Computing real solutions to polynomial equations

Supervisors: Simon Telen and Monique Laurent

Key words: Polynomial functions, Julia implementation

Problem description:

We consider the problem of solving a system of multivariate equations

$$f_1(x_1, \ldots, x_n) = \cdots = f_s(x_1, \ldots, x_n) = 0, \tag{1.1}$$

where $f_i \in \mathbb{R}[x_1, \ldots, x_n]$ are polynomial functions in the unknowns $x_1, \ldots, x_n$. Geometrically, each of the polynomials f_i defines a hypersurface V_i in $\mathbb{R}^n$, and our solutions form the intersection $V_1 \cap \cdots \cap V_s$ of all these hypersurfaces. In the real plane $\mathbb{R}^2$, we intersect curves.

Example 1.1 ($s = n = 2$). *Let $f_1 = x_1^2 + x_1x_2 + x_2^2 - 1$ and $f_2 = (x_1 - 1)^2 + x_2^2 - 1$. The solutions to $f_1 = f_2 = 0$ are the intersection points of an ellipse and a circle in the real plane. Figure 1 shows that there are two solutions. A more complicated situation is illustrated in the right part of that figure, where the equations each have degree 20.* ◇

Many standard algorithms for solving (1.1) intrinsically work over the complex numbers. That is, they compute all points $(x_1, \ldots, x_n) \in \mathbb{C}^n$ satisfying (1.1), and then select the real solutions. This holds for homotopy continuation methods, as well as for normal form methods. See for instance [3] for an overview. One way to avoid computing all complex solutions is to add equations $f_{s+1}, \ldots, f_t$ which are only satisfied by the real solutions.

Example 1.2. *The equations $f_1 = f_2 = 0$ from Example 1.1 have 4 complex solutions, 2 of which are real. Adding the equation $f_3 = 1 - (1 + \alpha)x_1 - \alpha x_2 = 0$, with $\alpha \approx 0.5652$, only the two real solutions are left. Geometrically, we add the dotted line to the picture.* ◇

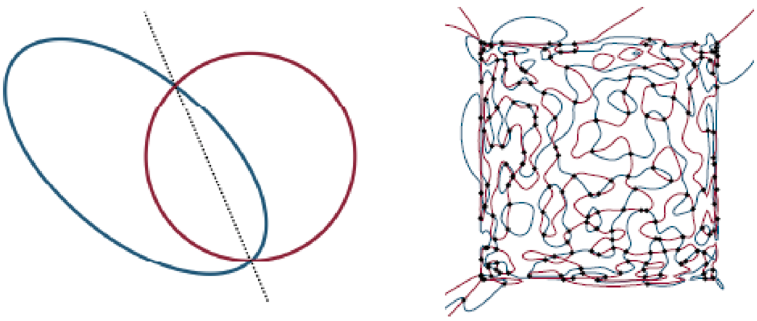

Figure 1: Systems of bivariate equations represent intersecting plane curves.

Project goals: The goal of this project is to compute the equations $f_{s+1}, \ldots, f_t$ needed to eliminate all nonreal solutions of (1.1), using adapted versions of the moment method [2]. We will compare with the recent Julia implementation in [1]. We will use this in a numerical method for finding all real solutions of (1.1), and test our algorithms on real-life problems.

Prerequisits: This project is suitable for students in mathematics or related subjects, with a strong knowledge of basic linear algebra and an interest in algebra and computation.

Contact: For questions, please send an e-mail to Simon.Telen@cwi.nl.

References

[1] L. Baldi and B. Mourrain. Computing real radicals by moment optimization. In Proceedings of the 2021 on International Symposium on Symbolic and Algebraic Computation, pages 43–50, 2021.

[2] M. Laurent and P. Rostalski. The approach of moments for polynomial equations. In Handbook on Semidefinite, Conic and Polynomial Optimization, pages 25–60. Springer, 2012.

[3] S. Telen. Solving Systems of Polynomial Equations. PhD thesis, KU Leuven, Leuven, Belgium, 2020. Available at https://simontelen.webnode.page/publications/.