Finding Roots of Polynomials

Tonicha Crook*& Arno Pauly
Department of Computer Science
Swansea University
Swansea, UK
t.m.crook15@outlook.com
Arno.M.Pauly@gmail.com

It is a classic result in computable analysis that we can compute the tuple of complex roots of a monic real polynomial (with each root appearing according to its multiplicity). Knowing the exact degree is clearly just as good as the polynomial being monic. If we only care about real roots, the problem ceases to be computable. Even asking for just one root, and guaranteeing that one exists, can solve LLPO: $((x-1)^2+\varepsilon)((x+1)^2-\varepsilon)$ is a polynomial of degree 4 with a real root, but whether the root is near $+1$ or near -1 tells us whether ε is non-negative or non-positive. LLPO is equivalent to finite choice C_2, and C^* suffices to select a real root amongst the tuple of complex root. Put together, we see that C^* is the Weihrauch degree of finding a real root of a polynomial of known degree (which has a real root).

However, a satisfactory representation of polynomials only provides an upper bound on the degree, not its exact value. Le Roux and the second author have shown that finding a zero of a continuous function with up to n local maxima can be done using C_{3^n} in [3]. Since $\mathrm{C}_{3^n} \leq_{\mathrm{W}} \mathrm{C}_2^*$, it follows that we only need to exclude the zero polynomial, and then C^* can find an existing real root (albeit in an exponentially less efficient way when it comes to counting the number of oracle calls).

If the polynomial is allowed to be 0, then already solving $bx = a$ for $0 \leq a \leq b$ has the degree $\mathrm{AoUC}_{[0,1]}$, and $\mathrm{AoUC}_{[0,1]} \not\leq_{\mathrm{W}} \mathrm{C}_2^*$ [5]. If we do not bound the location of the root, we even get the degree LPO. In the following, we thus only consider finding real roots which are guaranteed to exists in $[0,1]$.

Definition 1. Let $\mathrm{BRoot} : \mathbb{R}[X] \rightrightarrows [0,1]$ map real polynomials to a root in $[0,1]$, provided there is one.

Theorem 2. $\mathrm{BRoot} \leq_{\mathrm{W}} \mathrm{AoUC}_{[0,1]}^*$

Finding a root of a single polynomial does not provide much power however:

Proposition 3. $\mathrm{AoUC}_{[0,1]} \times \mathrm{C}_2 \not\leq_{\mathrm{W}} \mathrm{BRoot}$

We subsequently turn our attention to multivariate polynomials. Using cylindrical algebraic decomposition, one can compute some univariate polynomials from a multivariate polynomial such that if the latter has a root, then it has a root where the first component is a root of one of those univariate polynomials. We can recursively use this to find a root of the multivariate polynomial. Being allowed to make repeated calls to a Weihrauch oracle is captured by the operator $^\diamond$ introduced in [4] (see also [6]). If we use BMRoot to denote the multivariate counterpart of BRoot, this reasoning establishes:

Theorem 4. $\mathrm{BMRoot} \leq_{\mathrm{W}} \mathrm{AoUC}_{[0,1]}^\diamond$

*This work is supported by the UKRI AIMLAC CDT, cdt-aimlac.org, grant no. EP/S023992/1.

It is known that $\mathrm{AoUC}^*_{[0,1]} <_{\mathrm{W}} \mathrm{AoUC}^*_{[0,1]} \star \mathrm{AoUC}^*_{[0,1]} \equiv_{\mathrm{W}} \mathrm{AoUC}^{\diamond}_{[0,1]}$ [2], so iterative access to $\mathrm{AoUC}_{[0,1]}$ is indeed stronger than just parallel access (weirdly enough, two layers of oracle calls already provide the full power). We are thus left with the open question:

Open Question 5. Does $\mathrm{BMRoot} \leq_{\mathrm{W}} \mathrm{AoUC}^*_{[0,1]}$ hold?

For context and unexplained notation, see [1].

References

[1] Vasco Brattka, Guido Gherardi & Arno Pauly (2017): *Weihrauch Complexity in Computable Analysis.* arXiv 1707.03202.

[2] Takayuki Kihara & Arno Pauly (2016): *Dividing by Zero – How Bad Is It, Really?* In Piotr Faliszewski, Anca Muscholl & Rolf Niedermeier, editors: *41st Int. Sym. on Mathematical Foundations of Computer Science (MFCS 2016)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 58, Schloss Dagstuhl, pp. 58:1–58:14, doi:10.4230/LIPIcs.MFCS.2016.58.

[3] Stéphane Le Roux & Arno Pauly (2015): *Finite choice, convex choice and finding roots. Logical Methods in Computer Science*, doi:10.2168/LMCS-11(4:6)2015. Available at `http://arxiv.org/abs/1302.0380`.

[4] Eike Neumann & Arno Pauly (2018): *A topological view on algebraic computations models. Journal of Complexity* 44, doi:10.1016/j.jco.2017.08.003. Available at `http://arxiv.org/abs/1602.08004`.

[5] Arno Pauly (2010): *How Incomputable is Finding Nash Equilibria? Journal of Universal Computer Science* 16(18), pp. 2686–2710, doi:10.3217/jucs-016-18-2686.

[6] Linda Westrick (2020): *A note on the diamond operator.* arXiv:2001.09372.