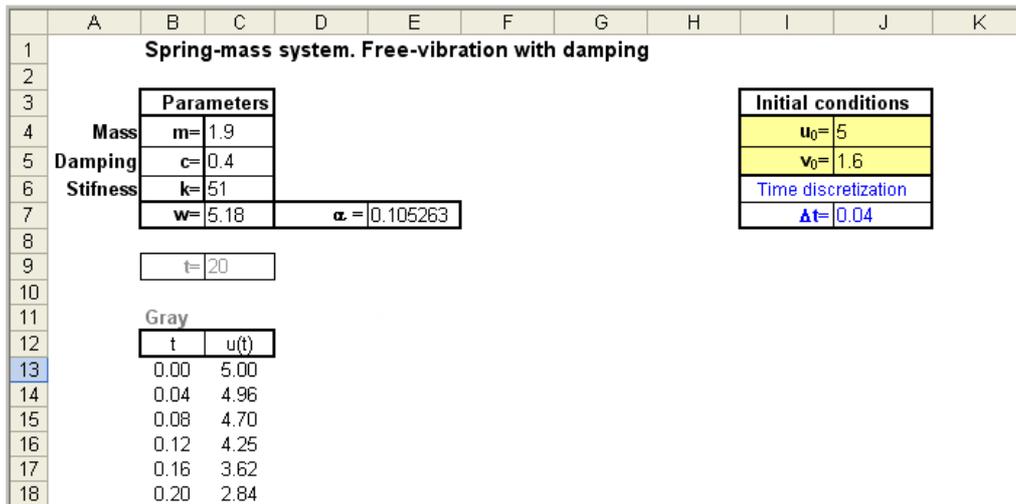


Figure 3: Using spreadsheet we can represent the mathematical situation in a discreet way using a table for the values of t and $u(t)$.

To build the graphic of the function presented in (13), using the values calculated in columns B and C, we should select cells B13 to C513 and choose Chart Wizard button. Inside we should pick a type XY scatter plot graphic. Then it is essential to format the graphic paying special attention to the scale.

Figure 4 shows the graphic for the displacements of the mass as a function of time, for the parameters and initial conditions previously defined.

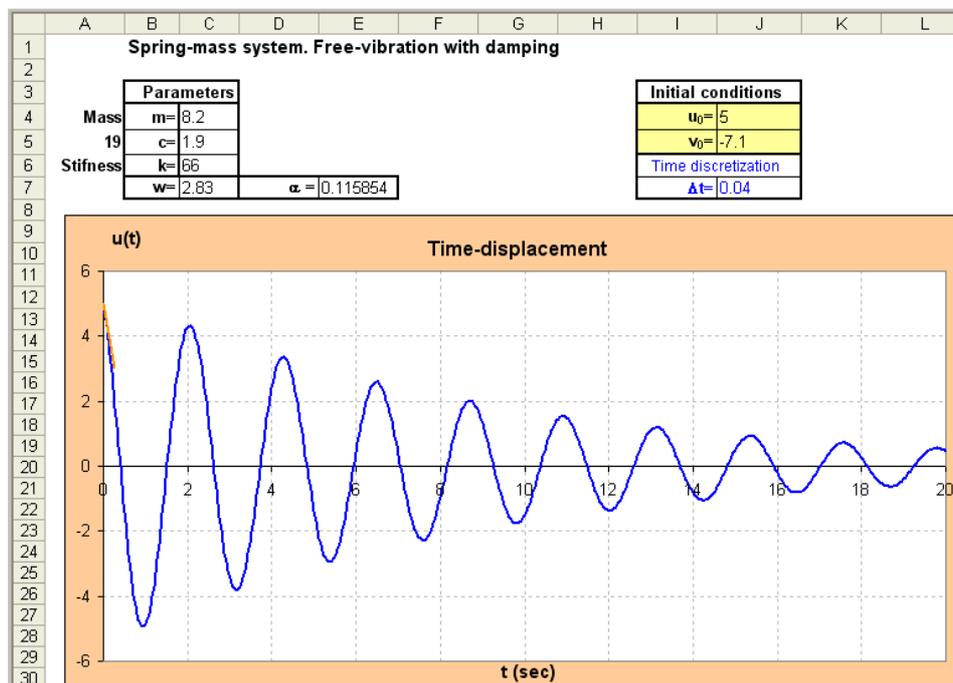


Figure 4: Graphic time versus displacements.

Finally, to include in the application, buttons of the type “Scrollbar” to control the parameters and the initial conditions, we must show the Control Toolbox bar, using the menu Tools-Customize-Toolbars.

Then we click on the Design Mode button which appears on the upper left hand corner of the Control Toolbar and we choose the scrollbar icon.

Then we have to work with the mouse to build the scrollbar and click on the scrollbar with the right button of the mouse which enables us to define the minimum and maximum values and the linked cell once we have chosen the option Properties. Another way to program the scrollbar is to use a code written in visual basic. For that we just need to double click on the scrollbar which opens the Visual Basic Programming Editor.

The following code corresponds to scrollbar1 which is linked to cell C4 where we can control the mass.

```
Private Sub ScrollBar1_scroll()
' Mass
Cells(4, 3) = ScrollBar1.Value / 10
End Sub
Private Sub ScrollBar1_Change()
' Mass
Cells(4, 3) = ScrollBar1.Value / 10
End Sub
```

We should follow the same procedure to program the rest of the scrollbars corresponding to the other parameters and initial conditions.

Each time we want to activate the scrollbar we just need to click on the Design Mode button.

In figure 5 we can see the application showing the five scrollbars.

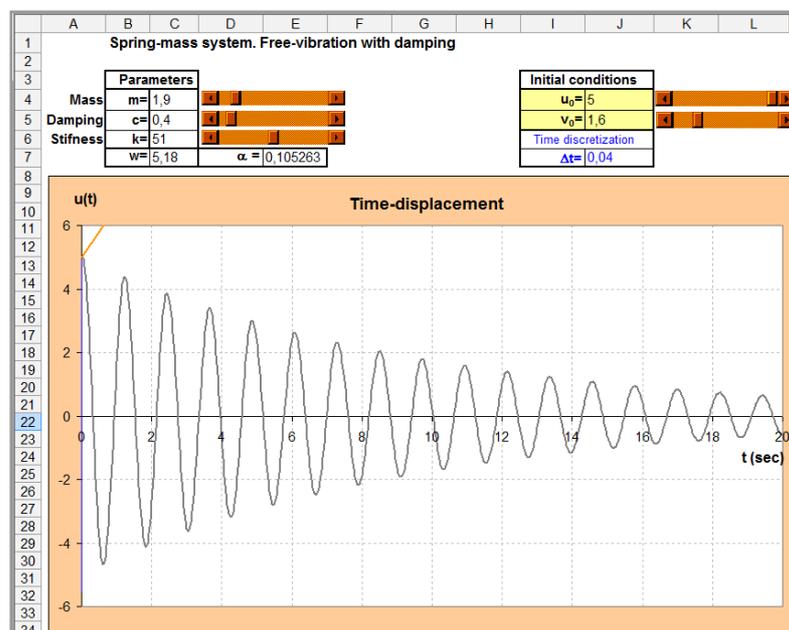


Figure 5: Students can engage in explorative activities that result from changing the values of parameters and the initial conditions within the representation, using scrollbars.

To complete the application we include the state-curve shown in figure 6. This graphic is constructed in a manner similar to the previous one. One needs to set a table of values and then use the option that allows Excel to make graphics. Then we must proceed to the respective chart formatting. This chart is also dynamic when the values of the parameters and initial conditions are changed.

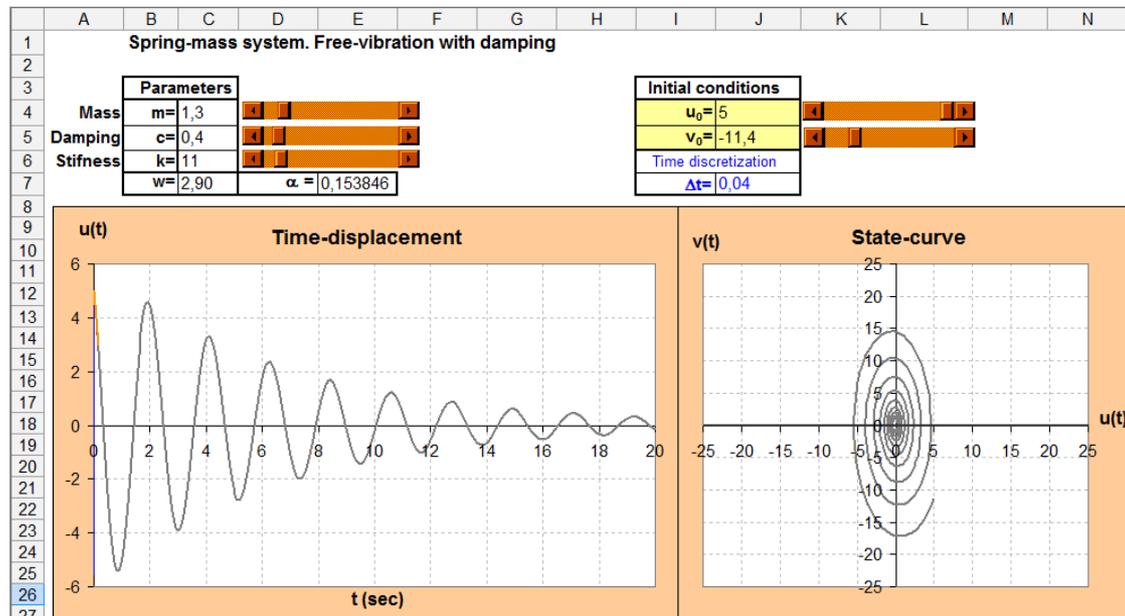


Figure 6: Time-displacement graphic and state-curve.

Then we show how to make the previous application more interactive and sophisticated through the use of VBA, Visual Basic for Applications.

VBA is a programming language embedded in Excel. VBA manipulates objects, and each application has its own unique object model [12]. The actions are executed after running the VBA code. To write the code uses the Visual Basic Editor pressing [Alt]F11. It will open an editor exhibiting the VBAProject structure [13].

You can use the Visual Basic for Applications to complete the aforementioned application, including a button that allows, when clicked to gradually see the graphics. This gives students the opportunity to monitor the movement over time. It is only necessary to click on the command button to fill a complete set of cells. If we plot a graphic with these values then we have the effect of movement.

To introduce the button type "Command Button" referred above (controlled with just a click) we use again the Control Toolbox bar and begin by drawing the button on the spreadsheet. We must use a For loop to repeat a operation a defined number of times.

The intention to submit the code is only to demonstrate that a simple subroutine can provide a short animation.

The code for the command button is the following:

```

Private Sub CommandButton1_Click()
Range("F13:F513").Clear ' Delete all cells from F13 to F513
Range("H13:I513").Clear ' Delete all cells from H13 to I513
Sheets(2).Range("E13:G513").Clear ' Delete all cells from E13 to G513
in sheets 2
Cells(9, 6) = 1 'Hide

For Line = 13 To 513
DoEvents
Cells(9, 3) = Cells(Line, 2) 'Be placed in cells C9, the value of
time that is in cells B13:B513
Cells(Line, 6) = Cells(Line, 3) 'Be placed successively in cells F13:
F513 values of displacement that is in cells B13: B513
Cells(Line, 8) = Cells(Line, 3) 'Be placed successively in cells H13:
H513 values of displacement that is in cells B13: B513
Cells(Line, 9) = Cells(Line, 4) 'Be placed successively in cells I13:
I513 values of velocity that is in cells D13: D513
Next Line
End Sub
    
```

In figure 7 we can see part of the application after insertion of a command button.

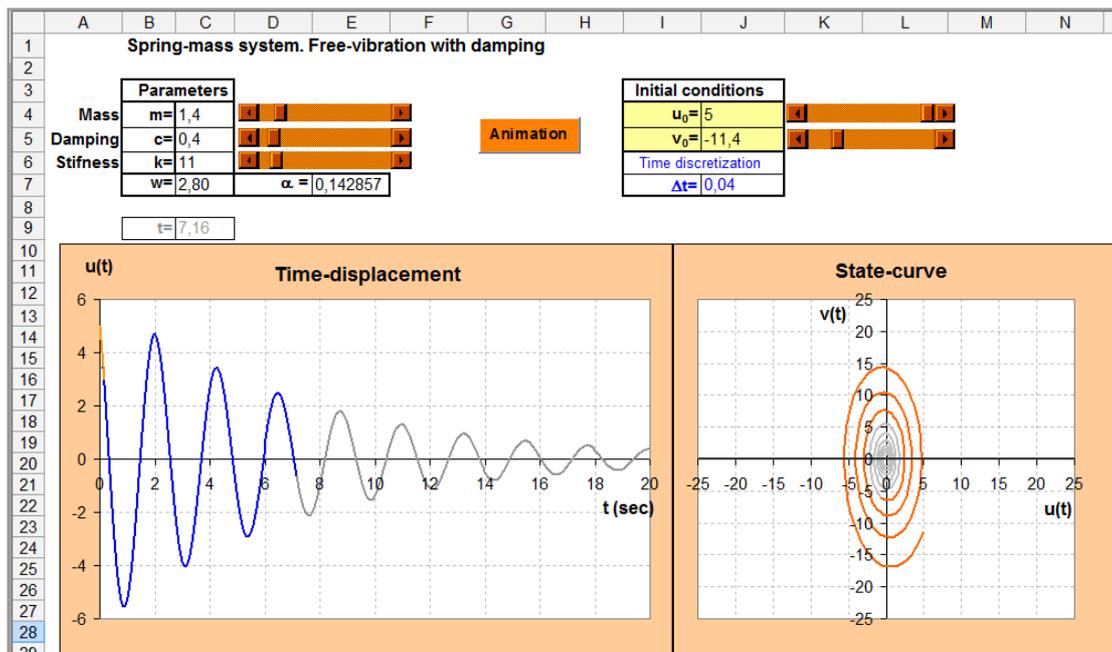


Figure 7: The command button "animation" when clicked lets you see the graphics appear.

4. Using Excel to build a spring and to simulate its oscillatory movement

To enrich the visual part of the application, the oscillatory spring movement has been included.

To do that it is necessary to make the design of a spring. Again it is necessary to work with rigorous planning and organization.

So, we are going to describe how to develop a computational module to draw a spring. We intend to reproduce in Excel the spring shown in figure 8.

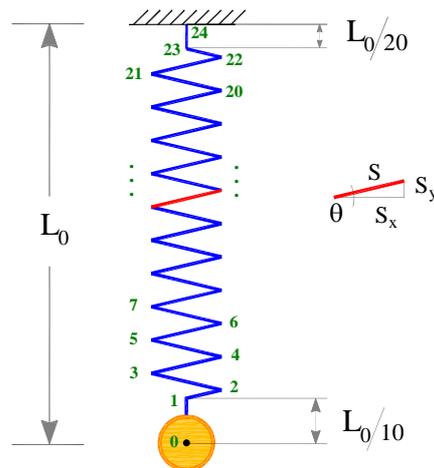


Figure 8 – Spring geometry.

We consider the origin of the reference frame is set to coincide with the centre of the mass when the spring is in its initial position ($F_e = 0$).

Next in the spreadsheet we choose the cells for the values: L_0 total length of the spring; S length of each segment and N the number of segments (between points 1 and 23. There are $N+1$ segments). These values are previously defined.

We also need to define a cell for the value of the length of the spring in each instant L_s , without the vertical segments, in which

$$L_s = (L_0 - u) - 3 \frac{L_0}{20}.$$

Notice that in each instant in which u is calculated, the L_s value gets updated.

In order to visualize the elongation and compression of the spring it should be noticed that the angle θ varies as well as S_y and S_x (see figure 8). The next expressions are used to the computation of these values in each instant.

$$S_y = \frac{L_s}{N+1} \quad \theta = \sin^{-1} \left(\frac{S_y}{S} \right) \quad S_x = \frac{S \cos \theta}{2}$$

All these values are organized in a table as shown in Figure 9.

MASS-SPRING SYSTEM

	J	K	L	M
10				
11		Geometry Spring parameters		
12				
13			L0= 6	
14			S= 1,5	
15			N= 20	
16			Ls= 5,72	
17			Sy= 0,27	
18			$\theta = 0,18$	
19			Sx= 1,48	
20				
21			u= -0,62	
22				

Figure 9: Table with the values previously defined.

After giving the cells the values shown in figure 9 it is necessary to build a table with the coordinates of the segments endpoints that form the spring (in figure 8 these correspond to the numbered points).

Figure 10 presents a scheme (a) which will help building the dynamic graphic of the spring in Excel. Notice that the expressions shown in figure 9 of some of the segments endpoints of the graphic as far as length S is concerned, correspond to the expressions of the ordinate of those points. All the expression depends on u .

(b) shows a scheme which demonstrates the behaviour of the spring along the time.

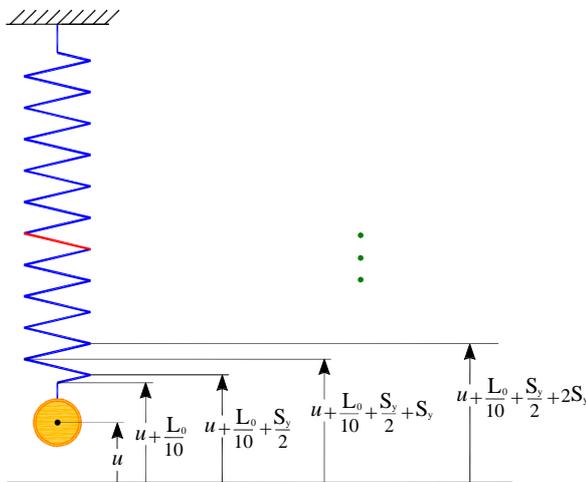


Figure 10 – (a) Dynamic spring graphic (position of the segments endpoints).

(b) The spring elongates and compresses itself along the time.

In order to simulate the oscillatory movement of the mass, the position of each vertex of the spring graphic should be computed in order to take into account a linear distribution of the displacement between the points 1 (maximum vertical displacement u) and 23 (zero vertical displacement).

The horizontal position of each vertex is adjusted in order to take into account the rotation of each segment around their middle point.

In table 1 we can see the formulas to calculate the coordinates of the first points. To calculate the coordinates of other points just use the feature of Excel formulas.

n	x	y
0	0	$= u$
1	0	$= \frac{L_0}{10} + u$
2	$= \frac{S_x}{2}$	$= \frac{L_0}{10} + u + \frac{S_y}{2}$
3	$= -\frac{S_x}{2}$	$= \frac{L_0}{10} + u + \frac{S_y}{2} + S_y$
4	$= \frac{S_x}{2}$	$= \frac{L_0}{10} + u + \frac{S_y}{2} + 2S_y$
⋮	⋮	⋮

Table 1: Expressions for the coordinates of the vertices of the spring.

	A	J	K	L	M	N	O	P
10								
11								
12			Geometry Spring parameters		Spring Geometry			
13						x	y	
14			L0=6		0	0,00	0,36	
15			T=1,5		1	0,00	0,96	
16			N=20		2	0,74	1,07	
17			Lm=4,74		3	-0,74	1,30	
18			Ty=0,23		4	0,74	1,52	
19			θ=0,15		5	-0,74	1,75	
20			Tx=1,48		6	0,74	1,98	
21			u=0,36		7	-0,74	2,20	
22					8	0,74	2,43	
23					9	-0,74	2,65	
24					10	0,74	2,88	
25					11	-0,74	3,10	
26					12	0,74	3,33	
27					13	-0,74	3,56	
28					14	0,74	3,78	
29					15	-0,74	4,01	
30					16	0,74	4,23	
31					17	-0,74	4,46	
32					18	0,74	4,68	
33					19	-0,74	4,91	
34					20	0,74	5,14	
35					21	-0,74	5,36	
36					22	0,74	5,59	
37					23	0	5,70	
38					24	0	6	

Figure 11 – Organization of the spreadsheet.

Finally it is possible, from the table in figure 11 to build the corresponding graphic and to conclude the application.

In figure 12 one can see the completed application containing the graphics time-displacement, the state-curve and the design of the spring.

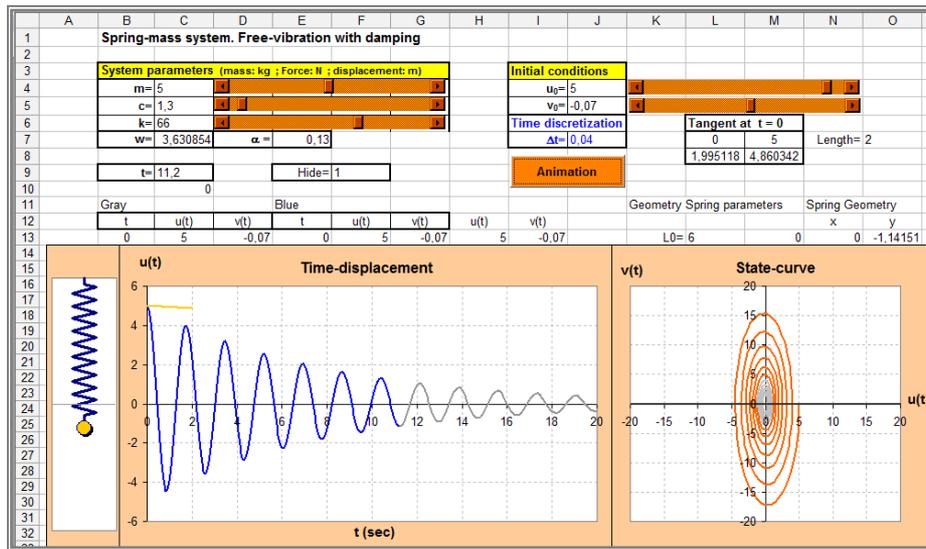


Figure 12 – Final application.

In order to visualize the animation of the time-displacement graphic and the state curve for any chosen set of spring parameters and initial conditions, a cell F9 controlled by the scrollbars and the animation button, was used. The cell can only assume one of two values, 0 or 1. When it is 0 the colour of the graphic is blue; when it is 1 the colour of the graphic is orange.

The values of the graphic tables are previously multiplied by the value of the cell F9 that turns 0 every time the user clicks the scrollbars and turns 1 when the animation button is pressed.

5. Parameter Studies

We are now able to show how the computational application we have constructed can be used to analyze the influence of the different parameters on the action of the mass-spring system that was mathematically described as an initial value problem

$$\begin{cases} m\ddot{u} + c\dot{u} + ku = 0 \\ u(0) = u_0 \quad \dot{u}(0) = v_0 \end{cases}$$

We are going to study the influence of the five parameters involved:

Mass, m

Damping, c

Stiffness, k

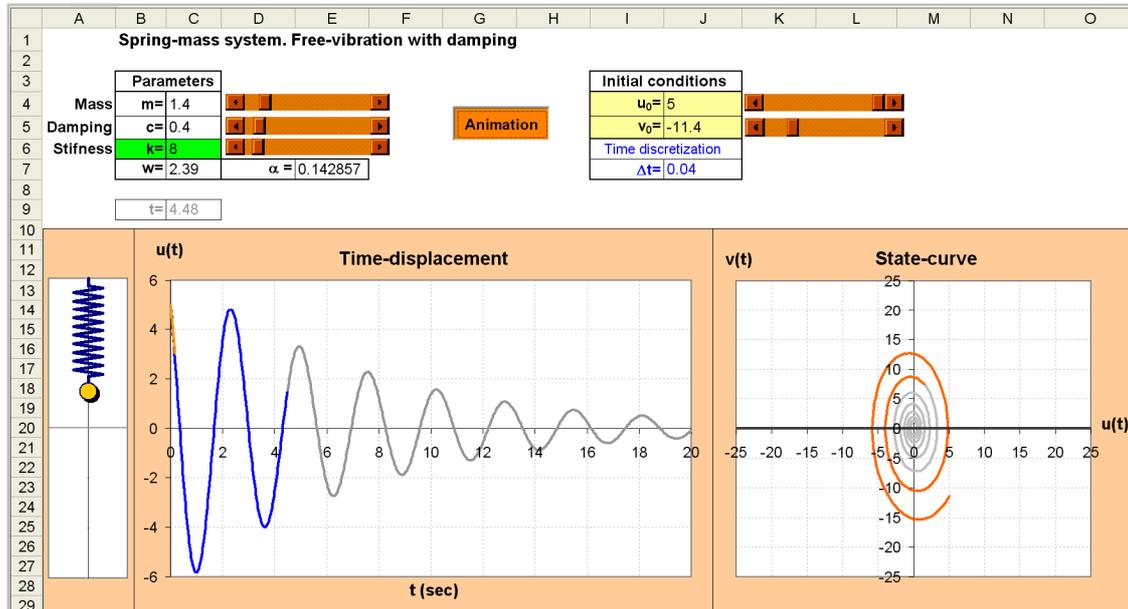
Initial displacement, u_0

Initial velocity, v_0

The introduction of scrollbar buttons in the computational application (one for each parameter) is of great interest as it allows analysis of the variation of the parameters in an interactive way, giving the immediate visualization of the solution for the initial values problem we are analyzing.

Influence of k (stiffness of the spring)

(a)



(b)

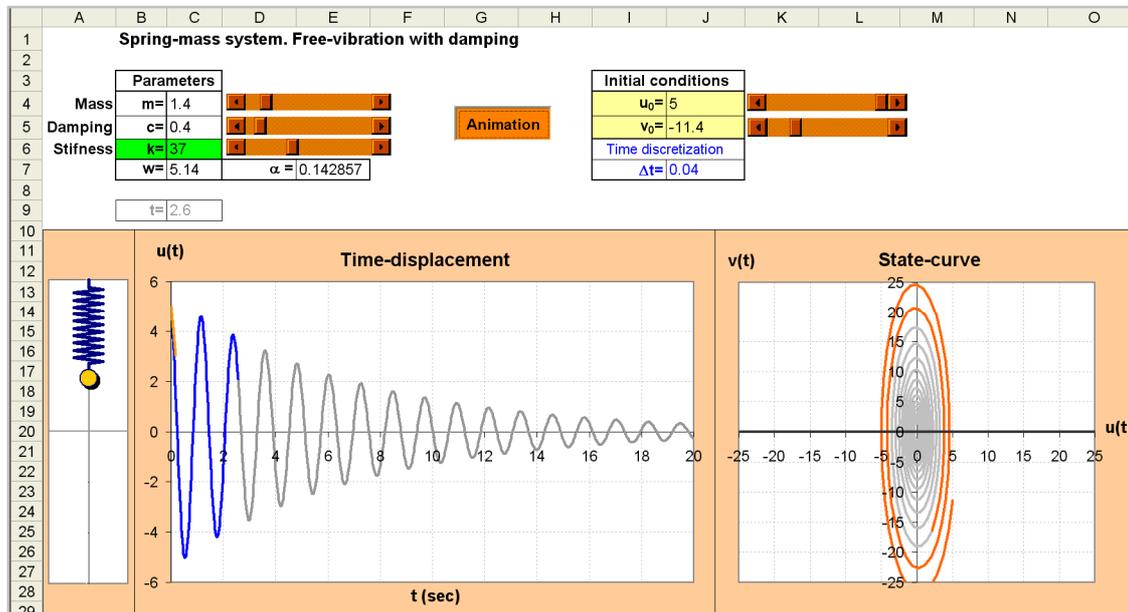


Figure 13 – Influence of k : (a) $k=8$ (b) $k=37$.

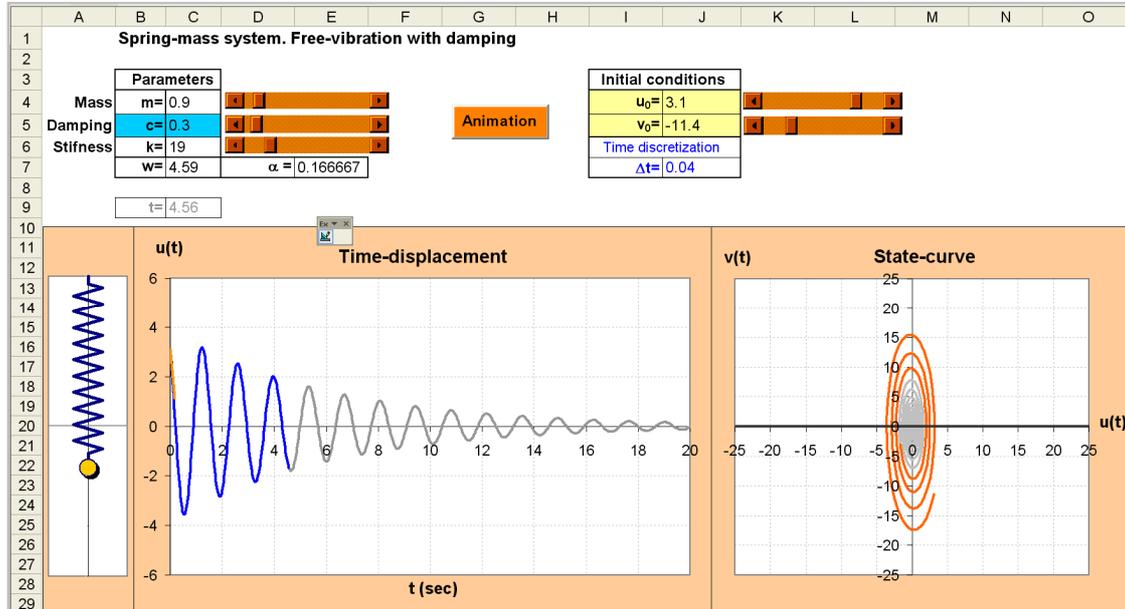
The comparison of figures 13 (a) and 13 (b) allows us to conclude that when we increase the stiffness of the spring, the oscillation frequency (the number of cycles per unit of time) also increases. Notice that we can easily obtain the frequency in

MASS-SPRING SYSTEM

cycle/sec (or Hertz) by counting in the graphic, the number of “peaks” between 0 and 20 seconds.

Influence of c (damping of the spring)

(a)



(b)

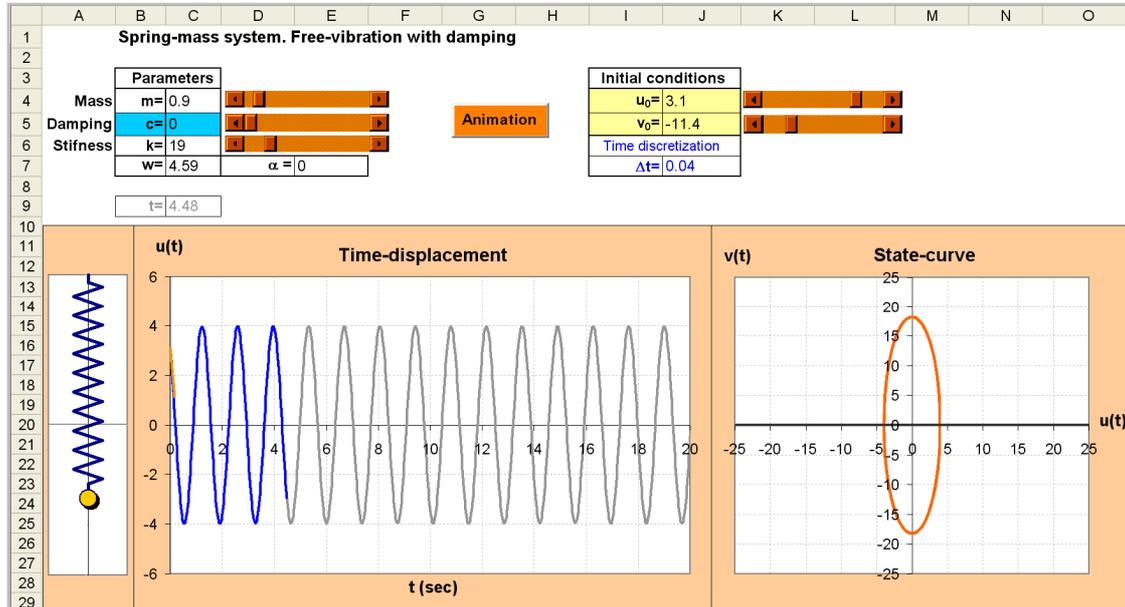


Figure 14 – Influence of c : (a) $c=0.3$ (b) $c=0$.

It is interesting to verify that the graphic of figure 14 (a) has the axis of time as horizontal asymptote which means that the amplitude of the oscillations is decreasing with time. In energetic terms, that corresponds to a dissipation of the elastic force that appeared initially in the system due to the initial displacement u_0

and initial velocity v_0 . In this case, the state-curve $(u(t), v(t))$ corresponds to a parametric spiral curve: For $t = 0$ the displacement and the velocity correspond to the initial values u_0 and v_0 and along with time the values $u(t)$ and $v(t)$ go to zero.

In figure 14 (b) we present the results corresponding to a situation of null weakening, $c = 0$ (it is an ideal situation, without dissipation of energy, that never occurs in real examples). In this case, the amplitude remains constant in the graphic $(t, u(t))$ and the state-curve is represented by a closed curve, different from the spiral we observe when there is damping.

Influence of u_0 and v_0 (initial conditions)

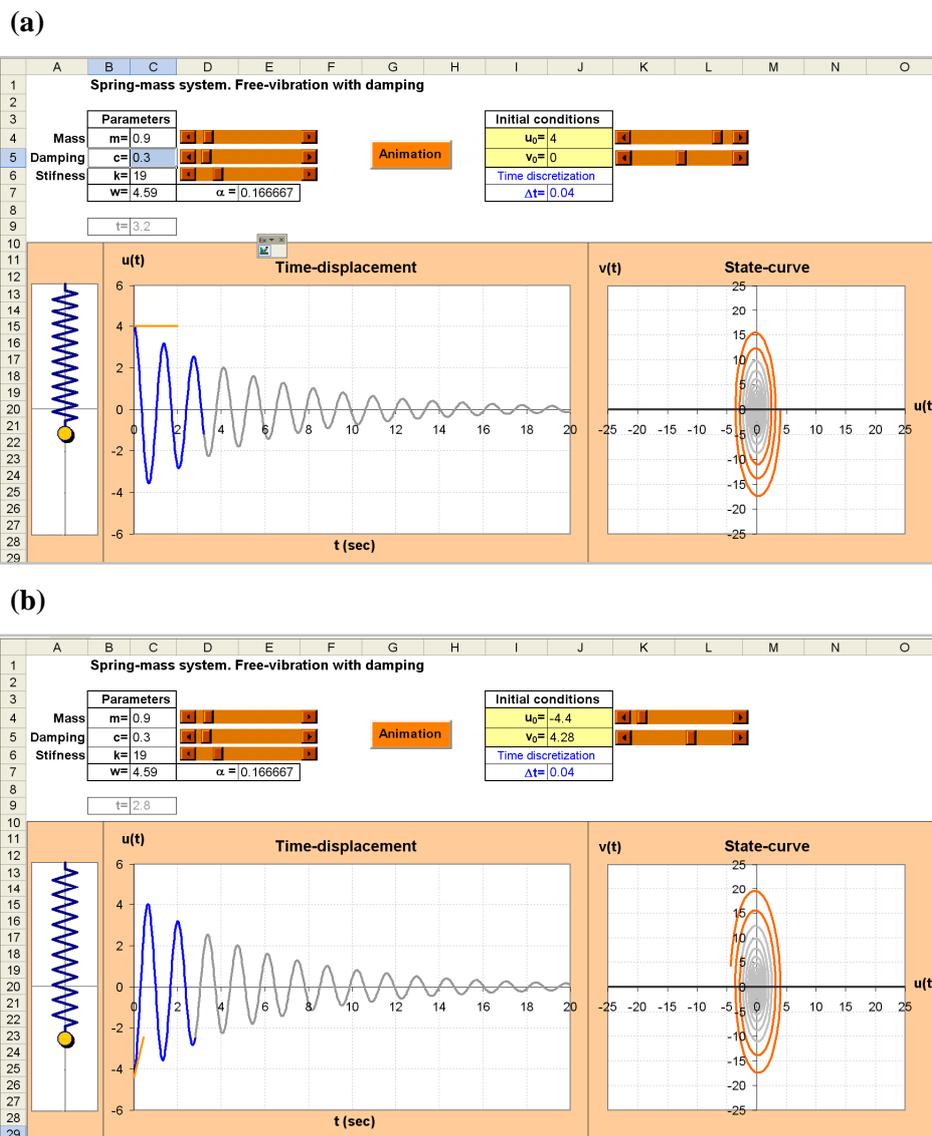


Figure 15: Influence of u_0 and v_0 : (a) $u_0 > 0$ and $v_0 = 0$ (b) $u_0 < 0$ and $v_0 > 0$.

In figure 15 (a) the tangent to the graphic at the initial instant has zero slope. In figure 15 (b) the slope is positive and equals the initial velocity v_0 .

6. Time-state curve with 3D animation (sheet 2)

For t in a given range, the point $(t, u(t), \dot{u}(t))$ traces out a curve in $tu\dot{u}$ space called a time-state curve (see figure 16) [4].

We can define the time-state curve associated with the solution shown in (13) of the damped spring model.

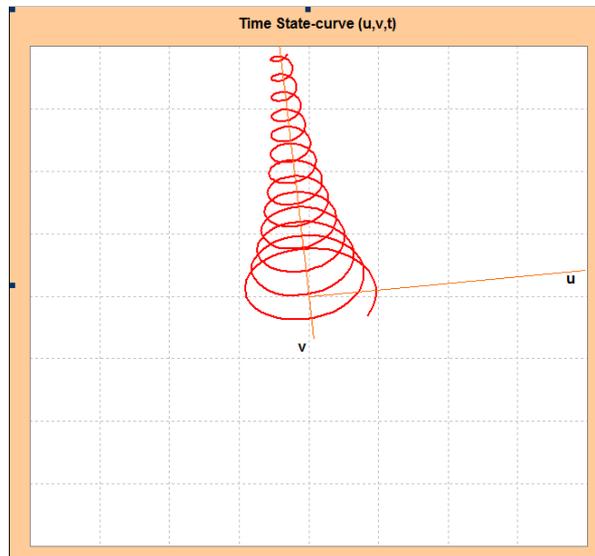


Figure 16 – Time-state curve for the damped spring model.

In order to build the time-state curve we have to bear in mind that the graphic is tridimensional so we have to use the matricial calculus. First we define the rotation angles around each of the axis v , R_u and R_t . In sheet 2 of the application, these values correspond respectively to cells F2, F3 and F4.

Next we define the rotation matrices.

$$R_v = \begin{bmatrix} \cos \theta_v & -\sin \theta_v & 0 \\ \sin \theta_v & \cos \theta_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_u = \begin{bmatrix} \cos \theta_u & 0 & -\sin \theta_u \\ 0 & 1 & 0 \\ \sin \theta_u & 0 & \cos \theta_u \end{bmatrix} \quad R_t = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_t & -\sin \theta_t \\ 0 & \sin \theta_t & \cos \theta_t \end{bmatrix}$$

The rotation matrix is the product of the three previous matrices

$$T = R_v \times R_u \times R_t$$

In Excel we define the matrices R_v , R_u and R_t . After that we apply function MMult which multiplies the two matrixes.

When we want to represent the tridimensional curve $(t, u(t), v(t))$ in a surface (computer screen) it is enough to determine the coordinates in a bidimensional surface (x', y', z') . It is enough to compute the following product:

$$[x' \ y' \ z'] = [x \ y \ z] \times T$$

After computing the product we must consider two matrix columns $[x' y' z']$ chosen according to the required projection plan. For example, we obtain the projection in plan $x' y'$ by drawing the graphic of the two first columns.

Figure 17 shows the organization of the spreadsheet (sheet 2) in order to represent the time-state curve. We can observe the three previous mentioned matrices as well as the final rotation matrix.

	A	B	C	D	E	F	G	H	I	J	K
1											
2		0.87462	-0.48481	0	Rv = 331						
3		0.48481	0.87462	0	Ru = 120						
4		0	0	1	Rt = 0						
5		-0.5	0	0.866025							
6		0	1	0							
7		-0.86603	0	-0.5	Rotation matrix						
8		1	0	0	-0.44	-0.48	0.76				
9		0	1	0	-0.24	0.87	0.42				
10		0	0	1	-0.87	0.00	-0.50				
11											

Figure 17 – Rotation matrix (blue).

Excel works with arrays but we need to know the technique to use them. For example to multiply two matrices we must first select a block of cells where we want to put the result. Then we write in the formula bar =MMult and then use the mouse to select the first array, put a comma and select the second array. Finally we press Ctrl-Shift-Enter rather than Enter [1]

To build the time state-curve graphic it is enough to multiply the values calculated in sheet 1 which refer to t , $u(t)$ and $v(t)$ and which we can find in cells B3:D513 (499 by 3 matrix), by the rotation matrix (3 by 3 matrix). The product we obtain is a 499 by 3 matrix.

We get a time-state curve when we graphically represent the solutions $u(t)$ and $v(t)$. Figure 18 shows in columns A, B and C, starting in line 13 the results as well as the corresponding graphic.

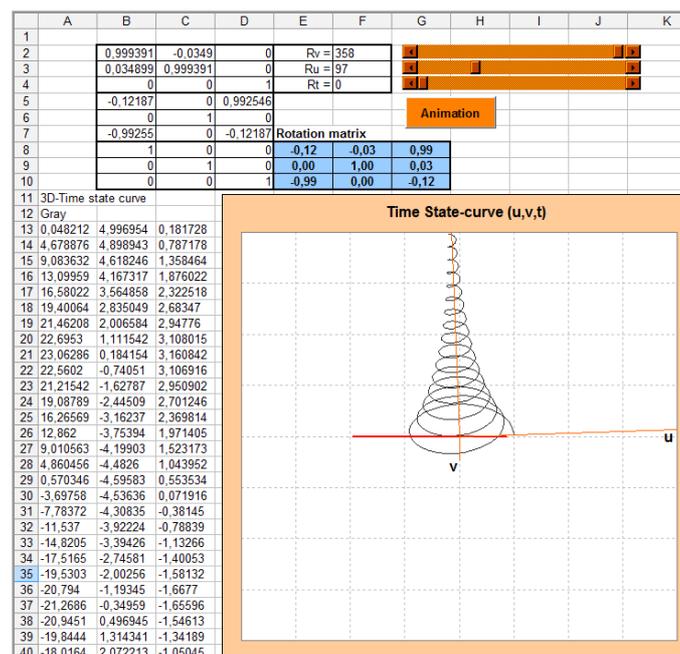


Figure 18: Calculations and the corresponding graphic.

In figure 19 we can see some results

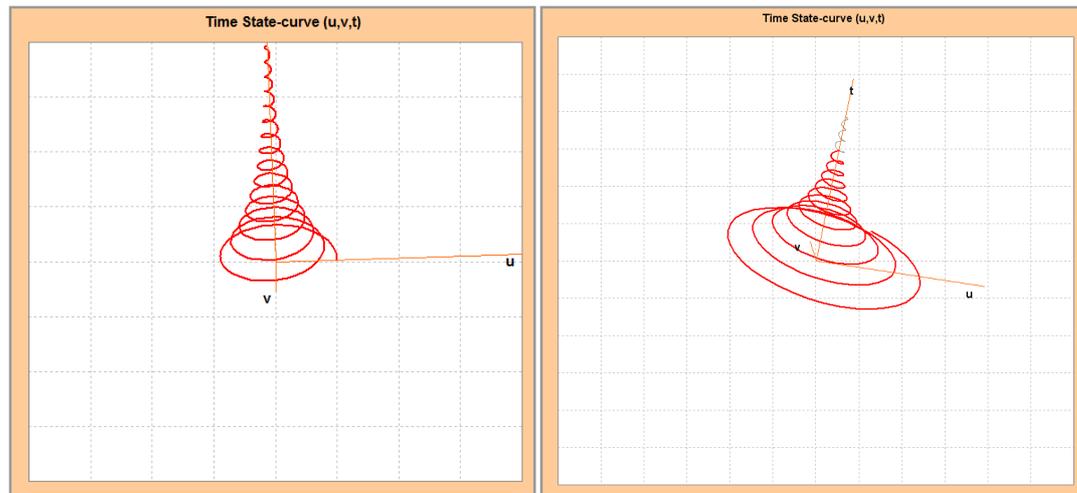


Figure 19: Examples of different time state-curves.

7. Conclusion

This paper intends to show the interest of computational tools and numerical approach to support the mathematical modelling of physical phenomena. In this case the mass spring system based on physical considerations that lead to the establishment of the differential equation that describes it.

The solution of the problem requires the use of a large number of mathematical concepts belonging to distinct areas, and the use of a spreadsheet allows the construction of computational interactive simulations, useful to support the interpretation of the modelling results. It is a mathematical work experiment to be developed by students to promote an integrated understanding of mathematical concepts of different areas and to increase the ability of using mathematics to study the several phenomena of the physical world.

The use of physical examples, of new technological abilities and current graphic computer skills can promote an important way for mathematics communication — the use of a spreadsheet is an original and simple way of computational implementation as it allows the adoption of a structure that raises the different concepts involved in mathematical modelling problems.

The construction and analysis of this type of models and computational algorithms develops skills for a better understanding of many mathematical concepts and for the development of the abilities for structuring the mathematical reasoning.

References

- [1] Arganbright, D. (2005). Enhancing Mathematical Graphical Displays in Excel through Animation, *Electronic Journal of Spreadsheets in Education*. Available online at <http://epublications.bond.edu.au/ejsie/vol2/iss1/8/>.
- [2] Baker, J., Sugden, S. (2003). Spreadsheets in Education - The first 25 years, *Electronic Journal of Spreadsheets in Education*. Available online at <http://epublications.bond.edu.au/ejsie/vol1/iss1/2/>.
- [3] Benacka, J. (2007). Spreadsheet Numerical Modelling in Secondary School Physics and Biology, *Electronic Journal of Spreadsheets in Education*. Available online at <http://epublications.bond.edu.au/ejsie/vol2/iss3/3/>.
- [4] Borreli, R.L. and Coleman, C.S. (1998). *Differential Equations. A Modelling Perspective*, John Wiley & Sons.
- [5] Borwein, J., Morales, M., Polthier, K. and Rodrigues, J.F. (Eds.) (2000). *Multimedia Tools for Communicating Mathematics*, Springer.
- [6] Boyce, W. and DiPrima, R. (2000). *Elementary Differential Equations and Boundary Value Problems*, John Wiley & Sons.
- [7] Braun, M. (1991). *Differential Equations and their applications*. Springer.
- [8] Confrey, J. and Maloney, A. (2007). A Theory of Mathematical modelling in technological settings, *Modelling and Applications in Mathematics Education, The 14th ICMI Study*, Springer.
- [9] Davis, P.J. and Hersh, R. (1981). *The Mathematical Experience*, Mariner Books.
- [10] Kreyszig, E., (2006). *Advanced Engineering Mathematics*. Wiley.
- [11] Sacristán, A.I., Calder, N., Rojano, T., Santos-Trigo, M., Friedlander, A., Meissner, H., Tabach, M., Moreno, L. and Perrusquia, E. (2009). The Influence and Shaping of Digital Technologies on the Learning - and Learning Trajectories - of Mathematical Concepts The 17th ICMI Study, Springer.
- [12] Walkenbach, J. (2004). *Excel 2003 Power Programming with VBA*. Wiley Publishing Inc. USA, John Wiley & Sons.
- [13] Wischniewsky, W.A.L. (2008). Movie-like Animation with Excel's Single Step Iteration Exemplified by Lissajous Figures. *Electronic Journal of Spreadsheets in Education*. Available online at <http://epublications.bond.edu.au/ejsie/vol3/iss1/4/>.