Using MAPLE's RegularChains library to automatically classify plane geometric loci

Francisco Botana[1] and Tomás Recio[2]

[1] Universidad de Vigo, Spain
fbotana@uvigo.es,
http://webs.uvigo.es/fbotana/
[2] Universidad de Cantabria, Spain
tomas.recio@unican.es,
http://www.recio.tk

Abstract. We report a preliminary discussion on the usability of the RegularChains library of MAPLE for the automatic computation of plane geometric loci and envelopes in graphical interactive environments. We describe a simple implementation of a recently proposed taxonomy of algebraic loci, and its extension to envelopes of families of curves is also discussed. Furthermore, we sketch how currently unsolvable problems in interactive environments can be approached by using the RegularChains library.

Keywords: parametrical systems solving, constructible sets, dynamic geometry.

1 Introduction

A Dynamic Geometry System (DGS) is a computer program that allows an accurate on–screen drawing of geometric diagrams and their interactive manipulation by mouse dragging or similar device. A key issue of these systems is their ability to display the trajectory of a point that depends on another one bound to a linear path, that is, a geometric locus. Traditionaly, DGS strategy to display loci consists of sampling the linear path and, for each sample, plotting the corresponding position of the locus point. Some ad–hoc heuristics are then applied to join contiguous points, ending with a visually continuous locus.

A locus can be seen as the projection on the space of its coordinates of the surface defined by the problem constraints, and Gröbner based elimination was proposed as a technique to find algebraic knowledge about loci [1]. Nevertheless, since Zariski closures are obtained as result, loci equations can include spurious points, and complete 1–dimensional objects can even appear due to construction degeneracies. A finer analysis of loci problems can be done through a recently proposed taxonomy [2], and it has been implemented using the GröbnerCover algorithm [3]. Here we describe an alternative implementation of the taxonomy using MAPLE's RegularChains library. The results seem to be competitive when dealing with the above class of loci. Despite license problems can emerge when linking Maple with widespread free DGSs as GeoGebra, the special attention given by RegularChains to constructible sets justifies considering this library. Furthermore, other subpackages will allow extending dynamic geometry issues to real geometry.

2 Loci as parametric problems

We consider a locus problem where all constraints can be described as polynomials. Let (x, y) the coordinates of a generic locus point, and $x_1, \ldots, x_n$ the remaining variables. So, the problem can be seen as a parametric polynomial system $F \subset \mathbb{Q}[x, y][x_1, \ldots, x_n]$, where x, y are parameters and $x_1, \ldots, x_n$ variables. See [4] for an precise setting description and a full account on the taxonomy. Roughly speaking, the taxonomy classifies locus points as

- normal: if for these parameter values the system has finite number of solutions;
- non-normal: if the system has infinite solutions for these parameter values.

Thus, in order to perform this classification we can count the number of solutions of F in RegularChains by using the command ComplexRootClassification. For the sake of illustration, we consider the limaçon of Pascal, an algebraic curve obtained as a locus as follows. Let $Q(x_1, x_2)$ be a point on the circle centered at the origin and with radius 2, $O(0, 2)$ and $P(x, y)$ a point at distance 1 from Q and lying on the line $\overline{OQ}$. As Q glides on the circle, P describes the limaçon. The polynomial system is

$$F = \{x_1^2+x_2^2-4, (y-2)x_1-x_2x+2x, x_1^2+x_2^2+(-2x)x_1+(-2y)x_2+x^2+y^2-1\},$$

and the Maple code for classifying the locus points,

```
with(RegularChains); with(ConstructibleSetTools):
with(ParametricSystemTools); R := PolynomialRing([x1, x2, x, y]):
F := [x1^2+x2^2-4,(y-2)*x1-x2*x+2*x,..
..x1^2+x2^2-2*x*x1-2*y*x2+x^2+y^2-1]:
crc := ComplexRootClassification(F, 2, R):
map(x -> [Info(x[1],R),x[2]],crc);
```

returns three constructible sets where the system has exactly one complex solution,

$$[[x^4 + (2y^2 - 9)x^2 + y^4 - 9y^2 + 4y + 12], [y - 2, x^2 + y^2 - 4y + 4, 2y - 3]],$$

$$[[4x^2 - 15, 2y - 3], [1]], [[x^2 + y^2 - 4y + 3], [2y - 3]],$$

and two other constructible sets determining two system solutions

$$[[x, y - 2], [1]], [[4x^2 - 3, 2y - 3], [1]].$$

Therefore, all locus points are normal. In order to further analyze these points (recall the taxonomy [4]), we count the number of solutions of the original system plus each constructible set, once inverted the roles of parameters and variables. If we get infinite solutions for a constructible set, we declare it as a special locus part. In the case we are dealing with, that happens for the third constructible set, $[[x^2+y^2-4y+3],[2y-3]]$:

```
R := PolynomialRing([x, y, x1, x2]):
F := [x1^2+x2^2-4,(y-2)*x1-x2*x+2*x,..
..x1^2+x2^2-2*x*x1-2*y*x2+x^2+y^2-1,x^2+y^2-4*y+3]:
H := [2*y-3]:
crc := ComplexRootClassification(F, H, 2, R):
map(x -> [Info(x[1],R),x[2]],crc);
```

returns

$$[[[[x_1, x_2-2],[1]],\infty]],$$

meaning that the circle centered at O must be labeled as a special locus part. Note that, although algebraically pertinent, this circle comes from a degeneracy in the construction, since it stems from the coincidence of points O and Q.

3 Envelopes as loci

Envelope computation can be also seen as solving a parametric system. As in loci, elimination can include spurious factors. Thus, applying the taxonomy can drive to improvements when automatically computing envelopes in a DGS. Envelope points are classified as normal or non–normal points as in loci. Consider, for instance, the envelope of horizontal lines through a point on the unit circle. Using the above Maple commands we obtain

```
R := PolynomialRing([x, y, x1, x2]):
F := [x1^2+x2^2-1, y-x2, x1]:
crc := ComplexRootClassification(F, 2, R):
map(x -> [Info(x[1],R),x[2]],crc);
```

$$[[[[y+1],[1]],[[y-1],[1]],1]],$$

that is, the envelope is $y=\pm 1$, where all points are normal. Nevertheless, studying the number of solutions of the system, interchanging roles of parameter and variables, we get for the first constructible set

```
R := PolynomialRing([x, y, x1, x2]):
F := [x1^2+x2^2-1, y-x2, x1,y+1]:
H := [1]:
crc := ComplexRootClassification(F, H, 2, R):
map(x -> [Info(x[1],R),x[2]],crc);
```

$$[[[[x_1, x_2 + 1], [1]], \infty]].$$

Thus, strictly following the taxonomy, the line $y = -1$ should be declared as special, while it is an ordinary part of the envelope. On the contrary, the envelope of lines passing through $A(0, 1)$ and a point gliding along the unit circle exactly consists of A, whereas MAPLE finds a constructible set where all points are normal, $[[y-1], [x]]$, and another one with a non–normal point, $[[x, y-1], [1]]$, that is, the point A. Note that the first constructible set should be removed from the envelope result (it comes from a construction degeneration) while the taxonomy would label it as special. Currently, we do not know how to automatically distinguish the two cases.

4 Further work

Many dynamic geometry constructions can only be expressed in a essential semialgebraic way. Each time a segment is used in a locus construction, for example, it is highly probable that the above computations include extra parts, since traditionally DGSs using these methods replace the segment by the whole line. Also, envelopes are defined in $\mathbb{R}$, while we work in $\mathbb{C}$. The subpackage SemiAlgebraicSetTools will allow to extend the class of dynamic geometry problems able to be automatically solved. As a simple illustration, consider two circles each one with a point gliding along it. The locus of their midpoint is a 2–dimensional part of the plane, currently only descriptible in any DGS by displaying a more or less accurate screen. There are techniques and algorithms capable of giving more precise answers. Our future work will study such problems in the context of dynamic geometry.

Acknowledgment

Both authors were partially supported by the Spanish Ministerio de Economía y Competitividad and by the European Regional Development Fund (ERDF), under the Project MTM2011–25816–C02–02.

References

1. Francisco Botana, José Valcarce. A Software Tool for the Investigation of Plane Loci. Math. Comput. Simul. 61, 139–152 (2003)
2. Miguel Abánades, Francisco Botana, Antonio Montes, Tomás Recio. An Algebraic Taxonomy for Loci Computation in Dynamic Geometry. Comput. Aided Des. submitted (2014)
3. Antonio Montes, Michael Wibmer. Gröbner Bases for Polynomial Systems with Parameters. J. Symb. Comput. 45, 1391–1425 (2010)
4. Miguel Abánades, Francisco Botana, Antonio Montes, Tomás Recio. Software Using the Gröbner Cover for Geometrical Loci Computation and Classification. these proceedings