MSEL
Modelling in Science Education and Learning

Modelling in Science Education and Learning
Volume 8(2), 2015 doi: 10.4995/msel.2015.3522.
Instituto Universitario de Matemática Pura y Aplicada
Universitat Politècnica de València

Learning Mechanical Vibrations with Wolfram Mathematica

Aprendiendo Vibraciones Mecánicas con Wolfram Mathematica

Mario Lázaro
Universitat Politècnica de València
malana@mes.upv.es

Abstract

Mechanical vibrations as subject can be found within many Engineering and Science Degrees. To achieve that the students understand the mathematics and its physical interpretation is the objective we should get as docents. In this paper we describe how to create a simple graphical model of a single degree of freedom vibrating system allowing us to visualize concepts like above concepts damping, resonance or forced vibrations. For that, we use the popular symbolic software Wolfram Mathematica with which, without an excessive programming complexity, we can obtain a very satisfactory visual model capable to move itself, controlled by parameters. In addition, the model incorporates the curve-response, something that links the mathematical results with reality

Las vibraciones mecánicas como asignatura están en un muchos estudios de ingeniería y ciencias. Lograr que los estudiantes entiendan las matemáticas y su interpretación física es el objetivo que debemos conseguir como docentes. En este artículo describimos la forma de crear un modelo gráfico sencillo de un grado de libertad dinámico que permita visualizar conceptos como amortiguamiento, resonancia o vibraciones forzadas. Para ello usamos el popular software simbólico Wolfram Mathematica. Sin un excesivo esfuerzo de programación se puede crear un modelo matemático vinculado a un modelo gráfico–dinámico que permite visualizar el movimiento de una masa unida a un muelle. Además, el modelo incorpora también de forma dinámica la curva de respuesta en el tiempo, algo que permite vincular los resultados matemáticos con la realidad.

Keywords: mechanical vibrations, mass-spring-dashpot system, graphical representation, Wolfram Mathematica, time domain animation
Palabras clave: vibraciones mecánicas, sistema masa-muelle-amortiguador, representación gráfica, Wolfram Mathematica, animación en el tiempo

1 Introduction

The analysis of mechanical vibrations becomes nowadays an area of special interest for scientists and engineers due to its wide applicability. Therefore, as subject, *vibrations* can be found approximately in the middle of a typical 4-year Engineering Degree (between the second and the third year). On one hand, subjects as mathematical analysis, mechanics or algebra are required before coursing any introductory course of vibrations. On the other, it is usually the firs stage in a set of specific subjects linked to each engineering area and strongly related with vibrations: isolation analysis, structural dynamics, earthquake engineering, aeroelasticity, control engineering or acoustics, as examples.

Figure 1: Vibrations is always present all around in our daily life

In mechanical vibrations, students meet with new concepts such as damping, resonance frequency, harmonic solution, frequency response function,... all of them based on mathematical results. The complexity increases for multiple degrees of freedom systems and for continuous structures (vibrations of beams). These results are usually presented together with graphical support, for instance in form of response curves in time or in frequency domain.

We can find many specialized books into mechanical vibrations in the bibliography. Thus, for example we find books that introduce vibrations for specific areas as Civil Engineering (Paz, 2013), Aerospace Engineering (Hodges & Pierce, 2001; Wright & Cooper, 2007; García-Fogeda Sanz-Andrés, 2014) or Modal Analysis (He & Fu, 2001). Other references where we find notes on computational and experimental methods in structural dynamics and vibrations are (Kelly, 2000; Thorby, 2008; Gatti & Ferrari, 2003; Meirovitch, 2001). We find of great didactic interest the Open Online Course on Mechanical Vibrations published by MIT (MIT Open Course Ware. Mechanical Vibrations, 2011)and available through a set of chapters covering from single degree of freedom (dof) system to multiple dof and continuous systems.

In order to learn the principles of mechanical vibrations we need to know the mathematical foundations. But as important as this latter is to understand physically how does it move a vibrating body. In this paper we propose to work with the computational symbolic software Wolfram Mathematica (WM) (Wolfram Mathematica, 2015) to reach the main challenges: mathematics and graphical representation of vibrations. WM allows simultaneously to solve and to represent a mathematical model symbolically or parametrically written. Our objective is to describe how to create a simple but illustrative single dof model which dynamically moves vibrating according to some parameters introduced as input data. Based on these graphical objects we will present concepts like forced/free vibrations, resonance or damping.

2 Theoretical foundations

The study of vibrations always begins with the analysis of a single degree of freedom dynamical system. We can imagine a mass m hanging from ceiling, joined with a elastic spring of linear constant k and with a dashpot with damping coefficient c. The latter is a device that acts when the mass has certain velocity. In fact, its force reaction is proportional to the mass velocity. Considering that the spring has a free length L_0, then when the mass is attached it suffers

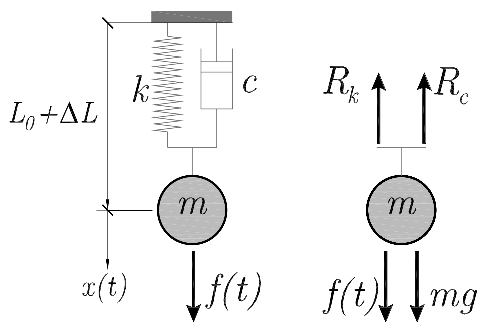

Figure 2: Mass–spring–dasphot setup for modeling a single degree of freedom vibrating system

a extra elongation ΔL. When the system is under static equilibrium in absence of external forces, the spring force equals the weight so that $k\Delta L = mg$. We assume this equilibrium level $L_0 + \Delta L$ as the origin for the vibration amplitudes $x(t)$. The differential equation that describes the movement of the mass is obtained from the dynamic equilibrium, that is the Newton's second law

$$f(t) + mg - R_k(t) - R_c(t) = m\,\ddot{x} \tag{1}$$

where $\ddot{x} = d^2x/dt^2$. The spring and dashpot reactions are $R_k(t) = k\,(\Delta L + x)$ and $R_c(t) = c\,\dot{x}$. Introducing these expressions in Eq. (1) and using the equality $mg = k\,\Delta L$ from the static equilibrium, we have

$$m\,\ddot{x} + c\,\dot{x} + k\,x = f(t) \tag{2}$$

In order to complete the mathematical problem, initial conditions must be introduced. We assume the system has certain initial position and velocity, so that

$$x(0) = x_0\ , \quad \dot{x}(0) = v_0 \tag{3}$$

The solutions of this equation are usually separated according to the values of c and $f(t)$ in the Table 1.

The four problems sequentially introduce very important concepts in the study of vibrations

1. The *natural frequency* of a sdof system is introduced with the free–undamped vibrations problem.

	FREE, $f=0$	FORCED $f\neq 0$
UNDAMPED, $c=0$	$m\,\ddot{x}+k\,x=0$	$m\,\ddot{x}+k\,x=f(t)$
DAMPED, $c\neq 0$	$m\,\ddot{x}+c\,\dot{x}+k\,x=0$	$m\,\ddot{x}+c\,\dot{x}+k\,x=f(t)$

Table 1: Nature of the vibrations depending on the external force $f(t)$ and damping c

2. The phenomenon of *resonance* is studied in the forced–undamped problem.
3. Losing of energy is modeled from the *viscous damping* in free–damped vibrations. This simple model produces an exponential decay of amplitudes, widely observed in Nature.
4. Finally, interaction between the three previous concepts is studied in forced–damped vibrations, introducing the *transfer function* and *frequency domain solutions*, very useful tools for analyzing general forms of the acting force.

To obtain the analytical solution of Eq. (2) for the different cases presented in Table (1) is not the objective of this work. This has already been developed by several authors using for that different methods like Laplace transform, Fourier transform or unit-impulse force among others (Kelly, 2000; Meirovitch, 2001). Since the software Wolfram Mathematica already obtains analytical solutions for a great number of forms of $f(t)$, we will focus on the advantages of this software for representing and visualizing solutions in order to stand out the physical insight of vibrations. In order to reduce the number of parameters of the problem, the Eq. (2) and the initial conditions (3) can be transformed so that the magnitudes become dimensionless. Thus, dividing by m we obtain

$$\ddot{x}+\frac{c}{m}\,\dot{x}+\frac{k}{m}\,x=\frac{f(t)}{m} \tag{4}$$

We denote $\omega_n=\sqrt{k/m}$ to the natural frequency of the system. The damping coefficient is replaced by the so called damping ratio $\zeta=c/2m\omega_n$. Rewriting the Eq. (4) with the new variables

$$\ddot{x}+2\omega_n\zeta\,\dot{x}+\omega_n^2\,x=\frac{f(t)}{m} \tag{5}$$

Now the introduce the variable $T_n=1/\omega_n$ with units of time to get a dimensionless time $\tau=t/T_n=\omega_n t$. The time derivatives must be transformed to the τ–derivatives $d/dt=\omega_n\,d/d\tau$. The τ–derivatives will be denoted shortly by $(\bullet)'=d(\bullet)/d\tau$. After some simplifications we have

$$x''+2\zeta\,x'+x=\mathfrak{f}(\tau)\ ,\quad x(0)=x_0\ ,\ x'(0)=\nu_0 \tag{6}$$

where $\mathfrak{f}(\tau)=f(\tau)/m\omega_n^2$ represents the external force and $\nu_0=v_0/\omega_0$ the initial velocity. Note that as $\mathfrak{f}(\tau)$ as ν_0 have units of length as $x(\tau)$, therefore the Eq. (7) has units of length too. This equation could be divided by some representative length of the problem in order to do it completely dimensionless, although for our proposes we can consider it reduced enough.

3 Working with Wolfram Mathematica

3.1 Solving the equation

The software Wolfram Mathematica (WM) allows to obtain a symbolic solution as function of all parameters of Eq. (7) (ζ, x_0 and ν_0), although for that a specific form of $\mathfrak{f}(\tau)$ is needed. In

Modelling in Science Education and Learning
http://polipapers.upv.es/index.php/MSEL

```
1 x0=0; v0=0; ζ=0.05; r=20; λe=0.80; tmax=100;
2 f1[t_]=1;f2[t_]=Sin[we*t];f3[t_]=4/r^2*t*(r-t)*Sin[we*t];f4[t_]=4/r^2*t*(r-t);f5[t_]=0;
3 f[t_]=-f4[t];
```

Table 2: Definition of parameters and types of external forces

our WM file the user can chose among five different mathematical forms for $\mathfrak{f}(\tau)$

1. Constant function $\mathfrak{f}_1(\tau) = \mathcal{H}(r-\tau)$
2. Armonic function $\mathfrak{f}_2(\tau) = \mathcal{H}(r-\tau)\,\sin(\lambda_e \tau)$
3. Parabolic function $\mathfrak{f}_3(\tau) = \mathcal{H}(r-\tau)\,\frac{4\tau}{r}\left(1-\frac{\tau}{r}\right)$
4. Armonic–Parabolic function $\mathfrak{f}_4(\tau) = \mathcal{H}(r-\tau)\,\frac{4\tau}{r}\left(1-\frac{\tau}{r}\right)\sin(\lambda_e \tau)$
5. Null function $\mathfrak{f}_5(\tau) = 0$

where $\mathcal{H}(\tau)$ denotes the unit–step function, defined as

$$\mathcal{H}(\tau) = \begin{cases} 1 & \text{if } \tau > 0 \\ 0 & \text{if } \tau < 0 \end{cases} \tag{7}$$

Hence, we consider that the force is applied from $\tau = 0$ to $\tau = r$. In the range $\tau > r$ there are not applied forces and the mass is left under free vibrations. Note also that the parameter λ_e represents the ratio between the excitation force and the natural force, that is $\lambda_e = \omega_e/\omega_n$. Therefore, if $\lambda_e = 1$ the excitation frequency is equals to the natural frequency, which means resonance. The fig. 3 shows the plot of each one of the non trivial considered time forces for $r = 60$ and $\lambda_e = 1/2$. The numerical parameters that control the mathematical problem are five:

- Initial conditions (2 parameters), x_0, ν_0
- Damping ratio (1 parameter), ζ
- Parameters of external force (2 parameters), r, λ_e

Let us see that programming the script for the dynamic representation of mechanical vibrations just takes few lines. In Table 2 the first command lines of the WM file are showed. We leave these first lines (1 to 3) to define the value of parameters and the choice of the external force. Note that more type of functions could be defined just adding their expressions in line 2. The parameter `tmax` defines the maximum value of τ used in the different response plots.

Now, we are ready to solve the differential equation (7) and to obtain the response $x(t)$. However, in this work we will use all computing power of WM, letting it do the needed calculations for solving analytically the differential equations with the `DSolve[...]` command. We must take into account that the external force has been defined in two different intervals, consequently also the solution must be separately calculated in these two intervals. Therefore, initial conditions for the second range are position and velocity calculated at the end of the first one in order to guarantee function and derivative continuity. Table 3 shows the needed lines to construct the response `xs[t]`, valid for the full range of (dimensionless) time.

With the analytical solution, each one of the particular cases shown in Table 1 can be calculated (and their solution represented) using the correct values of parameters. Thus, for example, free-undamped vibrations response is obtained with $\zeta = 0$ and `f[t_]=f5[t]` in lines

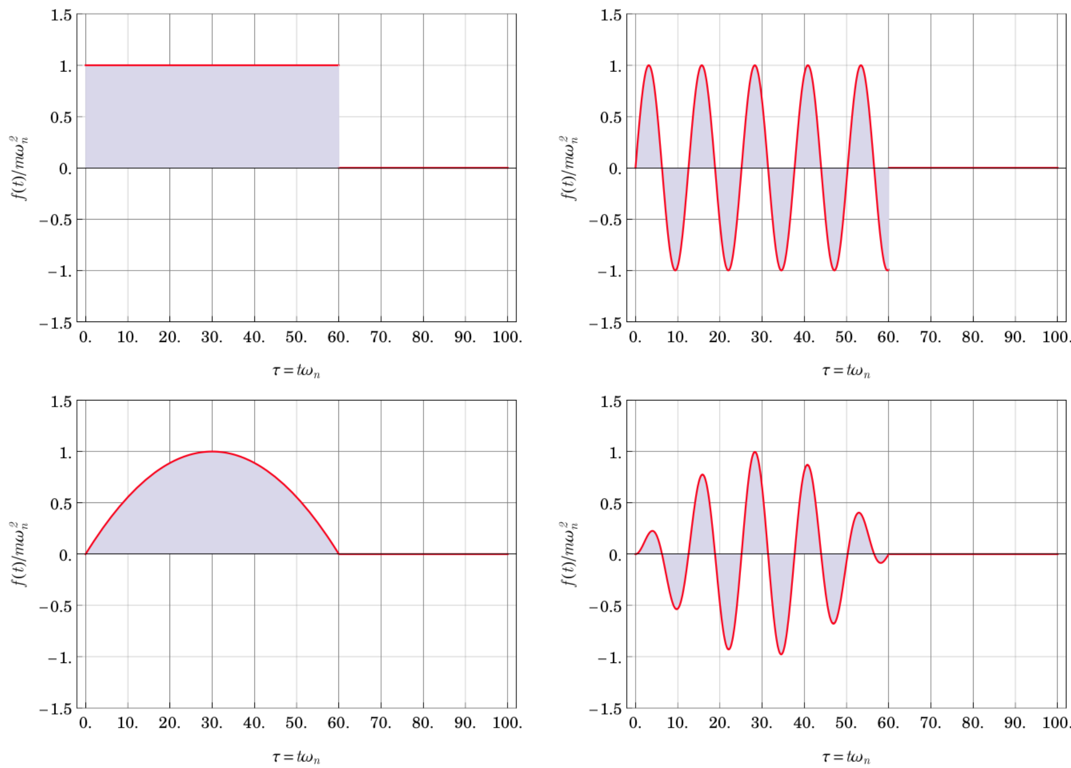

Figure 3: Plots of the different time–functions. $\mathfrak{f}_1(\tau)$, left-top. $\mathfrak{f}_2(\tau)$, right-top. $\mathfrak{f}_3(\tau)$, left-bottom. $\mathfrak{f}_4(\tau)$, right-bottom. The value of parameters: $r = 60$, $\lambda_e = 0.50$

```
4  x1[t_]=x[t] /. Flatten[DSolve[{x''[t]+2*ζ*x'[t]+x[t]==f[t],x[0]==x0,x'[0]==v0},x[t],t]];
5  x2[t_]=x[t] /. Flatten[DSolve[{x''[t]+2*ζ*x'[t]+x[t]==0,x[0]==x1[r],x'[0]==x1'[r]},x[t],t]];
6  xs[t_]=x1[t]*HeavisideTheta[r-t]+x2[t-r]*HeavisideTheta[t-r];
```

Table 3: Command lines for solving the differential equation, using WM function `DSolve`. The line 3 shows how to put together the two separated solutions using the unit step function of Heaviside, `HeavisideTheta[t]`

1 and 3. The mathematical expression is too long for writing here and itself is not of interest for students interested in the physical insight of mechanical vibrations. The objective of this work is not to derive the mathematical expressions of the solution, but visualizing and plotting the response together with external force. The graphic representations are directly related with our interpretation of vibrations and WM provides easy and powerful tools for it.

3.2 Plotting the solution

We consider the graphic output of the mechanical vibrations of special interest from a learning point of view. Sometimes, undergraduate students are overrun by mathematical machinery used in mechanical vibrations: differential equations, Fourier/Laplace transforms, harmonic series expansion, algebra and eigenvalue problems (for multiple dof problems), etc... Although they have been observing vibrations all around in real life, they find difficulties to relate their experiences with the new physical concepts shown in the equations. We suggest to exploit the plotting and animating tools of WM to cover this deficiency. First, we describe in this point how to create a high quality plot, ready for be introduced in any report or technical document.

Modelling in Science Education and Learning
http://polipapers.upv.es/index.php/MSEL

Secondly, in the next point we will create a short script to animate the solution of a single degree of freedom vibration problem (that one shown in Fig. 2).

```
7  ymin=-1.5; ymax=1.5; ny=6; Δy=(ymax-ymin)/ny; ylabels=Range[ymin,ymax,Δy];
8  xmin=0.0; xmax=tmax; nx=5; Δx=(xmax-xmin)/nx; xlabels=Range[xmin,xmax,Δx];
9  Ejex=Text[Style["τ = tωn", Black, FontFamily -> "LM Roman 12", Italic, 15]];
10 Ejey=Text[Style["x(t), f(t)/mωn^2", Black, FontFamily -> "LM Roman 12", Italic, 15]];
   figura = Import["C:/users/figuras/esquema.eps"];
```

Table 4: Plot parameters definition: axis, ticks, labels and legends

In Table 4 we define the numerical values of parameters which control the axis and their labels of the plots. In lines 7 and 8 we define the shown ranges and the associated labels. Lines 9 and 10 assign to variables `Ejex` and `Ejey` character strings which will be visualized near the axis. The last line shows how to use the import command to attach a figure within the graphic plot.

The command `Plot[...]` is used by WM for plotting mathematical functions. A detailed explanation of the different inputs and their syntax can be found in the Help of WM. In addition, many examples illustrating the different cases and situations are also very helpful. Since as the response $x(t)$ as the external force $f(t)/k$ have units of length, both can be plotted in the same axes. While $x(t)$ is said to be the dynamic response, $f(t)/k$ represents the static one, that is, the response if both velocity and acceleration were null. This affirmation is directly derived from Eq. (2) doing $m\ddot{x} = c\dot{x} = 0$. Thus, if curves $x(t)$ and $f(t)/k$ are close, the dynamic effects are negligible. In contrast, the more the discrepancy the higher the influence of inertial and damping forces. To represent simultaneously both curves helps us in the interpretation of the cause-effect relationship of the mechanical problem. The proposed script to generate the plots will be described and applied to three different examples.

Table. 5 shows the command lines used to draw curves $x(t)$ and $f(t)/k$. Line 11 contains the arguments of the function `Plot[...]`. It can be seen from above to below the instructions which control the different variables. We can remark, for instance, the argument `Prolog` used to overlay another imported figure (see Table. 4) on the plot (in this case we attach a sketch of the model). The rest of the arguments do not need to be described with detail since their particular syntax can be found in the Help of WM. The set of properties associated with this plot are saved in a variable named "`g`". To export the figure with different formats we use `Export[...]`. Thus, in lines 12 and 13 the plot saved as `g` is exported in eps and jpg formats, respectively. The eps format is widely used for scientific texts generated with LaTeXas for example the present paper. Other formats are available just changing the extension in the given file name.

To illustrate different scenarios, three examples will be constructed covering some of the proposed external functions (see Table 6 for more information on each example). In particular, in order to simplify the parameters influence, the initial conditions for the three examples are assumed to be null, $x_0 = \nu_0 = 0$.

In Fig. 4 the generated plots with response $x(\tau)$ (dynamic response) and external force $f(t)/k$ (static response) are shown. In the range $0 \leq \tau \leq r$ the external force excites the system with three different forms. Their values are highlighted with background on grey color.

```
g = Plot[{f[t]*HeavisideTheta[r-t], Re[xs[t]]}, {t, xmin, xmax},
     PlotRange -> {ymin, ymax},
     PlotStyle -> {{Red, Thickness[0.004]}, {Blue, Thickness[0.004]}},
     ImageSize -> 600,
     Frame -> True,
     FrameTicks -> {{ylabels, None}, {xlabels, None}},
     FrameLabel -> {Ejex, Ejey},
     LabelStyle -> Directive[Black, 11, FontFamily -> "Century SchoolBook"],
     PlotLegend -> {Style["Force,",Black,Bold,15] Style[" f(t)/k",Black,Italic,15],
                    Style["Response,",Black,Bold,15] Style[" x(t)",Black,Italic,15]},
     ShadowOffset -> 0.0,
     LegendPosition -> {0.15, 0.25},
     LegendTextOffset -> {-0.5, 0},
     LegendSize -> {0.6, 0.2},
     LegendBorder -> None,
     Prolog -> Inset[figura, {85, -0.5}, Top, 30],
     AspectRatio -> 0.5,
     GridLines -> {xlabels, ylabels},
     GridLinesStyle -> Directive[Gray],
     Filling -> {1 -> Axis}]
Export["C:/figures/PlotFigure.eps", g];
Export["C:/figures/PlotFigure.jpg", g];
```

Table 5: Plot command in Wolfram Mathematica to show and export high quality 2D graphic representations

	$f(\tau)/k$	r	ζ	$\lambda_e = \omega_e/\omega_n$	Response plot	Animation frames
Example 1	$-\mathfrak{f}_3(\tau)$	20	0.05	—	Fig. 4 (top)	Fig. 5
Example 2	$-\mathfrak{f}_1(\tau)$	40	0.10	—	Fig. 4 (middle)	Fig. 6
Example 3	$+\mathfrak{f}_4(\tau)$	40	0.10	1.00	Fig. 4 (bottom)	Fig. 7

Table 6: Values of the parameters for each example and figures where plots and animation frames are shown

As said before, the discrepancy between $x(\tau)$ and $f(\tau)/k$ curves give information about the dynamic effects from intertial and damping forces. Thus, In Example 1, these effects are much less important than Examples 2 and 3. In fact, this latter could be taken as a good example to describe the resonance effect. Indeed, $f_4(\tau)$ is an harmonic force with amplitude bounded by a parabolic function (see Fig. 3 right-bottom) and with relative frequency $\lambda_e = \omega_e/\omega_n = 1$. In Fig. 4-bottom we see that relatively low amplitudes of exciting force induces large amplitudes in the response. We note that the force is positive (upward direction) when velocity of mass is also positive. In the next oscillation the same occurs but in negative direction, that is, negative force act when velocity is downward, increasing the absolute amplitude more and more. This coupling is known as resonance. Understanding and interpreting this concept should be one of the main objectives of a first course on mechanical vibrations. Also the free-motion can be observed in the three examples in the range $\tau \geq r$ where the amplitude decay depend directly on the assigned value of damping ratio ζ.

3.3 Animating the results

Some guidelines to construct high-quality plots of the vibrating response have been given in the previous point. The new challenge is to relate the real movement of the mass with the graphical results derived from the solved differential equation. For that, we will use a very useful and easy function implemented in WM: `Manipulate[...]`. This function allows any graphical input depending on one or more parameters to be animated. In our particular case

Modelling in Science Education and Learning
http://polipapers.upv.es/index.php/MSEL

some graphical objects are created so that geometry depend on the response $x(t)$ and $f(t)$, which in addition depend on time. Consequently time (denoted by `t` in WM lines and by τ in equations, both represent dimensionless time) is the parameter whose variation produces the animation effect on the objects.

The Table 7 shows the complete script to generate the animated graphic. Two parts are clearly differentiated: lines 14 to 17 define the geometry of the system, whereas line 18 contains the necessary commands for `Manipulate[...]` function. We describe with more detail each line

```
   ANIMATION PARAMETERS;
14 L0=4.2; d0=0.5; R0=0.3; n0=50; v=0.9; e=3.5; ef=1.2; Xmax = 40; tex=44;
   FORCE DEFINITION;
15 VectorFuerza[y0_,long_,head_] = {Arrowheads[head],Arrow[{{0,y0},{0,y0+long}}]};
   MASS DEFINITION;
16 Masa[d_,U_] = {Black, Disk[{0,U}, d]};
   SPRING POINTS DEFINITION;
17 Muelle[L_,R_,U_,n_] =
       ListPlot[
         Table[{If[j==0||j==n, 0, If[OddQ[j], R, -R]], L-j/n (L+U)}, {j, 0, n}],
         Joined -> True,
         PlotRange -> {-3*Abs[L0],Abs[L0]}];
   ANIMATION - MANIPULATE;
18 ANIMA = Manipulate[
    Show[
a   Graphics[{Brown, Thickness[0.04], Line[{{-d0, L0+d0}, {d0, L0+d0}}]}],
b   Muelle[L0,R0,-e*Re[xs[t]], n0],
c   Plot[e*Re[xs[t-X/v]]*HeavisideTheta[t-X/v], {X,0,Xmax}, PlotRange -> {-Abs[L0],Abs[L0]}],
d   Plot[e*f[t-X/v]*(HeavisideTheta[t-X/v]-HeavisideTheta[t-r-X/v]), {X, 0, Xmax},
         PlotRange -> {- Abs[L0], Abs[L0]},
         PlotStyle -> {Red, Dashed}, Filling -> Axis],
e   Graphics[Masa[d0, e Re[xs[t]]]],
f   Graphics[{Red, Thickness -> 0.005,
              VectorFuerza[e*Re[xs[t]],
                           ef*f[t] (HeavisideTheta[t] - HeavisideTheta[t-r]),
                           0.022]}],
g    ImageSize -> 600, Frame -> True,
g    PlotRange -> {{-2*d0, 6*Abs[L0]}, {-1.2*Abs[L0], 1.2*Abs[L0]}}],
g    LabelStyle -> Directive[Black, 16, FontFamily -> "Century SchoolBook"]], "Time",
h    Row[{Control[{{t,tex,"τ = ωt"},0.001,tmax,.05,Appearance->"Labeled",ImageSize->Tiny}]}]]
```

Table 7: Code lines in Mathematica to define the animation graphics

Line 14. We define some parameters to control the geometries: sizes, scales, etc.... In this line, particular values are assigned to these parameters. For example `L0`, `R0` and `n0` represent, respectively, the length, radius and number of spirals of the spring. `e` and `ef` are two scale factors and `Xmax` is the visual horizontal range of the plot.

Line 15. The force is represented by a red arrow, created using a function named `Vectorfuerza[...]` whose arguments are the position, length and head size.

Lines 16 and 17. Two functions `Masa[...]` and `Muelle[...]` return the graphical objects representing the mass and the spring as function of different arguments like length `L`, radius `R` and number of spirals `n`. The input argument `U` denotes the response $x(\tau)$ and is used to locate the mass and the spring each time.

Line 18. Command `Manipulate[...]` animates a plot or graphic via a continuous variation of one or more parameters. Our variable parameter is obviously the time and all the graphics object previously defined depend on the displacement of the mass $x(t)$, denoted as `xs[t]` in Table 7. From line 18a to 18f, the different graphical objects are sequentially added: In 18a and 18b, a brown block and the spring is drawn, this latter with its end point attached to the mass (defined in line 18e) and animated by the $x(t)$. Lines 18c and 18d plot the curves response $x(t)$ and force $f(t)/k$ with different colors. The curves seem to be drawn by the mass creating an interesting dynamical effect. In line 18f we define the force arrow attached to the mass and dynamically sized depending on its absolute value, given by the curve. Lines 18g control the general aspect of the graphic plot: size, labels and plot range. Line 18h creates the control bar to manipulate the time domain, starting at the frame located at time `tex`, and ending at time `tmax` previously defined in lines 18a and 1 respectively

Figs. 5, 6 and 7 show different instantaneous frames of the animation for each one of the three examples presented at Fig. 4. In each example, the movement of the mass generates the response curve (blue) while simultaneously the force evolution (red) is also plotted. Both curves moves to the right like a wave in order to overview the complete force-response solution.

This form of representing models using command `Manipulate[...]` have become very popular to visualize a large number of mathematical models. Even there exists a free access platform named *Wolfram Demonstration projects* and linked to Wolfram Mathematica, where we can find a huge amount of uploaded files with dynamical and interaction demonstrations solved with Mathematica. The files are built under the CDF Technology (Computable Document Format). We can use their own words to describe it *"[...] launched by the Wolfram Group, the CDF standard is a computation-powered knowledge container-as everyday as a document, but as interactive as an app"*. (CFD Technology, 2015).

4 Conclusions

In this paper we describe how to program in Wolfram Mathematica a simple, but didactic, graphical single degree of freedom vibrating model. The objective is to complete the learning process of students in mechanical vibrations with a resource capable to move itself dynamically controlling some parameters. In addition to the graphical objects, also the curve-response and the force-input are embedded putting together the mathematical results with the physical insight of the phenomenon. In order to describe the process, some visual examples are also presented.

References

CFD technology. (2015).(2011).
http://www.wolfram.com/cdf/

García-Fogeda, P., & Sanz-Andrés, A. (2014).
Introducción a las vibraciones.
Garceta Grupo Editorial.

Gatti, P. L., & Ferrari, V. (2003).
Applied Structural and Mechanical Vibrations. Theory, Methods and measouring instrumentation.
Taylor & Francis Group.

He, J., & Fu, Z.-F. (2001).
Modal Analysis.
Butterworth Heinemann.

Hodges, D. H., & Pierce, G. A. (2001).
Introduction To Structural Dynamics And Aeroelasticity.
Cambridge Aerospace Series.

Kelly, S. G. (2000).
Fundamentals of Mechanical Vibrations.
McGraw Hill.

Meirovitch, L. (2001).
Fundamentals of Vibrations.
McGraw Hill.

MIT Open Course Ware.
Mechanical Vibrations. (2011).
http://ocw.mit.edu/courses/mechanicalengineering/2-003sc-engineering-dynamics-fall-2011/mechanical-vibration/

Paz, M. (2013).
Structural dynamics: theory and computation.
Springer.

Thorby, D. (2008).
Structural Dynamics and Vibration in Practice.
Butterworth Heinemann.

Wolfram Mathematica. (2015).
http://www.wolfram.com/mathematica/

Wright, J. R., & Cooper, J. E. (2007).
Introduction to Aircraft Aeroelasticity and Loads.
John Wiley & Sons.

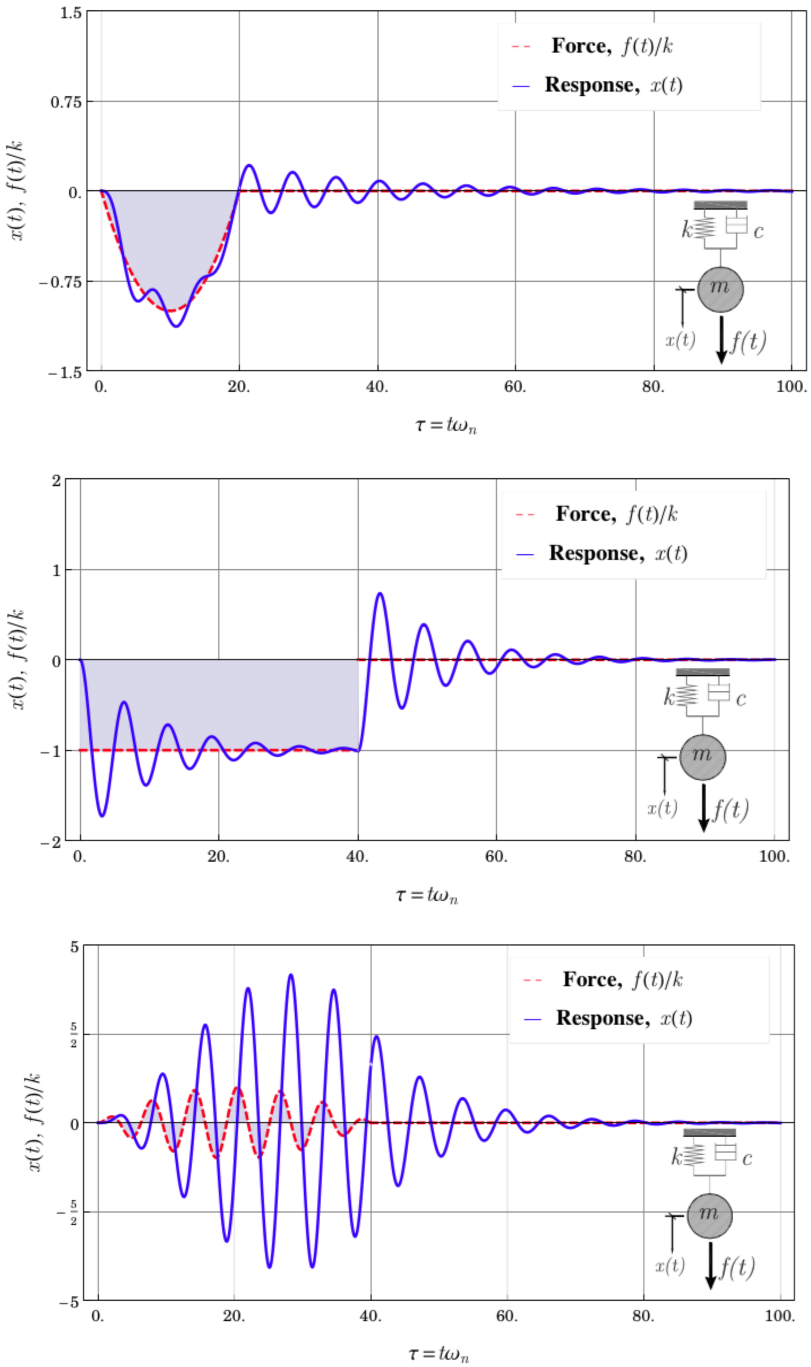

Figure 4: Output plots of response $x(t)$ and external force $f(t)/k$ according to script shown in Table 5. Top, parabolic force with downward direction up to $\tau = 20$. Middle, uniform downward force up to $\tau = 40$. Bottom, harmonic force with resonance frequency bounded by a parabolic function up to $\tau = 40$

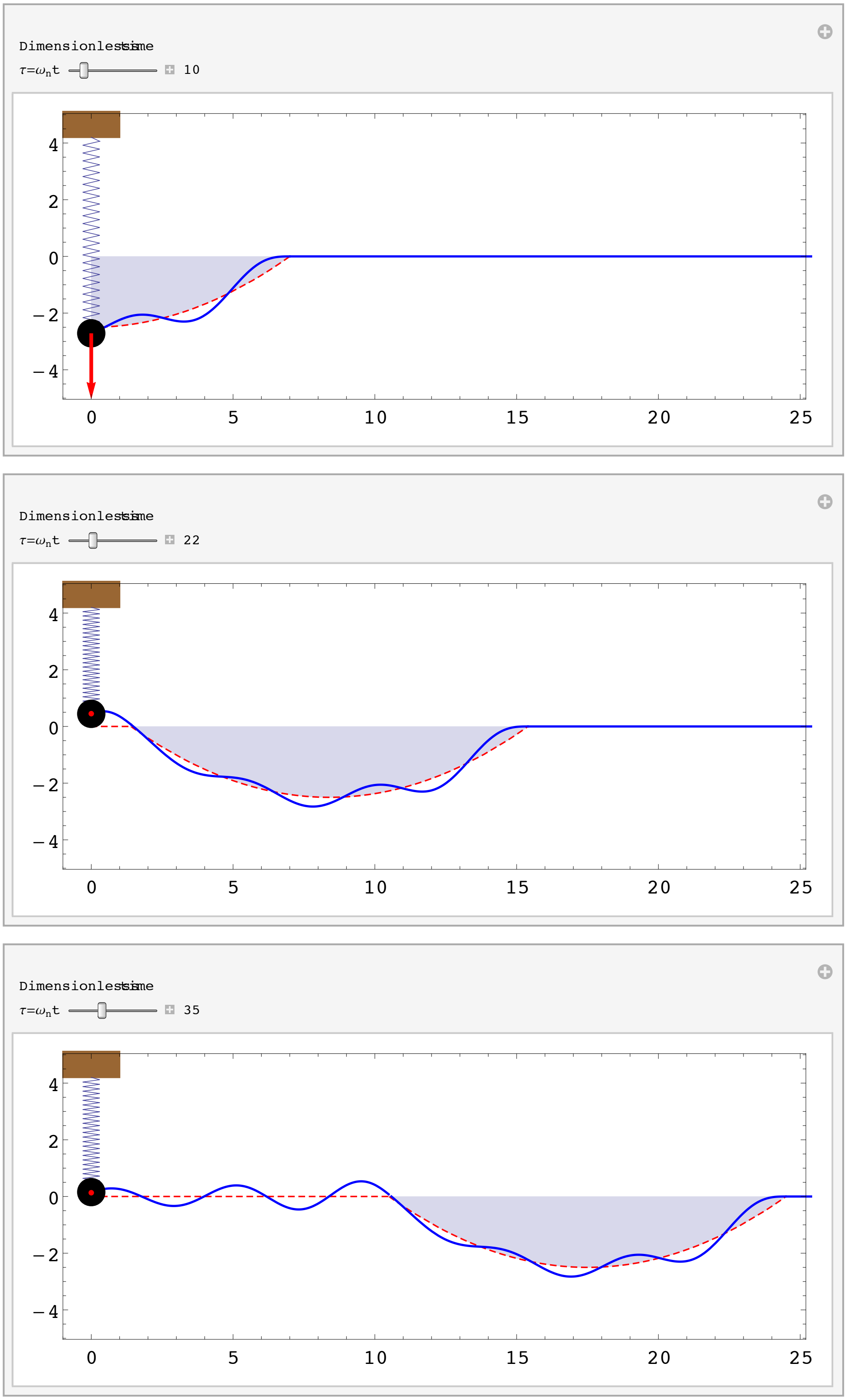

Figure 5: Three frames of animating video generated with the Mathematica file for parabolic-type external force. Upper: during applied force. Center: end of applying force. Bottom: free-damped vibrations

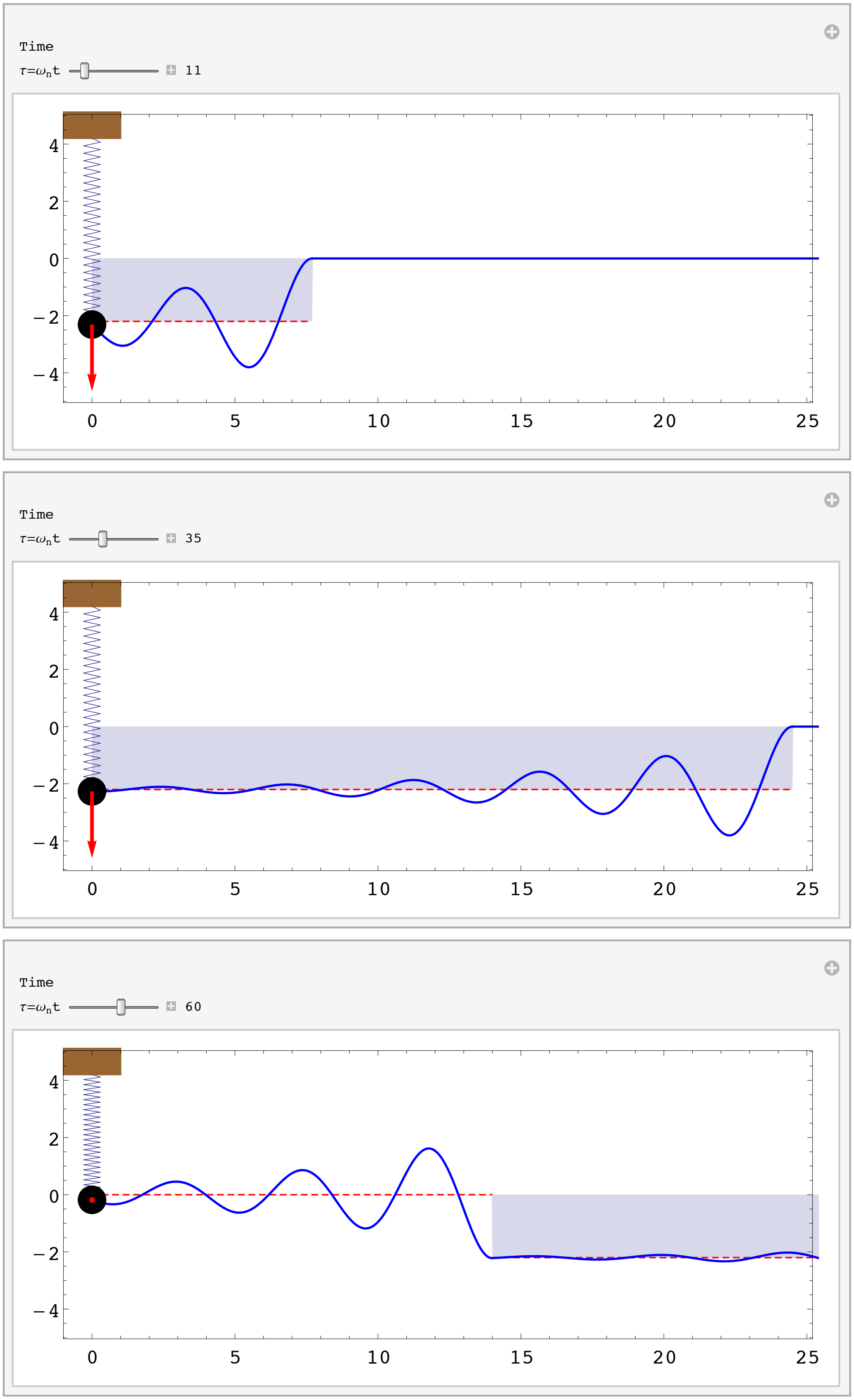

Figure 6: Three frames of animating video generated with the Mathematica file for constant-type external force. Upper: during applied force. Center: end of applying force. Bottom: free-damped vibrations

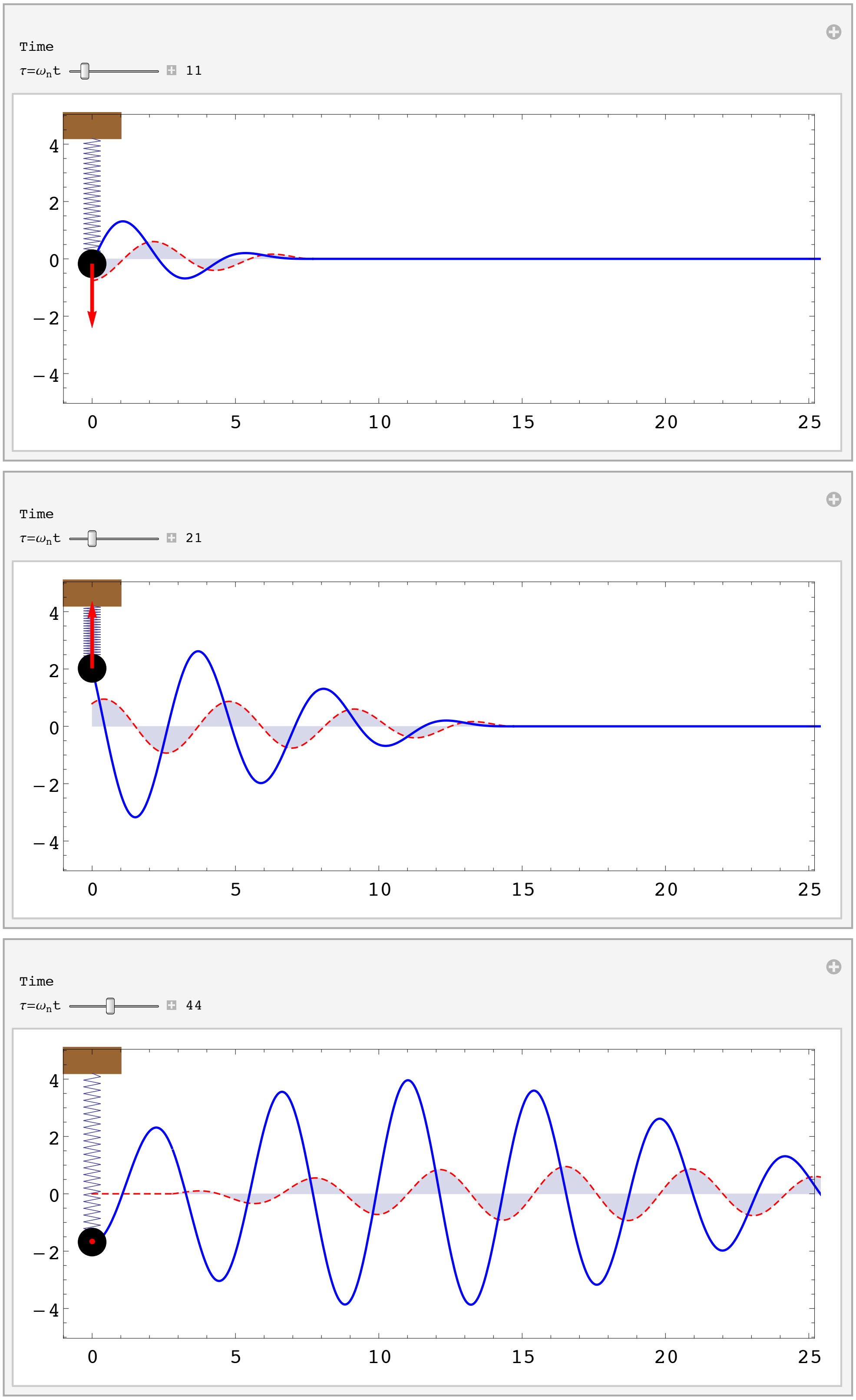

Figure 7: Three frames of animating video generated with the Mathematica file for harmonic-parabolic-type external force. Upper: force downwards. Center: force upwards. Bottom: free-damped vibrations

Modelling in Science Education and Learning
http://polipapers.upv.es/index.php/MSEL