

# On the Degree of Polynomials Computing Square Roots Mod $p$

Kiran S. Kedlaya   

Department of Mathematics, University of California San Diego, La Jolla, CA, USA  
School of Mathematics of the Institute for Advanced Study (2023–24 academic year),  
Princeton, NJ, USA

Swastik Kopparty  

Department of Mathematics and Department of Computer Science, University of Toronto, Canada

---

## Abstract

For an odd prime  $p$ , we say  $f(X) \in \mathbb{F}_p[X]$  *computes square roots* in  $\mathbb{F}_p$  if, for all nonzero perfect squares  $a \in \mathbb{F}_p$ , we have  $f(a)^2 = a$ .

When  $p \equiv 3 \pmod{4}$ , it is well known that  $f(X) = X^{(p+1)/4}$  computes square roots. This degree is surprisingly low (and in fact lowest possible), since we have specified  $(p-1)/2$  evaluations (up to sign) of the polynomial  $f(X)$ . On the other hand, for  $p \equiv 1 \pmod{4}$  there was previously no nontrivial bound known on the lowest degree of a polynomial computing square roots in  $\mathbb{F}_p$ .

We show that for all  $p \equiv 1 \pmod{4}$ , the degree of a polynomial computing square roots has degree at least  $p/3$ . Our main new ingredient is a general lemma which may be of independent interest: powers of a low degree polynomial cannot have too many consecutive zero coefficients. The proof method also yields a robust version: any polynomial that computes square roots for 99% of the squares also has degree almost  $p/3$ .

In the other direction, Agou, Deliglése, and Nicolas [1] showed that for infinitely many  $p \equiv 1 \pmod{4}$ , the degree of a polynomial computing square roots can be as small as  $3p/8$ .

**2012 ACM Subject Classification** Computing methodologies  $\rightarrow$  Representation of mathematical functions; Computing methodologies  $\rightarrow$  Number theory algorithms; Mathematics of computing  $\rightarrow$  Coding theory

**Keywords and phrases** Algebraic Computation, Polynomials, Computing Square roots, Reed-Solomon Codes

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2024.25

**Funding** *Kiran S. Kedlaya*: Research supported by NSF grant DMS-2053473, the UC San Diego Warschawski Professorship, the Simons Fellows in Mathematics program of the Simons Foundation (2023–24 academic year).

*Swastik Kopparty*: Research supported by an NSERC Discovery Grant.

**Acknowledgements** Both authors acknowledge support from IAS in 2018–19, where initial discussions towards this paper took place. We thank N. Carella and Igor Shparlinski for valuable comments and pointers to the literature.

## 1 Introduction

Let  $p$  be an odd prime, and let  $\mathbb{F}_p$  be the finite field with  $p$  elements.

We say  $f(X) \in \mathbb{F}_p[X]$  *computes square roots* in  $\mathbb{F}_p$ , if for all nonzero perfect squares  $a \in \mathbb{F}_p$ , we have:

$$f(a)^2 = a.$$

In other words, for each nonzero perfect square  $a \in \mathbb{F}_p$ ,  $f(a)$  is one of its two square roots.



© Kiran S. Kedlaya and Swastik Kopparty;  
licensed under Creative Commons License CC-BY 4.0  
39th Computational Complexity Conference (CCC 2024).

Editor: Rahul Santhanam; Article No. 25; pp. 25:1–25:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



25:2 On the Degree of Polynomials Computing Square Roots Mod  $p$

When  $p \equiv 3 \pmod{4}$ , then it is well known that  $f(X) = X^{(p+1)/4}$  computes square roots. This degree is surprisingly low, since we are essentially interpolating a polynomial from  $(p-1)/2$  evaluations (where the evaluations are specified up to sign). We are interested in whether there is a similar phenomenon for  $p \equiv 1 \pmod{4}$ .

Concretely, we study the question: what is the smallest degree of a polynomial that computes square roots? Despite being a basic and natural question, there were no nontrivial bounds known for this question for the case of  $p \equiv 1 \pmod{4}$ .

There is a very simple argument<sup>1</sup> that shows that the degree of such a polynomial  $f(X)$  must be at least  $\frac{p-1}{4}$ ; indeed, the nonzero polynomial  $f(X)^2 - X$  vanishes at all the  $\frac{p-1}{2}$  nonzero perfect squares in  $\mathbb{F}_p$ .

Our main result is that, unlike the case of  $p \equiv 3 \pmod{4}$ , the degree of any polynomial computing square roots in the case of  $p \equiv 1 \pmod{4}$  must be significantly higher, about  $\frac{1}{3} \cdot p$ .

► **Theorem 1.** *Let  $p \equiv 1 \pmod{4}$ . Then any polynomial that computes square roots in  $\mathbb{F}_p$  has degree at least  $\frac{p-1}{3}$ .*

Our proof is based on expressing the property of computing square roots as a polynomial relation (involving some unknown polynomials), and then eliminating the unknown polynomials through a combination of taking derivatives and truncations. Abstracting out the main steps, we get a general lemma (that can also be derived from the Mason-Stothers *abc*-theorem) which may be of independent interest: the powers of a low-degree polynomial cannot have too many consecutive zero coefficients (Lemma 5).

How does  $p \pmod{4}$  play a role in the proof? Our proof ends up showing that for all  $p$ , any polynomial  $f(X)$  of degree less than  $\frac{p}{3}$  that computes square roots must have  $f(X)^2 = X^{(p+1)/2}$  (as a polynomial identity), and this is not possible if  $p \equiv 1 \pmod{4}$ .

**A robust version**

The degree of a polynomial computing a certain function is quite a brittle notion. Changing just a single value of the function can change the degree drastically. By using the key idea of the Berlekamp-Welch algorithm for decoding Reed-Solomon codes, we can strengthen the above result to get a robust version, given below.

► **Theorem 2.** *Let  $p \equiv 1 \pmod{4}$ . Then any polynomial that computes square roots in  $\mathbb{F}_p$  on all but  $e \leq \frac{p-7}{12}$  nonzero perfect squares in  $\mathbb{F}_p$  must have degree at least  $\frac{p-1}{3} - e$ .*

The connection to decoding algorithms for Reed-Solomon codes is not such a surprise. The problem of whether a low-degree polynomial can compute square roots is in fact a list-recovery problem for Reed-Solomon codes [9]; our result effectively shows that a certain algebraic list recovery instance where each input list has size 2 has no solutions. The difficulty is that this lies beyond the regime where we have a good understanding of list-recoverability and list-decodability of Reed-Solomon codes.

More concretely, let  $C$  be the Reed-Solomon code of degree  $d$  polynomials over  $\mathbb{F}_p$  with evaluation set  $D$ . Suppose for each  $x \in D$  we are given a set  $S_x \subseteq \mathbb{F}_p$  with  $|S_x| \leq 2$ . How can we certify that there are no codewords  $c$  of  $C$  such that for each coordinate  $x \in D$ , we have  $c_x \in S_x$ ? It is not known how to give an efficiently verifiable certificate of this *in general* when  $d = (\frac{1}{2} + \Omega(1)) |D|$ . In our setting  $D$  is the set of perfect squares (so  $|D| = (p-1)/2$ ), and  $d$  is  $p/3$ , which is outside the range of known certification methods [9].

---

<sup>1</sup> This argument works for all  $p$ , and thus we get that  $X^{(p+1)/4}$  is the lowest degree polynomial computing square roots for  $p \equiv 3 \pmod{4}$ .

## Better upper bounds for special $p$

It turns out that for some  $p$  which are  $1 \pmod{4}$ , there are polynomials computing square roots with degree about  $\frac{3}{8} \cdot p$ .

► **Theorem 3.** *Let  $p \equiv 5 \pmod{8}$ . Then there is a polynomial that computes square roots in  $\mathbb{F}_p$  with degree at most  $\frac{3p+1}{8}$ .*

This was shown by Agou, Deliglése and Nicolas [1] (see also [5]), based on the Tonelli-Shanks algorithm for computing square roots. Since it is quite simple and short, we include it in this paper for completeness.

The method of Tonelli-Shanks also yields similar phenomena with degree  $(\frac{1}{2} - \Omega(1))p$  for  $p$  in special residue classes mod  $2^j$  with  $j$  constant. Theorem 5 of [1] gives another example of such a phenomenon for  $p \equiv 7 \pmod{12}$ , giving polynomials (different from  $X^{(p+1)/4}$ ) computing square roots of degree about  $\frac{5}{12} \cdot p$ .

## Upper bounds for general $p$

Finally, we discuss upper bounds for the case of general  $p$ . First, a heuristic. There are  $2^{(p-1)/2}$  different square root functions (the choice of sign for each perfect square). If the unique interpolating polynomials of degree  $< (p-1)/2$  for these functions had their coefficients behaving randomly, then we would expect a polynomial of degree at most  $\frac{1}{2}p - \Omega\left(\frac{p}{\log p}\right)$  that computes square roots.

Formalizing this intuition, we show that there is a polynomial computing square roots with degree  $\frac{1}{2}p - \tilde{\Omega}(\sqrt{p})$ . This is done by looking at the explicit formulas for the coefficients of the interpolants and arguing their pseudorandomness via the Weil bounds and some elementary Fourier analysis.

We also note that all our results have analogues for computing  $t$ -th roots.

## Related Work

The work of Agou, Deliglése and Nicolas [1] gave examples, for infinitely many  $p$ , of polynomials in  $\mathbb{F}_p$  of abnormally low degree that compute square roots. The focus there was on finding polynomials with few monomials, and they gave interesting upper and lower bounds for this. Chang, Kim and Lee [6] gave analogues of these results for computing  $t$ -th roots.

Another related line of research has studied lower bounds on the degree of polynomials computing interesting arithmetic functions. Coppersmith and Shparlinski [7] (following an error-free computation result of Mullen and White [12]), gave strong lower bounds on the degree of polynomials computing the discrete logarithm in prime fields  $\mathbb{F}_p$  with as many as  $(1 - o(1))p$  errors. Winterhof [20] later gave a generalization of this to all finite fields. These results are related to *list-decoding* of Reed-Solomon codes, since for each input there is only one “correct” value which we are hoping the polynomial will compute. As mentioned earlier, the problems we consider are related to *list-recovery* of Reed-Solomon codes, where there are multiple “correct” values for any given input, and we hope the polynomial computes one of them.

## Conclusions and Questions

Computing square roots and understanding quadratic residuosity are central topics in algebraic computation and pseudorandomness.

## 25:4 On the Degree of Polynomials Computing Square Roots Mod $p$

Perhaps the most interesting and fundamental open question in this area is that of deterministically computing square roots mod  $p$  in  $\text{poly}(\log p)$  time. As we already saw, when  $p \equiv 3 \pmod{4}$ , the simple deterministic  $\text{poly}(\log p)$  algorithm of raising  $x \in \mathbb{F}_p$  to the power  $\frac{p+1}{4}$  computes the square root of  $x$ . Our results show that the  $p \equiv 1 \pmod{4}$  case is qualitatively different in some respects. See [3, 18, 15] for what is known about this computational problem and related number theoretic issues.

Other important questions include the problem of determining the size of the least quadratic residue mod  $p$  (this is also connected to deterministic computation of square roots), and understanding the pseudorandomness of the Paley graph (for example, are Paley graphs Ramsey graphs?).

Finally, as mentioned above, our results can be viewed as showing that a certain list-recovery instance for Reed-Solomon codes has no solutions. We close with a conjecture about the list-recoverability of Reed-Solomon codes. The conjecture talks about prime fields; the results of Guruswami and Rudra [8] imply that this assumption cannot be dropped.

► **Conjecture 4.** *Let  $\mathbb{F}_p$  be a prime field. Let  $\ell \in \mathbb{N}, \epsilon > 0$  be constants. Suppose we are given, for each  $x \in \mathbb{F}_p$ , a set  $S_x$  with  $|S_x| \leq \ell$ . Then:*

$$|\{P(X) \in \mathbb{F}_p[X] \mid \deg(P) \leq (1 - \epsilon)p, \text{ and for all } x \in \mathbb{F}_p, P(x) \in S_x\}| \leq p^{O_{\epsilon, \ell}(1)}.$$

We hope that our methods can give some insight into understanding the list-recovery capacity of Reed-Solomon codes, and in particular the above conjecture.

### 2 Lower bound for polynomials computing square roots

We now prove our first theorem about polynomials computing square roots mod  $p$ .

► **Theorem 1.** *Let  $p \equiv 1 \pmod{4}$ . Then the degree of any polynomial that computes square roots in  $\mathbb{F}_p$  is at least  $\frac{p-1}{3}$ .*

**Proof.** Suppose  $f(X)$  is of degree  $d < \frac{p-1}{3}$  and computes square roots in  $\mathbb{F}_p$ . Then, since  $X^{(p-1)/2} - 1$  is the vanishing polynomial of the set of nonzero perfect squares in  $\mathbb{F}_p$ , we have:

$$f(X)^2 - X \equiv 0 \pmod{(X^{(p-1)/2} - 1)}.$$

Let  $A(X)$  be the polynomial of degree  $2d - (p-1)/2$  such that

$$f(X)^2 - X = A(X) \cdot (X^{(p-1)/2} - 1).$$

Let  $B(X) = X - A(X)$ . Then we get:

$$f(X)^2 = A(X) \cdot X^{(p-1)/2} + B(X), \tag{1}$$

where:

- $\deg(f(X)) = d$ .
- $\deg(A(X)), \deg(B(X)) \leq 2d - (p-1)/2$ .
- $A(X) \neq 0$ . If  $A(X) = 0$  then  $f(X)^2 = X$ , which is impossible for a polynomial  $f(X)$ .
- $B(X) \neq 0$ . Otherwise  $A(X) = X$ , and  $f(X)^2 = X^{(p+1)/2}$ , which is possible only if  $p \equiv 3 \pmod{4}$ .

These conditions together will give us our lower bound on  $d$ .

Taking derivatives<sup>2</sup> of both sides of (1), we get:

$$2f(X)f'(X) = A'(X) \cdot X^{(p-1)/2} - \frac{1}{2}A(X)X^{(p-3)/2} + B'(X). \quad (2)$$

Computing  $2f(X)^2f'(X)$  in two ways using (1) and (2), we get:

$$2f'(X)A(X)X^{(p-1)/2} + 2f'(X)B(X) = f(X) \left( X \cdot A'(X) - \frac{1}{2}A(X) \right) X^{(p-3)/2} + f(X)B'(X).$$

Now, using our assumption on  $d$ , the degrees of  $2f'(X)B(X)$  and  $f(X)B'(X)$  are both at most  $3d - (p-1)/2 - 1 < (p-3)/2$ , and thus taking the above equation mod  $X^{(p-3)/2}$ ,

$$2f'(X)B(X) = f(X)B'(X).$$

Since  $B(X) \neq 0$ , we get  $\frac{2f'(X)}{f(X)} = \frac{B'(X)}{B(X)}$ . Since  $2\deg(f), \deg(B) < p$ , by a basic property of logarithmic derivatives, this implies  $f(X)^2 = \lambda B(X)$  for some nonzero  $\lambda$ , contradicting the fact that  $A(X) \neq 0$ . (See Remark 1 in Section 2.2 for a precise statement and a proof.)

Thus our assumption that  $d < \frac{p-1}{3}$  is wrong, and the theorem follows. ◀

## 2.1 Consecutive zero coefficients in powers of polynomials

We isolate the key step above as the following lemma:

► **Lemma 5.** *Let  $\mathbb{F}$  be a field of characteristic  $p$ . Let  $f(X), A(X), B(X)$  be in  $\mathbb{F}[X]$ . Suppose*

$$f(X)^t = A(X) \cdot X^\ell + B(X) \quad (3)$$

where:

- $\deg(f(X)) \leq d < \frac{p}{t}$ ,
- $\deg(B(X)) \leq b < p$ ,
- $A(X) \neq 0$ ,
- $B(X) \neq 0$ ,

Then  $d + b \geq \ell$ .

In words, this says that if  $dt < p$ , the  $t$ -th power of a polynomial of degree  $d$  cannot have  $d$  consecutive 0 coefficients unless it is a single monomial (since the RHS of the above equation has at least  $\ell - b - 1$  consecutive 0 coefficients).

**Proof.** Suppose  $d + b < \ell$ . Observe that this implies that  $f(X) \neq 0$ .

Taking derivatives of both sides of (3), we get:

$$tf(X)^{t-1}f'(X) = C(X)X^{\ell-1} + B'(X), \quad (4)$$

for some  $C(X) \in \mathbb{F}[X]$ .

Computing  $tf(X)^t f'(X)$  in two different ways using (3), (4), we get:

$$tA(X)f'(X)X^\ell + tf'(X)B(X) = f(X)C(X)X^{\ell-1} + f(X)B'(X).$$

<sup>2</sup> Throughout this paper, we work with formal derivatives of polynomials.

## 25:6 On the Degree of Polynomials Computing Square Roots Mod $p$

Since  $\deg(tf'(X)B(X)), \deg(f(X)B'(X)) < d + b \leq \ell - 1$ , by taking this equation mod  $X^{\ell-1}$  we get:

$$tf'(X)B(X) = f(X)B'(X),$$

and since  $f(X), B(X)$  are nonzero, we get that:

$$t \frac{f'(X)}{f(X)} = \frac{B'(X)}{B(X)}.$$

By the logarithmic derivative, we get  $f(X)^t = \lambda B(X)$  for some nonzero  $\lambda$ , contradicting our assumption that  $A(X) \neq 0$ .

Thus  $d + b \geq \ell$  as claimed.  $\blacktriangleleft$

### 2.2 Remarks

1. The key fact about logarithmic derivatives that we are using is that if  $f(X), B(X) \in \mathbb{F}_p[X]$ ,  $t \cdot \deg(f), \deg(B) < p$ , and:

$$t \frac{f'(X)}{f(X)} = \frac{B'(X)}{B(X)},$$

then  $f(X)^t = \lambda \cdot B(X)$  for some constant  $\lambda \in \mathbb{F}_p$ .

We recap a quick proof. The hypothesis implies that  $\left(\frac{f(X)^t}{B(X)}\right)' = 0$ , and thus  $\frac{f(X)^t}{B(X)}$  must be a rational function in  $X^p$ . To see the last deduction, note that  $(f(X)^t \cdot B(X)^{p-1})' = \left(\frac{f(X)^t}{B(X)} \cdot B(X)^p\right)' = \left(\frac{f(X)^t}{B(X)}\right)' \cdot B(X)^p + \frac{f(X)^t}{B(X)} \cdot p \cdot B(X)^{p-1} \cdot B'(X) = 0 + 0 = 0$ . Thus the polynomial  $f(X)^t \cdot B(X)^{p-1}$  is a polynomial in  $X^p$ , which implies that  $\frac{f(X)^t}{B(X)} = \frac{f(X)^t \cdot B(X)^{p-1}}{B(X)^p}$  is a rational function in  $X^p$ . Once we know that  $\frac{f(X)^t}{B(X)}$  is a rational function in  $X^p$ , our assumption about the degrees implies the result.

2. The exact same proof also classifies when low-degree polynomials can compute square roots of very-low-degree polynomials on a multiplicative group.

► **Theorem 6.** Let  $G \subseteq \mathbb{F}_p^*$  be a multiplicative subgroup of size  $m$ , with  $m \leq \frac{p-1}{2}$ . Let  $C(X) \in \mathbb{F}_p[X]$  have degree at most  $\frac{m}{3}$ . Suppose  $f(X) \in \mathbb{F}_p[X]$  is such that  $f(a)^2 = C(a)$  for all  $a \in G$ . Then one of the following alternatives must hold:

- $f(X)^2 = C(X)$ ,
- $f(X)^2 = C(X) \cdot X^m$ ,
- $\deg(f) \geq \frac{2m}{3}$ .

The above statement for  $m = p - 1$  and  $C(X)$  being a constant follows from a result of Biro [4], who classified low-degree polynomials that take two values on  $\mathbb{F}_p^*$ . The proof from [4] is a delicate investigation of certain power sums. Our proof for  $m < p - 1$  is quite different, and has the flexibility of allowing for the robust version proved in the next section (which gives, for example, a classification of low-degree polynomials that take only 2 values on 99% of  $G$ ).

In this generality, the bound of  $\frac{2m}{3}$  is tight. If  $m$  is divisible by 3, then the polynomial  $f(X) = (X^{2m/3} + X^{m/3} - \frac{1}{2})$  is such that  $f(x)^2 = \frac{9}{4}$  for all  $x \in G$  (since  $x^{m/3}$  is a cube root of 1).

3. The proof of Lemma 5 also applies as is to *rational powers* of  $f(X)$ , where we now talk about consecutive 0 coefficients in the power series. We only state it for characteristic 0; it says that the power series expansion of  $f(X)^{r/s}$ , for  $f(X)$  of degree  $d$ , does not have  $d$  consecutive 0 coefficients. Precisely, we have:

► **Lemma 7.** *Let  $\mathbb{F}$  be a field of characteristic 0. Let  $t$  be a rational number. Let  $f(X) \in \mathbb{F}[X]$  be a polynomial of degree at most  $d$  with nonzero constant term. Then any (formal) power series expansion of  $f(X)^t$  in  $\mathbb{F}[[X]]$  does not have  $d$  consecutive zero coefficients.*

This is stronger than the usual bound for this situation (which shows up in polynomial factoring algorithms via the Hilbert irreducibility theorem [10] and the Arora-Sudan low degree test [2]), which goes as follows: Suppose  $f(X)^{1/s} = A(X)X^\ell + B(X)$ , where  $\deg(f) = d$ ,  $\deg(B) = b$  and  $A(X) \in \mathbb{F}[[X]]$  is nonzero, then  $f(X) - B(X)^s$  is a nonzero polynomial of degree at least  $\ell$ , and so  $\ell \leq \max(sb, d)$ . Thus if  $b$  is large, this bound only guarantees that there is a nonzero coefficient  $X^i$  for  $i \in [b+1, sb]$  (instead of  $[b+1, b+d]$  as guaranteed by Lemma 7).

4. Applying the same method, we can apply this method to the power series expansion of  $e^{f(X)}$  too.

► **Lemma 8.** *Let  $\mathbb{F}$  be a field of characteristic 0. Let  $f(X) \in \mathbb{F}[X]$  be a polynomial of degree at most  $d$  with constant term 0. Then the (formal) power series expansion of  $e^{f(X)}$  in  $\mathbb{F}[[X]]$  does not have  $d$  consecutive zero coefficients.*

5. The bounds of Lemma 5, Lemma 7 and Lemma 8 on the number of consecutive 0 coefficients are tight, for example when  $f(X)$  is of the form  $\alpha X^d + \beta$ .
6. We can give another proof of Lemma 5 (but not Lemma 7 or Lemma 8 as far as we know) using the Mason-Stothers *abc*-theorem for polynomials [16, 11].

Indeed, note that  $f(X)^t$ ,  $A(X) \cdot X^\ell$  and  $B(X)$  all have degree at most  $dt$ . Furthermore, the radical of their product divides  $f(X) \cdot A(X) \cdot X \cdot B(X)$ , and thus has degree at most  $d + (dt - \ell) + 1 + b$ . By the *abc*-theorem, we get that:

$$dt \leq (d + dt - \ell + 1 + b) - 1 = dt + d - \ell + b,$$

and thus  $d + b \geq \ell$ .

### 3 A robust version

Let  $p$  be a prime that is 1 mod 4. Let  $S$  be the set of nonzero perfect squares in  $\mathbb{F}_p$ .

We say a polynomial  $f(X) \in \mathbb{F}_p[X]$  computes square roots with error  $e$  if:

$$|\{a \in S \mid f(a)^2 \neq a\}| \leq e.$$

We show that any polynomial computing square roots even allowing  $\Omega(p)$  error cannot have degree much smaller than  $p/3$ .

► **Theorem 2.** *Let  $p \equiv 1 \pmod{4}$ . Suppose  $f(X) \in \mathbb{F}_p[X]$  is a polynomial of degree  $d$  that computes square roots with error  $e$ .*

Then

$$d \geq \begin{cases} \frac{p-1}{3} - e & e \leq \frac{p-7}{12} \\ \frac{p-1}{2} - 3e - 1 & e > \frac{p-7}{12}. \end{cases}$$

**Proof.** We use the idea of the Berlekamp-Welch Reed-Solomon decoding algorithm [19].

Let  $U \subseteq S$  be the set of  $a \in S$  where  $f(a)^2 \neq a$ .

Let  $E(X) \in \mathbb{F}_p[X]$  be the vanishing polynomial of  $U$ , given by:

$$E(X) = \prod_{u \in U} (X - u).$$

## 25:8 On the Degree of Polynomials Computing Square Roots Mod $p$

Note that  $E$  is a nonzero polynomial of degree at most  $e$ .

Then we have:

$$E(X)^2 \cdot f(X)^2 \equiv E(X)^2 \cdot X \pmod{(X^{(p-1)/2} - 1)}.$$

Let  $A(X)$  be the polynomial of degree at most  $2(e + d) - (p - 1)/2$  such that:

$$E(X)^2 \cdot f(X)^2 - E(X)^2 \cdot X = A(X) \cdot (X^{(p-1)/2} - 1).$$

Let  $g(X) = E(X) \cdot f(X)$ , and  $B(X) = E(X)^2 \cdot X - A(X)$ .

Then

$$g(X)^2 = A(X) \cdot X^{(p-1)/2} + B(X).$$

We have:

- $\deg(g) \leq d + e$ ,
- $\deg(B) \leq \max(2e + 1, 2(e + d) - (p - 1)/2)$ .
- $A(X) \neq 0$ . Otherwise  $E(X)^2 \cdot f(X)^2 = E(X)^2 \cdot X \implies f(X)^2 = X$ , which is impossible.
- $B(X) \neq 0$ . Otherwise  $E(X)^2 \cdot X = A(X)$ , and so  $E(X)^2 f(X)^2 = E(X)^2 X^{(p+1)/2}$ , which implies that  $f(X)^2 = X^{(p+1)/2}$ . This is only possible if  $p \equiv 3 \pmod{4}$ .

Plugging this into Lemma 5, we get:

■

$$(d + e) + \left(2(d + e) - \frac{p - 1}{2}\right) \geq \frac{p - 1}{2},$$

if  $2d - (p - 1)/2 \geq 1$ ,

■

$$(d + e) + (2e + 1) \geq \frac{p - 1}{2},$$

if  $2d - (p - 1)/2 \leq 0$ .

This tells us that either:

$$d \geq \frac{p - 1}{3} - e,$$

or:

$$d \geq \frac{p - 1}{2} - 3e - 1,$$

and thus:

$$d \geq \min\left(\frac{p - 1}{3} - e, \frac{p - 1}{2} - 3e - 1\right),$$

which gives us the desired claim. ◀

Note that there is another simple lower bound (which applies for all  $p$ ) of  $d + \frac{e}{2} \geq \frac{p-1}{4}$  (the simple lower bound is better for  $e > \frac{p-1}{10}$ ). This is proved by considering the number of roots of the degree  $2d$  polynomial  $f(X)^2 - X$ .



#### 4 Upper bound for special $p$

In this section, we give an upper bound on the degree of polynomials computing square roots mod  $p$ , for infinitely many  $p \equiv 1 \pmod{4}$ . The upper bound is best when  $p \equiv 5 \pmod{8}$ , and we only present this case. The result and proof of this section is due to Agou, Deliglése and Nicolas [1]. It remains in this paper only for completeness.

► **Theorem 3.** *Let  $p \equiv 5 \pmod{8}$ . Then there is a polynomial that computes square roots in  $\mathbb{F}_p$  with degree at most  $\frac{3p+1}{8}$ .*

**Proof.** Since  $p \equiv 1 \pmod{4}$ , we get that  $-1$  is a perfect square mod  $p$ . Let  $i \in \mathbb{F}_p$  be one of the square roots of  $-1$ . Our main ingredient is the Tonelli-Shanks algorithm [14, 17] computing square roots mod  $p$ . For  $p = 4\ell + 1$ , the algorithm essentially gives a formula for the square root depending on two cases. Specifically, let  $u : S \rightarrow \mathbb{F}_p$  given by:

$$u(a) = \begin{cases} a^{(p+3)/8} & a^{(p-1)/4} = 1, \\ i \cdot a^{(p+3)/8} & a^{(p-1)/4} = -1. \end{cases}$$

Then for all  $a \in \mathbb{F}_p$ ,  $u(a)$  is a square root of  $a$ .

This is already quite special; the set  $S$  is partitioned into two equal sized parts  $S_0$  and  $S_1$ , and on each  $S_i$  we have a polynomial  $f_i(X)$  computing the square root of degree about  $\frac{1}{2}|S_i|$ . (This is the lowest possible degree, since  $f_i(X)^2 - X$  is a nonzero polynomial that vanishes on all of  $S_i$ .)

Usually if we have this kind of setup, even though the  $f_i$  have unusually low degree, the unique polynomial  $f$  (obtained from the Chinese remainder theorem) which restricts to  $f_i$  on  $S_i$  has no reason to have unusually low degree. But in this case it does!

Using the usual Chinese remainder formula, we consider the polynomial  $f(X) \in \mathbb{F}_p[X]$  given by:

$$\begin{aligned} f(X) &= \frac{1}{2} \left( X^{(p+3)/8} (X^{(p-1)/4} + 1) - i \cdot X^{(p+3)/8} (X^{(p-1)/4} - 1) \right) \\ &= \frac{1-i}{2} X^{(3p+1)/8} + \frac{1+i}{2} X^{(p+3)/8}. \end{aligned}$$

By design, we have  $f(a) = u(a)$  for all  $a \in S$ . Finally, notice that  $\deg(f) \leq \frac{3p+1}{8}$ .

As a sanity check, we directly verify that  $f(X)^2 \equiv X \pmod{(X^{(p-1)/2} - 1)}$ . Indeed,

$$\begin{aligned} f(X)^2 &= \left( \frac{1-i}{2} \right)^2 X^{(3p+1)/4} + 2 \cdot \frac{(1-i)(1+i)}{4} X^{(4p+4)/8} + \left( \frac{1+i}{2} \right)^2 X^{(p+3)/4} \\ &= -\frac{i}{2} X^{(3p+1)/4} + X^{(p+1)/2} + \frac{i}{2} X^{(p+3)/4} \\ &= \left( -\frac{i}{2} X^{(p+3)/4} + X \right) \cdot (X^{(p-1)/2} - 1) + X, \end{aligned}$$

as desired. ◀

#### 5 Upper bounds for general $p$

In this section, we give a slightly nontrivial upper bound on the degree of polynomials computing square roots for all  $p$ . We will show that there is a polynomial with degree somewhat less than  $\frac{p}{2}$  which computes square roots.

## 25:10 On the Degree of Polynomials Computing Square Roots Mod $p$

► **Theorem 9.** For all odd primes  $p$ , for  $t = o\left(\frac{\sqrt{p}}{\log^2 p}\right)$ , there is a polynomial of degree  $\frac{p}{2} - t$  which computes square roots.

**Proof.** Let  $m = (p-1)/2$ . Let  $S \subseteq \mathbb{F}_p$  be the set of nonzero perfect squares, and note that  $|S| = m$ . For each  $\alpha \in S$ , let  $\delta_\alpha(X) \in \mathbb{F}_p[X]$  be the unique polynomial of degree  $\leq (m-1)$  such that for all  $\beta \in S$ :

$$\delta_\alpha(\beta) = \begin{cases} 1 & \beta = \alpha, \\ 0 & \beta \neq \alpha. \end{cases}$$

Explicitly, we have:

$$\delta_\alpha(X) = \frac{1}{m} \left( \left(\frac{X}{\alpha}\right)^{m-1} + \left(\frac{X}{\alpha}\right)^{m-2} + \dots + \frac{X}{\alpha} + 1 \right).$$

Then given a function  $u : S \rightarrow \mathbb{F}_p$ , the unique polynomial  $f(X) \in \mathbb{F}_p[X]$  of degree at most  $m-1$  such that  $f(\alpha) = u(\alpha)$  for all  $\alpha \in S$  is given by:

$$f(X) = \sum_{\alpha \in S} u(\alpha) \delta_\alpha(X).$$

Our goal is to pick  $u$  where each  $u(\alpha)$  is one of the two square roots of  $\alpha$  so that many of the leading coefficients of  $f(X)$  equal 0.

We now use the structure of  $S$ . Let  $g$  be a generator of  $\mathbb{F}_p^*$ . Then  $S = \{g^{2j} \mid 0 \leq j < (p-1)/2\}$ . Furthermore, for  $\alpha = g^{2j} \in S$ , one of the two square roots of  $\alpha$  is  $g^j$ .

Thus, our problem can be reformulated as choosing a function  $v : S \rightarrow \{\pm 1\}$  such that:

$$f(X) = \sum_{j=0}^{(p-1)/2} v(g^{2j}) \cdot g^j \cdot \delta_{g^{2j}}(X)$$

has many leading coefficients equal to 0.

Observe that the coefficient of  $X^{m-i}$  in  $f(X)$  equals:

$$\frac{1}{m} \sum_{j=0}^{(p-1)/2} v(g^{2j}) g^j \left(\frac{1}{g^{2j}}\right)^{m-i} = \frac{1}{m} \sum_{i=0}^{(p-1)/2} v(g^{2j}) g^{(2i+1)j}.$$

Thus, to get a polynomial  $f(X)$  of degree  $< m-t$ , we want to find a vector  $v \in \{\pm 1\}^m$  that lies in the kernel of the Vandermonde-type matrix  $M \in \mathbb{F}_p^{t \times m}$ , where:

$$M_{i,j} = g^{(2i+1)j}.$$

(The row index  $i$  runs from 1 to  $t$ , the column index  $j$  runs from 0 to  $m-1$ .)

For later use, for a vector  $y \in \mathbb{F}_p^t$ , we define  $P_y(Z) \in \mathbb{F}_p[Z]$  to be the polynomial:

$$P_y(Z) = \sum_{i=1}^t y_i Z^{2i+1}.$$

Thus for  $j \in \{0, 1, \dots, m-1\}$ , the  $j$ th entry of  $M^T y$  equals  $P_y(g^j)$ .

To show that there exists the desired  $\pm 1$  vector, we count the number of such vectors using Fourier analysis. Let  $\omega$  be a  $p$ th root of unity in  $\mathbb{C}$ . The number of such  $\pm 1$  vectors equals:

$$\begin{aligned}
N &= \sum_{v \in \{\pm 1\}^m} \mathbf{1}_{Mv=0} \\
&= \sum_{v \in \{\pm 1\}^m} \mathbb{E}_{y \in \mathbb{F}_p^t} \left[ \omega^{\langle y, Mv \rangle} \right] \\
&= \mathbb{E}_y \left[ \sum_v \omega^{\langle M^T y, v \rangle} \right] \\
&= \mathbb{E}_y \left[ \sum_v \omega^{\sum_{j=0}^{m-1} (M^T y)_j \cdot v_j} \right] \\
&= \mathbb{E}_y \left[ \sum_v \prod_{j=0}^{m-1} \omega^{(M^T y)_j \cdot v_j} \right] \\
&= \mathbb{E}_y \left[ \sum_v \prod_{j=0}^{m-1} \omega^{P_y(g^j) \cdot v_j} \right].
\end{aligned}$$

For  $y = 0$ , the expression inside the expectation equals  $2^m$ . We will show that for the remaining  $p^t - 1$  values of  $y$ , the expression inside the expectation is very small.

Fix any  $y \neq 0$ . The expression inside the expectation equals:

$$\sum_{v \in \{\pm 1\}^m} \prod_{j=0}^{m-1} \omega^{P_y(g^j) \cdot v_j} = \prod_{j=1}^m \left( \omega^{P_y(g^j)} + \omega^{-P_y(g^j)} \right). \quad (5)$$

The next lemma (which uses the Weil bounds on mixed character sums) shows that for any nonzero  $y$ , the evaluations of the polynomial  $P_y$  at  $\{1, g, g^2, \dots, g^{m-1}\}$  are well distributed in  $\mathbb{F}_p$ .

► **Lemma 10.** *Let  $0 \leq \alpha < \beta \leq 1$ . Let  $y \in \mathbb{F}_p^t \setminus \{0\}$ . Then:*

$$\Pr_{j \in \{0, 1, \dots, m-1\}} [P_y(g^j) \in [\alpha p, \beta p]] = (\beta - \alpha) + O\left(\frac{t \log^2 p}{\sqrt{p}}\right).$$

Assuming the lemma, we get that for  $t = o\left(\frac{\sqrt{p}}{\log^2 p}\right)$ , the product in Equation (5) is at most  $2^m \cdot \exp(-m)$ . Thus:

$$\begin{aligned}
N &\geq \frac{2^m}{p^t} - \max_{y \neq 0} \left| \prod_{j=1}^m \left( \omega^{P_y(g^j)} + \omega^{-P_y(g^j)} \right) \right| \\
&\geq \frac{2^m}{p^t} - O(2^m \exp(-m)) \\
&\geq 2^m \left( \frac{1}{p^t} - \exp(-m) \right) \\
&> 0,
\end{aligned}$$

where the penultimate inequality holds since

$$t = o\left(\frac{\sqrt{p}}{\log^2 p}\right) \ll \frac{p}{\log p} = \Theta\left(\frac{m}{\log p}\right).$$

This completes the proof. ◀

### 5.1 Distribution of values of $P_y(g^j)$

We now prove Lemma 10.

**Proof.** Let  $I$  be the interval  $[\alpha p, \beta p]$ . Let  $J$  be the set  $\{1, g, g^2, \dots, g^{m-1}\}$ . Then the probability in the statement of the lemma equals:

$$\frac{1}{m} \sum_{z \in \mathbb{F}_p} \mathbf{1}_I(P_y(z)) \mathbf{1}_J(z). \quad (6)$$

$I$  is an interval in the additive group of  $\mathbb{F}_p$ . By standard results, if we expand  $\mathbf{1}_I$  in its additive Fourier expansion:

$$\mathbf{1}_I = \sum_{\psi} \widehat{\mathbf{1}}_I(\psi) \psi$$

(where the  $\psi$  are the additive characters), then:

$$\sum_{\psi} |\widehat{\mathbf{1}}_I(\psi)| \leq O(\log p). \quad (7)$$

Similarly,  $J$  is an interval in the multiplicative group of  $\mathbb{F}_p$ . If we expand  $\mathbf{1}_J$  in its additive Fourier expansion:

$$\mathbf{1}_J = \sum_{\chi} \widetilde{\mathbf{1}}_J(\chi) \chi$$

(where the  $\chi$  are the multiplicative characters), then:

$$\sum_{\chi} |\widetilde{\mathbf{1}}_J(\chi)| \leq O(\log p). \quad (8)$$

Then the probability in Equation (6) equals:

$$\begin{aligned} & \frac{1}{m} \sum_z \left( \sum_{\psi} \widehat{\mathbf{1}}_I(\psi) \psi(P_y(z)) \right) \left( \sum_{\chi} \widetilde{\mathbf{1}}_J(\chi) \chi(z) \right) \\ &= \frac{1}{m} \sum_{\psi, \chi} \widehat{\mathbf{1}}_I(\psi) \widetilde{\mathbf{1}}_J(\chi) \left( \sum_z \psi(P_y(z)) \chi(z) \right) \\ &= \frac{1}{m} \left( \frac{|I|}{p} \cdot \frac{|J|}{p} \cdot p + O \left( \sum_{(\psi, \chi) \neq (1, 1)} |\widehat{\mathbf{1}}_I(\psi)| \cdot |\widetilde{\mathbf{1}}_J(\chi)| \cdot \left| \sum_z \psi(P_y(z)) \chi(z) \right| \right) \right) \\ &= (\beta - \alpha) + \frac{1}{m} \cdot O \left( \sum_{(\psi, \chi) \neq (1, 1)} |\widehat{\mathbf{1}}_I(\psi)| \cdot |\widetilde{\mathbf{1}}_J(\chi)| \cdot \left| \sum_z \psi(P_y(z)) \chi(z) \right| \right) \end{aligned}$$

The Weil bound for mixed character sums (see [13]) shows that whenever  $(\psi, \chi)$  are not both trivial characters, the inner expression is bounded:

$$\left| \sum_z \psi(P_y(z)) \chi(z) \right| \leq O(t\sqrt{p}).$$

Combined with the bounds in Equations (7), (8), we get the desired bound on the probability.  $\blacktriangleleft$

## 6 $t$ -th roots

Now we discuss  $t$ -th roots in place of square roots. We think of  $t$  as a constant, and the prime  $p$  growing. Let  $p \equiv 1 \pmod t$ , so that the set of nonzero  $t$ -th powers in  $\mathbb{F}_p$  has size  $\frac{p-1}{t}$ .

Just like in the case  $t = 2$ , for special  $p$  there is a simple formula for computing the  $t$ -th root; when  $p \equiv 1 - t \pmod{t^2}$ , then  $a^{\frac{p+t-1}{t^2}}$  is a  $t$ -th root of  $a$  whenever  $a$  is a perfect  $t$ -th power in  $\mathbb{F}_p$ . Thus there is a polynomial of degree  $\frac{1}{t^2} \cdot p + O(1)$  computing  $t$ -th roots. This matches the trivial lower bound of  $\frac{p-1}{t^2}$  on the degree of polynomials computing  $t$ -th root (proved by counting zeroes of the nonzero polynomial  $f(X)^t - X$ ).

An immediate generalization of Theorem 2 shows that for all other  $p$  (namely,  $p \not\equiv 1 - t \pmod{t^2}$ , but we still preserve the condition that  $p \equiv 1 \pmod t$ ), any polynomial of degree  $d$  computing  $t$ -th roots in  $\mathbb{F}_p$  with error  $e \leq \frac{t-1}{t^2(t+1)} \cdot (p-1) - 1$  must satisfy

$$d \geq \frac{2}{t(t+1)} \cdot (p-1) - e.$$

This is  $\frac{2t}{t+1}$  times (which is about double for large  $t$ ) the trivial lower bound, but quite far from the obvious upper bound (from Lagrange interpolation) of  $\frac{1}{t} \cdot (p-1)$ .

The best upper bound we know for  $p$  not of the special form  $p \equiv 1 - t \pmod{t^2}$  is for  $p$  such that  $2p \equiv 2 - t \pmod{t^2}$  (there are infinitely many such  $p$  for any odd  $t$ ), and in this case the polynomial  $f(X) = X^{\frac{2p+t-2}{t^2}}$  computes  $t$ -th roots. This is of the form  $\frac{2}{t^2} \cdot p + O(1)$ , and thus somewhat close to our lower bound for large  $t$ .

Closing these gaps seems like a very basic and interesting open question.

---

### References

- 1 Simon Joseph Agou, Marc Deléglise, and Jean-Louis Nicolas. Short polynomial representations for square roots modulo  $p$ . *Designs, Codes and Cryptography*, 28(1):33–44, 2003.
- 2 Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 485–495, 1997.
- 3 Eric Bach and Jeffrey Outlaw Shallit. *Algorithmic number theory: Efficient algorithms*, volume 1. MIT press, 1996.
- 4 András Biró. On polynomials over prime fields taking only two values on the multiplicative group. *Finite Fields and Their Applications*, 6(4):302–308, 2000.
- 5 NA Carella. Formulas for the square root modulo  $p$ . *arXiv preprint*, 2011. [arXiv:1101.4605](https://arxiv.org/abs/1101.4605).
- 6 Seunghwan Chang, Bihtnara Kim, and Hyang-Sook Lee. Polynomial representations for  $n$ -th roots in finite fields. *Journal of the Korean Mathematical Society*, 52(1):209–224, 2015.
- 7 Don Coppersmith and Igor Shparlinski. On polynomial approximation of the discrete logarithm and the Diffie–Hellman mapping. *Journal of Cryptology*, 13:339–360, 2000.
- 8 Venkatesan Guruswami and Atri Rudra. Limits to list decoding reed-solomon codes. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 602–609, 2005.
- 9 Venkatesan Guruswami and Madhu Sudan. Improved decoding of reed-solomon and algebraic-geometric codes. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No. 98CB36280)*, pages 28–37. IEEE, 1998.
- 10 Erich Kaltofen. Effective noether irreducibility forms and applications. In *Proceedings of the twenty-third annual ACM symposium on Theory of Computing*, pages 54–63, 1991.
- 11 Richard Clive Mason. *Diophantine equations over function fields*, volume 96. Cambridge University Press, 1984.
- 12 Gary Mullen and David White. A polynomial representation for logarithms in  $\text{gf}(q)$ . *Acta arithmetica*, 3(47):255–261, 1986.

## 25:14 On the Degree of Polynomials Computing Square Roots Mod $p$

- 13 Wolfgang M Schmidt. *Equations over finite fields: an elementary approach*, volume 536. Springer, 2006.
- 14 Daniel Shanks. Five number-theoretic algorithms. In *Proceedings of the Second Manitoba Conference on Numerical Mathematics (Winnipeg), 1973*, 1973.
- 15 Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge university press, 2009.
- 16 W Wilson Stothers. Polynomial identities and hauptmoduln. *The Quarterly Journal of Mathematics*, 32(3):349–370, 1981.
- 17 Alberto Tonelli. Bemerkung über die Auflösung quadratischer Congruenzen. *Nachrichten von der Königl. Gesellschaft der Wissenschaften und der Georg-Augusts-Universität zu Göttingen*, 1891:344–346, 1891.
- 18 Joachim Von Zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge university press, 2013.
- 19 Lloyd R Welch and Elwyn R Berlekamp. Error correction for algebraic block codes. *US patent*, 4,633,470, 1983.
- 20 Arne Winterhof. Polynomial interpolation of the discrete logarithm. *Designs, Codes and Cryptography*, 25(1):63–72, 2002.