UNIVERZITA KOMENSKÉHO
V BRATISLAVE

FAKULTA MATEMATIKY, FYZIKY
A INFORMATIKY

RNDr. Mária Gemeranová

**Autoreferát dizertačnej práce**

# APROXIMÁCIA MNOŽINY BODOV POMOCOU KRIVKOVÝCH A PLOŠNÝCH LEMNISKÁT

na získanie akademického titulu philosophiae doctor
v odbore doktorandského štúdia

9.1.7 Geometria a topológia

Bratislava 2016

Dizertačná práca bola vypracovaná v dennej forme doktorandského štúdia na Katedre algebry, geometrie a didaktiky matematiky Fakulty matematiky, fyziky a informatiky Univerzity Komenského v Bratislave.

**Predkladateľ:** RNDr. Mária Gemeranová
Katedra algebry, geometrie a didaktiky matematiky
FMFI UK, Mlynská Dolina, 842 48 Bratislava

**Školiteľ:** doc. RNDr. Pavel Chalmovianský, PhD.
Katedra algebry, geometrie a didaktiky matematiky
FMFI UK, Mlynská Dolina, 842 48 Bratislava

**Oponenti:** ...................................................................
...................................................................

...................................................................
...................................................................

...................................................................
...................................................................

**Obhajoba dizertačnej práce sa koná ..................... o ............. h**
pred komisiou pre obhajobu dizertačnej práce v odbore doktorandského štúdia vymenovanou predsedom odborovej komisie .......................................

Geometria and topológia – 9.1.7 Geometria a topológia

na Fakulte matematiky, fyziky a informatiky Univerzity Komenského
Mlynská Dolina, 842 48 Bratislava

**Predseda odborovej komisie:**

prof. RNDr. Július Korbaš, CSc.

Katedra algebry, geometrie a didaktiky matematiky
FMFI UK, Mlynská Dolina, 842 48 Bratislava

# Contents

# Approximation of a point set using lemniscate curves and surfaces

## 1 Introduction

Approximation of a set of points in Euclidean plane is possible in many ways. It can be performed for example by polylines, splines or by algebraic curves e.g. *multifocal lemniscates*, which are the topic of our research. For 3D data, meshes, splines or algebraic surfaces can be used for the reconstruction of some surfaces. In this case, we chose *3D lemniscates* for approximating data in real space.

*Multifocal lemniscate* is a set of points in Euclidean plane, whose product of distances, to a finite set of fixed points (lemniscate foci), is equal to a constant (lemniscate radius). A space analogues of these algebraic curves are algebraic surfaces named *3D lemniscates*. Such curves and surfaces have a lot of interesting properties and usage in many scientific/technology areas.

## 2 Theoretical background

### 2.1 Polar coordinates in Euclidean plane

Polar coordinates determine point in Euclidean plane $\mathbb{E}^2$ by a coordinate pair $(r, \theta)$, where $r \in \mathbb{R}_0^+$ represents a distance of the point from a fixed pole and $\theta \in \langle 0, 2\pi)$ represents a polar angle measured with respect to an axis passing through the pole. A special case of polar coordinates, where the polar angle is substituted by a distance to another fixed pole are bipolar coordinates, which determine a point in $\mathbb{E}^2$ by a bipolar coordinate pair $(r_1, r_2)$, where $r_1, r_2 \in \mathbb{R}_0^+$ represent distances of the point from two different fixed poles. Dimensional inhomogeneous generalization of bipolar coordinates are multipolar coordinates, which determine a point in $\mathbb{E}^2$ by a redundant system of multipolar coordinates $(r_1, r_2, \ldots, r_n)$, $r_i \in \mathbb{R}_0^+$, $i = 1, \ldots, n$, where $r_i$ are distances of the point to $n$ fixed poles. [5], [7], [8]

### 2.2 Multifocal lemniscates

A set of points $\{z \in \mathbb{E}^2 | r_1 r_2 \ldots r_n = R\}$, where $z = (r_1, r_2, \ldots, r_n)$ is the point defined by its multipolar coordinates, $r_i = |z - z_i|$, $i = 1, \ldots, n$ are distances of the point $z$

from $n$ fixed poles $z_1, z_2, \ldots, z_n \in \mathbb{E}^2$ and $R \in \mathbb{R}^+$ is a selected constant, is called multifocal lemniscate in $\mathbb{E}^2$. This compact algebraic curve of degree $2n$ is determined by its foci $z_1, z_2, \ldots, z_n$ and its radius $R$. It may consist of one or more connected components which number changes according to the positions of the foci and the value of the radius. There are no more components than the number of lemniscate foci and each component contains inside at least one focus. [3], [9], [5], [6], [11], [2]

### 2.3 3D lemniscates

Similarly to the curve case, 3D lemniscates are bounded algebraic surfaces, whose degree is twice the number of their foci. They are spatial analogues of classical multifocal lemniscates in $\mathbb{E}^2$. Such a surface is a set of points in $\mathbb{R}^3$, whose product of squared distances, to a finite set of fixed points is equal to a constant. 3D lemniscates may also consist of one or more connected components of a surface. The number of such components changes according to the positions of the foci and the value of the radius. Each focus of the 3D lemniscate is enclosed in some of these connected components but, in the contrast to the plane multifocal lemniscates, there might be a component enclosing inside no focus of the 3D lemniscate. [12], [11], [1]

## 3 Approximation of a set of points using lemniscates

A motivation for our research focused on approximating the input set of points using lemniscates is a fact, that with multifocal lemniscates, we are able to achieve a good approximation using a small amount of data. The idea of our work is based on the *Hilbert theorem* [10] which says, that an arbitrary curve in plane, smooth enough, can be approximated by a multifocal lemniscate with a desired accuracy. In our work, we are looking for a solution of the following task: try to achieve the most accurate approximation of an input set of points $\{p_1, p_2, \ldots, p_n\} \in \mathbb{E}^2$ using multifocal lemniscate given by the equation $L(z_1, z_2, \ldots, z_N; R) = 0$. In other words, we are looking for a collection of optimally located foci and a suitable radius value, so that the resulting lemniscate approximates the input data set sufficiently accurate in Euclidean metric.

We have examined a few iterative algorithms of doubling, removing and adding lemniscate's foci and a heuristic estimation function for location of new foci. Our aim is to minimize the number of used foci and the error of achieved lemniscate approximation. As an approximating criteria, we use *Sampson's distance* [13]. It is the first order approximation of an oriented Euclidean distance from a regular part of an implicitly defined curve $f = 0$ by $\frac{f}{|\nabla f|}$. Approximation lemniscate parameters are optimized using quasi-Newton's method BFGS [4]. The methods from the class of quasi-Newton's methods are iterative optimization methods in numerical analysis for solving unconstrained minimization problems. They are derived from the known Newton's descend method. Instead of computing a Hessian matrix of a function or its inverse, they use only approximation of the Hessian inverse matrix.

## 3.1 Algorithm of doubling lemniscate's foci

The first of our methods is the *Algorithm of doubling lemniscate's foci*. The method is looking for an appropriate approximation lemniscate by cyclically doubling its foci, modifying the radius value and optimizing all parameters using method BFGS. For an illustration of its behaviour see fig.1. Steps of the method are described here:

1. **Determination of parameters for a starting circle**
   The center $z_1$ of the starting circle $k(z_1, R)$, which is also the first focus of an approximating lemniscate, is computed as a center of gravity of an input set of points. For the starting circle, an appropriate radius $R$ has to minimize the product of distances of the input set of points from $z_1$. The radius is computed from the equation $\frac{d}{dR}F(R) = 0$, where

$$F(R) = \sum_{i=1}^{n} \left( \frac{k(z_1, R)(p_i)}{\| \nabla k(z_1, R)(p_i) \|} \right)^2, \tag{1}$$

   the number of points in the input data set is $n$ and the points $p_i$, $i = 1, \ldots, n$ are from the input data set.

2. **Optimization of the starting circle parameters**
   Parameters are optimized using our own implementation of the BFGS method.

3. **Doubling the foci**
   Consider a set of foci $z_1, z_2, \ldots, z_{2^m}$, where $m$ stands for the number of iterations of our algorithm. Next, the algorithm looks for the new set of foci $z_1, \ldots, z_{2^m}, z_{2^m+1}, \ldots, z_{2^{m+1}} = z_N$, which are derived by doubling the original foci $z_1, z_2, \ldots, z_{2^m}$ i.e. $z_{2^m+j} = z_j$, $j = 1, \ldots, 2^m$. Since the multiplicity of the foci has changed, the old radius has to be squared in order to preserve the same curve shape.

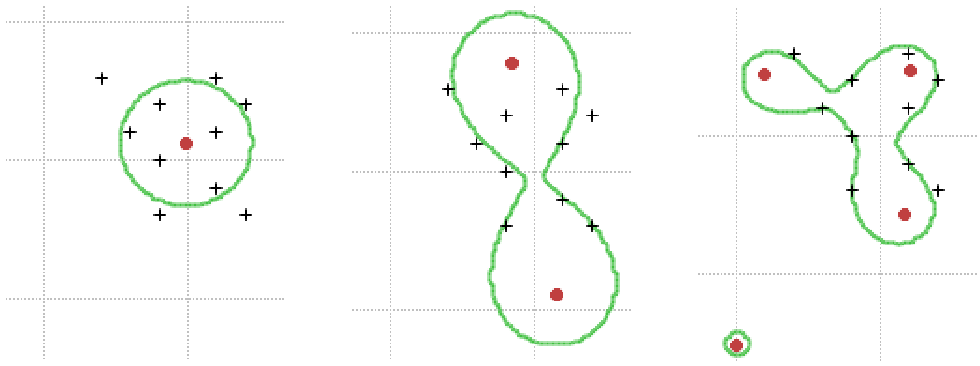
4. **Foci motion step**
The algorithm is passing through an actual set of foci and choosing one of them. The chosen focus is moved in a direction of the axes $x$ and $y$ about a specific constant. The algorithm finally chooses the direction for the focus motion, in which the error value is the smallest. The error is computed as

$$F(z_1, z_2, \ldots, z_N; R) = \sum_{i=1}^{n} \left( \frac{L(z_1, z_2, \ldots, z_N; R)(p_i)}{\| \nabla L(z_1, z_2, \ldots, z_N; R)(p_i)\|} \right)^2, \quad (2)$$

where $L(z_1, \ldots, z_N; R) = 0$ stands for the lemniscate equation with the foci $z_1, \ldots, z_N$ and the radius $R$, the points $p_i$, $i = 1, \ldots, n$ are from the input data set. When the algorithm has completely modified the positions of all the foci, a new radius value is computed similarly as it was done for the starting circle above.

5. **Optimization of the approximation lemniscate parameters**
Parameters are optimized using our own implementation of the BFGS method.

6. **Iterating from doubling the foci step**
If the total error is too high, the algorithm jumps to the step three and the number of the lemniscate foci is increased.



(a) 1 focus, error $\doteq$ 7.01037 (b) 2 foci, error $\doteq$ 2.01015 (c) 4 foci, error $\doteq$ 0.30875

Figure 1: An illustration of behaviour of the *Algorithm of doubling lemniscate's foci*: black crosses – the input set of 10 points, red points – foci of approximation lemniscates, green curves – approximation lemniscates. (The lemniscate error is rounded to five decimal points.)

## 3.2 Algorithm of doubling and removing lemniscate's foci

With the *Algorithm of doubling lemniscate's foci*, we have achieved quiet good results, but in some cases, the final approximation seemed to have certain redundant foci (see fig. 1 (c)). This fact was motivation for our second algorithm the *Algorithm of doubling and removing lemniscate's foci*, in which we have extended previous method by part, where one focus can be removed, even at the cost of possible increase of local approximation error and change of lemniscate topology. In each its cycle, the method decides whether to double the lemniscate foci or to remove one focus. The steps of the extended algorithm are described here (for a more detailed description of some steps see previous section 3.1):

1. **Determination of parameters for a starting circle**
2. **Optimization of the starting circle parameters**
3. **Main loop**
   The main loop of the algorithm is repeating till the approximation error is too high and the number of lemniscate foci is under the chosen threshold (denote the current approximation as *'Current'*):

   A. If the number of lemniscate foci is smaller than four, i.e. the approximation lemniscate has one or two foci and so none of them can be redundant, then execute the algorithm in unextended way:

      i. **Doubling the foci**
      ii. **Foci motion step**
      iii. **Optimization of the approximation lemniscate parameters**
      iv. Jump to the main loop.

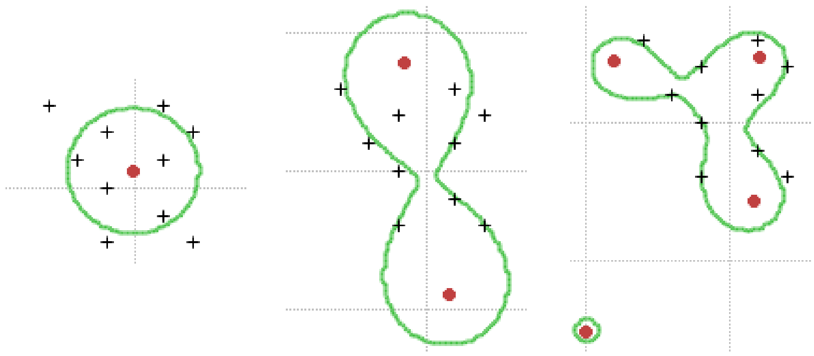
   B. Else, the number of lemniscate foci is at least four, i.e. the approximation lemniscate has four or more foci and so it is possible to remove some foci to get a better approximation, so do:

      i. **Remove focus**
         The algorithm searches for a lemniscate focus to be removed. Lemniscate foci are one by one tested how much is the approximation error getting better when one focus is removed and the lemniscate radius is appropriately modified by the estimation function. The algorithm chooses the focus to remove, so that the approximation error improves the most.

ii. **Foci motion step**
iii. **Optimization of the approximation lemniscate parameters**
iv. If the error of this approximation after removing of a focus (denote it *'Remove'*) is better than the error of the *'Current'*, then:
   a. **Doubling the foci** of the *'Remove'*
   b. **Foci motion step** with the *'Remove'*
   c. **Optimization of the approximation lemniscate parameters** on the *'Remove'*
   d. Jump to the main loop.
v. Else, the *'Current'* is better than the *'Remove'*, so do:
   a. **Doubling the foci** of the *'Remove'*
   b. **Foci motion step** with the *'Remove'*
   c. **Optimization of the approximation lemniscate parameters** on the *'Remove'*
   d. **Doubling the foci** of the *'Current'*
   e. **Foci motion step** with the *'Current'*
   f. **Optimization of the approximation lemniscate parameters** on the *'Current'*
   g. Choose better approximation among the *'Remove'* and the *'Current'*

For a better illustration of behaviour of the extended method, let us mention an example of the algorithm run with the corresponding illustrations in fig. 2:

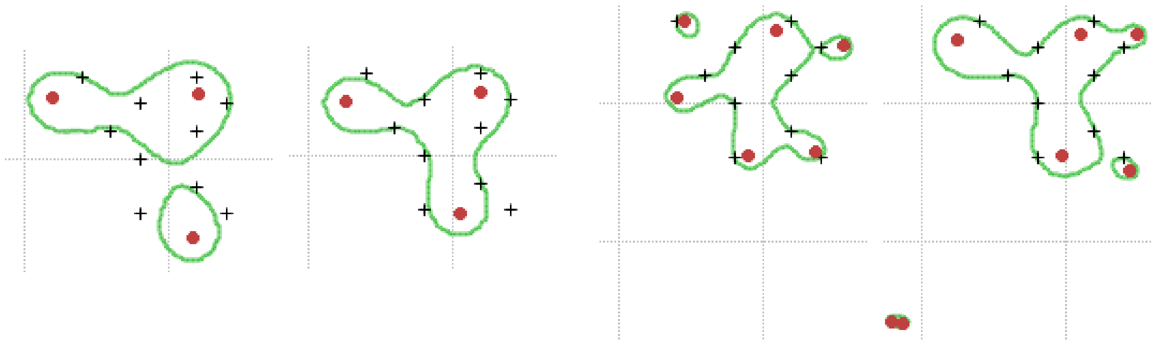
- At the beginning, the algorithm determines parameters for the starting circle and optimizes them (a).
- In the first pass through the main loop, the algorithm doubles the foci to 2, performs the 'Foci motion step' and BFGS optimization (b).
- In the second pass through the main loop, the algorithm doubles the foci to 4 and again performs the 'Foci motion step' and BFGS optimization (c).
- In the next pass through the main loop, the lemniscate has 4 foci, and so it is enough to the 'Remove focus' step be performed. The algorithm removes focus (d), performs the 'Foci motion step' and BFGS optimization and we get an approximating lemniscate with 3 foci (e).



(a) 1 focus, error $\doteq$ 7.01037

(b) 2 foci, error $\doteq$ 2.01015

(c) 4 foci, error $\doteq$ 0.30875



(d) 3 foci, error $\doteq$ 1.74065

(e) 3 foci, error $\doteq$ 0.82157

(f) 6 foci, error $\doteq$ 0.00005

(g) 8 foci, error $\doteq$ 0.0

Figure 2: An illustration of behaviour of the *Algorithm of doubling and removing lemniscate's foci*: black crosses – the input set of 10 points, red points – foci of approximation lemniscates, green curves – approximation lemniscates. We get (d),(e),(f) by removing foci extension of the previous algorithm (see fig. 1). (The lemniscate error is rounded to five decimal points.)

Now, the algorithm compares the achieved approximations with 3 and 4 foci. The lemniscate with 4 foci has smaller error than the one with 3 foci, but our algorithm does not take the 4 foci lemniscate as a winner and throw the 3 foci lemniscate away, instead of this, it works further with both of these approximations.

(If the 3 foci lemniscate were better than the one with 4 foci, the algorithm would work with the 3 foci lemniscate further, i.e. double foci, the 'Foci motion step', BFGS optimization, jump to the main loop and throw the 4 foci lemniscate away.)

The algorithm doubles the foci of the 3 foci lemniscate, performs the 'Foci motion step' and BFGS optimization and gets a lemniscate with 6 foci (f). The algorithm next doubles the foci of the 4 foci lemniscate, performs the 'Foci motion step' and BFGS optimization and gets a lemniscate with 8 foci (g).

At this moment, the algorithm performs a comparison between the approximation with 6 and 8 foci and chooses the best. In our example, the winner is the 6 foci lemniscate although its error is a bit worse than the one with 8 foci, but the difference is at insignificant decimals and so globally counts more the number of used foci.

- In this case, the algorithm now ends and the final approximation is the one with 6 foci.

The *Algorithm of doubling and removing lemniscate's foci* allows local worsening of the approximation error to get potential better global approximation. In fact, global improvement of the approximation can not be assessed only by improving the approximation error but also by proportion between improvement of the error and increasing number of the lemniscate foci.

## 3.3 Adding one focus method

The removing one focus process with an appropriate modification of the radius value from the *Algorithm of doubling and removing lemniscate's foci* described in the previous section 3.2 has inspired us to examine another method for approximating points in $\mathbb{E}^2$. So in this section, we have proposed an iterative approximation method, where only one new lemniscate focus is added in each algorithm cycle. Even at the cost of possible change of lemniscate topology, with this *Adding one focus method* we are able to achieve approximation lemniscates with one, two, three, . . . foci. Steps of the method are described bellow (the steps without a more detailed description are similar as in the section 3.1). For an illustration of behaviour of the method see fig. 3.

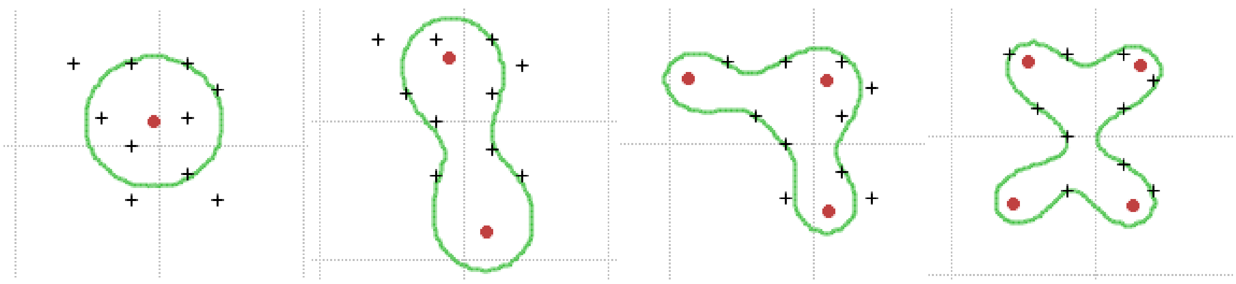
1. **Determination of parameters for a starting circle**
2. **Optimization of the starting circle parameters**
3. **Adding one focus**
   Now, the algorithm adds to current approximating lemniscate one new focus and is searching for its suitable location. The positions of the foci we already have are one by one tested and the lemniscate radius is appropriately modified.

The algorithm finally chooses the position for location of new focus, so that the approximation error improves maximally.

4. **Foci motion step**

5. **Optimization of the approximation lemniscate parameters**

6. **Iterating from adding one focus step**
   If the total error is too high, the algorithm jumps to the step three and the number of the lemniscate foci is increased.



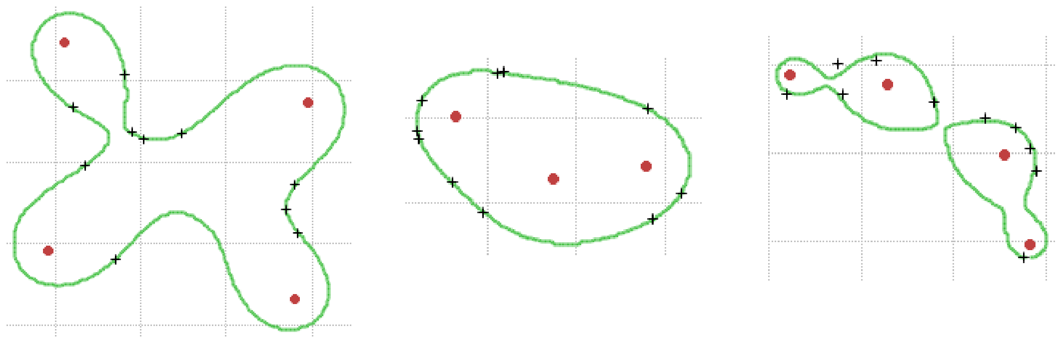
(a) 1 focus, error $\doteq 6.373$ (b) 2 foci, error $\doteq 1.91085$ (c) 3 foci, error $\doteq 0.61198$ (d) 4 foci, error $\doteq 0.08572$

Figure 3: An illustration of the *Adding one focus method*: black crosses – the input set of 10 points, red points – foci of approximation lemniscates, green curves – approximation lemniscates. (The lemniscate error is rounded to five decimal points.)

## 3.4 Algorithms testing and evaluating

We have performed some experiments with our three mentioned approximation algorithms. Some results achieved on the input set of 10, 25 and 50 points are presented in fig. 4. (Note: in subfigures descriptions, there can be found the number of the input points and some lemniscate approximation parameters – the number of used foci, the number of performed algorithm iterations and also the approximation error value.)

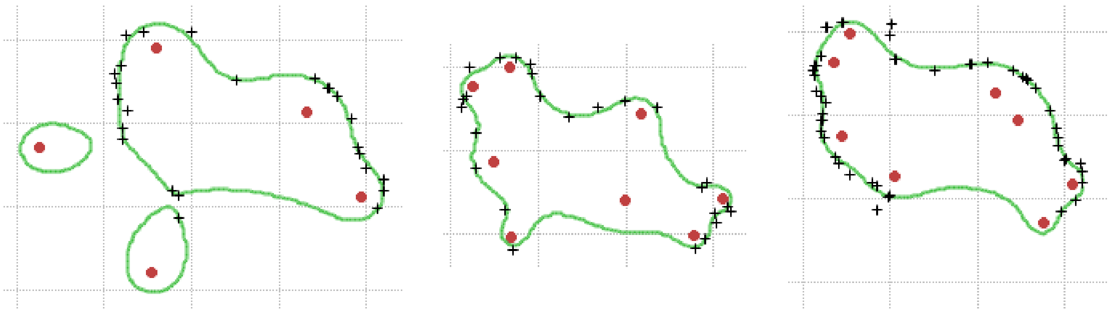
Based on the knowledge, that it is possible to achieve a good approximation of an input set of points in $\mathbb{E}^2$ with multifocal lemniscates using only a small amount of data and also on the fact, that 3D lemniscates are the space analogues of these curves, we have successfully tried to approximate points in $\mathbb{R}^3$ using 3D lemniscates. We have adapted previously mentioned approximation methods from the planar case to the spatial case. Some results achieved by these methods are presented in fig. 5.



(a) 10 points, 4 foci, 6 iterations, error ≐ 0.01873

(b) 10 points, 3 foci, 3 iterations, error ≐ 0.02146

(c) 10 points, 4 foci, 4 iterations, error ≐ 0.43251



(d) 25 points, 5 foci, 6 iterations, error ≐ 0.70222

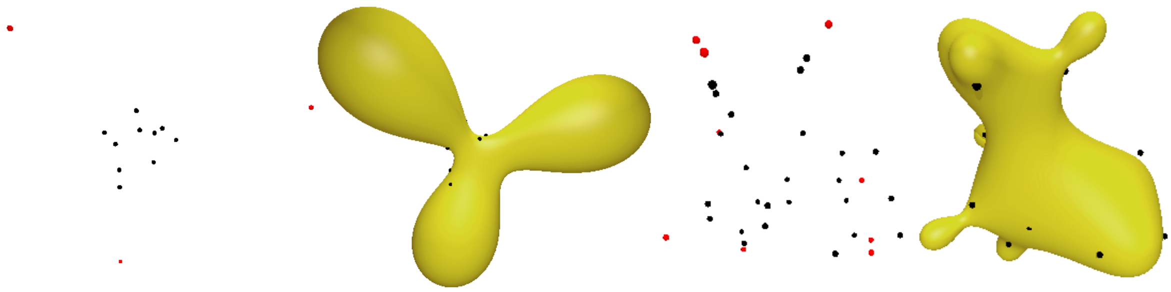
(e) 25 points, 8 foci, 5 iterations, error ≐ 0.667

(f) 50 points, 8 foci, 5 iterations, error ≐ 2.12294

Figure 4: The input sets of points approximated by the *Algorithm of doubling and removing lemniscate's foci* (a), the *Adding one focus method* (b),(d) and the *Algorithm of doubling lemniscate's foci* (c),(e),(f): black crosses – input points, red points – foci of approximation lemniscates, green curves – approximation lemniscates. (The lemniscate error is rounded to five decimal points.)

Our algorithms are now implemented in such a way, that they are not able to handle a large number of points – already the input set of 100 points may cause some troubles. The methods are capable to achieve only partially approximations for the input sets with so many elements, because our way of implementation is not always able to cope with approximation lemniscates with 16, 18, 32 or more foci.

Computational time expectable depends on the number of the input points, their distribution and also on the used algorithm. For the planar case it means: 10 points can be approximated in at maximum few minutes, 25 points in about tens of minutes and 50 points can take even some hours. For the spatial case: 10 points can be approximated in at maximum tens of minutes, 25 points in at maximum few hours and 50 points can take even a few days. So our methods are not proper for real time applications in current setup. Because of the number of needed time consuming

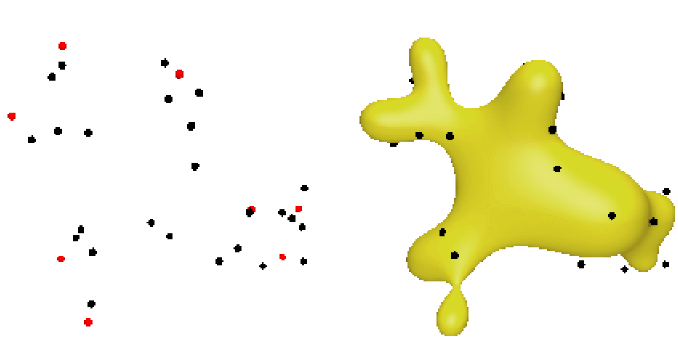


(a) 10 points, 3 foci

(b) 3 iterations, error $\doteq$ 0.02997

(c) 25 points, 9 foci

(d) 9 iterations, error $\doteq$ 0.95209

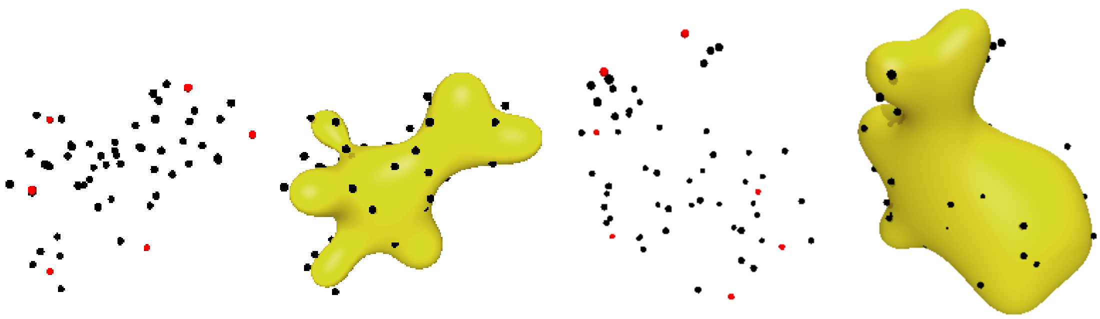


(e) 25 points, 8 foci

(f) 5 iterations, error $\doteq$ 3.79249

(g) 25 points, 9 foci

(h) 10 iterations, error $\doteq$ 2.43892



(i) 50 points, 8 foci

(j) 5 iterations, error $\doteq$ 0.13869

(k) 50 points, 7 foci

(l) 9 iterations, error $\doteq$ 8.48506

Figure 5: The input sets of points approximated by the *Adding one focus method* (a),(b), the *Algorithm of doubling and removing lemniscate's foci* (c),(d),(g),(h),(k),(l) and the *Algorithm of doubling lemniscate's foci* (e),(f),(i),(j) : black points – input points, red points – foci of approximation lemniscates, yellow surfaces – 3D approximation lemniscates. Parts (a),(c),(e),(g),(i),(k) illustrate the input points and computed foci placements and parts (b),(d),(f),(h),(j),(l) illustrate corresponding 3D lemniscate approximations. In subfigures descriptions, there can be found the number of the input points and some 3D lemniscate approximation parameters: the number of used foci, the number of performed algorithm iterations and also the approximation error value. (The lemniscate error is rounded to five decimal points.)

iterations, the *Algorithm of doubling and removing lemniscate's foci* takes longest. Another both algorithms take comparable much time for small input sets of points, the *Adding one focus method* is a bit faster. For bigger point sets is the *Algorithm of doubling lemniscate's foci* faster.

In general, the *Algorithm of doubling lemniscate's foci* achieves the most appropriate results for small or larger input sets of points even though it does not always achieve the best possible approximation, the approximation with a suitable small number of foci. The *Algorithm of doubling and removing lemniscate's foci* is sometimes able to find more appropriate approximation than the *Algorithm of doubling lemniscate's foci*, but it takes longer. And the *Adding one focus method* has a potential to produce good results for a small number of input points in the case that it does not stuck in some of a local minimum of its estimation function.

In the computed examples, the approximation error less or around the number 1.0, 0.5 resp. 0.1 was appropriate resp. sufficient. But the acceptability of the reached approximation error also depends on the shape of the achieved lemniscate and the number of the input points – for input sets with a bigger number of elements, there can be tolerated the error value less than 5.0 if the input points which are not approximated well by the lemniscate can be neglected. For 3D lemniscates, we have to be more permissive and the error value less than 10.0 may be tolerated.

## 3.5 Lemniscate blending

When we were testing our approximation methods, we have found out, that they are able to achieve a good approximation in a short time mainly for an input set of small number of points while these points are approximated by a lemniscate with a small number of foci or by some segment/component of such a lemniscate. Based on these observations we have performed an experiment, in which we have approximated an input set of points in $\mathbb{E}^2$ using blended segments of some lemniscates:

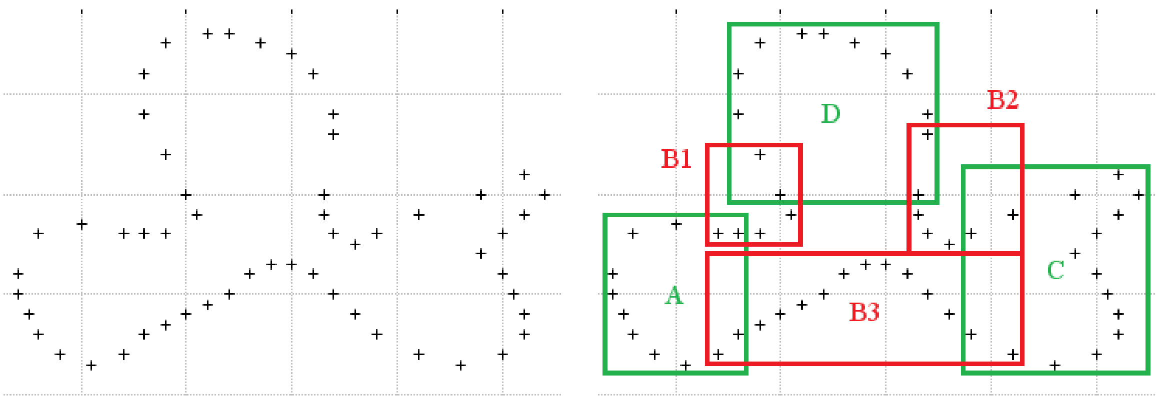
- At the beginning, we have divided the input set of 56 points (see fig. 6 (a)) sampled from a chosen curve to smaller subsets (sectors) (see fig. 6 (b)) for which is easier to be approximated by a lemniscate with only a few foci. The points in the individual subsets were selected in such a way, that they altogether form a 'continuous segment of the original curve', their number is in the range from at least 5 to 15, two adjacent segments have 2 common 'consecutive' points and there might be possibility to approximate these points by a lemniscate with 3-5 foci (such a number of used foci is small enough and a good approximation can be reached).

- Next, approximations of the subsets were performed. We have chosen the *Adding one focus method* for this purpose. Since we have needed only particular segments from the achieved approximations, not whole lemniscates, we picked only that parts, which are contained inside our sectors $A, \ldots, D$ (from fig. 6 (b)). In fig. 7, there are illustrated whole approximation lemniscates of particular sectors and also their picked segments.

- Finally, we have blended obtained segments together in the following way. Consider two adjoining segments $Seg_i, Seg_j$ which are to be glued together. They have a common part between two approximated points $P_i = (x_i, y_i)$, $P_j = (x_j, y_j)$, which is unnecessary for our work, because it will be substituted by a new join. For creating such a join we have used only a simple combination of lemniscate approximations of these segments:

$$\left(\frac{t_1 - x}{t_1 - t_0} + \frac{t_3 - y}{t_3 - t_2}\right) * L_i(x, y) - \left(\frac{x - t_0}{t_1 - t_0} + \frac{y - t_2}{t_3 - t_2}\right) * L_j(x, y), \qquad (3)$$

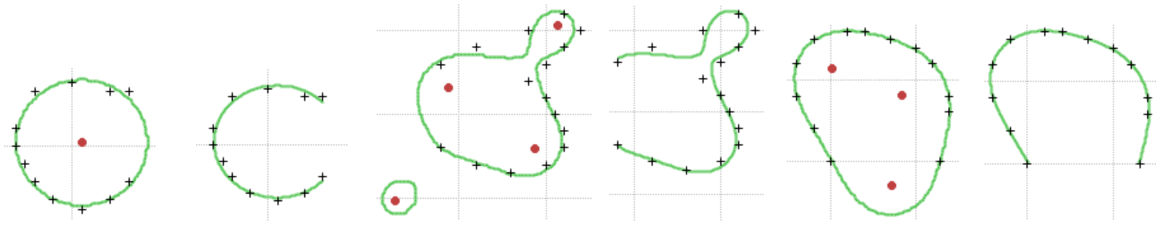
  where $L_i(x, y), L_j(x, y)$ are lemniscates equations, $t_0, t_1$ are minimum and maximum from $x$-coordinates of points $P_i, P_j$ and so $t_2, t_3$ are minimum and maximum from $y$-coordinates and variables $x \in \langle t_0, t_1 \rangle, y \in \langle t_2, t_3 \rangle$. This kind of a 'blending function' was sufficient for our example. (See fig. 8 (a) for an illustration of the picked segments, (b) the segments blended with the new joins and (c) the final blended approximation.)



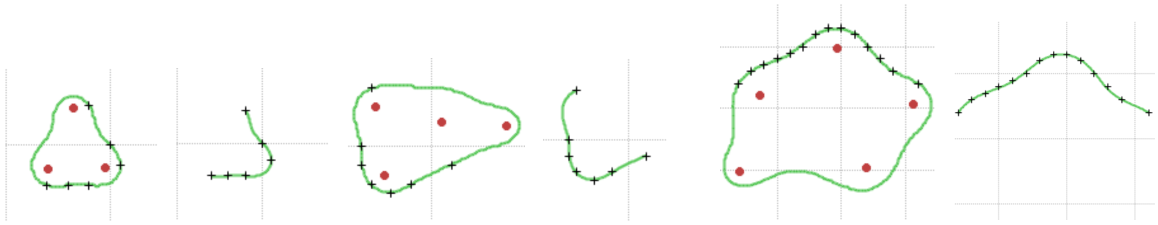

(a) the input points

(b) the input points divided into the sectors

Figure 6: An illustration of the input points (a) and their division to the sectors (b) for the lemniscate blending: black crosses – the input points, green/red rectangles – the sectors $A, \ldots, D$.



(a) 1 focus, error ≐ 0.43599 (b) 12 points, segment A (c) 4 foci, error ≐ 1.01161 (d) 16 points, segment C (e) 3 foci, error ≐ 0.02993 (f) 13 points, segment D

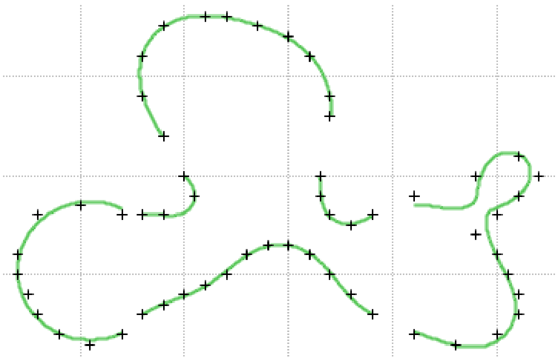


(g) 3 foci, error ≐ 0.0 (h) 6 points, segment B1 (i) 4 foci, error ≐ 0.01359 (j) 7 points, segment B2 (k) 5 foci, error ≐ 0.05634 (l) 14 points, segment B3

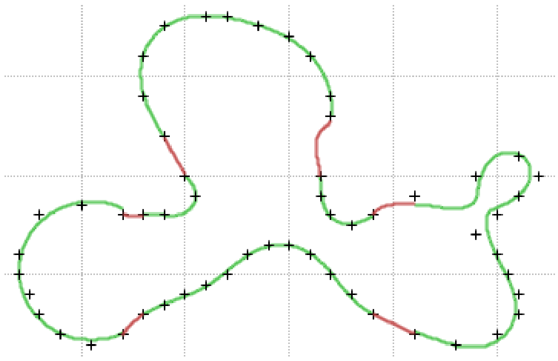
Figure 7: Approximation lemniscates of the particular sectors (a),(c),(e),(g),(i),(k) and their corresponding picked segments (b),(d),(f),(h),(j),(l): black crosses – the input points, red points – lemniscate foci, green curves – approximation lemniscates and their picked segments. (The lemniscate error is rounded to five decimal points.)

Although, we have achieved an acceptable final approximation of the input points using blended segments of lemniscates, there are some shortcomings in our result:

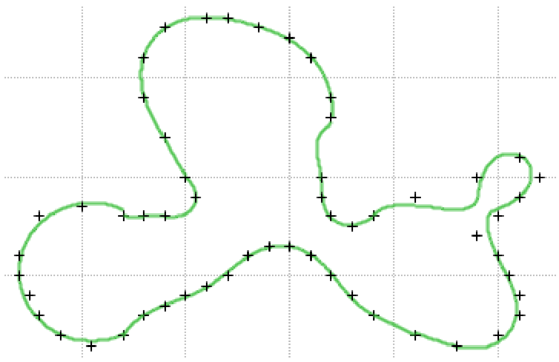
- The points in the sector A were approximated by a circle, a lemniscate with only 1 focus which is not a suitable shape for the lemniscate blending process. As shown in fig. 9 (a), the joins blending the circular segment A with the adjacent segments B3 but mainly B1 have noticeable turning points.
- The segment in the sector D (see fig. 9 (b)) was approximated with some kind of a 'lobe' lemniscate which was not able to reach sufficient approximation of the points in the sector and so this inaccuracy has worsened also the final approximation of the input data. This fault was caused by inappropriate choice of the points in the sector D – points selected to one sector should have potential to be approximated without such lobes.



(a) the final segments to be blended with new joins

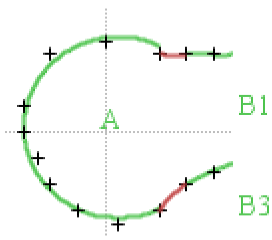


(b) the final segments blended with red joins

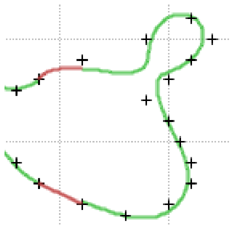


(c) the final approximation

Figure 8: An illustration of the picked segments (a) to be blended with the new joins (b) and the final blended approximation of the input points (c): black crosses – the input points, green curves – the segments of approximation lemniscates, red/green curves – the segment's joins.




(a) segment A



(b) segment D

Figure 9: An example of not suitable particular approximations: (a) a circular approximation, (b) a 'lobe' approximation. (black crosses – the input points, green curves – the segments of approximation lemniscates, red curves – the segment's joins)

# 4 Conclusion

We have introduced algebraic curves called multifocal lemniscates defined using multipolar coordinates and their spatial analogues, algebraic surfaces called 3D lemniscates. Such theoretical basis of our work has served as a preparation for the research in the area of approximating the input set of points in Euclidean plane using multifocal lemniscates, resp. approximating the input points in real space using 3D lemniscates.

Main part of our research was focused on looking for a suitable lemniscate approximation for the input data by finding an appropriate foci displacement and adjusting the corresponding radius value. Our aim was to minimize the error of the achieved lemniscate and the number of used foci. For this purpose, we have proposed three main algorithms of doubling, removing and adding lemniscates foci and examined a suitable estimation function for location of the new foci.

The first proposed method, the *Algorithm of doubling lemniscate's foci*, is able to find an appropriate lemniscate approximation only by doubling the actual set of lemniscate's foci and squared the current radius value. This approach leaves lemniscate's topology unchanged but sometimes it brings into a play certain redundant foci.

The second method, the *Algorithm of doubling and removing lemniscate's foci*, has solved the issue with redundant foci from the previous method by the extended version of this algorithm, in which it can decide to remove one redundant focus even at the cost of possible increase of local approximation error and change of lemniscate topology.

And the last algorithm, the *Adding one focus method*, which may add only one new focus per cycle can produce approximation lemniscates with one, two, three, . . . foci and so it can achieve potentially the best approximation for the input data set, but it is also at the cost of change of lemniscate topology.

We have performed some tests with these methods and found out that the algorithms were capable to achieve a good approximation for only a small number of input points at the time. For sets with a larger number of elements, the algorithm needed a lot of time to reach a lemniscate approximation, which was in some cases even not suitable.

Based on this tests, we have successfully performed an experiment, in which we have gluing pieces of some 2D lemniscates together to get the lemniscate approximation for the set of a larger number of the input points.

# References


[1] G. Arcos, G. Montilla, J. Ortega, and M. Paluszny. Shape control of 3D lemniscates. *Mathematics and Computers in Simulation*, 73(1-4):21–27, 2006.

[2] P. Chalmovianský. Multifocal Lemniscates – A Short Introduction. In *Proceedings on Symposium on Computer Geometry*, volume 21, Bratislava, 2012. Slovak University of Technology in Bratislava, Publishing House of STU.

[3] P. J. Davis. *Interpolation and Approximation*. Dover Publications Inc., New York, 1975. Republication, with minor corrections, of the 1963 original published in Blaisdell Publishing Co., New York, with a new preface and bibliography.

[4] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Soc. for Industrial and Applied Math., 1996.

[5] R. Farouki and H. P. Moon. Bipolar and multipolar coordinates. In *The Mathematics of Surfaces IX*, pages 348–371. Springer, London, 2000.

[6] R. T. Farouki and C. Y. Han. Robust plotting of generalized lemniscates. *Applied Numerical Mathematics*, 51(2-3):257–272, 2004.

[7] J. D. Lawrence. *A Catalog of Special Plane Curves*. Dover, New York, 1972.

[8] E. H. Lockwood. *A Book of Curves*. Cambridge University Press, Cambridge, 1967.

[9] M. Marden. *Geometry of Polynomials*. American Mathematical Society, Providence, R.I., 1966.

[10] A. I. Markushevich. *Theory of Functions of a Complex Variable*. PrenticeHall, 1965.

[11] M. Paluszny. Contour representation in 2D and 3D with lemniscates. available online: http://www.researchgate.net/publication/228605816_CONTOUR_REPRESENTATION_IN_2D_AND_3D_WITH_LEMNISCATES, 2005. last time accessed: 26/01/2016.

[12] M. Paluszny, G. Montilla, and J. R. Ortega. Lemniscates 3D: a CAGD primitive? *Numerical Algorithms*, 39(1-3):317–327, 2005.

[13] P. D. Sampson. Fitting conic sections to "very scattered" data: An iterative refinement of the Bookstein algorithm. *Computer Graphic and Image Processing 18*, pages 97–108, 1982.

# Publications

## Conference and journal publications

Gemeranová, M.: *3D lemniscates and their usage in data approximation.* In: G – slovenský časopis pre geometriu a grafiku, Vol. 12, No. 24: Journal. Bratislava: Slovak University of Technology, 2016. pp. 15-26. ISBN 13336-524X.

Gemeranová, M. – Chalmovianský, P.: *Data Approximation Using Lemniscates – Adding One Focus Method With Sequentially Adding Input Points.* In: Proceedings of Conference on Applied Mathematics APLIMAT'2016, Vol. 15: Conference Proceedings. Bratislava: Slovak University of Technology, 2016. pp. 406-416. ISBN 978-80-227-4530-7. SCOPUS

Gemeranová, M. – Chalmovianský, P.: *Data Approximation Using Lemniscates – Adding One Focus Method.* In: Proceedings of Slovak-Czech Conference on Geometry and Graphics (Symposium on Computer Geometry SCG'2015), Vol. 24: Conference Proceedings. Plzeň: Publishing service, 2015. pp. 105-111. ISBN 978-80-86843-48-3.

Gemeranová, M. – Chalmovianský, P.: *Data approximation using multifocal lemniscates – Algorithm of doubling and removing lemniscate's foci.* In: Proceedings of Conference on Applied Mathematics APLIMAT'2015, Vol. 14: Conference Proceedings. Bratislava: Slovak University of Technology, 2015. pp. 332-340. ISBN 978-80-227-4143-3. SCOPUS

Gemeranová, M. – Chalmovianský, P.: *Planar reconstruction using multifocal lemniscates.* In: Proceedings of Symposium on Computer Geometry SCG'2013, Vol. 22: Conference Proceedings. Bratislava: Slovak University of Technology, 2013. pp. 61-66. ISBN 978-80-227-4041-8.

Gemeranová, M. – Chalmovianský, P.: *Approximation of points in Euclidean plane using multifocal lemniscate.* In: Proceedings of Spring Conference on Computer Graphics SCCG'2013: Conference Materials and Posters, Vol. 29: Conference Proceedings. Bratislava: Comenius University, 2013. pp. 41-44. ISBN 1335-5694.

Gemeranová, M.: *Využitie viacohniskových lemniskát pri aproximácii dát v euklidovskej rovine.* In: Geometria a jej aplikácie 2013: Seminar Proceedings. Bratislava: Slovenská technická univerzita, 2013. ISBN 978-80-227-3894-1.
http://www.math.sk/gaja/abstract/2013/Abstrakt_Gemeranova_GAJA2013.pdf

## Research projects

*Využitie lemniskát pri aproximácii dát.*
UK/298/2015. 2015. Comenius University grant (€1000).
Principal investigator: Mgr. Mária Gemeranová.

*Aproximácia množiny bodov v rovine pomocou lemniskát.*
UK/357/2013. 2013. Comenius University grant (€1000).
Principal investigator: Mgr. Mária Gemeranová.

*COMENIANA – metódy a prostriedky digitalizácie a prezentácie 3D objektov kultúrneho dedičstva.* ITMS: 26240220077. 2012 – 2015. Operational program ASFEU.
Participated in the project as a part of the research team – investigator.
(Note: Project was co-financed by the European regional development fund.)

## Theses

Gemeranová, M.: *Aproximácia dát zmenou parametrov vhodnej viacohniskovej lemniskáty.* Rigorous thesis (RNDr.). Comenius University, 2015.

Gemeranová, M.: *Curve and Surface Reconstruction Using Multifocal Lemniscates.* Project of dissertation. Comenius University, 2014.