Maplesoft
Mathematics • Modeling • Simulation
Application Demonstration
www.maplesoft.com

Classroom Tips and Techniques: Sliders for Parameter-Dependent Curves

Robert J. Lopez
Emeritus Professor of Mathematics and Maple Fellow
Maplesoft

Introduction

The polar curve $r = b + a\cos(\theta)$ is called a limaçon except when $b = a$, in which case it is called a cardioid. (To be exceedingly precise, the limaçon becomes a trisectrix when $b = a/2$.) Since both the limaçon and the cardioid appear regularly as examples of polar curves in the typical integral calculus course, I thought it would be useful, if not interesting, to graph the limaçon with sliders controlling the values of the parameters a and b. This month's article shows several methods for producing such graphs.

Many articles on the internet provide definitions for both the limaçon and the cardioid, and also point out that the limaçon is the *conchoid* of a circle with respect to a point on the circle. (The "ch" in *conchoid* is hard, pronounced as "k" because the conchoid of a line resembles the shape of a conch (pronounced "kongk"), a spirally shaped seashell.)

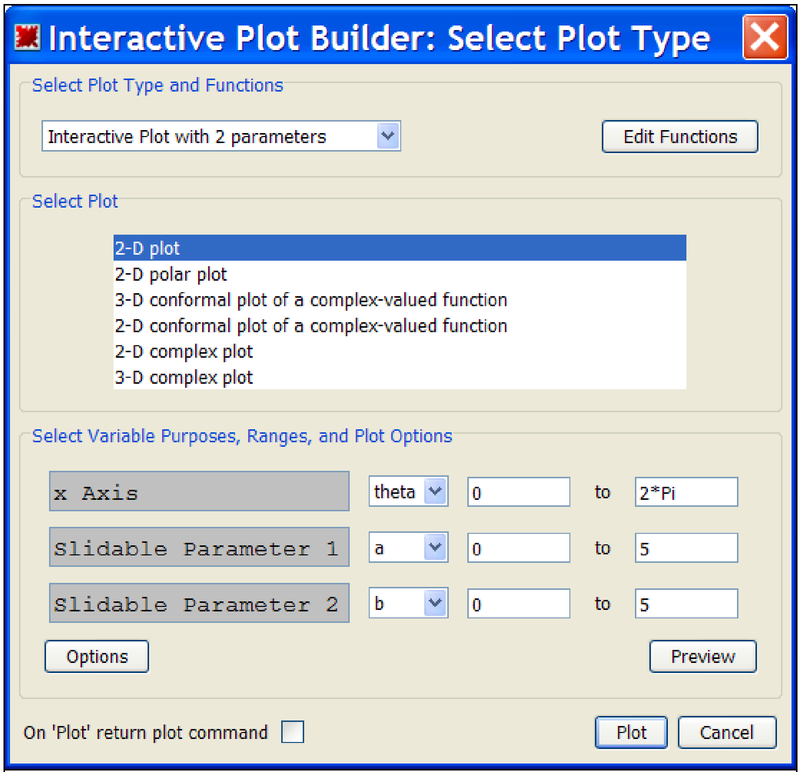
The conchoid of a curve C with respect to a point P is not an easy curve to visualize. A line L through P is drawn so it intersects C in point Q. The locus of points P_1 and P_2 on L for which $|QP_1| = |QP_2| = k$, where k is a constant, is a conchoid of C with respect to P.

To visualize a conchoid, a line of intersection, the point of intersection, and two points equidistant from this point of intersection all have to be imagined. Of course, this calls for graphing with sliders! So this month's article also explores ways to visualize a conchoid.

The Limaçon

The Plot Builder

In math mode, type $b + a\cos(\theta)$ and bring up the Plot Builder from the Context Menu. In the section "Select Plot Type and Functions" choose "Interactive Plot with 2 parameters." (See Figure 1.) In the "select Plot" section, the Plot Builder defaults to "2-D plot" but in the "Select Variable Purposes, Ranges, and Plot Options" section, adjust as per Figure 1.


Interactive Plot Builder: Select Plot Type
Select Plot Type and Functions
Edit Functions
Select Plot
Select Variable Purposes, Ranges, and Plot Options
Options
Preview
Plot
Cancel

Interactive Plot with 2 parameters

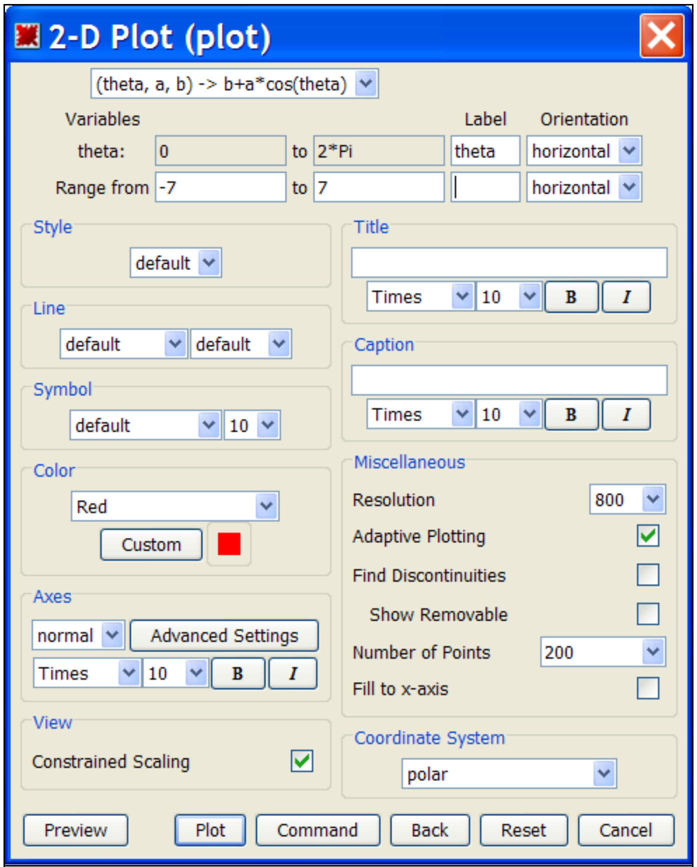
- 2-D plot
- 2-D polar plot
- 3-D conformal plot of a complex-valued function
- 2-D conformal plot of a complex-valued function
- 2-D complex plot
- 3-D complex plot

x Axis	theta	0	to	2*Pi
Slidable Parameter 1	a	0	to	5
Slidable Parameter 2	b	0	to	5

On 'Plot' return plot command □

Figure 1 First interactive panel in the Plot Builder

Click on the Options button and in the ensuing dialog, Adjust the "Range from - to" options as shown in Figure 2. Also, check the box for Constrained Scaling, and in the section "Coordinate System", select "polar". Then click either the Preview or Plot buttons to obtain the 2-slider graph shown in Figure 3.



2-D Plot (plot)

(theta, a, b) -> b+a*cos(theta)

Variables				Label	Orientation
theta:	0	to	2*Pi	theta	horizontal
Range from	-7	to	7		horizontal

Style: default

Line: default, default

Symbol: default, 10

Color: Red

Axes: normal; Times, 10

View: Constrained Scaling ☑

Title: Times, 10

Caption: Times, 10

Miscellaneous:

Resolution 800

Adaptive Plotting ☑

Find Discontinuities ☐

Show Removable ☐

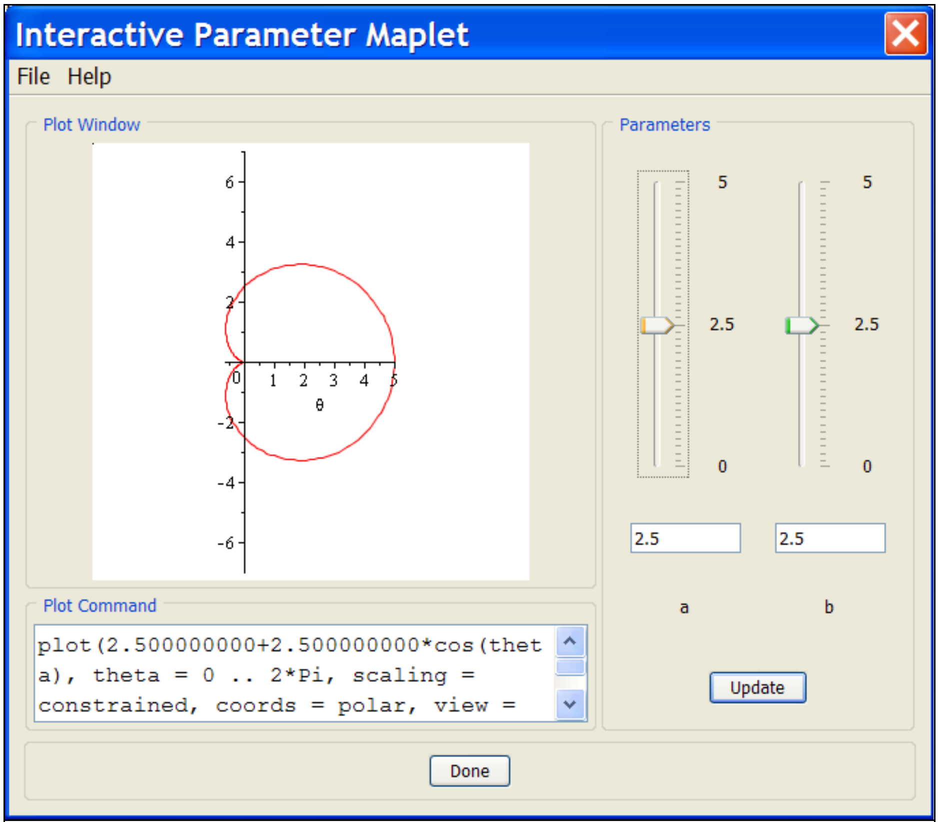
Number of Points 200

Fill to x-axis ☐

Coordinate System: polar

Figure 2 The Options panel in the Plot Builder

The 2-slider graph produced by the Plot Builder is shown in Figure 3. In exchange for the simplicity with which this tool was built, there are two shortcomings. First, it is a Maplet, so it cannot be saved or stored, and as long as it's "up," there is no access to the underlying copy of Maple. Second, as the sliders are moved, it will become apparent that the "Range from - to" setting applies only to the vertical axis. As the horizontal width of the curve changes, so too does the horizontal axis. It is just not possible to fix both the horizontal and vertical axes of this graph using just the Plot Builder.



```
plot(2.500000000+2.500000000*cos(thet
a), theta = 0 .. 2*Pi, scaling =
constrained, coords = polar, view =
```

a: 2.5

b: 2.5

Figure 3 Interactive Parameter Maplet generated by the Plot Builder

Note the plot command in the "Plot Command" window. This command draws a single curve according to the values of a and b set by the sliders. In fact, if the "Done" button is clicked, a single curve such as shown in Figure 4 results. As has already been pointed out, there is no way to "save" the 2-slider tool constructed via the Interactive Parameter Maplet.

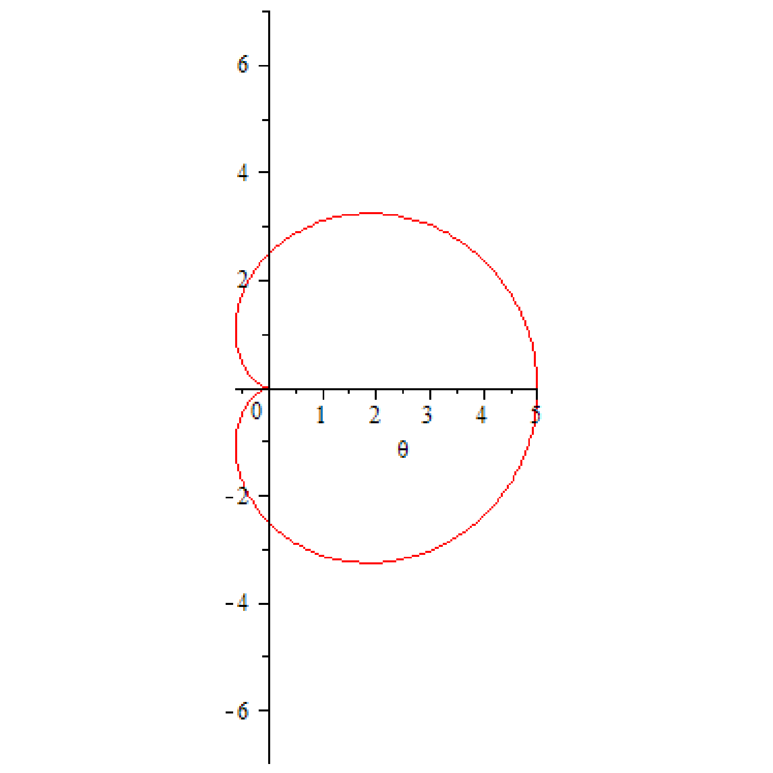


Figure 4 Single curve obtained from the Interactive Parameter Maplet

If, instead of returning the Interactive Parameter Maplet, the command

$$plots_{interactiveparams}('plot', [b + a\cos(\theta), \theta = 0..2\pi, scaling = constrained, coords = \text{polar}, view = [DEFAULT, -7..7], labels = [\theta, \text{""}]], [\,], a = 0..5, b = 0..5)$$

is returned. (Either select "On 'Plot" return plot command" - see Figure 1, or press the Command button in the Options panel - see Figure 2.) This suggests examining the **interactiveparams** command in the *plots* package.

The interactiveparams Command

With a slight modification, the **interactiveparams** command in the *plots* package will generate an improved 2-slider graph that bounds the horizontal range as well as the vertical.

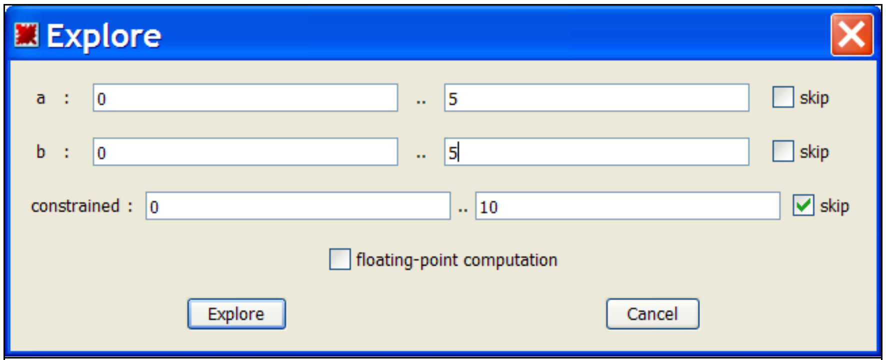
$$plots{:-}interactiveparams(plot, [b + a\cos(\theta), \theta = 0..2\pi, coords = \text{polar}, scaling = constrained, view = [-5..10, -7..7]], a = 0..5, b = 0..5)$$

Unfortunately, no matter how the Interactive Parameter Maplet is launched, it is still a Maplet, and has the shortcoming of disappearing once it is closed.

The Explore Command

The **Explore** command can be applied to a complete graphing command either as a command, or as an option selected from the Context Menu. The result is a slider-controlled graph resident in a separate worksheet. For example, the following command will first generate the dialog shown in Figure 5, then, in a new worksheet, create the application shown in Figure 6.

$$Explore\left(plot\left(b + a\cos(\theta), \theta = 0..2\pi, coords = \mathrm{polar}, scaling = constrained, view = [-5..10, -7..7]\right)\right)$$



a : 0 .. 5 □ skip

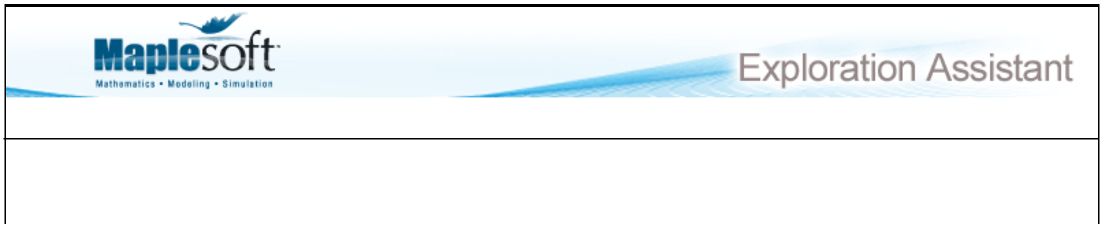
b : 0 .. 5 □ skip

constrained : 0 .. 10 ☑ skip

□ floating-point computation

Figure 5 Dialog for the **Explore** command

Notice that in the dialog Maple has been instructed to skip creating a slider for "constrained." (Obviously, some of the options to the **plot** command are interpreted as variables.) The default ranges of 0 to 10 have been modified. Pressing the "Explore" button generates (in a new worksheet) the application pictured in Figure 6.



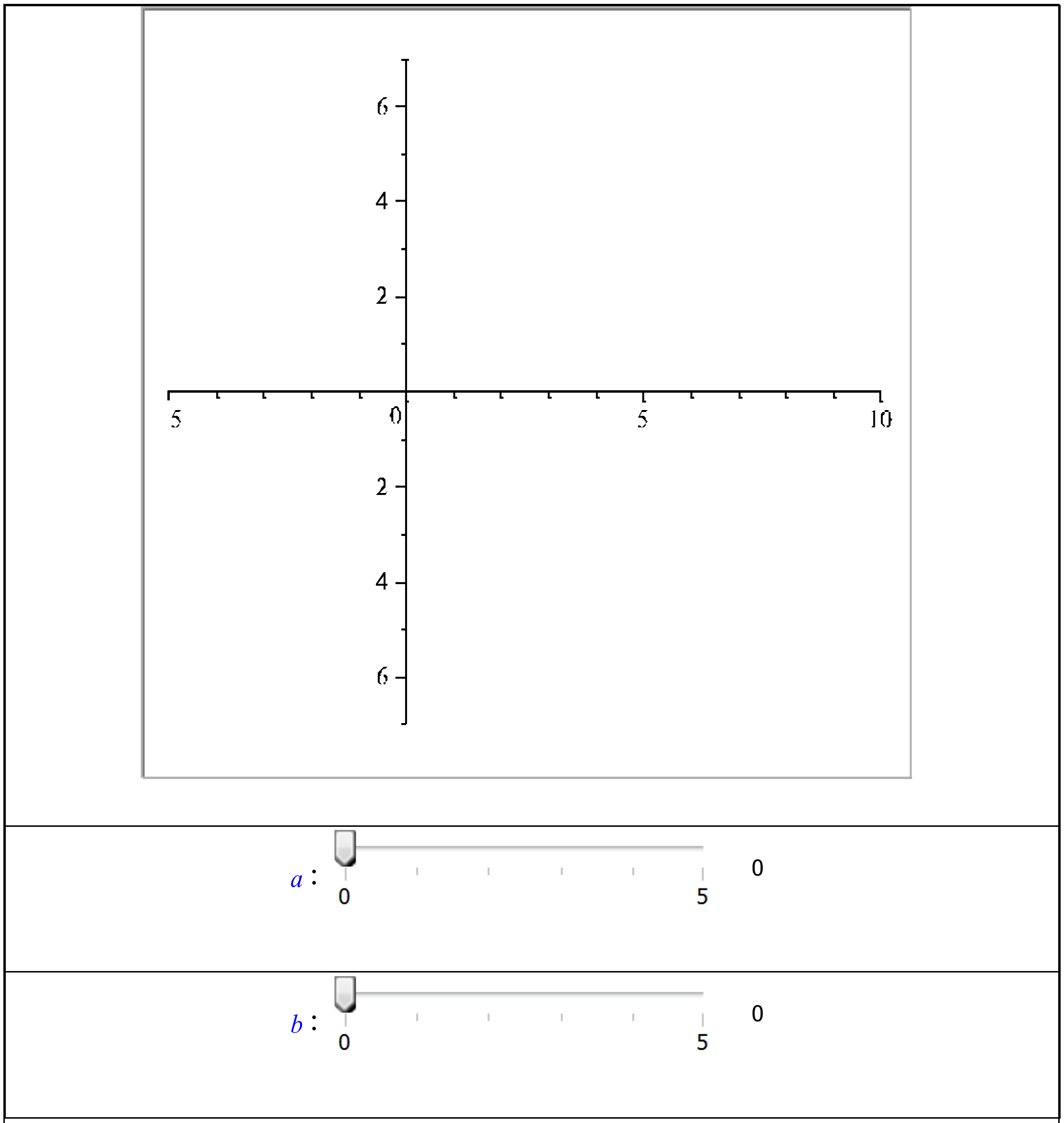
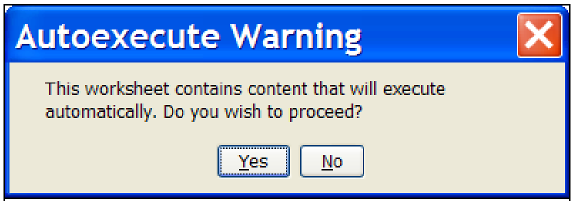


Figure 6 Slider-controlled graph produced by the **Explore** command

The great advantage of the application produced by the **Explore** command is that it consists of Embedded Components, so is not subject to the major defect of the Interactive Parameter Maplet: it can be made to "persist." However, there is a trick to this.

It is not enough to copy and paste the application created by the Explore command. When the application is inserted into a new Maple worksheet, code written by the command is executed. This code is what "programs" the sliders. Copying and pasting the application back into the launching worksheet copies this hidden code, but does not re-execute it. Hence, the pasted application will not function.

To re-execute this hidden code for the on-going session, select Execute≻Repeat Autoexecute from the Edit menu. The Autoexecute Warning dialog in Figure 7 will appear, a dialog magically suppressed when the application is first created. Clicking "Yes" will execute the hidden code and allow the pasted application to function. Autoexecute for this worksheet has been suppressed by selecting Autoexecute≻Clear All from the Format menu.)



This worksheet contains content that will execute automatically. Do you wish to proceed?

Figure 7 Autoexecute warning dialog

But where is this code hidden? It's in *autoexecute regions*, regions that can be found if the "Markers" option has been selected in the View menu. This opens a column on the left of the workspace where opposing triangles delineate Document Blocks. The Document Block corresponding to the blank line beneath the header (where it says "Maplesoft" and "Exploration Assistant") contains the bulk of the hidden code. Place the cursor on the Marker column at the location of this Document Block, and right-click. In the pop-up, select the option "Expand Document Block" to see the code. Of course, going back to the Marker column and selecting "Collapse Document Block" will again hide this code. There is also autoexecute code in the Document Blocks containing the two sliders. This code simply prints the slider labels a and b.

Right-clicking on the Marker for a Document Block that contains autoexecute code provides the Autoexecute option. Selecting the "Set" option and saving the worksheet means that when the worksheet is relaunched, the Autoexecute dialog shown in Figure 7 will appear. The Autoexecute option for each of the three Autoexecute regions inherited from the pasted Explore worksheet have been set to "Clear" so that the Autoexecute dialog does not appear upon the initial launch of this worksheet.

The astute reader will realize that the Explore functionality is an efficient way to generate a variety of applications controlled by sliders. Simply create the application with the **Explore**command (or select it from the Context Menu), copy/paste the application back into the launching worksheet, and activate its code by selecting Execute≻Repeat Autoexecute from the Edit menu.

Using the Components Palette

Of course, the 2-slider graph generated with the **Explore** command could be built interactively by inserting and coding the appropriate components from the Components palette. One advantage of such a construction would be a bit more flexibility over what is to be graphed. Simply add a MathContainer into which a function can be entered, and code the sliders to read this component for its content. However, building a robust graphing application means providing access to a multitude of plotting options. Just look at the Options panel in the Plot Builder to see just how many options have to be addressed. Each option requires at least one component, and each such component takes up space on the screen. The Plot Builder, which uses Maplet technology, handles the space constraint with secondary Maplets. This option is not yet available for embedded components, so screen space becomes the limiting factor in the design and construction of applications based on this technology.

For these reasons, we pass over the option of building a 2-slider graph with the Component palette, and proceed to an investigation of the conchoid of a curve.

The Conchoid of a Curve

Almost every web site that I looked at for a discussion of the conchoid of a curve referenced the text "A Catalog of Special Plane Curves," J. Dennis Lawrence, Dover Publications, 1972. Fortunately, I happen to own a copy of this book, and from it, I extracted the following formulae for the conchoid of a curve given parametrically by the equations $x = f(t), y = g(t)$.

$$x = f(t) \pm \frac{k\left(f(t) - x_0\right)}{\sqrt{\left(f(t) - x_0\right)^2 + \left(g(t) - y_0\right)^2}}$$

$$y = g(t) \pm \frac{k\left(g(t) - y_0\right)}{\sqrt{\left(f(t) - x_0\right)^2 + \left(g(t) - y_0\right)^2}}$$

The fixed point P through which the line L is drawn has coordinates $\left(x_0, y_0\right)$, so in my first experiments I clearly took P as the origin. Hence, the formulas I used were

$$x = \left(1 \pm k/\sqrt{f^2 + g^2}\right)f \quad \text{and} \quad y = \left(1 \pm k/\sqrt{f^2 + g^2}\right)g$$

If $f = t$ and $g = t^3$, the equations for the conchoid of this cubic with respect to the origin are

$$x(t) = t\left(1 \pm k/\sqrt{t^2 + t^6}\right) \quad \text{and} \quad y(t) = t^3\left(1 \pm k/\sqrt{t^2 + t^6}\right)$$

The following realizations came slowly, after looking at a lot of puzzling graphs. Both branches for $x(t)$ have a jump discontinuity of $2k$ at $t = 0$; both branches for $y(t)$ have a removable singularity at $t = 0$, with the limit being zero in both cases. Figure 8 shows the branches of $x(t)$ when $k = 2$; Figure 9, $y(t)$. In each, the branch with the plus sign is in red; the minus sign, in green.

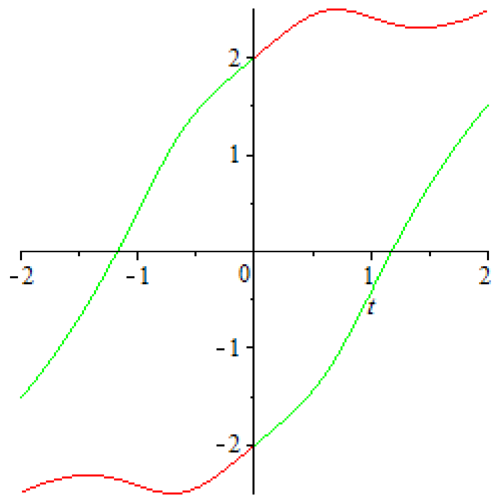


Figure 8 Branches of $x(t)$: x_+ in red, x_- in green

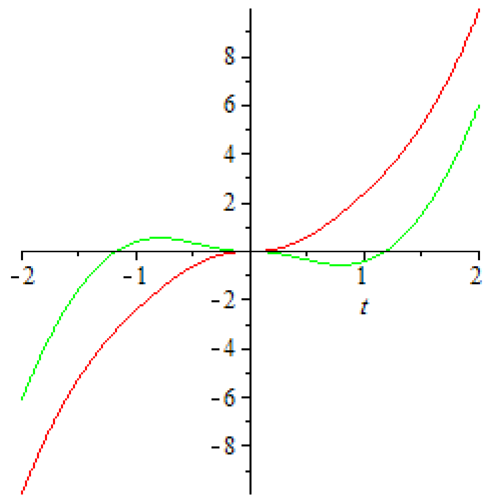


Figure 9 Branches of $y(t)$: y_+ in red, y_- in green

It's not easy to visualize the parametric branches from these two figures. Figure 10, an animated trace of the two branches of the conchoid, (the red corresponding to the plus sign; the green, to the minus sign), helps. Click on the graph to access the animation toolbar. Move the slider to trace the graphs. (Incidentally, for a single parameter, this is a viable method of creating a graph with one slider.) The code for this animation, based on the **animate** command, is hidden behind its table; similarly, for Figures 8 and 9.

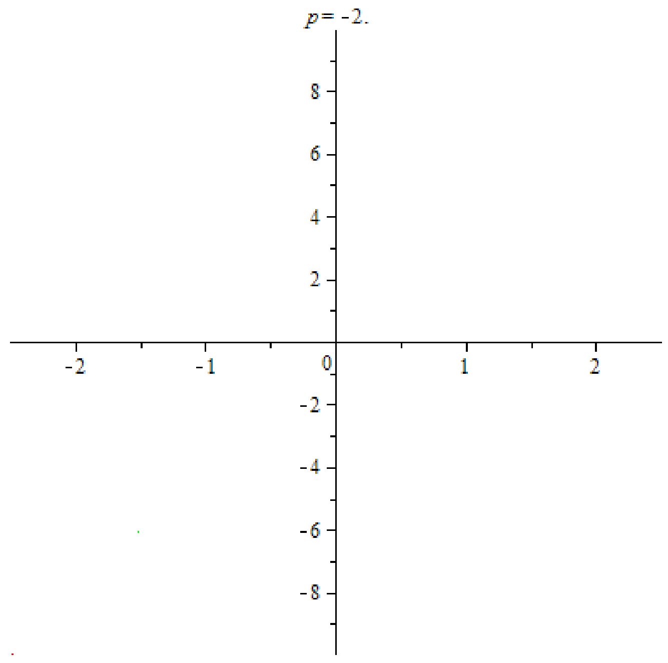
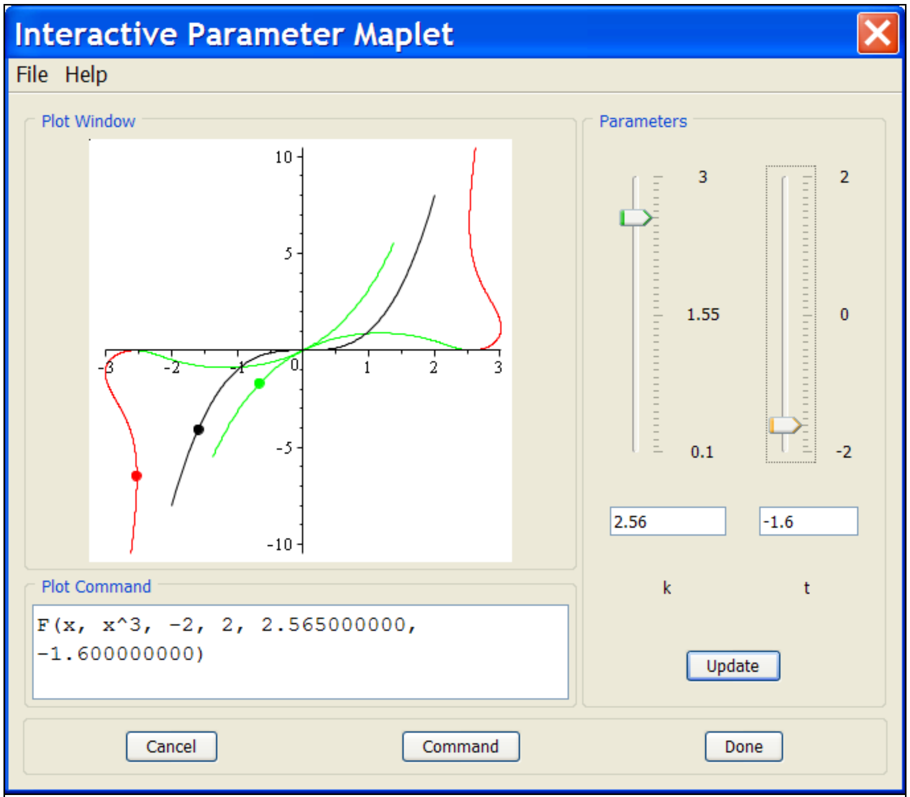


Figure 10 Animated trace of the two branches of the conchoid of a cubic

In Figure 10, notice how each branch of the conchoid (as defined by (x_+, y_+) in red and (x_-, y_-) in green) has both a red and green component, adding to the difficulty of visualizing the conchoid. Finally, Figure 11 shows an image of the Interactive Parameter Maplet in which the cubic is graphed in black, and the conchoid, in red and green. The black dot represents the intersection point P; the red and green dots, the points P_1 and P_2.



k: 2.56

t: -1.6

```
F(x, x^3, -2, 2, 2.565000000,
-1.600000000)
```

Figure 11 An image of the Interactive Parameter Maplet for the conchoid of a cubic

The value of k is adjusted with the leftmost slider; the value of the parameter t along the curve C is controlled by the other slider. The point on C determined by t is Q, the intersection of C with line L. The black dot represents point Q. The red and green dots represent the points P_1 and P_2 on the conchoid. Since the three dots are on line L, they are collinear. Given the complexity of the conchoid, the line L itself has not been included in the graph.

The Interactive Parameter Maplet in Figure 11 is launched by applying the **interactiveparams** command to $F(f, g, a, b, k, t)$, where the function F is defined in the Code-Edit region in Table 1. Click the icon in this table to define the function F, then execute the **interactiveparams** command in the next row of the table.

$$plots[interactiveparams]\left(F, [x, x^3, -2, 2, k, t], k = 0.1 ..3, t = -2 ..2\right)$$

Table 1 Define F and use it to generate the 2-slider graph in Figure 11

To see the code that defines the function F, select "Expand Code Edit Region" from the Context Menu for the icon in Table 1.

Legal Notice: © Maplesoft, a division of Waterloo Maple Inc. 2012. Maplesoft and Maple are trademarks of Waterloo Maple Inc. This application may contain errors and Maplesoft is not liable for any damages resulting from the use of this material. This application is intended for non-commercial, non-profit use only. Contact Maplesoft for permission if you wish to use this application in for-profit activities.