Algebraic Number Comparisons for Robust Geometric Operations

John Keyser, Koji Ouchi
Department of Computer Science
Texas A&M University

In this talk, we describe a method for exact comparison of algebraic numbers, with application to geometric modeling.

Motivation: Computations with algebraic numbers are of key importance in several geometric computations. Algebraic numbers arise as solutions to systems of polynomials—a common operation in many geometric applications, particularly those involving curved objects. Polynomials regularly describe the relationships between even basic geometric objects, for example the (squared) distance between two points. For more complex curved geometric objects, polynomials are often used to describe the actual shapes. Finding solutions to systems of polynomials thus becomes a key operation in numerous geometric applications.

Unfortunately, the robustness issues well-known in traditional computational geometry become even more significant when dealing with algebraic numbers and curved geometry. Bounding and handling the numerical error that arises in these computations becomes more problematic, as does the detection and elimination of degeneracy problems. For reliable computation on curved geometry, we therefore need robust operations on algebraic numbers. We choose to use an exact-computation approach, achieving robustness by eliminating numerical error, while supporting straightforward operations even in the presence of degeneracies.

Our work is particularly motivated from the field of computer-aided geometric design. Finding intersections of geometric objects involves solving systems of polynomials, usually of moderate degree in a few variables. Our methods apply, however, to a far wider range of problems.

Background: Our earlier work focused on techniques for exact manipulation of algebraic curves and 2D points [MAPC00], and applied these to solid modeling, producing the first exact boundary evaluation system [ESOLID04]. Although this work yielded greater robustness by eliminating numerical error, degeneracies could not be handled effectively.

Computations with algebraic numbers has been a topic of recent research interest among a variety of other researchers. LEDA supports a limited set of constructions for algebraic numbers, though it does not solve polynomial systems [LEDA]. The Core library supports a wider variety of number types, including real algebraic numbers [CORE]. Such exact computation approaches have been incorporated in larger projects, such as Exacus [EXACUS] and CGAL [CGAL]. Recently, Emiris and Tsigaridas have developed an approach for exact comparison of algebraic numbers of relatively small degree (at most 4) that is asymptotically faster than an expliit solution [ET04].

Major Results: We describe a method for comparing complex algebraic numbers exactly. More precisely, one can know whether or not the real and imaginary parts of given a pair of complex algebraic numbers are identical. In particular, one can test whether or not the real and imaginary parts of a given complex algebraic number vanish.

Our method is based on the rational univariate reduction (RUR). The RUR computes common roots of systems of multivariate polynomials with rational coefficients. The roots are represented in terms of a set of univariate polynomials with rational coefficients. These polynomials, when evaluated at the roots of another univariate polynomial, yield the coordinates of the common roots of the original system. The RUR can be computed exactly, i.e. the coefficients of these polynomials can be computed to full precision.

As a key advantage over our earlier methods, this RUR method works even in the presence of "degenerate" situations. For example, the RUR approach handles roots of high multiplicity, finds roots at singularities, and works even when the underlying set of roots is positive dimensional. As such, it offers a general method for achieving robust calculations with algebraic numbers.

Our RUR implementation gives an exact representation of points with algebraic coordinates. This allows us to exactly determine geometric predicates such as whether a point lies on a curve or surface. We can also determine how surfaces intersect, e.g. whether the surfaces meet in "general position." Thus, our representation is very general, and can serve as a single representation for all such algebraic points. Although this point representation is more robust than our earlier approach that could not represent degeneracies, it is also less efficient. We therefore propose the use of the RUR computation in a hybrid fashion, using it in cases where there are likely to be difficulties due to degeneracies.

We describe several applications, with special emphasis on geometric modeling. In particular, we describe a new implementation that has been used successfully on certain degenerate boundary evaluation problems. We describe how the RUR can be used to detect when degeneracies occur, and how it can then be combined with a numerical perturbation scheme to achieve an overall more robust boundary evaluation..

References:

[CGAL] Computational Geometry Algorithms Library: http://www.cgal.org

[CORE] Core library: http://www.cs.nyu.edu/exact/core

[ESOLID04] John Keyser, Tim Culver, Mark Foskey, Shankar Krishnan, Dinesh Manocha, "ESOLID: A System for Exact Boundary Evaluation," *Computer Aided Design*, vol. 36, no. 2, pp. 175-193, 2004.

[ET04] Ioannis Emiris, Elias Tsigaridas, "Comparing Real Algebraic Numbers of Small Degree," Proceedings of European Symposium on Algorithms, 2004.

[EXACUS] Efficient and Exact Algorithms for Curves and Surfaces: http://www.mpi-sb.mpg.de/projects/exacus/

[LEDA] Library for Efficient Data Types and Algorithms: http://www.algorithmic-solutions.com/enleda.htm

[MAPC00] John Keyser, Tim Culver, Dinesh Manocha, Shankar Krishnan, "Efficient and Exact Manipulation of Algebraic Points and Curves," *Computer Aided Design*. Vol 32, No. 11. pp. 649-662. 2000. Special Issue on Robustness.