ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1996
Seria: GEOMETRIA I GRAFIKA INŻYNIERSKA z. 1 Nr kol. 1318

ARNOŠT **ŠARMAN**
Department of Computer Science, Technical University of Ostrava
Tř. 17 listopadu, 708 33 Ostrava-Poruba, Czech Republic
e-mail, arnost Sarman @vsb. cz

CONICS IN GRAPHIC SYSTEMS

Summary. In this paper, an alogorithm for drawing the conics is outlined. The proposed algorithms are not based on the use of numerical methods. They use purely the methods of projective geometry.

STOŻKOWE W SYSTEMACH OPROGRAMOWANIA GRAFICZNEGO

Streszczenie. W pracy rozpatruje się problem procedur umożliwiających kreślenie elipsy, paraboli i hiperboli. Inaczej niż w szeroko stosowanych systemach, np. AutoCAD, Microstation, krzywe te nie byłyby aproksymowane łukami okręgów, lecz kreślone na podstawie klasycznych metod geometrii rzutowej.

1. Introduction

In the graphical systems which are widely used in Czech Republic, there is not given a high attention to conica. These system use mostly circles and arcs. Using the arc, there is solved not only the drawing of ellipse but possible polylines are constructed from arc and lines. Arcs are used as parts of osculate and hyperosculate circles. There are several reasons for that. The most frequently given reason is that arc has alredy been solved and programmed so why to complicate the system with other procedures which are not actually asked for. But here is one solution which is not complicated neither for numerical computation nor for PC speed.

The other simple conic sections, parabolae and hyperbolae are not implemented in many graphic systems. Even if these curves can be widely used, e.g., in building industry. It is of course possible to object that using the graphical systems in different professions can be supported by various libraries. Absent parts of the systems can be programmed later. But we do not accept that explanation. As further mentioned, the algorithms do not demand much for implemention. They can be easily used in the basic menu of graphical systems.

We can also use these algorithms for the determination of parameters for showing the corresponding entity – for finding the centre, the size and the location of axis, etc. The drawing itself can be done by using the traditional algorithms. Which conic sections and party of geometry are we going to consider?

2. Ellipse, hyperbola and parabola in graphical systems

In the graphical systems Autocad and Microstation PC, which are widely used in our republic, only the ellipse as a whole is solved. It is, of course, possible to draw a circle and also its parts. However, for the construction of ellipse, we must know the center and the axis.

The basic element for the construction of ellipse is the arc and the whole ellipse is constructed. Ellipse (not only in those two mentioned systems) is drawn with the help of hyperosculation circles in vertex, and another osculation circles which connect these arc. This „connection", in certain ratio of semiaxis lenght, is even possible to see. So when the user of systems draws the ellipse, she has to show it by its center, size and the location of axis. The ellipse is then shown in four hyperosculation circle arcs in vertex and four connectioning arcs. If we want to draw only a part of ellipse, it is neccesary to eliminate the other part. When the ellipse or its parts are required to be used in other than in the basic position, it is possible to use other procedures. For example, rotation, scale and move, must be used for making the final effect. Many systems are not capable of solving parabolic or hyperbolic conic sections. If we need to draw them, they must be further programmed. The solution by the circle arcs has perhaps the only advantage: it is fast because it uses only few procedures that are well done from the point of view of programming. From the point of view of geometry, they are not good enough. One reason is that not everything is possible to be substituted with the circle arc. Namely, if the curvature of the curve is approaching to zero.

So it is necessary to look for other algorithms which will solve the drawing of the whole conic sections (simple ones – ellipses, parabolae, hyperbolae and, of course, circles) and their parts as well. At the same time, there is the demand for speed and also for the curve to be the real curve not only a collection of circle arcs.

Which means for building of these procedures have we had?

3. Algorithms for drawing conic section by means of projective geometry

The algorithms that we will use for drawing the conic sections belong to the basic parts of projective geometry. They use only the basic elements as is the ratio, cross ratio Brianchon's and Pascal's theorem, polar properties of conic sections, etc. By the means, it is possible to draw the circle arc which will connect the given points without the necessity to draw or count the whole conic and to trim the useless rest after that. It is also possible to draw the tangents in points of conic section and tangents from points that do not belong to conic. This can be also used for zooming, etc.

Example 1

A conic section is given by 5 points *A, B, C, D* and *E*. The problem is to construct another point.

For solving this example, it is possible to use the definition of conic section in projective geometry. We will find another point of conic section as an intersection of two correspondig pencils of lines. These pencils of lines are the projective pencils of lines that have the center points *S* and *S'* [*S* # *S'*], respectively. In Figure 1 *A* = *S* and *B* = *S'*. The lines connecting the points *C, D, E* with the center *S* create one pencil of lines *S (c, d, e,...)*. The lines connecting the points *C, D, E* with center *S'* create the second pencil of lines *S' (c', d', e',...)*.

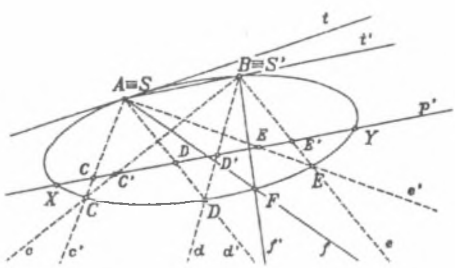

Fig. 1

The lines *f* and *f'* that are the lines of the beams *S* and *S'* intersect in the next point of the conic. Figure 1 shows the construction of points of a conic. The construction is based on the definition. There is also shown the possible solution of tangents in given points.

If we want to program this task, we will need the following programming procedures:

a) Determine the equation of line given by its two endpoints. (I)

b) Determine the intersection of two lines. (II)

c) Given three points *A, B, X*, find the fourth point *Y* so that the cross ratio *(A, B, X, Y)* equals to a given value (the points *A, B, X, Y* are collinear). (III)

Durring the programming, we need only the common means typical for computer languages without special mathematic means. We do not need, e.g., the procedures for solving the linear or nonlinear equations, etc.

The important advantage of this solution is the small number of steps. That is why we can simplify the result making it more accurate and faster.

Figure 1 shows the proposed solution works:

The intersection is found by using the operations I, II and III and by other notion from projective geometry (involution, etc.) There is no need (if we do not demand it) to draw the conics or to solve their equations.

Example 2

Connect two points with the elliptic arc that passes through a given third point, and that is tangent to given lines in its end points.

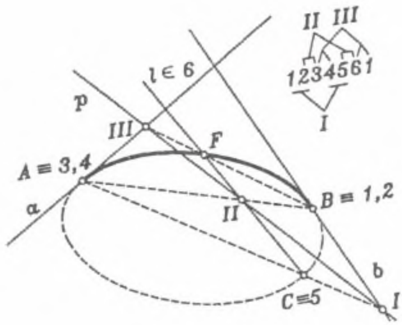

Fig. 2

In this case, it is possible to continue in similar way as in Example 1. We will construct other points of conic and we will test which part of conic is to be drawn. It is good not to use here the definition of conic but the Pascal's or Brianch's theorem instead. Figure 2 depicts an example of how the process can proceed. We choose two points. The first one is the given point determining the conic. The second is chosen as a point on the line segment defined by those points that are to be connected by the arc. Let us denote *b* the tangent line determined by the points 12, and I the intersection of *b* with the line 45. Thus the line determined by the points I and II is the Pascal's line.

Now, the algorithm can be briefly outlined as follows:

a) Choice the point *II* on the line segment *AB*.

$$II \in (A, B)$$

b) The intersection of the line *I II* with the tangent a is the third point denoted *(III)*, of the Pascal's line.

$$(I\ \ II * a) = III.$$

c) The intersection of the line connecting the points *III* and *B* with the line connecting the points *C* and *II* determines the new point, denoted *F*, of the conic.

$$(III\ B * C\ II) = F \ldots \text{ next point of conic section.}$$

Here are more possible ways of speeding up or slowing down the computation in dependence on how the programs are worked out.

But this is not the topic of this contribution.

For the solution to the polar properties, it is possible to use effectively other parts of projective geometry, in the similar way (involution, system of quadratic points series or lines, etc.). For making algorithms in graphical system, it is very advantagenous to combine these algorithms with traditionally used numerical or iterative methods.

The algorithms must be extended as follows:

1) Construction of tangent (tangent point) of conic. (Pascal's or Brianchon's theorem)
2) Construction of center of conic (Polar properties of conic.)
3) Construction of conjugate diameters of conic (Polar properties of conic.)
4) Construction of axis of conic (Rytz's construction.)

As you can see, the number of procedures for computing the mentioned tasks is relatively low. Except the case of looking for the fix points, when the projection of points on the circle is used, the procedures mentioned in introduction fully satisfy our needs.

4. Conclusion

In this paper I wanted to recall some forgotten methods of projective geometry which can be found in curriculums of PhD studies and maybe also elsewhere. The above mentioned algorithms were programmed by students of the 3rd year of Engineering informatic course of study at FEI of Technical University of Ostrava in subject Geometry for computer graphics. I think that, after professional re-programming, these algorithms would suit many graphical systems. I can show the examples of these procedures for those possible interested.

Streszczenie

Praca dotyczy procedur umożliwiających kreślenie elipsy, paraboli i hiperboli. Autor proponuje, by inaczej niż w szeroko stosowanych systemach (AutoCAD, Microstation), krzywe te nie były aproksymowane łukami okręgów, lecz by były kreślone klasycznymi metodami geometrii rzutowej. W odnośnych algorytmach zostałyby wykorzystane podstawowe konstrukcje geometrii syntetycznej związane z zachowaniem dwustosunku podziału odcinka oraz twierdzeniami Pascala i Brianchona.

Przedstawione dwa przykłady konstrukcji krzywych stopnia drugiego określonych alternatywnie pięcioma punktami bądź jednym punktem i parą stycznych wraz z punktami styczności.

Precyzując związane z konstrukcją najprostsze procedury Autor zauważa, że oparte na nich przykładowe, choć nieprofesjonalne, programy zostały napisane przez studentów informatyki Uniwersytetu Technicznego w Ostrawie.