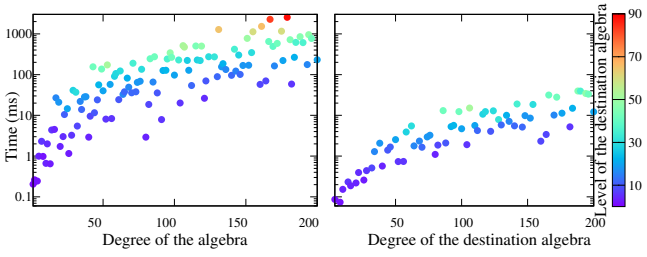

Figure 1: Timings for computing decorated fields $(\mathbb{F}_{p^l}, \alpha_l)$ (left, log scale), and for computing the standard Kummer embedding from $\mathbb{F}_{p^2}$ to $\mathbb{F}_{p^l}$ (right) for $p = 3$.

It is even evident that computing our standard polynomials is essentially equivalent to computing Conway polynomials; indeed from α_l one can immediately deduce $(\zeta_{p^a-1})^a$, and by taking an a-th root (doable in polynomial time in l), deduce ζ_{p^a-1} and the associated Conway polynomial. Hence, an efficient algorithm for computing our polynomials (for arbitrary degrees) would imply an efficient algorithm to compute Conway polynomials, which would be unexpected.

However, our proposed implementation is not the only possible way to exploit our definitions. It would be interesting, indeed, to find some middle ground between the flexibility of the Bosma–Steel–Cannon framework and the rigidity of Conway polynomials, for example by lazily enforcing the conditions required to have a standard solution of (H90), while incrementally constructing the lattice of roots of unity.

Another line of work would be to give a complete implementation of a lattice of finite fields, not limited to extensions of degree coprime to p. We leave these questions for future work.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their useful comments. We thank Éric Schost for fruitful discussions and for helping bootstrap this work during a visit by two of the authors to the University of Waterloo. We acknowledge financial support from the French ANR-15-CE39-0013 project *Manta*, the *OpenDreamKit* Horizon 2020 European Research Infrastructures project (#676541), and from the French Domaine d'Intérêt Majeur *Math'Innov*.

REFERENCES

[1] Bill Allombert. 2002. Explicit Computation of Isomorphisms between Finite Fields. *Finite Fields and Their Applications* 8, 3 (2002), 332 – 342.
[2] Wieb Bosma, John Cannon, and Catherine Playoust. 1997. The MAGMA algebra system I: the user language. *Journal of Symbolic Computation* 24, 3-4 (1997), 235–265. https://doi.org/10.1006/jsco.1996.0125
[3] Wieb Bosma, John Cannon, and Allan Steel. 1997. Lattices of compatibly embedded finite fields. *Journal of Symbolic Computation* 24, 3-4 (1997), 351–369. https://doi.org/10.1006/jsco.1997.0138
[4] Alin Bostan, Philippe Flajolet, Bruno Salvy, and Éric Schost. 2006. Fast computation of special resultants. *Journal of Symbolic Computation* 41, 1 (2006), 1–29.
[5] Alin Bostan, Grégoire Lecerf, and Éric Schost. 2003. Tellegen's principle into practice. In *ISSAC'03*. ACM, 37–44. https://doi.org/10.1145/860854.860870
[6] Richard P. Brent and H.-T. Kung. 1978. Fast Algorithms for Manipulating Formal Power Series. *J. ACM* 25, 4 (1978), 581–595. https://doi.org/10.1145/322092.322099
[7] Ludovic Brieulle, Luca De Feo, Javad Doliskani, Jean-Pierre Flori, and Éric Schost. 2017. Computing isomorphisms and embeddings of finite fields (extended version). *arXiv preprint arXiv:1705.01221* (2017). https://arxiv.org/abs/1705.01221
[8] Ludovic Brieulle, Luca De Feo, Javad Doliskani, Jean-Pierre Flori, and Éric Schost. 2019. Computing isomorphisms and embeddings of finite fields. *Math. Comp.* 88 (2019), 1391–1426. https://doi.org/10.1090/mcom/3363
[9] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. 1997. *Algebraic Complexity Theory*. Springer.
[10] Jean-Marc Couveignes and Reynald Lercier. 2013. Fast construction of irreducible polynomials over finite fields. *Israel Journal of Mathematics* 194, 1 (01 Mar 2013), 77–105. https://doi.org/10.1007/s11856-012-0070-8
[11] Luca De Feo, Javad Doliskani, and Éric Schost. 2013. Fast algorithms for ℓ-adic towers over finite fields. In *ISSAC'13*. ACM, 165–172.
[12] Luca De Feo, Javad Doliskani, and Éric Schost. 2014. Fast Arithmetic for the Algebraic Closure of Finite Fields. In *ISSAC '14*. ACM, 122–129. https://doi.org/10.1145/2608628.2608672
[13] Luca De Feo and Éric Schost. 2012. Fast arithmetics in Artin-Schreier towers over finite fields. *Journal of Symbolic Computation* 47, 7 (2012), 771–792. https://doi.org/10.1016/j.jsc.2011.12.008
[14] Javad Doliskani and Éric Schost. 2015. Computing in degree 2^k-extensions of finite fields of odd characteristic. *Designs, Codes and Cryptography* 74, 3 (01 Mar 2015), 559–569. https://doi.org/10.1007/s10623-013-9875-7
[15] Claus Fieker, William Hart, Tommy Hofmann, and Fredrik Johansson. 2017. Nemo/Hecke: Computer Algebra and Number Theory Packages for the Julia Programming Language. In *ISSAC '17*. ACM, 157–164. https://doi.org/10.1145/3087604.3087611
[16] William Hart, Fredrik Johansson, and Sebastian Pancratz. 2013. *FLINT: Fast Library for Number Theory*. http://flintlib.org Version 2.4.0.
[17] Lenwood S. Heath and Nicholas A. Loehr. 1999. New algorithms for generating Conway polynomials over finite fields. In *SODA '99*. SIAM, 429–437.
[18] Kiran S. Kedlaya and Christopher Umans. 2011. Fast Polynomial Factorization and Modular Composition. *SIAM J. Comput.* 40, 6 (2011), 1767–1802. https://doi.org/10.1137/08073408X
[19] Hendrik W. Lenstra. 1991. Finding isomorphisms between finite fields. *Math. Comp.* 56, 193 (1991), 329–347.
[20] Hendrick W. Lenstra Jr. and Bart de Smit. 2013. *Standard models for finite fields*. Chapman and Hall/CRC, Chapter 11.7 in Handbook of Finite Fields, 401–404. https://doi.org/10.1201/b15006
[21] Anand Kumar Narayanan. 2018. Fast Computation of Isomorphisms Between Finite Fields Using Elliptic Curves. In *WAIFI 2018 (LNCS)*, Vol. 11321. Springer.
[22] Werner Nickel. 1988. Endliche Körper in dem gruppentheoretischen Programmsystem GAP. (1988). https://www2.mathematik.tu-darmstadt.de/~nickel/
[23] David Roe, Jean-Pierre Flori, and Peter Bruin. 2013. Implement pseudo-Conway polynomials. Trac ticket #14958. (Oct. 2013). https://trac.sagemath.org/ticket/14958
[24] Victor Shoup. 1994. Fast construction of irreducible polynomials over finite fields. *Journal of Symbolic Computation* 17, 5 (1994), 371–391. https://doi.org/10.1006/jsco.1994.1025
[25] Victor Shoup. 1999. Efficient computation of minimal polynomials in algebraic extensions of finite fields. In *ISSAC'99*. ACM, 53–58. https://doi.org/10.1145/309831.309859
[26] The GAP Group 2018. *GAP – Groups, Algorithms, and Programming, Version 4.9.2*. The GAP Group. https://www.gap-system.org
[27] The Sage Developers 2019. *SageMath, the Sage Mathematics Software System (Version 8.7)*. The Sage Developers. https://www.sagemath.org
[28] Joachim von zur Gathen and Jurgen Gerhard. 1999. *Modern Computer Algebra*. Cambridge University Press, New York, NY, USA.