The `ConstructibleSetTools` and `ParametricSystemTools` modules of the `RegularChains` library in `Maple`

Changbo Chen[1], Marc Moreno Maza[1], François Lemaire[2]
Wei Pan[1], Liyun Li[1], Yuzhen Xie[1]
[1] University of Western Ontario, London ON, N6A 5B7, Canada
[2] Université de Lille 1, 59655 Villeneuve d'Ascq Cedex, France

Solving systems of parametric polynomial equations symbolically is in demand for an increasing number of applications such as program verification, optimization and the study of dynamical systems. Gröbner bases and triangular decompositions are classical techniques for processing parametric systems. Recent research has focused on enhancing theories and algorithms to meet the practical requirement of these systems. The `ParametricSystemTools` is a new module of the `RegularChains` library [7] in `Maple` which implements *comprehensive triangular decompositions* (CTD), a new algorithmic approach introduced in [1] for studying polynomial systems with parameters.

Constructible sets are the geometrical objects naturally attached to triangular decompositions, as polynomial ideals are the algebraic concept underlying the computation of Gröbner bases. This relation becomes even more complex and essential in the case of systems with infinitely many solutions, and in particular with parametric polynomial equations. The `ConstructibleSetTools` module of the `RegularChains` library is, up to our knowledge, the first computer algebra package providing constructible set as a type and exporting a rich collection of operations for manipulating constructible sets. Meanwhile, this module provides routines in support of solving parametric polynomial systems.

During this software presentation, we give a brief overview of these new modules, see Section 1. Then, we provide an experimental comparison with related software packages, see Section 2. Finally, we present three applications: an algorithmic realization of *Chevalley's Theorem* for constructible sets, a classification problem in *Classical Invariant Theory* and a *verification software tool* for polynomial system solvers, Section 3 gives one of them.

1 Specification and Implementation

The interplay between constructible sets and triangular decompositions was underlying since the early work of W.T. Wu [11] and through the work of his followers such as D.M. Wang [10]. This relation became explicit in [1] where the authors develop procedures for computing the set theoretical difference and the intersection of two constructible sets represented by triangular decompositions. These operations, which are at the core of the `ConstructibleSetTools` module, have led us to the following implementation design.

We represent a constructible set C by a list $[[T_1, h_1], \ldots, [T_e, h_e]]$ of so-called *regular systems*, where each T_i is a regular chain and each h_i is a polynomial regular w.r.t. the saturated ideal of T_i. Then the points of C are formed by the points that belong to at least one quasi-component $W(T_i)$ without canceling the associated polynomial h_i. Since the zero set of a regular system is always nonempty (in fact unmixed), a constructible set is empty if and only if it is given by an empty list of regular systems. However, such representation may contain redundant (or superfluous) components: the zero sets $W(T_i) \setminus V(h_i)$ and $W(T_j) \setminus V(h_j)$ of two of the defining regular systems of C may not be disjoint. The operation `MakePairwiseDisjoint` of the module `ConstructibleSetTools` replaces the representation of C by an irredundant one. Therefore our design allows lazy evaluation (or unevaluated expressions) for efficiency reasons while providing efficient simplification tools, as reported in [3].

Redundant components appear naturally when decomposing polynomial systems with both symbolic and numerical methods. A typical situation of this phenomenon is when computing the union of (non-disjoint) constructible sets $C_1, \ldots, C_e$. In this case one may want to replace the C_i's by pairwise disjoint constructible sets $D_1, \ldots, D_f$ such that each C_i can be written as a union of some of the D_j's. The operation `RefiningPartition` of `ConstructibleSetTools` computes such "intersecion-free basis".

As mentioned before, one of the purposes of this module is to support the implementation of `ParametricSystemTools` and thus the solving of parametric polynomial systems. In particular, the two simplification tools `MakePairwiseDisjoint` and `RefiningPartition` are used for partitioning the parameter space during the computation of a CTD. In return, the CTD is the back-engine for advanced operations of `ConstructibleSetTools` such as the image (or pre-image) of a constructible set by a rational map, providing an algorithmic realization of Chevalley's Theorem for constructible sets (Corollary 14.7 in [5]).

2 Comparison with Related Packages

Several software packages, many of them in the computer algebra system `Maple`, are available for solving parametric polynomial systems. Among them: the `Epsilon` library by D.M. Wang, the `DISPGB` by A. Montes [8] and `SACGB` by A. Suzuki and Y. Sato [9]. In [1] we report on comparative benchmarks between `Epsilon`, `DISPGB` and our implementation of the CTD in the `ParametricSystemTools`. The CTD can solve all the test systems that we use (all taken from A. Montes, D.M. Wang and D. Lazard) whereas the other packages fail (generally for memory consumption reasons) on some of them. In [2] we report on comparative benchmarks between `SACGB` and our CTD; using the examples of [9] we reach again conclusions favorable to the CTD.

During our software presentation, we aim at demonstrating that the output decompositions produced by our `ComprehensiveTriangularize`, that is, our CTD command, are often more concise than those produced by the other methods. Moreover, they are easy to handle thanks to the `ConstructibleSetTools` module.

3 Applications

During our software presentation, we will discuss the three applications mentioned above. Below, as an example of CTD, we show one from *Classical Invariant Theory*, which is taken from [6].

While regarding u and v as parameters, the following polynomials g_1 and g_2 define two families of elliptic curves:

$$g_1 = x^3 + ux - y^2 + 1 \text{ and } g_2 = x^3 + vx - y^2 + 1.$$

In invariant theory, a classical question is to ask whether there exists a linear rational map f between these two curves:

$$f : (x, y) \mapsto \left(\frac{A\,x + B\,y + C}{G\,x + H\,y + K}, \frac{D\,x + E\,y + F}{G\,x + H\,y + K} \right).$$

Assuming for simplicity that the origin is mapped to the origin, which sets $C = F = 0$, we obtain the following polynomial equations:

$$(\star)\begin{cases} K^3-1=0 \\ (vA+3G)K^2-u=0 \\ (vB+3H)K^2=0 \\ A^3+G^3+vAG^2-D^2G-1=0 \\ B^3+H^3+vBH^2-E^2H=0 \\ (E^2-2vBH-3H^2)K-1=0 \\ (D^2-2vAG-3G^2)K=0 \\ 3A^2B+3G^2H+vBG^2+2vAGH-2GDE-HD^2=0 \\ 3AB^2+3GH^2+vAH^2+2vBGH-2HDE-GE^2=0 \\ (3GH+vBG+vAH-DE)K=0. \end{cases}$$

A CTD of the system $(\star)$, as computed by our command `ComprehensiveTriangularize`, is

```
[regular_chain,regular_chain,regular_chain,regular_chain,regular_chain,regular_chain,
 regular_chain,regular_chain,regular_chain,regular_chain,regular_chain],
[[constructible_set, [1, 2, 3, 10, 11]],
 [constructible_set, [4, 5, 6, 7, 8, 9]],
 [constructible_set, [1, 2, 3]]]
```

There are three constructible sets in this output:

$$\begin{cases} C_1 & : & u^3=v^3=9, \\ C_2 & : & u\ =v\ =0, \\ C_3 & : & u^3=v^3,\ u\neq 0,\ v^3\neq 9. \end{cases}$$

The union of the C_i's is the answer to our question; taking the union produces a single component, with equation $u^3=v^3$. In the above output the eleven regular chains $T_1,\ldots,T_{11}$ are used as follows to describe the solutions for the unknowns A,B,D,E,G,H,K: for $i=1,2,3$ the solutions arising from the parameter values in C_i are given by the regular chains whose indicies are in the list associated with C_i. Above each C_i the "geometry" of the solution set is different (different degrees), which explains the partition $\{C_1,C_2,C_3\}$.

4 Conclusion and Future Work

This software presentation introduces the audience to two new modules of the `RegularChains` library for manipulating constructible sets and solving parametric polynomial systems. One of the main motivations for developing a rich collection of commands for handling constructible sets was the need to partition the parameter space during the computation of a comprehensive triangular decomposition. However, the `ConstructibleSetTools` module is also of great interest as an independent package. For example, as shown in [4], it serves well as a *program verifier*.

Another ongoing project is the development of a module dedicated to *parametric semi-algebraic sets*, allowing the manipulations of parametric polynomial systems with equations, inequations and inequalities. The mathematical theory of comprehensive triangular decomposition of such sets is actually well engaged. We hope that in a near future this new module will provide a helpful support for problems in real algebraic geometry.

References

[1] C. Chen, O. Golubitsky, F. Lemaire, M. Moreno Maza, and W. Pan. *Comprehensive Triangular Decomposition*, volume 4770 of *LNCS*, pages 73–101. Springer Verlag, 2007.

[2] C. Chen, F. Lemaire, L. Liyun, M. Moreno Maza, W. Pan, and Y. Xie. The constructiblesettools and parametricsystemstools modules of the regularchains library in maple. In *Proc. of the International Conference on Computational Science and Applications*. IEEE Computer Society Press, 2008.

[3] C. Chen, L. Li, M. Moreno Maza, W. Pan, and Y. Xie. On the representation of constructible sets. In *Proc. of Milestones in Computer Algebra*, pages 103–108, 2008.

[4] C. Chen, M. Moreno Maza, W. Pan, and Y. Xie. On the verification of polynomial system solvers. *Frontiers of Computer Science in China*, 2(1):55–66, 2008.

[5] D. Eisenbud. *Commutative Algebra with a View Toward Algebraic Geometry*. Springer-Verlag, New York-Berlin-Heidelberg, 1995.

[6] I. A. Kogan and M. Moreno Maza. Computation of canonical forms for ternary cubics. In Teo Mora, editor, *Proc. ISSAC 2002*, pages 151–160. ACM Press, July 2002.

[7] F. Lemaire, M. Moreno Maza, and Y. Xie. The `RegularChains` library. In Ilias S. Kotsireas, editor, Maple Conference 2005, pages 355–368, 2005.

[8] A. Montes. A new algorithm for discussing Gröbner bases with parameters. *J. Symb. Comput.*, 33(2):183–208, 2002.

[9] A. Suzuki and Y. Sato. A simple algorithm to compute comprehensive Gröbner bases using Gröbner bases. In *ISSAC*, pages 326–331, 2006.

[10] D. M. Wang. *Elimination Methods*. Springer, New York, 2000.

[11] W. T. Wu. Basic principles of mechanical theorem proving in elementary geometries. *J. Sys. Sci. and Math. Scis*, 4:207–235, 1984.