Sujet de stage de L3 - CRIStAL, Lille - 2021

## Certified numerical approximation of multiple roots

- Team : CFHP `cfhp.univ-lille.fr`. Head François Boulier.
- Supervisors : Adrien Poteaux, `adrien.poteaux@univ-lille.fr` et Florent Bréhard, `florent.brehard@math.uu.se`, `florent.brehard@univ-lille.fr`.
- Laboratory : CRIStAL, Lille (France). Head Olivier Colot, `cristal.univ-lille.fr`.

**Key words:** algorithm, numerical solver, multiple roots, certified computations.

## 1 Our problem: Recovering polynomial multiplicity structures

A wide range of applications in scientific computing require to solve univariate polynomial equations of the form $f(x) = 0$, where the coefficients of $f$ are known only approximately, due to modeling or numerical errors. Although robust algorithms can be used to compute good approximations of the roots of $f$, one loses track of their *multiplicity*, which may be crucial information depending on the context. Consider for example the polynomial $f(x) = x^6 - 2\,x^5 - 4\,x^4 + 6\,x^3 + 7\,x^2 - 4\,x - 4$, which easily factors as $f(x) = (x-2)^2\,(x+1)^3\,(x-1)$. We denote by $[f]$ its *multiplicity structure*, that is, $[3, 2, 1]$ in this example. Slight perturbations of the coefficients of $f$ most often result into 6 distinct roots (some of them might even be complex), that is, multiplicity structure of the perturbed polynomial is $[1, 1, 1, 1, 1, 1]$. It is therefore tedious to infer $[f]$ for the exact polynomial $f$ from these numerical observations.

Our problem is as follows. Assume we are given:

- a set $\mathcal{L} = \{(\tilde{f}_i, \epsilon_i)\}_{1\le i\le s}$ of approximations of univariate polynomials $f_1, \cdots, f_s$ of same degree $d$, together with error bounds $\epsilon_i$ on their coefficients, and such that we can ask for arbitrarily precise $\tilde{f}_i$, that is $\epsilon_i$ as small as desired;
- a set $\mathcal{M} = \{\mathbf{m}_1, \cdots, \mathbf{m}_s\}$ of multiplicity structures, such that each $\mathbf{m}_j$ corresponds to one $f_i$ (and vice-versa), but the corresponding matching is unknown.

We want to solve the following two problems:

1. Find which multiplicity structure $\mathbf{m}_j$ corresponds to which approximation $\tilde{f}_i$, or ask for more precision if one cannot certificate the matching.
2. For each $1 \le i \le s$, compute an approximation of the roots of $f_i$ according to the multiplicity structure $[f_i]$, together with a rigorous bound on the distance between the computed approximations and the (exact) roots of $f_i$.

## 2 Application to Puiseux series

This problem originates from the study of singularities of plane algebraic curves, for which symbolic-numeric algorithms are designed to compute *Puiseux series*, that is, generalized Taylor series above singular points. More details can be found in [6, Chapter 2], especially in Section 2.4 (in French), or [5] (in English).

The problem itself corresponds to the “second filter”: sorting a set of approximations of univariate polynomials according to a set of multiplicity structures.

## 3 Existing works and suggestions

Following the ideas developed in [9, 10, 11] for computing approximate numerical gcd's of polynomials, one of us already designed and implemented a first solution (see [6, Section 2.4.2]). The key point is that the degree of the gcd of two polynomials is related to the rank defect of so-called *Sylvester matrices* defined from the coefficients of the polynomials. Hence, by using numerical SVD (Singular Value Decomposition) algorithms, we can tell how many roots a polynomial $f$ and its derivative $f'$ have in common, and by iterating this process, we recover the desired multiplicity structure $[f]$.

However, we only used purely numerical algorithms so far, so that we are forced to fix a threshold under which a small approximate singular value is said to be zero. As a consequence, this procedure offers no guarantee concerning the computed $[f]$. Since we already have bounds on the numerical approximations of the coefficients of $f$, a possible solution, proposed in [2], may be to use techniques from *interval arithmetics* and *validated numerics* [4, 8] to compute certified enclosures of the singular values. We thus need to understand how existing works on rigorous SVD, e.g. [7], could efficiently solve our problem.

Another solution, proposed in [3], would be to use SOS (sums of squares) techniques: the idea is to check whether there exists a polynomial close enough (i.e. at a distance less than $\epsilon_0$) to the given approximation, whose exact roots correspond to a candidate multiplicity structure. It is possible to certify a negative answer to this question by using sums of squares : one first computes a rational function $\frac{p}{q}$ from $\tilde{f}$ and $[f]$; then, if $p - \epsilon_0 q$ is a sum of squares (thus always positive), we can answer no to our question. Nevertheless, such a strategy needs some improvement: this uses multivariate polynomials (one variable per multiplicity), leading to expensive computations. Starting with the highest multiplicity root in the given multiplicity structure may be a good strategy, which we propose to investigate.

Finally, to improve the accuracy of the computed approximate roots, one can use some well-suited Newton-like method (that can be adapted once we know the multiplicity of the root, as proposed in [9]).

## 4 A roadmap for this internship

Several aspects need to be investigated in order to provide an entirely satisfactory solution to this problem :

- The first point is to find a valid and efficient strategy to solve the problem. That means making a literature survey on this topic to decide which existing tools to use, and which improvements need to be worked out.
- An important goal for our problem consists in designing an algorithm together with a certified implementation, relying on existing libraries for validated numerics, e.g. Arb [1].
- Finally, another important task is to investigate the bit complexity of this algorithm and also its accuracy in order to predict how precise the input polynomial coefficients need to be to solve this problem.

In view of this, such a topic specifically targets students interested in both computer science and computational mathematics. Being at the intersection between computer algebra and numerical analysis, this internship provides an introduction to the emerging trend of certified symbolic-numeric algorithms to efficiently tackle problems with strong reliability requirements. A basic background in mathematics as well as programming skills (and motivation!) are needed.

## References

[1] Arb – a C library for arbitrary-precision ball arithmetic. `https://arblib.org/`.

[2] Zhe Li and Qi Liu. A heuristic verification of the degree of the approximate gcd of two univariate polynomials. *Numerical Algorithms*, 67(2):319–334, 2014.

[3] Zhi Lihong and Wu Wenda. Nearest singular polynomials. *Journal of Symbolic Computation*, 26(6):667 – 675, 1998.

[4] Ramon E. Moore, R. Baker Kearfott, and Michael J. Cloud. *Introduction to interval analysis*, volume 110. Siam, 2009.

[5] Adrien Poteaux. Computing Monodromy Groups defined by Plane Algebraic Curves. In *Proceedings of the 2007 International Workshop on Symbolic-numeric Computation*, pages 36–45, New-York, 2007. ACM.

[6] Adrien Poteaux. *Calcul de développements de Puiseux et application au calcul de groupe de monodromie d'une courbe algébrique plane.* PhD thesis, Université de Limoges, 2008.

[7] Siegfried M Rump. Verified bounds for singular values, in particular for the spectral norm of a matrix and its inverse. *BIT Numerical Mathematics*, 51(2):367–384, 2011.

[8] Warwick Tucker. *Validated numerics: a short introduction to rigorous computations*. Princeton University Press, 2011.

[9] Zhonggang Zeng. A method computing multiple roots of inexact polynomials. In *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, ISSAC '03, pages 266–272, New York, NY, USA, 2003. ACM.

[10] Zhonggang Zeng. The approximate gcd of inexact polynomials part 1: a univariate algorithm. In *ISSAC '04: Proceedings of the 2004 international symposium on Symbolic and algebraic computation*, pages 320–327. ACM Press, 2004.

[11] Zhonggang Zeng. Computing multiple roots of inexact polynomials. *Mathematics of Computation*, 74:869–903, 2005.