Software using the GRÖBNER COVER for geometrical loci computation and classification

Miguel A. Abánades[1], Francisco Botana[2], Antonio Montes[3], and Tomás Recio[4]

[1] Universidad Complutense de Madrid, Spain
abanades@ajz.ucm.es,
[2] Universidad de Vigo, Spain
fbotana@uvigo.es,
http://webs.uvigo.es/fbotana/
[3] Universitat Politècnica de Catalunya, Spain
antonio.montes@upc.edu,
http://www-ma2.upc.edu/montes/
[4] Universidad de Cantabria, Spain
tomas.recio@unican.es,
http://www.recio.tk

Abstract. We describe here a properly recent application of the GRÖBNER COVER algorithm (GC) providing an algebraic support to Dynamic Geometry computations of geometrical loci. It provides a complete algebraic solution of locus computation as well as a suitable taxonomy allowing to distinguish the nature of the different components. We included a new algorithm LOCUS into the Singular grobcov.lib library for this purpose. A web prototype has been implemented using it in Geogebra.

Keywords: Locus, Taxonomy, Dynamical Geometry, Groebner Cover.

1 Introduction

[5] One of the defining characteristics of Dynamical Geometry (DG) is obtaining geometrical loci problems. Neverthless, the existing DG software are not able to give a satisfactory answer. This is the case for the first standard DG systems developed in the late 80's (such as Cabri and The Geometer's Sketchpad), as well as for more recent ones, such as GeoGebra or Java Geometry Expert. In DG systems, it is often the case that locus determination is purely graphical, producing an output that is not robust enough and not reusable by the given software. Moreover, extraneous objects are frequently appended to the true locus as side products of the locus determination process.

Using the GRÖBNER COVER we are able to give an exact algebraic result, allowing moreover to give a taxonomy of the different locus components. In

[5] Authors 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. The third author was supported by projects Gen. Cat. DGR 2009SGR1040 and MICINN MTM2009-07242.

section 2 we give a summary of the Gröbner Cover. In section 3 we give the locus taxonomy obtained using it, and describe the new Locus algorithm. Finally, in section 4 we show characteristic examples justifying the taxonomy and showing its functionality. A web prototype [3] has been implemented using the new algorithm in Geogebra.

2 The Gröbner Cover

The Gröbner Cover algorithm for discussing parametric polynomial ideals gives a canonical description, classifying the solutions by their characteristics (number of solutions, dimension, etc.). This is used here for defining a taxonomy of geometrical loci and to implement it in the new algorithm Locus. It is included in the Singular grobcov.lib library allowing its use by DG software.

The Gröbner Cover provides the analog of the *reduced Gröbner basis* of an ideal for parametric ideals. Its existence was proved by Wibmer's Theorem [6], and the method and algorithms were developed in [5]. Montes implemented in Singular the grobcov.lib library [7], whose actual version incorporates Kapur-Sun-Wang algorithm [2] for computing the initial Gröbner System used in grobcov algorithm, as described in [4], and recently also the Locus algorithm described here. A more detailed description can be seen in [1].

Let $\mathbf{y} = y_1, \dots, y_n$ be the set of variables and $\mathbf{u} = u_1, \dots, u_m$ the set of parameters. Given a generating set $F = \{f_1, \cdots, f_s\} \subset \mathbb{Q}[\mathbf{u}][\mathbf{y}]$ of the parametric ideal $I = \langle F \rangle$ and a monomial order $\succ_{\mathbf{y}}$ in the variables, the grobcov algorithm determines

- the unique *canonical partition* of the parameter space $\mathbb{C}^m$ into locally closed sets (*segments*) with associated *generalized reduced Gröbner basis*:

$$GC = \{(S_1, B_1, \mathrm{lpp}_1), \dots, (S_r, B_r, \mathrm{lpp}_r)\}.$$

- The segments S_i are disjoint locally closed subsets of $\mathbb{C}^m$ and $\oplus_i S_i = \mathbb{C}^m$.
- The basis B_i of a segment S_i has *fixed set of leading power products* (lpp), who ensures that the type of solutions is the same over all points of the segment, and is the *generalized reduced Gröbner basis* of $\langle F \rangle$ over the segment S_i.
- The lpp's are included in the output, even if they can be seen on the basis, to characterize the segments and facilitate the applications.
- Moreover, if the ideal is homogeneous, the lpp's are characteristic of the segment, no other segment having the same lpp's.

The generalized reduced Gröbner basis B_i of a segment S_i is formed by a set of monic I-regular functions over S_i. An I-regular function, representing an element of the basis, allows a full-representation in terms of a set of polynomials that specialize for every point $\mathbf{u}_0$ of the segment, either to the corresponding element of the reduced Gröbner basis of the specialized ideal $I_{\mathbf{u}_0}$ after normalization, or to zero. It also allows a generic representation given by a single polynomial that specializes well on an open subset of the segment and to zero on the remaining points of it. Usually it is sufficient with the generic representation, and we can, if needed, compute the full representation from it using the EXTEND algorithm.

The segments S_i are given in canonical P-representation, given by a set of prime ideals of the form

$$\mathrm{Prep}(S) = \{\{\mathfrak{p}_i, \{\mathfrak{p}_{ij} : 1 \leq j \leq r_i\}\} : 1 \leq i \leq s\}$$

representing the set:

$$S = \bigcup_{i=1}^{s} \left(\mathbb{V}(\mathfrak{p}_i) \setminus \bigcup_{j=1}^{r_i} \mathbb{V}(\mathfrak{p}_{ij}) \right).$$

Each $\mathbb{V}(\mathfrak{p}_i) \setminus \bigcup_{j=1}^{r_i} \mathbb{V}(\mathfrak{p}_{ij})$ is a *component* of the segment, and its representative $\{\mathfrak{p}_i, \{\mathfrak{p}_{ij} : 1 \leq j \leq r_i\}\}$, by abuse of language, is also denoted a component when there is no ambiguity. $\mathfrak{p}_i$ is called the *top* of the component, and $\{\mathfrak{p}_{ij} : 1 \leq j \leq r_i\}$ the *holes*.

3 Locus taxonomy

A geometric locus is a set of points satisfying some conditions. Locus computation is an important issue in Dynamic Geometry, where the term locus generally refers to loci of the following kind: determine the trajectory determined by the different positions of a point P (tracer), corresponding to the different positions of a second point M (mover) along the path where it is constrained by the construction. Nevertheless, the actual existing DG software do not provide correct algebraic solutions.

Using the GRÖBNER COVER we are able to give a precise algebraic answer. We shall consider more general locus problems in the plane with a tracer point $P(u_1, u_2)$, whose coordinates $\mathbf{u} = (u_1, u_2)$ are considered as parameters and the remaining coordinates, distances, etc. $\mathbf{y} = y_1, \ldots, y_n$ of the construction as variables. The locus problem will give rise to a system $F \subset \mathbb{Q}[\mathbf{u}][\mathbf{y}]$. The locus determination consists now in obtaining the conditions over the parameters $\mathbf{u}$ for which there are solutions for the $\mathbf{y}$.

In the GRÖBNER COVER the values of the parameters and variables are considered over $\mathbb{C}$. Thus we can provide only locus solutions over the complex $\mathbb{C}$, whereas DG is interested in the real projection, who is not always obvious. Moreover, we consider only problems that can be be formulated exactly in terms of equations with coefficients in $\mathbb{Q}$. We also restrict the study to plane loci problems, even if it can be generalized to higher dimensional spaces. Let

$$\mathbb{V}(F) = \{(\mathbf{u}, \mathbf{y}) \subset \mathbb{C}^{2+n} : \forall f \in F, f(\mathbf{u}, \mathbf{y}) = 0\}$$

be the set of solutions of F. Denote π_1 and π_2 the projections onto the parameter and variable spaces:

$$\begin{array}{rcl} \pi_1 : \mathbb{C}^{2+n} & \longrightarrow & \mathbb{C}^2 \\ (\mathbf{u}, \mathbf{y}) & \mapsto & \mathbf{u} \end{array} \qquad \begin{array}{rcl} \pi_2 : \mathbb{C}^{2+n} & \longrightarrow & \mathbb{C}^n \\ (\mathbf{u}, \mathbf{y}) & \mapsto & \mathbf{y} \end{array}$$

```
$L \leftarrow$ **Locus**$(G)$
Input: $G = \{(S_i, B_i, \mathrm{lpp}_i) : i \leq i \leq s\}$ the Gröbner cover of an ideal in $\mathbb{Q}[\mathbf{u}][\mathbf{y}]$,
    where $S_i = \cup_j C_{ij}$ and $C_{ij} = \{(\mathfrak{p}_{ij}, \{\mathfrak{p}_{ijk} : 1 \leq k \leq r_{ij}\}) : 1 \leq j \leq r_i\}$.
Output: $L$, the components of the P-representation of the locus
    $L = \{\{\mathfrak{q}_i, \{\mathfrak{q}_{ij} : 1 \leq j \leq s_i\}, \mathrm{type}_i\} : 1 \leq i \leq s\}$
begin
  $C_1$ = Select the segments of $G$ with $\dim(\mathrm{lpp}_i) = 0$ # normal-segments
  $C_1$ = Specialize the basis on every component of $C_1$ and mark the component
    'Normal' if the basis continues to depend on the $\mathbf{u}$'s and 'Special' if not
  $C_2$ = Select all the components of the segments of $G$ with $\dim(\mathrm{lpp}_i) > 0$
    # non-normal segments
  $L_1$ = **LCUnion**$(C_1)$,      marking the components of $L_1$ as
      'Normal' or 'Special' inheriting the character of the full
  $L_2$ = **LCUnion**$(C_2)$;
  Mark the components of $L_2$ of $\dim(C) = 0$ and $\dim(C) > 0$ respectively as
    'Accumulation' and 'Degenerate' components
  $L = L_1 \cup L_2$
end
```

Table 1. Locus algorithm

The taxonomy that we give is motivated by the interpretation of the solutions in a lot of loci problems of different kind (see [1]):

Generic locus L associated to the parametric polynomial system $F(\mathbf{u}, \mathbf{y})$ is the set $L = \pi_1(\mathbb{V}(F)) \subset \mathbb{C}^2$,(i.e. the set of values of the coordinates of the tracer for which there exist solutions).

Taxonomy:

- **Normal locus**: are the points $\mathbf{u} \in \mathbb{C}^2$ of the locus L for which $\dim(\pi_2(\mathbb{V}(F) \cap \pi_1^{-1}(\mathbf{u}))) = 0$ (i.e. the set of points in the parameter space that correspond to a single (or a finite number of) positions of the variables).
  - **Normal components**: A component C_s of the normal locus is *normal* if $\dim(C_s) = \dim(\pi_2(\mathbb{V}(F) \cap \pi_1^{-1}(C_s)))$ (i.e. the components of the normal locus whose different points are generated by different points of the variables).
  - **Special components**: A component C_s of the normal locus is *special* if $\dim(C_s) > 0$ and $\dim(\pi_2(\mathbb{V}(F) \cap \pi_1^{-1}(C_s))) = 0$ (i.e. the components of the normal locus of dimension 1 that are generated by a single (or a finite number of) points of the variables).
- **Non-normal locus**: are the points $\mathbf{u} \in \mathbb{C}^2$ of the locus L for which $\dim(\pi_2(\mathbb{V}(F) \cap \pi_1^{-1}(\mathbf{u}))) > 0$ (i.e. the set of points in the parameter space that correspond to infinite positions of the variables).
  - **Degenerate components**: are the components C_d of the non-normal locus with $\dim(C_d) > 0$ (i.e. the components of the non-normal locus of dimension 1).

- **Accumulation components (points)**: are the components C_a of the non-normal locus with $\dim(C_a) = 0$ (i.e. the zero dimensional components of the non-normal locus).

Problems in section 4 are chosen to justify the taxonomy. The geometric relevance of this algebraic classification of the different components of a locus is open to interpretation by the user. Dynamic Geometry systems could present the collection of different parts (with the corresponding typology) of the computed locus, letting to the user the decision of which pieces are to be considered or discarded in a particular context.

Based on our experience (see section 4), we tend to discard the degenerate components as geometrically irrelevant, as they usually correspond to degenerate instances of a construction, such as two coincident vertices in a triangle. However, we consider the accumulation points as forming part of the (geometric) locus, since they represent special points that are determined by infinitely many values of the variables. The special components are generally also discarded, as they are generated by a single position of the mover, but sometimes they can be useful.

We designed the Locus algorithm (Table 1) that takes the output of the grobcov and classifies the appropriated components of the segments following the defined taxonomy. We show now some examples justifying our taxonomy and its functionality.

4 Applications and Functionality

4.1 Pascal's Limaçon

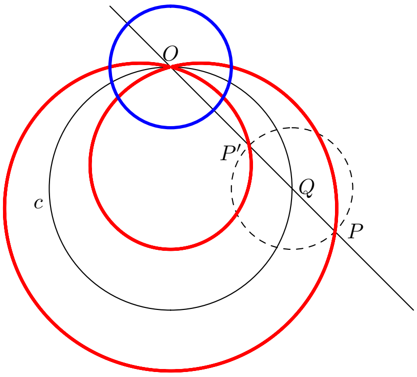

Fig. 1. Pascal limaçon

The following Problem is considered here to justify the definitions of "Normal" and "Special" components of the "Normal Locus". Consider the following locus problem. Let O be a fixed point on a circle c of radius r, and l be a line passing through $O(0,r)$ and $Q(y_1,y_2)$, a general point on c. Let $P(u_1,u_2)$ be a point on l such that $distance(P,Q)=k$, where k is a constant. The limaçon of Pascal is the locus set traced by P as Q moves along c, as shown on Figure 1. Setting $r=2$ and $k=1$, the ideal (set of equations) determining the locus is:

$$F=\langle y_1^2+y_2^2-4,\ (u_1-y_1)^2+(u_2-y_2)^2-1,\ (2-y_2)u_1+y_1(u_2-2)\rangle$$

Computing the solution using grobcov and locus algorithms in Singular we do the following:

Input:

```
LIB "grobcov.lib";
ring R=(0,u1,u2),(y1,y2),lp;
short=0;
ideal F= y1^2+y2^2-4, (u1-y1)^2+(u2-y2)^2-1, (2-y2)*u1+y1*(u2-2);
def L=locus(grobcov(F));
"locus(grobcov(F))="; L;
```

Output:

```
locus(grobcov(F))=
[1]:
   [1]:
      _[1]=(u1^4+2*u1^2*u2^2-9*u1^2+u2^4-9*u2^2+4*u2+12)
   [2]:
      [1]:
         _[1]=1
   [3]:
     Normal
[2]:
   [1]:
      _[1]=(u1^2+u2^2-4*u2+3)
   [2]:
      [1]:
         _[1]=1
   [3]:
      Special
```

Locus algorithm determines and characterizes two components: a "Normal" component that is the Pascal limaçon's concoid, and a "Special" circle generated by the single mover point O, for which the construction is degenerate. Usually these Special components are to be discarded by Dynamical Geometry software, and the algorithm returns it as "Special" letting to the user the decision about its consideration.

4.2 Offset of a circle

Now we want to justify the definition of "Accumulation" and "Degenerate" components of a locus considering the locus of the offset of a circle of radius 1 at distance 1. The given circle has as equation $g : y_1^2 + y_2^2 - 1$. The family of circles who generate the envelope is $F = (u_1 - y_1)^2 + (u_2 - y_2)^2 - 1$, where (u_1, u_2) is some point of the envelope. To compute the envelope we have to add the equation $\frac{\partial F}{\partial y_1}\frac{\partial g}{\partial y_2} - \frac{\partial F}{\partial y_2}\frac{\partial g}{\partial y_1}$, ensuring that the envelope is tangent in each point to a curve of the family. We have to consider thus the following ideal:

$$H = \langle y_1^2 + y_2^2 - 1, (u_1 - y_1)^2 + (u_2 - y_2)^2 - 1, y_1 u_2 - y_2 u_1 \rangle.$$

The standard method will eliminate (y_1, y_2) to obtain the envelope. But we can consider (y_1, y_2) as the mover and take (u_1, u_2) as the tracer using the ring `R=(0,u1,u2),(y1,y2),lp;` and the command `locus(grobcov(H))`.

Doing so we also obtain two components: The "Normal" component consisting of the circle $u_1^2 + u_2^2 - 4$, plus the "Accumulation" point component (u_1, u_2) at the center generated by all the circles. The accumulation points are to be considered as part of the offset by Dynamical Geometry software,.

4.3 Improvements: Detecting *Bad* Mover Positions

Locus algorithm as described in Table 1 is incomplete, and must be improved. It assumes that the generic segment of the Gröbner Cover has basis $\{1\}$, as we do not expect a locus that contents the whole plane except some curves. Nevertheless, in the next example the generic segment does not have basis $\{1\}$.

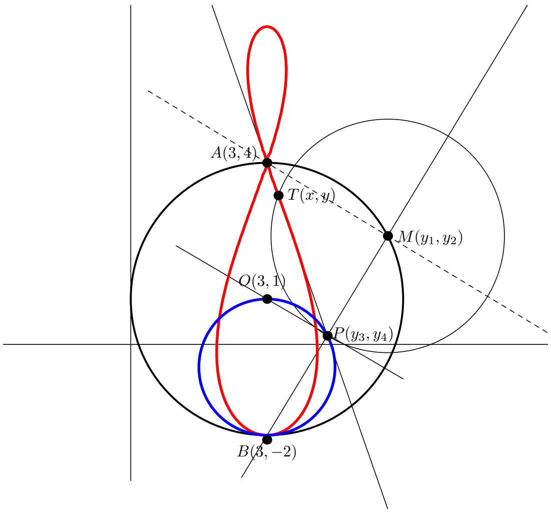

Fig. 2. Locus described by T (and P) as M runs along its circle.

The reason is that there is a point of the mover for which the construction is degenerate, and would give rise to a bidimensional solution. It is necessary to eliminate this point of the mover to obtain the correct expected locus.

We programmed our algorithm to be able to detect and eliminate such bad mover point positions producing degenerate solutions in the locus.

Consider the following locus construction (see Figure 2). The mover $M(y_1, y_2)$ runs over the circle with center at $O(3,1)$ and radius OA, where $A = (3,4)$. We consider the line parallel to the line AM passing through O and the line perpendicular to it passing through the point $B = (3, -2)$. Both lines intersect at point $P(y_3, y_3)$. Consider the line AP and the circle with center M and radius MP. We consider this intersection as the tracer point(s): $T(x, y)$ and $T' = P$.

The polynomial system describing the problem is the ideal F given by

$$\begin{aligned}F = \langle &(y_1-3)^2+(y_2-1)^2-9,\ (4-y_2)(y_3-3)+(y_1-3)(y_4-1),\\ &(y_1-3)(x_1-y_1)-(4-y_2)(y_4-y_2),\\ &(y_4-4)x-(y_3-3)y+4y_3-3y_4,\\ &(x-y_1)^2+(y-y_2)^2-(y_1-y_3)^2-(y_2-y_4)^2\rangle\end{aligned}$$

For the computation we use the ring `ing R=(0,x,y),(x1,x2,y1,y2),lp;`. With this improvement we obtain the proper two irreducible "Normal" components

$$\begin{aligned}&\mathbb{V}(x^2-6x+y^2+y+7),\\ &\mathbb{V}(x^4-12x^3+2x^2y^2-13x^2y+236x^2-12xy^2+78xy-1200x+y^4\\ &\quad -13y^3+60y^2-85y+1495).\end{aligned}$$

The output produced by the algorithm includes a message informing of the removal of *bad* mover positions (point $A(3,4)$ in our case).

Geometrically, the problem is that when the mover is on the point $(y_1 = 3, y_2 = 4)$, the line AM is not defined. The algorithm eliminates the segments of the Gröbner Cover containing this point that is detected on the generic segment.

This locus is example 9 in our prototype [3]. By clicking the *Find locus* button, we obtain the locus description shown on Figure 2.

References

1. Miguel Abanades, Francisco Botana, Antonio Montes, Tomas Recio. "An Algebraic Taxonomy for Locus Computation in Dynamic Geometry". Computer Aided Design (2014), submitted.
2. Deepak Kapur, Yao Sun, Dingkang Wang. "A new algorithm for computing comprehensive Gröbner systems". Proceedings of ISSAC'2010, (2010), 29–36. ACM Press, New York.
3. F. Botana, Locus Prototype: http://webs.uvigo.es/fbotana/LocusGC/ (2012).
4. Antonio Montes. "Using Kapur-Sun-Wang algorithm for the Gröbner Cover". Proceedings of EACA 2012, (2012), 135–138. ed: R. Sendra, C. Villarino. pub: Universidad de Alcalá de Henares
5. Antonio Montes, Michael Wibmer. "Gröbner bases for polynomial systems with parameters". *Jour. Symb. Comp.* **45** (2010), 1391–1425.
6. Michael Wibmer. "Gröbner bases for families of affine or projective schemes". *Jour. Symb. Comp.* **42**: (2007), 803–834.
7. Singular. http://www.singular.uni-kl.de/ (Last accessed February 2014).