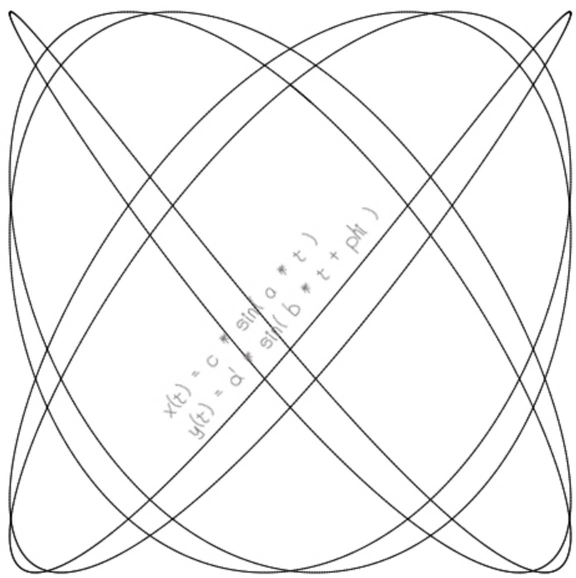
3𝒟 ℳodelling

Assignment 2; Lissajous Curves

S.J. van den Elzen (F072754)
M.K. Lewiński (F072790)



Universiteit Utrecht

Game and Media Technology

March 4, 2009

Contents

1 Lissajous Curves

First Assignment
"Implement a utility that displays a lissajous curve on the screen. It must be possible to set the various parameters interactively. The settale range of the parameters must be reasonable and not restricted to integers. Display must be such that the lissajous curve appears smooth and has no visible holes in its curve. Examine the effect of changing the parameters. What is the meaning of the five parameters? When is the curve closed?"

The lissajous curve consists of 5 parameters, (a, b, c, d and φ), which are used in the following formulae to calculate the x and y coordinates of the points on the curve:

$$x(t) = c \times \sin(a \times t) \tag{1}$$

$$y(t) = d \times \sin(b \times t + \varphi) \tag{2}$$

1.1 Parameters

The parameters all have a different influence on the lissajous curve. The a and b parameters control the period of the x and y coordinate respectively. The φ parameter only has influence (together with b) on the period of the y coordinate. The c and d have an influence on the amplitude of the x and y coordinate respectively. The shape of the lissajous curve heavily depends on the ratio $\frac{a}{b}$. If $\frac{a}{b} = 1$ the curve is an ellipsoid. When also $c = d$ and $\varphi = \frac{\pi}{2}$ the curve is a circle. The effect of changing the amplitude (c and d parameters) can be viewed very good when the curve is a circle. By changing the c parameter the circle becomes and ellipsoid because it stretches in the x direction. The same thing applies for changing the d parameter, the circle will stretch in the y direction. By keeping $\frac{a}{b} = 1$ and setting $\varphi = 0$, the curve will change to a line. The orientation of the line then depends on the c and d parameters. Another special curve that can be constructed by setting the parameters to the right values, is the parabola. This can be achieved by setting $\frac{a}{b} = \frac{1}{2}$ and $\varphi = \frac{\pi}{2}$. The 'width' and 'height' of the parabola can be changed by changing the c and d values again.

curve	parameters
ellipsoid	$\frac{a}{b} = 1$
circle	$\frac{a}{b} = 1, c = d, \varphi = \frac{\pi}{2}$
line	$\frac{a}{b} = 1, \varphi = 0$
parabola	$\frac{a}{b} = \frac{1}{2}, \varphi = \frac{\pi}{2}$

1.2 Smoothness

The smoothness of the curve highly depends on the distance of the calculated sample points. The lissajous curve is drawn by calculating (x, y) coordinates for uniform t values. Between these sampled points a line is drawn. The highter the t value the further the distance between one point and the second will be. As a consequence the lissajous curve will not be smooth but very sharp-edged. Because we want the lissajous curve to be as smooth as possible, a small t value is preferred. But because a small t values means, many points have to be calculated, this will always be a tradeoff between smoothness and speed. The value for t, and the number of sample points to be calculated are made user settable parameters. The initial values for these parameters are 0.05 and 450 respectively. These values are chosen because they gave good results in practice for most lissajous curves.

1.3 Closedness of the curve

Wheter the lissajous curve is closed or not depends again on the ratio $\frac{a}{b}$. If this ratio is **rational**, the curve is closed. The c, d and φ parameters have no influence on the closedness of the curve. So for example $\frac{a}{b} = \frac{1}{4}$ is closed whilst $\frac{a}{b} = \sqrt{2}$ is not (independent of c, d and φ).

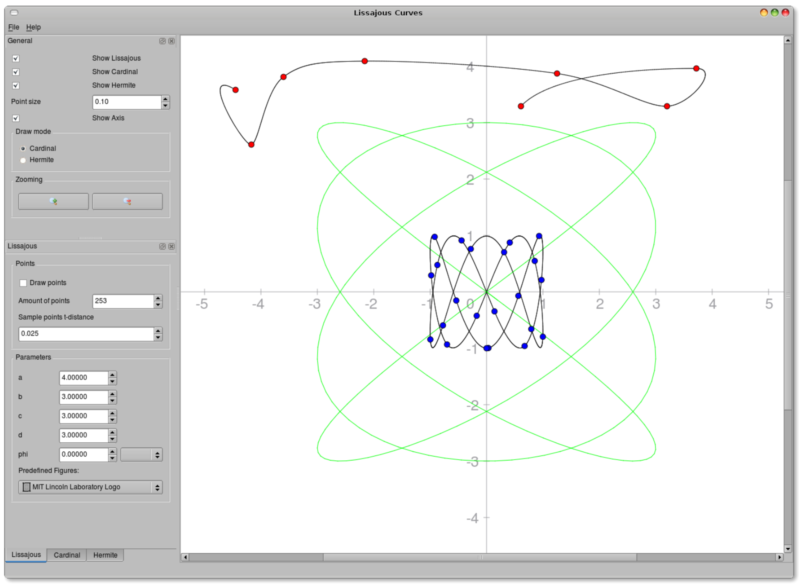


Figure 1: The graphical user interface with a Lissajous curve (green line), Cardinal spline (line with red control points) and Hermite spline (line with blue control points)

2 Hermite and Cardinal splines

Second Assignment
"implement a utility that can draw Hermite and Cardinal splines, given a set of control points (and derivatives in the case of Hermite splines)"

In the graphical user interface (see Figure 1) there can be chosen to set the draw mode to either cardinal or hermite spline. With the right mouse button the user is able to place control points. If the cardinal spline draw mode is selected the derivatives at the control points are calculated. The user has influence on the curvedness of the curve by changing the t value. When the hermite spline draw mode is selected the user has to specify the derivatives in each of the control points.

The expression for both a cardinal and hermite spline seqment between two control points is:

$$Q(u) = \mathbf{a}u^3 + \mathbf{b}u^2 + \mathbf{c}u + \mathbf{d} \tag{3}$$

where **a,b,c** and **d** are calculated by:

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_k \\ P_{k+1} \\ P'_k \\ P'_{k+1} \end{bmatrix}$$

In case of a Cardinal spline the derivative is computed from neighbouring control points (P_{k-1}, P_{k+1}):

$$P'_k = \frac{1}{2} \times (1 - t) \times (P_{k+1} - P_{k-1}) \quad (4)$$

where t is the tension parameter.

Initial t is set to 0 to create a *Catmull-Rom* spline. The most usefull values for the tension values are within the range $-1 \leq t < 1$. The user is able to adjust the tension parameter to suit his/her needs within the range $-10 \leq t \leq 10$.

Third Assignment
"Sample points and derivatives uniformly (i.e. with fixed t-distance) from the lissajous curve and consider these samples control points of a Hermite spline curve. Add functionality to draw the spline on the screen along with the lissajous curve. Now sample your points in such a way that the final Hermite spline never has an error exceeding 5% anywhere on the spline. A 5% error is defined as meaning that the distance of any point on the spline is never farther than 5% of the lissajous curve width away from the nearest lissajous curve point. If computing exact errors is difficult, you may aproximate them. How many control points did you need for a lissajous curve with $\{a, b, c, d, \varphi\} = \{2, 5, 1, 1, \frac{\pi}{2}\}$*? And how many the same parameters with a 1% error?"*

First functionality is added to draw the cardinal and hermite spline on the screen along with the lissajous curve. Then points from the lissajous curve are sampled with a uniform t-distance and used as control points for the hermite spline. The derivative for the sample points is directly calculated from the lissajous function. The derivative for the lissajous function is:

$$x'(t) = a \times c \times \cos(a \times t) \quad (5)$$

$$y'(t) = b \times d \times \cos(b \times t + \varphi) \quad (6)$$

The derivative of the lissajous function is multiplied with the sampling t-distance. This implies that the more points are taken as control points the smaller the length of the derivative becomes. This solves the problem of having derivatives with very large length. When the derivatives have a large length the interpolation of the points between the lines becomes strange because the derivative vectors overlap. The effect of this is that a curl in the line originates which gives a large approximation error. To avoid this, the derivative lengths with respect to the number of points are taken into account.
The width of the lissajous curve is defined as the absolute value of the c parameter. From this width the approximation error is calculated and displayed in the user interface. The user is able to specify the number of points to approximate the lissajous curve with either a cardinal or hermite spline.

For the lissajous curve with parameters $\{a, b, c, d, \varphi\} = \{2, 5, 1, 1, \frac{\pi}{2}\}$, 14 control points where needed to have an approximation error of just below 5%, namely 4.06264% (see Table 2). For an approximation error of just below 1%, 20 control points were needed. If one extra control point is added to the Hermite spline, the approxitmation error significantly drops to 0.58754%. See Figure 2 for the results of the approximated lissajous curve by using a Hermite spline with sampled control points. For a 4.91365% approximation error by using a **Cardinal spline**, 22 control points are needed.

Number of Hermite spline control points	Approximation error
12	7.61641%
13	5.4017%
14	4.06264%
⋮	
19	1.15248%
20	0.938636%
21	0.58754%

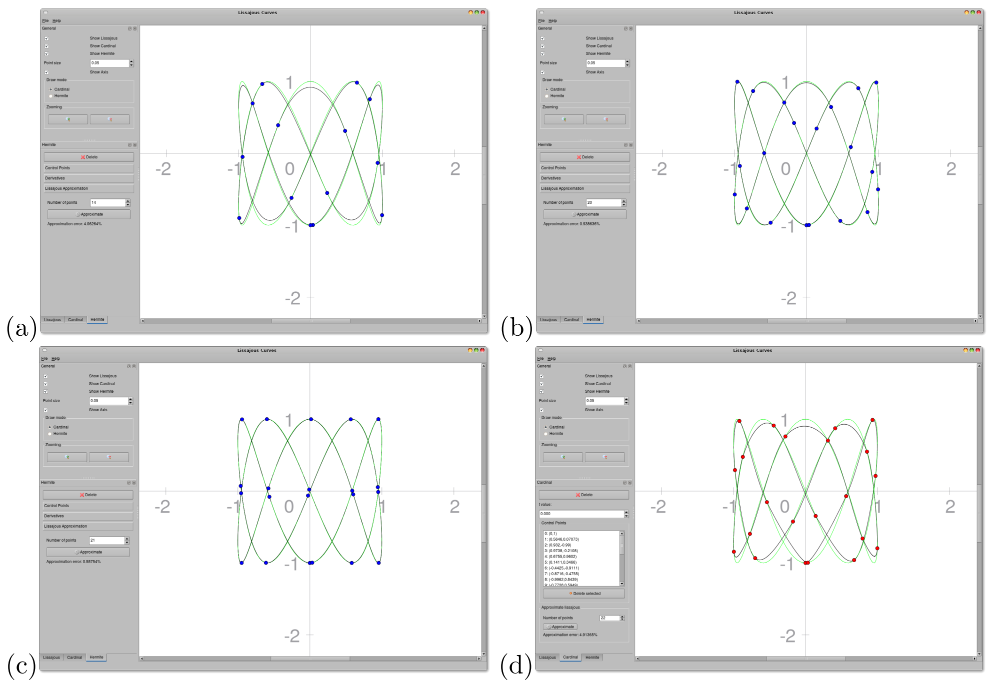


Figure 2: (a) Hermite spline with a 4.06264% approximation error w.r.t. the lissajous curve. To achieve this, 14 control points are needed. (b) Hermite spline with a 0.938636% approximation error w.r.t. the lissajous curve. To achieve this, 20 control points are needed. (c) Hermite spline with a 0.58754% approximation error w.r.t. the lissajous curve. To achieve this, 21 control points are needed. (d) Cardinal spline with a 4.91365% approximation error w.r.t. the lissajous curve. To achieve this, 22 control points are needed.

Fourth Assignment
"Fourth, sample points as before, but consider them control points for a Cardinal spline. Try varying your sampling density (a large error is probably ok here) and the tension parameter of the spline to obtain an esthetically pleasing spline result."

The control points for the cardinal spline are, as before, sampled from the lissajous curve. The derivatives for the control points are calculated from neighbouring control points as before. A tension parameter with value 0 gives in general the most pleasing spline results. These splines are known as *Catmull-Rom* splines, and fit the lissajous curve very good. In general the approximation error for the cardinal spline is low, when the tension parameter is set to 0.

A Compile instructions

We chose to make the program compile with the following tools:

- **G**NU **C**ompiler **C**ollection (GCC) 4.3.2
  http://gcc.gnu.org/
- Qt 4.4.3
  http://trolltech.com/products/qt

Qt is a cross-platform application framework for C++. The main choice to work with this setup, is to provide cross-platform compilation. With this setup we can easily compile the constructed code to run on Linux, Mac and Windows. Another great advantage of Qt is that it has an extensive library with graphical user interface components which can easily be adapted to serve our needs.

This section first explains how to build and install the requirements for the `Lissajous curve` program, which is Qt 4.4.3. After that, the build procedure for the Lissajous curve program itself is given.

In this section it is assumed that the code is build for Linux. The build procedures for Windows and Mac OS X are almost the same, but will not be discussed here. The main differences for compiling on the different platforms is how to included and link the external libraries.

On each platform the program can easily be build without adjustments by loading the project file into Qt creator (free downloadable from http://trolltech.com/developer/qt-creator) and pressing the *build-all* button. At code level, platform dependent differences are taken into account (for example mouse handling) but will not be discussed here.

A.1 Qt 4.4.3

Qt 4.4.3 can be downloaded at ftp://ftp.trolltech.com/qt/source. The file needed is called "qt-x11-opensource-src-4.4.3.tar.gz". Save this file to a known directory, say "/qt". After the file has been downloaded, open up a "terminal" and run the following code (please note that for the last command you need to enter your password):

```
user@host:~$ cd /qt
user@host:/qt$ tar -zxvf qt-x11-opensource-src-4.4.3.tar.gz
user@host:/qt$ cd qt-x11-opensource-src-4.4.3
user@host:/qt/qt-x11-opensource-src-4.4.3$ ./configure --prefix=/usr
user@host:/qt/qt-x11-opensource-src-4.4.3$ make
user@host:/qt/qt-x11-opensource-src-4.4.3$ sudo make install
```

After this Qt 4.4.3 is installed. This can be verified by running "qmake -v", which outputs something like:

```
user@host:~$ qmake -v
QMake version 2.01a
Using Qt version 4.4.3 in /usr
```

A.2 Lissajous

After Qt 4.4.3 is installed, The Lissajous curve program can be compiled. Assuming the source code of the program is placed in "/src", run the following commands in a "terminal":

```
user@host:~$ cd /src
user@host:/src qmake lissajous.pro -config release
user@host:/src$ make
```

Lissajous curve is now compiled. To run Lissajous, you can run the executable found in the directory where the source of lissajous is by typing in a "terminal" (from the "/src" directory):

```
user@host:/src$ ./lissajous
```