

# Modelling the landing of a plane in a calculus lab

Antonio Morante and José A. Vallejo  
*Facultad de Ciencias, Universidad Autónoma de San Luis Potosí*  
*Lat. Av. Salvador Nava s/n, CP 78290*  
*San Luis Potosí (SLP), México*

August 16, 2011

## Abstract

We exhibit a simple model of a plane landing that involves only basic concepts of differential calculus, so it is suitable for a first year calculus lab. We use the computer algebra system Maxima and the interactive geometry software GeoGebra to do the computations and graphics.

Keywords: *Mathematical modelling; calculus labs; GeoGebra; CAS Maxima.*

## 1 Introduction

Usually the problems posed in freshman calculus courses are just formal in character and, therefore, not quite exciting. Students are not satisfied when we say that mathematics can be used in different fields of knowledge, they want to see math applied to real world situations!

Obviously, the problem is that most of real mathematical applications need the study of more advanced subjects than the ones offered at freshman level, but with a little bit of imagination is possible to adapt complex real problems to tractable case studies requiring a minimum of mathematical formalism. An example of this approach is Richard Barshinger's paper *How not to land at Lake Tahoe* (American Mathematical Monthly **99** n5 (1992) 453-455), of which we present an adaptation using Maxima and GeoGebra. The choice of this couple is dictated by two main reasons: their power and easiness of use, and the fact that they are free software, so the students can work at home with the same programs with which they learn at the classroom, at no cost. We include a brief appendix on the basic usage of both programs, to make the lab self-contained.

Before June 1965, when autolandings were first used in a commercial flight, there were several accidents of landing aircrafts, like that of Paradise Airlines Flight 901A, where a Constellation crashed with Genoa peak on approach to Lake Tahoe airport (see, for example, <http://aviation-safety.net/database/record.php?id=19640301-1>). Nowadays, even with good visual conditions, most of the medium/large range airliners are able to land by means of the so called instrument landing systems (or ILS for short). There is a well defined protocol to decide when to use these ILS, indeed, landings are classified taking into account several conditions related to visibility (see <http://www.kel.com/aero/pamila/category/three/>). In our classes (first calculus courses for mathematicians, physicists and electrical engineers), before doing this lab we suggest the students to look in the Internet for information regarding CAT III landings, which are the most difficult ones. Usually, they find the webpages we are referring to in this paper by themselves, and also several videos of actual landings taken by professional pilots (a particularly shocking one can be found here: [http://www.youtube.com/watch?v=Fe\\_pEK\\_kRVw](http://www.youtube.com/watch?v=Fe_pEK_kRVw)). Then, it is easy to motivate them by saying that in the next calculus lab we will see how the automated system can compute the trajectory of best approach!. We also promise the students to explain the landing process and the meaning of the expressions they can hear in those videos, to make things more interesting<sup>1</sup>.

As mentioned, all the computations in this lab are done with the aid of Maxima and Geogebra. This combination of research in the Internet, getting videos of real situations, and doing computations with software, has proven to be very stimulating for our students. We sincerely believe that inserting this kind of projects into the curriculum contributes to maintain the student's interest in calculus courses.

## 2 Background

Suppose we are modelling the landing of a plane. We would like to do this in such a way that, eventually, the model could be implemented in an ILS (see, for example, <http://en.wikipedia.org/wiki/Autoland>), so what we want is a step-by-step way to solve the problem. The model we will present is quite simple but includes the basic ideas of the process, with

---

<sup>1</sup>For instance, the words “retard, retard” (frequently cited in the forums, asking for their meaning) remind the pilot to put the throttles to idle; otherwise, the system will keep the plane along the runway at a speed too high to turn right or left, a manoeuvre commonly needed to reach the airport gates.

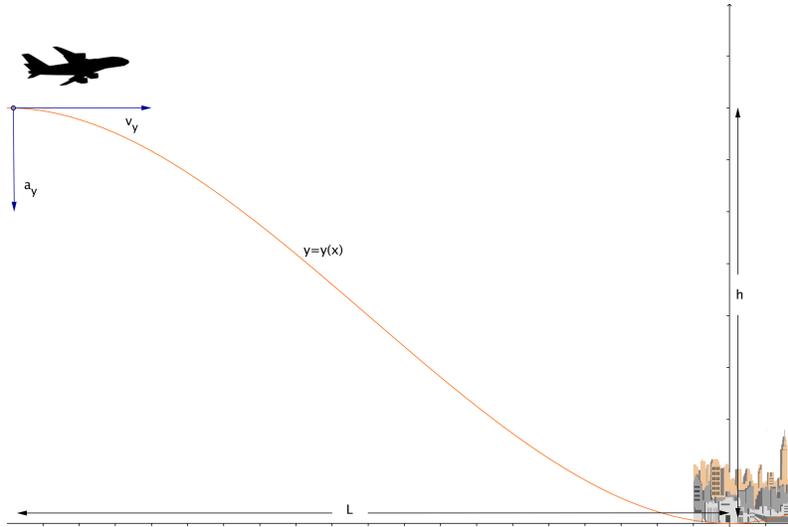


Figure 1: A graphical description of the problem.

the advantage of needing no more than freshman calculus. It is based on the wonderful exposition of the aforementioned Barshinger's paper given by E. W. Packel and S. Wagon (Animating calculus: Mathematica notebooks for the laboratory. W.H. Freeman, 1994), but using more modern resources, actual data from the Internet and free software instead of licensed (and expensive) products.

The problem is as follows: an aircraft is approaching its destination, with  $h$  the altitude above the airport. We want to know the most appropriate distance  $L$ , measured from the airport, to start descent and the trajectory (the approximation curve  $y = y(x)$ ) that the aircraft should follow for a smooth landing. Figure 1 illustrates the situation.

This is a quite complex problem, so we will make some assumptions in order to simplify it. We also need to take into account some physical constraints, in particular, the absolute value of the vertical component of the acceleration,  $a_y$ , must be lesser than some fixed constant  $k$ . This is to avoid the discomfort (motion sickness) that passengers feel when the aircraft suddenly ascents or descents. Fixing a coordinates system, with the origin corresponding to the beginning of the runway, the boundary conditions are:

- (a) The aircraft starts to descent at position  $(-L, h)$ , therefore,  $y(-L) = h$ .

- (b) The aircraft starts descent in horizontal position, so  $y'(-L) = 0$  (in other words, the aircraft is not making a “dive”).
- (c) The aircraft concludes landing in an horizontal position, then  $y'(0) = 0$ .
- (d) Finally, we consider that the landing is done when the aircraft touches down, that is,  $y(0) = 0$ .

These conditions impose by themselves some restrictions on the approximation curve  $y = y(x)$ . Note that the curve  $y = y(x)$  can not be a straight line  $y = mx + n$  or a parabola  $y = ax^2 + bx + c$ . Indeed, for the straight line conditions (b) and (c) are not fulfilled. For the parabola, either condition (b) or condition (c) fails.

As a consequence of this remark, the most simple polynomial curve that we can use to model the approximation path is a cubic one:  $y = ax^3 + bx^2 + cx + d$ .

### 3 The model

We are going to use the CAS Maxima to perform calculations in our model for an aircraft landing. We start by defining the basic equation, in terms of some parameters  $a$ ,  $b$ ,  $c$  and  $d$  to be determined:

```
(%i1) y:a*x^3+b*x^2+c*x+d;
```

```
(%o1)          a x3 + b x2 + c x + d
```

We need the derivative of this function, because of conditions (b) and (c):

```
(%i2) dy:diff(y,x);
```

```
(%o2)          3 a x2 + 2 b x + c
```

Now, we impose conditions (a), (b), (c) and (d), in order to determine the parameters for the curve  $y = y(x)$ :

```
(%i3) ec1:subst(x=-L,y)=h;
```

```
(%o3)          -a L3 + b L2 - c L + d = h
```

```
(%i4) ec2:subst(x=-L,dy)=0;
```

(%o4) 
$$3 a L^2 - 2 b L + c = 0$$

(%i5) `ec3:subst(x=0,dy)=0;`

(%o5) 
$$c = 0$$

(%i6) `ec4:subst(x=0,y)=0;`

(%o6) 
$$d = 0$$

(%i7) `solve([ec1,ec2,ec3,ec4],[a,b,c,d]);`

(%o7) 
$$\left[ \left[ a = \frac{2h}{L^3}, b = \frac{3h}{L^2}, c = 0, d = 0 \right] \right]$$

Replacing these values for  $a$ ,  $b$ ,  $c$  and  $d$  in equation (%o1), we obtain the approximation curve  $y = y(x)$ :

(%i8) `subst([a=2*h/L^3,b=3*h/L^2,c=0,d=0],y);`

(%o8) 
$$\frac{3 h x^2}{L^2} + \frac{2 h x^3}{L^3}$$

The  $(x, y)$  coordinates of the aircraft vary with time. Therefore we define the function (implicitly depending on time):

(%i9) `y(x):=(3*h*x(t)^2)/L^2+(2*h*x(t)^3)/L^3;`

(%o9) 
$$y(x) := \frac{3 h x(t)^2}{L^2} + \frac{2 h x(t)^3}{L^3}$$

We assume that the aircraft reduces its horizontal velocity along the descent (in fact, this is done in practice). In order to keep calculations more or less simple we suppose that the aircraft horizontal speed depends linearly on time; the aircraft starts the descent with horizontal velocity  $u_0$  and touches down with horizontal velocity  $u_1$ , where  $u_1 < u_0$ . If the descent last  $T$  seconds, the equation for the horizontal velocity is,

(%i10) `u(t):=u[0]-t*(u[0]-u[1])/T;`

(%o10) 
$$u(t) := u_0 - \frac{t(u_0 - u_1)}{T}$$

Integrating this equation, we get the aircraft horizontal displacement formula:

(%i11) `integrate(u(t),t);`

(%o11) 
$$u_0 t - \frac{(u_0 - u_1) t^2}{2T}$$

This define a new function, but we don't call it  $x(t)$  since is not yet the equation we need (some suitable substitutions are going to be done), we call it  $X(t)$ :

(%i12) `X(t):=u[0]*t-((u[0]-u[1])*t^2)/(2*T)+c;`

(%o12) 
$$X(t) := u_0 t - \frac{(u_0 - u_1) t^2}{2T} + c$$

We added an integration constant  $c$  (Maxima *does not* include integration constants when the command `integrate` is used). When the descent starts, the horizontal position of the aircraft relative to the airport is  $-L$  (this determines  $c$ ):

(%i13) `solve(X(0)=-L,c);`

(%o13) 
$$[c = -L]$$

(%i14) `X(t):=u[0]*t-((u[0]-u[1])*t^2)/(2*T)-L;`

(%o14) 
$$X(t) := u_0 t - \frac{(u_0 - u_1) t^2}{2T} - L$$

And at the instant  $T$ , the aircraft touches the runway at the origin of coordinates:

(%i15) `solve(X(T)=0,T);`

(%o15) 
$$[T = \frac{2L}{u_1 + u_0}]$$

Finally, we substitute this last condition in  $X(t)$  to obtain the aircraft horizontal displacement formula  $x(t)$  as an explicit function of time:

(%i16)  $x(t) := \text{subst}(\%o15, X(t));$

(%o16)  $x(t) := \text{subst}(\%o15, X(t))$

(%i17)  $x(t);$

(%o17) 
$$-L - \frac{(u_0 - u_1)(u_1 + u_0)}{4L} t^2 + u_0 t$$

Thus, the explicit equation for the approximation curve  $y = y(x)$  as function of time  $t$  is:

(%i18)  $y(x);$

(%o18)

$$\frac{3h \left( -L - \frac{(u_0 - u_1)(u_1 + u_0)}{4L} t^2 + u_0 t \right)^2}{L^2} + \frac{2h \left( -L - \frac{(u_0 - u_1)(u_1 + u_0)}{4L} t^2 + u_0 t \right)^3}{L^3}$$

The aircraft's vertical velocity  $v_y$  is given by the derivative  $\frac{dy}{dt}$ , and the vertical acceleration  $a_y$  is given by the second derivative  $\frac{d^2y}{dt^2}$ . Let us compute these:

(%i19)  $v[y] : \text{radcan}(\text{diff}(y(x), t));$

(%o19)

$$\begin{aligned} & -(96 u_0^2 h t L^4 + (72 u_0 u_1^2 - 168 u_0^3) h t^2 L^3 \\ & + (12 u_1^4 - 120 u_0^2 u_1^2 + 108 u_0^4) h t^3 L^2 \\ & + (-30 u_0 u_1^4 + 60 u_0^3 u_1^2 - 30 u_0^5) h t^4 L \\ & + (-3 u_1^6 + 9 u_0^2 u_1^4 - 9 u_0^4 u_1^2 + 3 u_0^6) h t^5) / (16 L^6) \end{aligned}$$

(%i20)  $a[y] : \text{radcan}(\text{diff}(y(x), t, 2));$

(%o20)

$$\begin{aligned} & -(96 u_0^2 h L^4 + (144 u_0 u_1^2 - 336 u_0^3) h t L^3 \\ & + (36 u_1^4 - 360 u_0^2 u_1^2 + 324 u_0^4) h t^2 L^2 \\ & + (-120 u_0 u_1^4 + 240 u_0^3 u_1^2 - 120 u_0^5) h t^3 L \\ & + (-15 u_1^6 + 45 u_0^2 u_1^4 - 45 u_0^4 u_1^2 + 15 u_0^6) h t^4) / (16 L^6) \end{aligned}$$

At this moment it seems that our problem has become quite complicated to be worked out “by hand”, but with Maxima to perform the previous calculations is a simple task. We will find the extreme values of vertical acceleration. To determine the maximum and minimum of  $a_y$  we equal its derivative to 0, solving for  $t$ :

```
(%i21) solve(diff(%,t)=0,t);
(%o21)
```

$$\left[ t = -\frac{\sqrt{30} \sqrt{u_1^2 + u_0^2} L + 10 u_0 L}{5 u_1^2 - 5 u_0^2}, \right. \\ \left. t = \frac{\sqrt{30} \sqrt{u_1^2 + u_0^2} L - 10 u_0 L}{5 u_1^2 - 5 u_0^2}, \right. \\ \left. t = \frac{2 u_0 L}{u_0^2 - u_1^2} \right]$$

(After this calculation the reader should be sure about how useful a CAS like Maxima can be at the classroom).

In order to have an idea about when there is a local maximum or a local minimum for  $a_y$ , we use GeoGebra to make an active graph of the function  $a_y$  in the time interval  $[0, T]$ , using parameters  $u_0$ ,  $u_1$ ,  $h$  and  $L$  varying between standard actual values (see Figure 2).

In the interval  $[0, T]$ , the vertical acceleration  $a_y$  has an absolute maximum for  $t = (\sqrt{30} \sqrt{u_1^2 + u_0^2} L - 10 u_0 L) / (5 u_1^2 - 5 u_0^2)$  and an absolute minimum for  $t = 0$ . We can explicitly compute these extreme values:

```
(%i22) a[ymin]:subst(t=0,a[y]);
```

```
(%o22)
```

$$-\frac{6 u_0^2 h}{L^2}$$

and

```
(%i23) a[ymax]:radcan(subst(t=(sqrt(30)*sqrt(u[1]^2+u[0]^2)*L-10*u[0]*L)/(5*u[1]^2-5*u[0]^2),a[y]));
```

```
(%o23)
```

$$-\frac{(27 u_1^4 - 6 u_0^2 u_1^2 + 27 u_0^4) h}{(20 u_1^2 - 20 u_0^2) L^2}$$

For  $u_1 < (4/5) u_0$ , a restriction which includes most of the acceptable values for horizontal velocities  $u_0$  and  $u_1$ , the maximum of the magnitude of the vertical acceleration,  $|a_y|$ , is reached for  $t = 0$ , that is, when the aircraft starts the descent (which is very reasonable, physically).

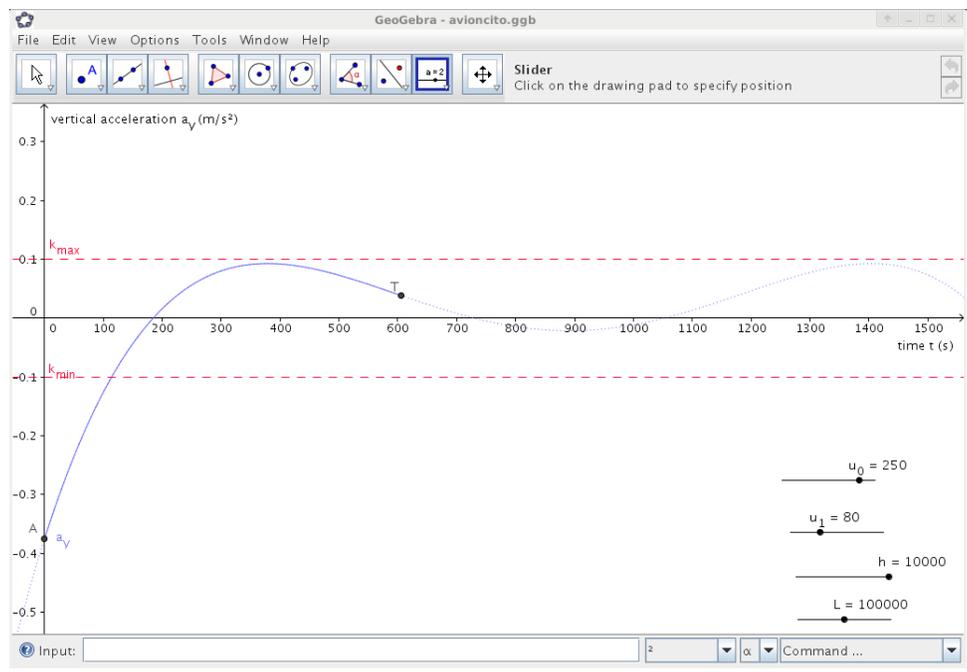


Figure 2: Determination of extrema of the vertical acceleration.

```
(%i24) subst(u[1]=(4/5)*u[0],radcan(-a[ymin]-a[ymax])) > 0;
```

```
(%o24) 
$$\frac{1871 u_0^2 h}{1500 L^2} > 0$$

```

For an acceptable comfort level, the modulus of vertical accelerations must be lesser than some constant  $k$  (see, for example, <http://ftp.rta.nato.int/public/FullText/RT0/MP/RT0-MP-036/MP-036-17.pdf>), so we impose an upper bound for the magnitude of the maximum vertical acceleration:

```
(%i25) -a[ymin] < k;
```

```
(%o25) 
$$\frac{6 u_0^2 h}{L^2} < k$$

```

and solve for the distance  $L$  at which the descent should start:

```
(%i26) L>sqrt((6*u[0]^2*h)/k);
```

```
(%o26) 
$$L > \sqrt{6} u_0 \sqrt{\frac{h}{k}}$$

```

In a transcontinental flight (with an Airbus 340, for example), its cruise speed is about 900km/h (that is, 250m/s) with a cruise altitude of about 10000m (see [http://en.wikipedia.org/wiki/Airbus\\_A340#Specifications](http://en.wikipedia.org/wiki/Airbus_A340#Specifications)). Note that we have rounded the actual values). Taking  $k = 0.1\text{m/s}^2$ , the distance from the aircraft to the airport to start descent should be:

```
(%i27) float(subst([u[0]=250,h=10000,k=0.1],%));
```

```
(%o27) 
$$L > 193649.1673103709$$

```

That is, about 190km from the airport.

The practical procedure for computing the trajectory followed by the pilots in actual flights, is called the “3-degree glide slope”, that is: The trajectory must approximate as close as possible the ideal straight line leading to the airport with a declination angle of 3 degrees (see, for example, <http://www.navfltsm.addr.com/gs.htm> or <http://www.ohio.edu/avionics/research/nav/ils/glideslope.cfm>). Using the previous calculation,  $h = 10000\text{m}$  and  $L = 193649\text{m}$ , we get a declination angle,

$$\theta = \tan^{-1} \left( \frac{h}{L} \right) \simeq 2.96^\circ$$

that is very close to the empirical value.

## 4 Special cases

Singular cases appear for airports surrounded by mountains. In this case the most important parameter is not the distance to start the descent, because an obstacle must be avoided first. What we need to determine is the aircraft speed at the moment that descent begins, just after passing over the mountains. Therefore, in equation (25) we solve for  $u_0$ :

$$(28) \quad u_0 < \sqrt{(k \cdot L^2) / (6 \cdot h)};$$

$$(28) \quad u_0 < \frac{\sqrt{\frac{k}{h}} |L|}{\sqrt{6}}$$

### 4.1 Mexico City international airport

We consider the above calculations using as example Mexico City International airport, with an elevation about 2200m above the mean sea level (amsl), and about 70km away from the Popocatepetl volcano (about 5400m amsl):

$$(29) \quad \text{float}(\text{subst}([k=0.1, h=5400-2200, L=70000], 29));$$

$$(29) \quad u_0 < 159.7524126056735$$

This is the aircraft speed in m/s. Converting to km/h:

$$(30) \quad u_0 = 160 \cdot 0.001 \cdot 3600;$$

$$(30) \quad u_0 = 576.6$$

The aircraft's speed needs to be reduced to about 2/3 of its cruise speed just before starting the descent. In a Madrid-Mexico City transcontinental flight driven by Iberia, for example, the screens in the cabin show information like time to destination, estimated arrival time, distance to destination and the aircraft speed with respect to the ground (see Figure 3). In the experience of one of the authors, who makes this trip frequently (JAVR), the data for a standard flight when approaching Mexico City International Airport from Puebla (which is near Popocatepetl volcano), show a relative good approximation with the values obtained here.



Figure 3: Flight data as seen in the passenger's cabin.

## 4.2 Lake Tahoe airport

For the Lake Tahoe airport, which is located at an elevation about 1910m amsl and about 18km away from the Genoa peak (2790m amsl), the previous calculations give an initial horizontal speed to start the descent of:

```
(%i31) float(subst([k=0.1,h=2790-1910,L=18000],%o28));
```

```
(%o31)  $u_0 < 78.33494518006403$ 
```

Again, this is the speed in m/s. Converting to km/h:

```
(%i32) u[0]=78*0.001*3600;
```

```
(%o32)  $u_0 = 280.8$ 
```

Thus, the aircraft's speed needs to be reduced to about 1/3 of its cruise speed just before starting the descent. This result, of course, was to be expected since the plane has less space for descending.

## 5 Concluding remarks

Calculus teachers can suggest to their students interesting case studies based on realistic problems, some of them requiring only basic techniques or results. The use of software can help with complicated and boring calculations but, most important, can attract the attention of students. A standard math project, like the one presented here, can integrate: Searching information in the world wide web, downloading videos, and performing calculations or graphing functions with the help of suitable software (Maxima and GeoGebra being good examples for that). The experience of using this approach in our classes is quite rewarding. In addition to this, we believe that modelling real problems that involve actual data and require the use of computational tools, should be part of the training of any math student.

## Acknowledgements

JAVR benefited from useful discussions with Dagoberto Salazar on topics related to air navigation, when he was a lecturer at the Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels EETAC (Aerospatial and Telecommunications Engineering College, UPC, Spain); also, thanks are due to the National Council of Science and Technology (CONACyT, Mexico), for the grant CB-JB2-78791, which partially supported this project. Finally, both authors express their gratitude to Fabrizio Fresia, who took the photograph appearing in Figure 3 and shared it with us.

## Appendix A: Introduction to Maxima

In this Appendix, we give a very brief introduction to Maxima. Of course, the program has many more possibilities than those listed here. We restrict ourselves to explaining the commands used in the paper.

Maxima is a Computer Algebra System (CAS). It has several graphical interfaces, and we will assume that the reader is using wxMaxima (available at <http://andrejv.github.com/wxmaxima/>), which is very similar to that of the commercial package Mathematica<sup>®</sup> by Wolfram. When you start a Maxima session you are presented a blank document and you can start by directly typing commands (see figure 4). For example, you type “2+2”, make Shift+Enter and Maxima returns the answer:

```
(%i1) 2+2;
```

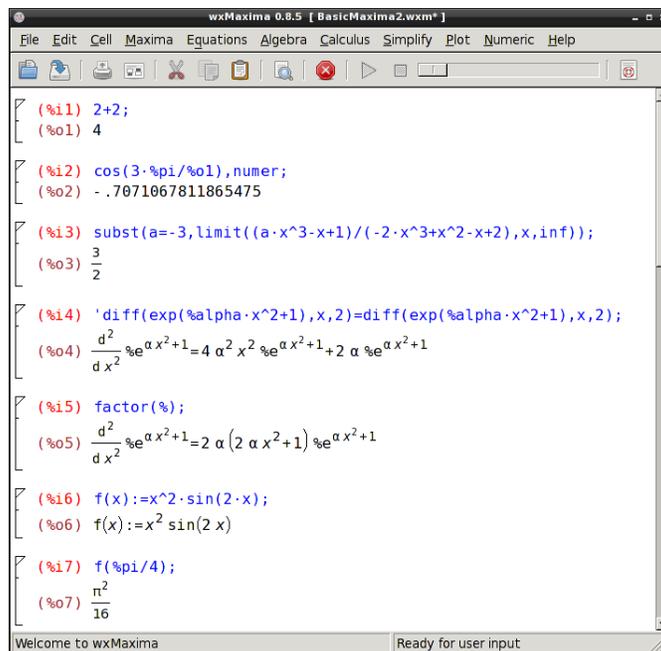


Figure 4: A wxMaxima session.

(%o1) 4

Note that to every input, an indicator like %i1 is assigned. The same is done with the output. This makes very easy to refer to previous computations. For example, we can now evaluate the cosine of  $\frac{3\pi}{4}$  by concatenating (piping) commands. Note also that the constant  $\pi$  is written %pi (other built-in constants are %i, %e):

(%i2) `cos(3*%pi/%o1),numer;`

(%o2) `- 0.70710678118655`

Another way to concatenate commands is by composition; this is closer to the standard mathematical notation. In the following example we first compute the limit of  $\frac{ax^3-x+1}{-2x^3+x^2-x+2}$  when  $x \rightarrow +\infty$  (through the command `limit(expression,variable,value)`) and then evaluate the result ( $-\frac{a}{2}$ ) when  $a = -3$  by making a substitution (this is done with the command `subst(variable=value,expression)`):

```
(%i3) subst(a=-3,limit((a*x^3-x+1)/(-2*x^3+x^2-x+2),x,inf));
```

```
(%o3) 3/2
```

This example shows that Maxima knows how to do limits. It is no surprise that it also knows how to do derivatives (with the command `diff(function,variable,order of the derivative)`), even of a function depending on parameters ( $\alpha$  in this case; note the way to write greek symbols in wxMaxima):

```
(%i4) 'diff(exp(%alpha*x^2+1),x,2)=diff(exp(%alpha*x^2+1),x,2);
```

```
(%o4)  $\frac{d^2}{dx^2} e^{\alpha x^2+1} = 4\alpha^2 x^2 e^{\alpha x^2+1} + 2\alpha e^{\alpha x^2+1}$ 
```

The apostrophe ' in front of a command prevents evaluation, so in the left hand side we have the derivative in formal notation. We can simplify the result by grouping factors with the command `factor` (the % refers to the last output):

```
(%i5) factor(%);
```

```
(%o5)  $\frac{d^2}{dx^2} e^{\alpha x^2+1} = 2\alpha (2\alpha x^2 + 1) e^{\alpha x^2+1}$ 
```

Functions are defined exactly with the same notation we use in the blackboard:

```
(%i6) f(x):=x^2*sin(2*x);
```

```
(%o6)  $f(x) := x^2 \sin(2x)$ 
```

We can evaluate a function on a certain value of its argument in a straightforward manner:

```
(%i7) f(%pi/4);
```

```
(%o7)  $\frac{\pi^2}{16}$ 
```

Once we have a function defined, we can do any computation on it. For example, we can integrate it:

```
(%i8) integrate(f(x),x);
```

$$(\%o8) \quad \frac{4x \sin(2x) + (2 - 4x^2) \cos(2x)}{8}$$

Or integrate over a certain interval, in this case the  $[0, 1]$ :

(%i9) `integrate(f(x),x,0,%pi/2);`

$$(\%o9) \quad \frac{\pi^2 - 2}{8} - \frac{1}{4}$$

Another possibility to simplify expressions, particularly rational ones, is the `ratsimp` command:

(%i10) `ratsimp(%);`

$$(\%o10) \quad \frac{\pi^2 - 4}{8}$$

Functions can have subscripts:

(%i11) `f[x](x):='diff(f(x),x);`

$$(\%o11) \quad f_x(x) := \frac{d}{dx} f(x)$$

In this example we get the derivative:

(%i12) `f[x](x);`

$$(\%o12) \quad \frac{d}{dx} (x^2 \sin(2x))$$

To finish this brief introduction, let us indicate the use of the `solve` command to solve systems of equations. We first define the equations:

(%i13) `eq1:a*x+2*y=3;`

$$(\%o13) \quad 2y + ax = 3$$

(%i14) `eq2:-x+y=1;`

$$(\%o14) \quad y - x = 1$$

And then solve the system `(eq1,eq2)` for `(x,y)` as the indeterminates. Note that the equations and the indeterminates are passed to Maxima in the form of lists, enclosed in squared brackets; the solution is given as a list, also:

(%i15) `solve([eq1,eq2],[x,y]);`

$$(\%o15) \quad \left[ \left[ x = \frac{1}{a+2}, y = \frac{a+3}{a+2} \right] \right]$$

## Appendix B: Introduction to Geogebra

GeoGebra is an interactive geometry software oriented to be used in high school math courses, and it was developed by M. Hohenwarter, M. Borchers and Y. Kreis. GeoGebra is a free, multilanguage and multiplatform software, it can be obtained at <http://www.geogebra.org> (in this paper we have used version 3.2.0.0).

In general, to each algebraic object in GeoGebra, it corresponds a geometric object and vice versa. These objects can be constructed with a click in the appropriate icon, or writing the corresponding command in the input window. In order to construct the active graph presented in Section 3, you can perform the following steps (see Figure 5):

1. Introduce the parameter  $u_0$ , selecting the **Slider** tool and making click on the drawing pad. In the contextual menu that appears, define the name, min, max and increment values for the parameter. Repeat this procedure to define the parameters  $u_1, h$  and  $L$ .
2. In the input window assign the value for the modulus of the vertical acceleration threshold (here we used  $k = 0.1$ ).
3. In the input window define the function  $a(x)$  with the following command:

$$a(x) = -(96*u_0^2*h*L^4 + (144*u_0*u_1^2-336*u_0^3)*h*L^3*x) + (36*u_1^4-360*u_0^2*u_1^2+324*u_0^4)*h*L^2*x^2 + (-120*u_0*u_1^4+240*u_0^3*u_1^2-120*u_0^5)*h*L*x^3 + (-15*u_1^6+45*u_0^2*u_1^4-45*u_0^4*u_1^2+15*u_0^6)*h*x^4) / (16*L^6)$$

4. Restrict the previous function to the interval  $[0, 2L/(u_0 + u_1)]$  with the command

```
Function[a(x),0,2*L/(u_0+u_1)]
```

5. Plot the point  $A$  at the left extreme of the graph of  $a_y$  with the command

```
A = (0, a(0))
```

6. Plot the point  $T$  at the right extreme of the graph of  $a_y$  with the command

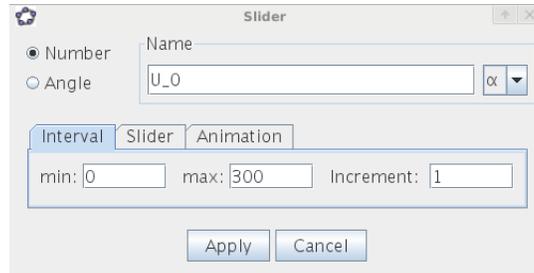


Figure 5: Slider menu in GeoGebra.

$$A = (2*L/(u_0+u_1), a(2*L/(u_0+u_1)))$$

7. Finally, plot the limits for the maximum and minimum vertical acceleration

$$y = k \quad \text{and} \quad y = -k$$

Once the construction is done, it is possible to change dynamically the value of any parameter by selecting the **Move** tool and dragging the corresponding slider button.