JOSAI MATHEMATICAL MONOGRAPHS
Vol. 2 *(2000) pp. 77 - 86*

Faithful plotting on a two dimensional pixel space

Tomokatsu SAITO*
Sophia University

Yuji KONDOH†
Takuma National College of Technology

Yoshihiko MIYOSHI‡
Saitama Women's Junior College

Taku TAKESHIMA§
Fujitsu Laboratories Limited

Abstract

Drawing the figure of the real zeros of a bivariate polynomial, in other words, the graph of a real algebraic function, is an old problem not easy to answer[4]. The problem to obtain the real solution curves of a general bivariate function exactly is still a very hard one. This paper presents a method to plot the solution curves precisely, of an equation in terms of a bivariate polynomial with rational coefficients. The method is based on a principle for a graph to be mathematically precise. According to the principle, an algorithm is given to obtain the solution curves. The effectiveness of the proposed principle and the practicality of the algorithms are demonstrated by examples. The presented algorithm is implemented as an alternative implicit function plotting package for Risa/Asir, a non-commercial computer algebra system developed at Fujitsu Labs.

1 Introduction

We study how to draw the zeros of a mathematically defined bivariate function on a display device which has a small but finite resolution, such as a CRT display or paper printer. The problem is commonly recognized to be an easy one. By the limitation of resolution of a display device, a curve is usually drawn, regardless of connected or non-connected, as a set of "spots" that occupy finite areas. A curve mathematically can not have a width, however. Thus, it is not easy to give a mathematically well defined answer to the question: In what sense and to what extent the drawn figure is precise? Usually a graph, we know, shows only the outline of the true mathematical curve. By employing

*saito@mm.sophia.ac.jp
†kondoh@dc.takuma-ct.ac.jp
‡miyoshi@saitama.email.ne.jp
§takesima@flab.fujitsu.co.jp

algebraic computation, however, we can give a mathematically definite meaning to the plotted figure of the solution curves.

Let f be a function $D \to \mathbb{R}$, where $D \subset \mathbb{R}^2$. We propose a *Plotting Principle*[10] for plotting the solutions of the equation $f = 0$.

Principle

1. **A plotted "spot" has a two dimensional volume.**

2. **A plotted "spot" has a point common with the solution.**

3. **A not-plotted "spot" has no common points with the solution.**

Here, "plot" means to put a foreground color on a pixel of display device.

A plotting algorithm that satisfies the above principles is not easy to obtain in general. The difficulty lies in the construction of effective procedure to decide whether a root exists or not in a given area.

For general bivariate equations, no theories and algorithms that meet the principles is known. For a univariate polynomial equation, Sturm's theorem provides the answer. For a multivariate polynomial equation, however, mathematical theory is not yet completed. For a bivariate polynomial equation, a theory presented by Saito [9] is an answer to the problem. Improvements of computational complexity is still left for the actual algorithm, however.

In this paper, we present, on the basis of Saito's work, a basic algorithm that satisfies the **Principle** for bivariate polynomials with rational coefficients.

The outline of the algorithm is the following.

- Divide the domain over which the graph is plotted into a family of small sets, called *Cells.*

- Decide whether the given bivariate polynomial has isolated singular points, and if it does, locate the Cells in which the isolated singular points exist.

- Decide whether a Cell contains solutions on its boundary.

- Decide whether a Cell contains solutions which have no intersections on the boundary, i.e., entirely contained components.

Currently predominant algorithms practically used for plotting planer curves are mostly based on numerical computing. On the other hand, algorithms based on algebraic computing have been mostly developed for mathematical studies such as locating singular points exactly and also knowing exact geometric structures, e.g., connected components and branches, and only few are employed to graph the curves[2].

Tracing type is widely used in numerical plotting algorithms and many variations are studied. In the tracing algorithms, however, appropriate initial points is required but not easy to obtain in general, and careful examination are necessary not to miss all the branches. Moreover, those algorithms risk to miss very small structures unless they dare to use sufficiently small step size of tracing which always costs too much for such rare cases. Mitra[6] proposed a scanning algorithm based on the theorem of midpoint. His algorithm is similar in a sense that it checks whether a rectangular contains a solution. Numerical errors due to fixed size floating point operations is unavoidable; this limits the numerical plotting algorithms applicable only for polynomials with small degrees and small coefficients.

On the other hand, the proposed algorithm based on algebraic computing has the following advantages.

- Plotting the zeros of a bivariate polynomial at any required accuracy.
- Detecting and locating singular points, especially isolated ones.
- Detecting and locating small structures.
- Able to enlarge the obtained figure by reusing the processing computed data.

2 Criterion of Drawing Graph

2.1 Cells and Resolution

Let D be a connected compact subset of $\mathbb{R}^2$. Let $f : D \to \mathbb{R}$ be defined on D, and continuous.

By the proposed principle, plotting a solution of $f(x, y) = 0$ is to decide for every Cells defined on D whether or not it intersects the solution set $\{(x, y) \in D | f(x, y) = 0\}$.

Let $\{C_i\}_{i=1,\ldots,m}$ be a family of m subsets of D.

Definition 1
We call $\{C_i\}_{i=1,\ldots,m}$ a family of Cells, or resolution, defined on D if it satisfies the following conditions.

1. *For every i C_i is a connected closed subset of D.*
2. $D = \bigcup_{i=1}^{m} C_i$.
3. $C_i^\imath \bigcap C_j^\imath = \emptyset \ (i \neq j)$.

In the above definition, $C_i^\imath$ and $C_j^\imath$ are a set of all interior points of C_i and C_j respectively.

2.2 Plotting Functions

Our main concern in plotting the zeros of a function is to compute the following function called a Character.

Definition 2
We call a function $\chi : \{\mathcal{C}_i\} \to \{0, 1\}$ a Character of f with resolution $\{\mathcal{C}_i\}$ defined on D if it satisfies the condition:
If $\chi(\mathcal{C}_i) = 0$, then $f(x, y) \neq 0$ for every point (x, y) in $\mathcal{C}_i$.

The above Character function χ guarantees that if $\chi(\mathcal{C}_i) \neq 0$ then the given function f does never vanish all over the Cell $\mathcal{C}_i$. It does not guarantee the existance of zeros of f in the Cell $\mathcal{C}_i$ when $\chi(\mathcal{C}_i) = 1$, instead.

Many of the conventional plotting algorithms have the strategy that they first determine several points roughly on the solution curve by some means and then interpolate in between those points. A problem of such algorithms is that they can not guarantee the accuracy of the points lying in the interpolated intervals, no matter how they guarantee the accuracy of the first solution points.

Note that the Character function approach can reduce the computational cost if a resolution is set as coarse as is required. Also note that the definition of χ is very general here.

2.3 Faithful Character

The Character defined above is too general to be used for plotting zeros of a function. We propose the following stronger definition called Faithful Character. The Faithful Character gives a plotting function that we really desire.

Definition 3
A Character χ of f with resolution $\{\mathcal{C}_i\}$ on D is a Faithful Character if the following condition is satisfied:
If $\chi(\mathcal{C}_i) = 1$ then in $\mathcal{C}_i$ there exits a point (x, y) such that $f(x, y) = 0$.

For general bivariate functions, there are no known algorithms to compute Faithful Character, except for the one which applies to bivariate polynomials with rational coefficients [9, 11].

3 Implementation

In the rest of this paper, let f be a bivariate polynomial with rational coefficients. We assume, without generality, that $f \in \mathbb{Q}[x, y]$ is square free, since the square free part of f determine the same algebraic curves as that of f does.

3.1 Style of Cells

As a practical convention to reduce the computational complexity, we deal with cases where D is a rectangular with its boundary. The accuracy of plotting depends on the resolution. To increase the accuracy for interested parts of D is possible by further dividing the Cells with coarse resolution into smaller Cells with finer resolution.

The division boundaries are taken from rational points to avoid unnecessary hard computation: If a boundary is an irrational point, operations over algebraic extension is needed. This is very heavy load, especially when applying Sturm's theory to deciding the existence of zeros on the boundaries of Cells.

According to the above consideration, we put

$$D = \{(x, y) | a_0 \le x \le a_m, \quad b_0 \le y \le b_n\}.$$

where a_0, a_m, b_0 and b_n are rational numbers. Dividing the rectangular D into $m \times n$ small rectangular Cells $C_{i,j}$, $(i = 1, ..., m; j = 1, ..., n)$, viz.

$$C_{i,j} = \{(x, y) \in \mathbb{R}^2 \mid a_{i-1} \le x \le a_i, \quad b_{j-1} \le y \le b_j\}.$$

where $a_0 < a_1 < \cdots < a_m$ and $b_0 < b_1 < \cdots < b_n$.

3.2 Detection of Zeros on Cell Boundaries

For every line from (a_i, b_0) to (a_i, b_n) $(i = 0, 1, ..., m)$, and (a_0, b_j) to (a_m, b_j) $(j = 0, 1, ..., n)$, we compute Sturm sequence of $f(a_i, y)$ and $f(x, b_j)$ to locate the Cell boundaries that intersect the zeros of $f(x, y)$. Sturm sequence is computed once per each line. By this process we can determine exactly the existence of the zeros on the boundaries of each Cell $C_{i,j}$.

3.3 Detection of Singular Points

A singular point of an algebraic curve defined by $f(x, y) = 0$ is defined as a point on the curve such that $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ both vanish. Thus, in order to obtain the singular points, we just have to solve the following polynomial equations.

$$\begin{aligned} \frac{\partial f}{\partial x} &= 0 \\ \frac{\partial f}{\partial y} &= 0 \\ f(x, y) &= 0 \end{aligned}$$

As we are dealing only a fraction free polynomial $f(x, y)$, the ideal generated by the left-hand-side polynomials of the above equations is zero dimensional. It is comparably easy to compute the solution at a desired accuracy, if we compute lexicographic Gröbner basis or Rational Univariate Representation (RUR) [8]. In case of special need, we can obtain strict boundaries of the solution by operating in the algebraic extension fields.

A Cell determined to be newly plotted (be put foreground color on) at this stage contains isolated singular points or singular points on a entirely interior solution.

3.4 Detection of Interior Solutions

After applying boundary solution detection and singular point detection, the remaining possibility for the zeros of f is to form a small structure that is entirely contained in a Cell. Therefore, form now on we assume the following without generality: If $\mathcal{C}_{i,j}$ is not plotted yet, f does not have zeros on $\partial\mathcal{C}_{i,j}$ and it does not have isolated singular points within the interior of $\mathcal{C}_{i,j}$.

Thus, the remaining task is to determine whether or not a small structure that is entirely contained in a (not yet plotted) Cell $\mathcal{C}_{i,j}$.

If such a solution exists, it must be a Jordan curve from the structure of solutions of algebraic equation. Therefore,

$$\begin{aligned}\frac{\partial f}{\partial y} &= 0\\ f(x,y) &= 0\end{aligned}$$

must have solution in $\mathcal{C}_{i,j}$. And if no such solutions exist, the above simultaneous equations does never have a solution in $\mathcal{C}_{i,j}$. The process for this detection is similar to the process of detecting singular points.

4 Examples

The algorithm is implemented in C language as an external function of Risa/Asir[7] developed at Fujitsu Labs. By an algorithmic similarity, singular points and entirely interior solutions are simultaneously obtainable; in practice they are integrated into one.

To exemplify by the real data, we treat the following polynomial named "Heart."

Table 1: Computing Time at 100dpi (400 × 400 points)

Equation	Boundary Solutions	Singular Points + Interior Solutions
Heart	17.7074	1.4137
Heart−1/5	17.7386	3.5683

Table 2: Computing Time at 600dpi(2400×2400 points)

Equation	Boundary Solutions	Singular Points + Interior Solutions
Heart	117.08	1.4137
Heart$-1/5$	117.40	3.5683

From the timing we notice that the computing time spent for singular point detection and interior solution detection is very little compared to the total time.

$$\begin{aligned} Heart(x,y) &= \frac{93392896}{15625}x^6 \\ &+ (\frac{94359552}{625}y^2 + \frac{91521024}{625}y - \frac{249088}{125})x^4 \\ &+ (\frac{1032192}{25}y^4 - 36864y^3 - \frac{7732224}{25}y^2 - 207360y + \frac{770048}{25})x^2 \\ &+ 65536y^6 + 49152y^5 - 135168y^4 - 72704y^3 \\ &+ 101376y^2 + 27648y - 27648 \end{aligned}$$

The $Heart(x,y)$ function has isolated singular points: They are located at the center of small circles in Figure 1.

In order to locate isolated singular points by Risa/Asir, the plotting system computed the prime decomposition of the ideal

$$< Heart(x,y), \frac{\partial Heart(x,y)}{\partial x}, \frac{\partial Heart(x,y)}{\partial y} > .$$

The solution is the following.

$$\left\{ \begin{array}{l} 351y + 386 \\ 123201x^2 - 17150 \end{array} \right.$$

$$\left\{ \begin{array}{l} 76y + 41 \\ 2888x^2 - 8575 \end{array} \right.$$

Figure 2 is the graph of $Heart(x,y) - 1/5 = 0$. The isolated singular points of $Heart(x,y) = 0$ have to become small ellipses. The two center points is barely recognized as elipses at 600 dpi; however, the two points at the sides entirely buried within a Cell at 600 dpi. (So they are circled.) If we enlarge one of the latter by 256 times, we can recognize a ellipse as is shown in Figure 3.

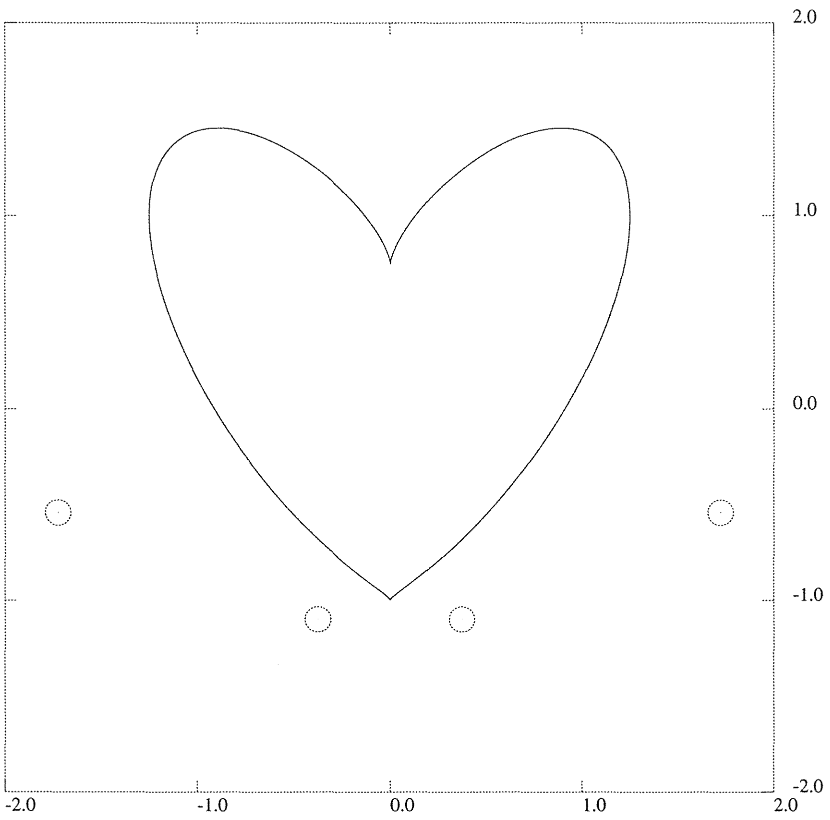

Figure 1: Heart function

5 Conclusion

A new concept of plotting faithful figures of planer curves defined implicitly by bivariate functions is proposed. The essence of the concept is to locate the Cell which intersects with the planer curve instead locating the points on the planer curve. Such algorithm is based on deciding whether the Cell in question shares a common point with the planer curve by exact rational computation. According to the principle, algorithms to obtain faithful figures for algebraic functions is presented. The algorithm is implemented on a computer algebra system and is used for plotting difficult functions hard to plot by ordinary means. The performance timing is shown for an interesting algebraic curve in heart shape. As was expected, it detected singular points, isolated points and very small structures which is smaller than the resolution. Many variations of the algorithm can be designed to compromise the time-faithfulness contention.

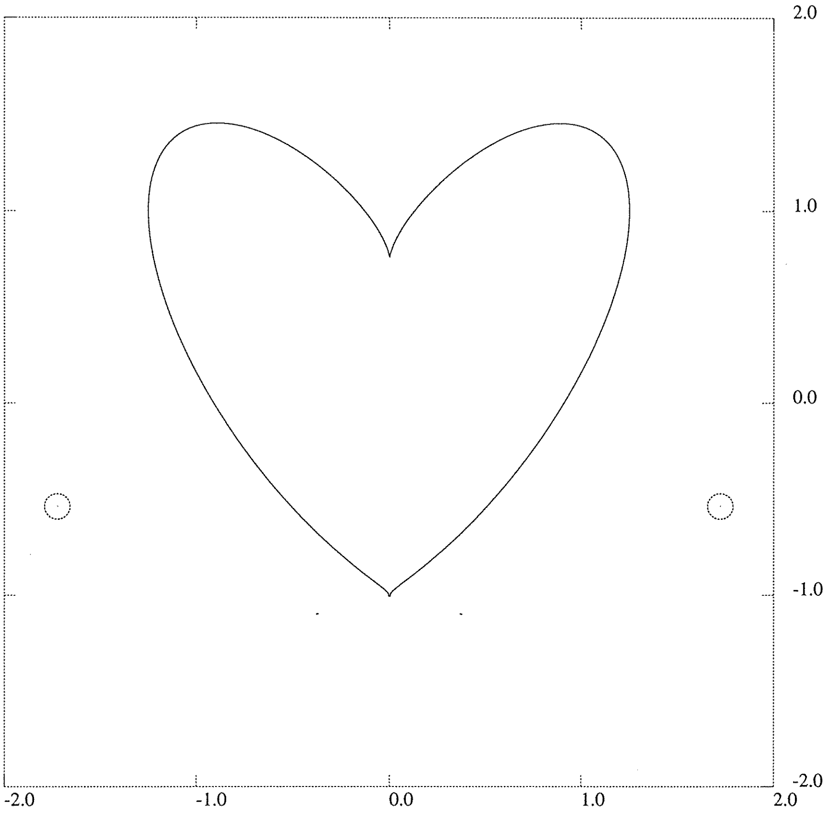

Figure 2: Heart function $-1/5$

References

[1] Becker,T.,Weispfenning,V.:*Gröbner Bases*, Springer-Verlag, New York, 1993

[2] Gebauer, R., Kalkbrener, B., Wall, F., Winkler, F.: CASA: A Computer Algebra Package for Constructive Algebraic Geometry, In *Proceedings of ISSAC'91*, ACM, New York, 1991.

[3] Collins,G.E.,Loos,R.:Real Zeros of Polynomials, *Computer Algebra Symbolic and Algebraic Computation* (Buchberger,B.,Collins,G.E. and Loos,R. eds.), Springer-Verlag, New York, 1983, 83–94

[4] Fateman,R:Honest Plotting, Global Extrem, and Interval Arithmetic, In *Proceedings of ISSAC '92*, ACM, New York,1992, pp. 216–223

[5] Knuth,D.E.:*The Art of Computer Programming second edition*, Vol. 2. Addison-Wesley, 1981

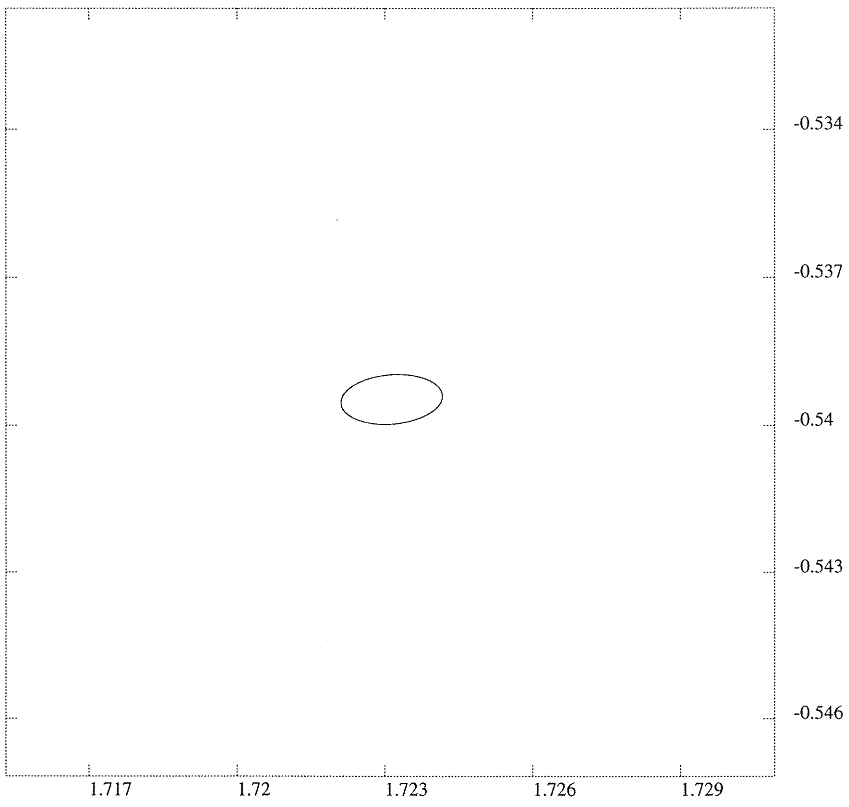

Figure 3: Expand for Fig 2

[6] Mitra, A. K.:Graphing Implicit Functions $f(x, y) = 0$, In *Applied Mathematics and Computation*, Vol. 39, pp. 199–205, 1990.

[7] Noro,M.,Takeshima,T.:Risa/Asir — A Computer Algebra System, In *Proceedings of ISSAC '92*,New York,1992, pp. 387–396

[8] Gonzalez-Vega, L., Trujillo, G.:Using Symmetric Function to Describe the Solution Set of a Zero Dimensional Ideal, In *Proc. AAECC-11 (LNCS 948)*, pp.232-247.

[9] Saito,T.:An extension of Sturm's theorem to two dimensions, In *Proceedings of the Japan Academy*,73,Ser.A(1997),pp. 18–19

[10] Saito, T and Kondoh, Y and Miyoshi, Y and Takeshima, T, :Displaying Real Solution of Mathematical Equations, In *J. JSSAC*, Vol.6, pp. 2–21, 1998.

[11] Saito, T and Kondoh, Y and Miyoshi, Y and Takeshima, T, :Faithful plotting of real curves defined by bivariate rational polynomials, In *Trans. of Information Processing Society of Japan*, to appear.