ISSAC 2011 Software Demonstrations

Communicated by Mike Stillman

Computing the real solutions of polynomial systems with the `RegularChains` library in Maple

Changbo Chen[1], James H. Davenport[2], François Lemaire[3],
Marc Moreno Maza[1], Bican Xia[4], Rong Xiao[1], Yuzhen Xie[1]
[1] University of Western Ontario, Canada; [2] University of Bath, UK;
[3] Université de Lille 1, France; [4] Peking University, China

Computing and manipulating the real solutions of polynomial systems is a requirement for many application areas such as biological modeling, robotics, program verification, to name a few. The `RegularChains` library in Maple provides a collection of tools for dealing with systems of polynomial equations, inequations and inequalities. These tools include isolating and counting the real solutions of zero-dimensional systems, describing real solutions of positive dimensional systems, classifying the number of real roots of parametric systems, finding sample points (thus determining emptiness) of semi-algebraic sets, performing set theoretical operations on semi-algebraic sets as well as computing cylindrical algebraic decompositions. The theory and algorithms underlying these tools are described in [9, 10, 5, 1, 3, 4]. Most commands implementing these tools are part of the `SemiAlgebraicSetTools` module while the others can be found in the `ParametricSystemTools` module or at the top level of the `RegularChains` library itself. All of these commands but one are already present in Maple 15, that is, in the current version of Maple.

1 Design and Specification

One design features of the `RegularChains` library is the use of types for a few key algebraic structures such as *regular chains*, *constructible sets*, *semi-algebraic sets*, etc. This feature, unusual for a Maple package, forces the user to provide command input in an unambiguous manner and eases the manipulation of complex output values. Let us illustrate this design feature with one example. Based on the algorithms of [3], the `RealTriangularize` command decomposes an input semi-algebraic system into finitely many so-called *regular semi-algebraic systems*. An object of type `regular_semi_algebraic_system`. consists of a regular chain, a quantifier-free formula and positive inequalities. The `RegularChains` library provides types for the former two whereas inequalities form a Maple primitive type.

The values of these algebraic types are encoded by expression trees whose leaves are polynomials. For non-trivial examples, these expressions are likely to be large. Hence, by default, the output format of such a value is simply the name of its type, for instance `quantifier_free_formula`. However, two commands `Display` and `Info` provide a pretty printer and parsable printer for the values of the algebraic types exported by the `RegularChains` library. The fact that `RealTriangularize` decomposes any semi-algebraic system into finitely many regular semi-algebraic systems leads to a convenient representation of semi-algebraic sets. Indeed, regular semi-algebraic systems enjoy remarkable properties, which enable an easy implementation of set theoretical operations on semi-algebraic sets, like `Difference` and `Intersection`.

Another design feature is the use of Maple piecewise structure for formatting the output of commands producing a set of "components" (for instance regular semi-algebraic systems). This has at least two advantages. First, this highlights the relations between components. Secondly, this supports lazy evaluation in the form of unevaluated recursive calls, see [3] for details.

Below, we list our main functions, with their specifications, to be illustrated during the demonstration.

- `RealRootIsolate`. For any semi-algebraic system (i.e. system of polynomial equations, inequalities) with finitely many complex solutions, this command isolates all the real solutions by so-called *boxes*; in the the real space $\mathbb{R}^n$, a box is a Cartesian product of n bounded intervals of $\mathbb{R}$.
- `RealTriangularize`. For any semi-algebraic system, this command returns a decomposition into regular semi-algebraic systems. This "solve" command for semi-algebraic systems has no restrictions or limitations. Up to our knowledge, this is the first such command in a computer algebra system.
- `SamplePoints`. For any semi-algebraic system S, this command computes at least one sample point per connected component of S. The sample points are encoded by boxes.
- `RealRootClassification`. For a generically zero-dimensional parametric semi-algebraic system S, this command computes conditions for S to have a prescribed number of real solutions.
- `CylindricalAlgebraicDecompose`. For a set of polynomials F in n variables, this command computes an F-sign invariant cylindrical algebraic decomposition of the real space $\mathbb{R}^n$.
- `Difference`. For any two semi-algebraic sets A and B (represented by regular semi-algebraic systems) this command computes the set-theoretical difference $A \setminus B$.

2 Applications

Our software demonstration is articulated around four application problems.

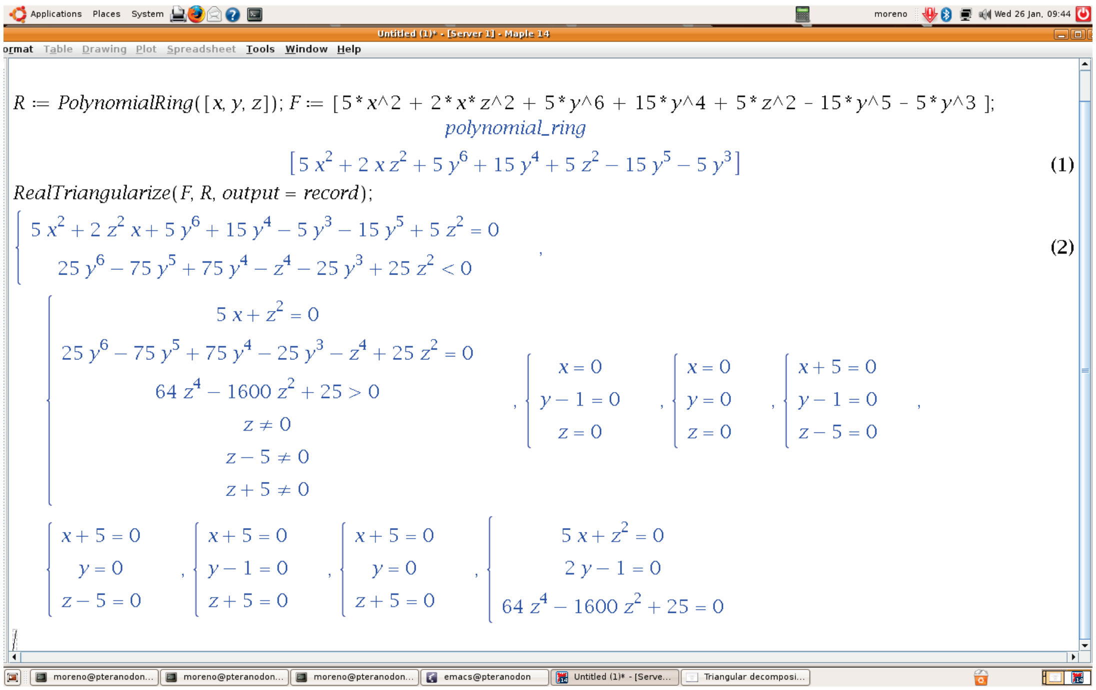
2.1 Branch cut computations. In analysis, a major challenge is the manipulation of "multivalued functions". Regarding them as single-valued functions requires the imposition of branch cuts, which are normally semi-algebraic sets in $\mathbb{C}^n = \mathbb{R}^{2n}$ across which the functions are not continuous. In [6], the authors show how the connectivity of the complement of the branch cuts becomes the question of interest. Since cell adjacency in a cylindrical algebraic decomposition (CAD is explicit, one way (the only practical one known to us) of exploring these connectivity questions is to compute a CAD of $\mathbb{R}^{2n}$ induced by the branch cuts, and construct connected components from this. The *CAD* algorithm of [5] starts with a triangular decomposition of the set of polynomials occurring in the branch cuts, irrespective of how they are linked, whereas the QEPCAD approach [2] takes advantage of knowing how the equalities and inequalities are connected. Nevertheless, [8] shows that the approach of [5] often produces no more cells than QEPCAD.

2.2 Verification of real solvers. On a given input polynomial system, two solving tools may produce correct results that look fairly different. Proving that these two results are equivalent can be a very complex task. Here's an example. Given a triangle with edge lengths a, b, c (denoting the respective edges a, b, c too) the following two conditions C_1, C_2 are both characterizing the fact that the external bisector of the angle of a, c intersects with b on the other side of a than the triangle: $C_1 = a > 0 \wedge b > 0 \wedge c > 0 \wedge a < b + c \wedge b < a + c \wedge c < a + b \wedge \left(b^2 + a^2 - c^2 \leq 0 \vee c(b^2 + a^2 - c^2)^2 < ab^2(2ac - (c^2 + a^2 - b^2))\right)$, $C_2 = a > 0 \wedge b > 0 \wedge c > 0 \wedge a < b + c \wedge b < a + c \wedge c < a + b \wedge c - a > 0$. With the set-theoretical operations on semi-algebraic sets, we can verify the equivalence of C_1 and C_2 by computing $C_1 \setminus C_2$ and $C_2 \setminus C_1$.

2.3 Realization of matroids. Consider a rank n ordinary matroid M and a field $\mathbb{K}$. A classical problem is to ask whether M arises from a finite subset P of the affine space $\mathbb{K}^{n-1}$ as the matroid of the affine dependencies among P. When $\mathbb{K}$ is an ordered field, it is natural to ask whether M is orientable, that is, whether there exists an oriented matroid $\mathcal{M}$ representable over $\mathbb{K}$ and with M as underlying ordinary matroid. Via the notion of a chirotope, one can turn this question into testing the consistency of a semi-algebraic system, which can be done via our `SamplePoints` command.

2.4 Study of the equilibria of biological systems. Many biological system can be modeled as dynamical systems. The library MABsas [7], developed by our third author and his colleagues, can automatically convert a biochemistry reaction network to a system of nonlinear ordinary differential equations governing the reaction. After building the model, the existence of a unique positive equilibrium is an important question, which can be answered with our commands `RealRootClassification` and `RealTriangularize`.

In the screen shot below, the purpose of the MAPLE session is to obtain a description of the real points of the hypersurface *EVE* from the *Algebraic Surface Gallery*. The output of `RealTriangularize` consists of 9 regular semi-algebraic systems. The first one is two-dimensional, the second one is one-dimensional and the other 7 components give points. The entire software demo will be available at `www.csd.uwo.ca/~moreno/`



References

[1] F. Boulier, C. Chen, F. Lemaire, and M. Moreno Maza. Real root isolation of regular chains. In *Proc. of ASCM'09*, 2009.

[2] C. W. Brown. QEPCAD B: a program for computing with semi-algebraic sets using cads. *SIGSAM Bull.*, 37(4):97–108, 2003.

[3] C. Chen, J.H. Davenport, J. May, M. Moreno Maza, B. Xia, and R. Xiao. Triangular decomposition of semi-algebraic systems. In S.M. Watt, editor, *Proceedings ISSAC 2010*, pages 187–194, 2010.

[4] C. Chen, J.H. Davenport, M. Moreno Maza, B. Xia, and R. Xiao. Computing with semi-algebraic sets represented by triangular decomposition. In *Proceedings ISSAC 2011*, 2011.

[5] C. Chen, M. Moreno Maza, B. Xia, and L. Yang. Computing cylindrical algebraic decomposition via triangular decomposition. In *ISSAC'09*, pages 95–102, 2009.

[6] J.H. Davenport. The Challenges of Multivalued "Functions". In *Proceedings AISC/Calculemus/MKM 2010*, volume 6167 of *Lecture Notes in Computer Science*, pages 1–12. Springer Verlag, 2010.

[7] François Lemaire and Asli Ürgüplü. Modeling and Analysis of Biological Systems, 2008. www.lifl.fr/∼urguplu.

[8] N. Phisanbut, R.J. Bradford, and J.H. Davenport. Geometry of Branch Cuts. *Communications in Computer Algebra*, 44:132–135, 2010.

[9] L. Yang and B. Xia. Real solution classifications of a class of parametric semi-algebraic systems. In *Proc. of the A3L'05*, pages 281–289, 2005.

[10] T. Zhang and B. Xia. A new method for real root isolation of univariate polynomials. *Mathematics in Computer Science*, 1:305–320, 2007.