Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 179 (2021) 838–844

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on Computer Science and Computational Intelligence 2020

An Algorithms for Finding the Cube Roots in Finite Fields

Faisal[a,*], Rojali[a], Mohd Sham Bin Mohamad[b]

[a]*Mathematics Department, School of Computer Science, Bina Nusantara University, Jl. K.H. Syahdan No. 9 Palmerah, Jakarta Barat 11480, Indonesia*

[b]*Faculty of Industrial Sciences and Technology, Universiti Malaysia Pahang, Lebuhraya Tun Razak, Gambang, 26300 Kuantan, Pahang, Malaysia*

Abstract

Let F_q be a finite field with q elements. Quadratic residues in number theory and finite fields is an important theory that has many applications in various aspects. The main problem of quadratic residues is to find the solution of the equation $x^2 = a$, given an element a. It is interesting to find the solutions of $x^3 = a$ in F_q. If the solutions exist for a we say that a is a cubic residue of F_q and x is a cube root of a in F_q. In this paper we examine the solubility of $x^3 = a$ in general finite fields. Here, we give some results about the cube roots of cubic residue, and we propose an algorithm to find the cube roots using primitive elements.

Keywords: cubic residue, finite field, cube root, primitive element;

1. Introduction

One of the important problem in computational number theory is finding n-th roots in the finite field F_q. This problem is the generalization of quadratic residues of congruence modulo. The problem of finding n-th root has many applications in other area. In cryptography, pairing-based cryptography [1,2] using elliptic and hyper-elliptic curves over F_q require cube root computations. Another application of cube roots in F_q can be found in some methods of point compression [3] for elliptic curves. A method of calculation n-th root is used in some polynomial factorization algorithms [4]. In dimensional analysis, the problem concerns volume suggests cube roots as a simplifying transformation. The cube root has often been applied to precipitation data, which are characteristically right-skewed and sometimes include zeros [5].

Let $(a, m) = 1$ and $m > 0$. We say that a is **cubic residue** mod m if $x^3 \equiv a(\mod m)$ has a solution. Otherwise, a is a **cubic nonresidue** mod m. If a is a cubic residue, then x is called a **cube root** of a mod m. For example, if $m = 13$ then there are exactly four cubic residues $(\mod 13)$, that is $1, 5, 8$ dan 12. Cube roots of $1, 5, 8$ and 12 mod 13 are

* Corresponding author. Tel.: +62-21-534-5830 ext 2230
E-mail address: faisal@binus.edu

1877-0509 © 2021 The Authors. Published by Elsevier B.V.

10.1016/j.procs.2021.01.072

3, 7, 6 and 4, respectively. Note that in mod 13, each cubic residue exactly have three cube roots. For example, cube roots of 1 are 1, 3, and 9. By Hensel's Lemma and Chinese Remainder Theorem (for more details, see Rosen [6]) we can reduce the problem of finding cubic residues and cube roots mod m into mod p with p is a prime.

Cubic residues is the development theory of quadratic residues in number theory. Quadratic residue theory starts from the problem of whether there are integers x such that $x^2 \equiv a(\mod p)$ for an integer a and a prime p. The primary source for basic information about quadratic residues is the *DisquisitionesArithmeticae* [7]. Quadratic residues has been applied extensively in modern cryptology. Quadratic residues also used to maintain security when verifying identification numbers using electronic cards, electronic money, electronic banking and other similar types of communication based on a zero-knowledge proof discovered by Adi Shamir [8].

Cubic residues have been studied extensively by several authors in Namli [9], Sun [10], Sun [11], Xing et al. [12] and Ireland and Rosen [13]. It is well known that we can determine whether the integer a is a cubic residue by Euler's criterion. It was proved by L. Euler in 1761 (see Euler [14])

Theorem 1. *[Euler's Criterion] If p is an odd prime and x is a positive integer with $(x, p) = 1$, then x is a cubic residue $\mod p$ if and only if*

$$x^{(p-1)/3} \equiv 1(\mod p)$$

Proof. See Namli [15] □

A more general result is also given by Euler as written in the following theorem.

Theorem 2. *A number $a \not\equiv 0(\mod p)$ is a* ***power residue of degree*** *n modulo a prime number p if and only if*

$$a^{\frac{p-1}{\delta}} \equiv 1(\mod p)$$

where $\delta = lcm(p-1, n)$.

We may divide the study of cubic residue in congruence modulo prime number into two major problems, the first is finding elements which is a cubic residue. The second problem is computing cubic roots from a cubic residue. The study of quadratic residue and cubic residue can be generally developed in finite field structure. This can occur because of numbers in congruent mod p with p is a prime can be regarded as an element of a finite field. In order to compute cube roots in finite fields, there are two standards algorithms, that is the Adleman- Manders-Miller [16] whose complexity is $O(\log^4 q)$, and the Cipolla-Lehmer [17,18] algorithm whose complexity is $O(\log^4 q)$. The first algorithm is a generalization of the Tonelli-Shanks square root algorithm [19,20]. In Faisal and Gazali [21], they provide an algorithm for finding square root in finite fields using the properties of primitive element. Based on their result, we develop an algorithm for computing the cube root in finite fields.

Based on the importance of finding cube root, we interested to solve the cubic root problems in finite fields. Using the property of finite fields, the cubic residue and cube root problem can be solved algebraically. In this paper, we also propose an algorithm for finding cube roots in general finite field.

The remainder of this paper is organized as follows: In Section 2, we collect some basic properties of finite field F_q to support this research. In Section 3, we write the research methodology. In Section 4, we introduce the notion and results about cubic residues in finite field F_q and we give the root extraction formula in F_q and we present a cube root algorithm based on primitive elements of finite field. Finally, in Section 5, we give a summary of our result.

2. Literature Review

In this section we discuss the fundamental theory and results of finite fields. The next result plays important role in the problem of finding the number of cubic residues of finite fields.

Theorem 3. *Let F be a field and let $f(x)$ be a nonzero polynomial of degree n. The polynomial f has at most n distinct roots in F.*

Proof. See Irving [22] □

We collect some important properties of finite fields.

Theorem 4. *Every finite field has prime power order.*

Proof. See Irving [22] □

If F is a field, we denote by F^*, the set of all nonzero element of F. It is clear that F^* is a group under the multiplication. In particular, for finite fields we have the following property.

Lemma 1. *If F is a finite field, the group F^* is cyclic.*

Proof. See Howie [23]. □

If F is a finite field with q elements, then by lemma 1 we can write $F^* = \{g, g^2, \ldots, g^{q-1}\}$ for some $g \in F^*$. Such element g is called a primitive element of F. The following result guarantee the existence of a primitive element in finite fields.

Theorem 5. *Every finite field has at least one primitive element.*

Proof. See Irving [22] □

Since F^* is cyclic for every finite field F. We have the following consequence.

Lemma 2. *If F_q is a finite field with q elements and $x \neq 0 \in F_q$, then $x^q = x$, for all x in F_q.*

Proof. See Ling and Xing [24]. □

Theorem 6. *If F is a finite field, then F is isomorphic to $F_p[x]/(\pi(x))$ for some prime p and some monic irreducible in $F_p[x]$.*

Proof. See Irving [22] □

Suppose p is a prime number, by Theorem 6, a finite field F_p with p elements is isomorphic to $\mathbb{Z}_p$.

3. Methodology

This type of the research is a qualitative research. The results of this study are obtained by proving the properties of a finite field, that is a mathematical system which satisfy some certain conditions. We derive the formula the cube root by exploring the theory of quadratic residues and cubic residues in finite fields. This research method is divided into several stages:

1. Finding similar results for n-root problems.
2. Collect basics and important properties of finite fields.
3. Finding the number of cubic residues in finite fields.
4. Characterize cubic residues in finite fields.
5. Finding the formula of cube root ini finite fields.
6. Design the algorithm from the formula of cube roots.
7. Implementation the algorithm in Python programme.

4. Results

4.1. Cubic residues of Finite Fields

Generally, we can extend the cubic residue notion into finite fields.

Definition 1. Let a be a nonzero element of a finite field with q elements F_q. We say a is a **cubic residue** of F_q if there exists $x \in F_q$ such that $x^3 = a$. Otherwise, a is a **cubic nonresidue** of F_q.

We denote the set of cubic residues of a finite field F_q by $CR(q)$. It is well known that for a prime number p with $p \equiv 2 \mod 3$, we have $|CR(p)| = p - 1$. We also have similar result for a finite field with p^n elements for some prime number p.

```
print("Finding the cube root in Z_q, with q is an odd prime")
q = input("Input an odd prime q: ")
c = input("Input an integer of Z_q: ")
cr = []
ncr = []

for a in range (2,q): # find primitive element of fields Z_q
    n = 0
    p = 1
    while (p != 1 or n==0):
        p = p*a % q
        n = n+1

    if  n == q-1:
                cr.append(a)
    else:
                ncr.append(a)

primitive = cr[0]
primitive3 = (primitive**3)%q
qmod= q % 3
test = primitive3
if qmod == 0: # case q congruent 0 modulo 3
    k = q/3
    root = c**k % q
    print("the modular cube root of this integer: ",root)
else:
    if qmod == 2: # case q congruent 2 modulo 3
        k=(q-2)/3
        t=q-2
        cp= c**k % q
        i= cp**t % q  # inverse modulo of cp
        i=int(i)
        root = i
        print("the modular cube root of this integer: ",root)
    else : # case q congruent 1 modulo 3
        if (test == c):
            root = primitive
        else :
            r=1
            while (test !=c):
                r = r+1
                test = (primitive3**r)%q
                root = primitive**r%q
                if r==(q-1)/3:
                    break

        if  (test !=c):
            print(c," is a cubic nonresidue")
        else:
            print("a modular cube root of this integer: ",root)
```

Fig. 1. cube root in $\mathbb{Z}_q$

Theorem 7. *Let $q = p^n$ with p is a prime number. If $q \not\equiv 1 \mod 3$, then every nonzero element of F_q is a cubic residue.*

Proof. If $q \equiv 0 \mod 3$, then $q = 3k$ for an integer k. Let a be an element of F_q with $a \neq 0$. By Lemma 2, $(a^k)^3 = a^{3k} = a$. It follows that a is cubic residue. Now, suppose that $q = 3k + 2$ for a non-negative integer k. By lemma 2 we obtain that

$$a^{3k} = a^{-1}.$$

Equivalently, $a = (a^{-k})^3$. We conclude that a is a cubic residue in F_q. □

Note that $a^{-k} = (a^{-1})^k = (a^k)^{-1}$ for all element $a \in F_q$

Theorem 8. *Let $q = p^n$ with p is a prime number. If $q \equiv 1 \mod 3$, then the number of cubic residues in finite field F_q is $\frac{q-1}{3}$.*

Proof. Since F_q is a finite field, there exist a primitive element $g \in F_q$. Suppose that $q = 3k + 1$ for an integer k. We can write $F_p^* = \{g, g^2, \ldots, g^{3k}\}$. If $a \in F_p^*$ is equal to g^m with m is divisible by 3, then $a \in CR(q)$. Note that the number of such elements in F_q^* is k. It follows that $\frac{q-1}{3} = k \leq |CR(q)|$. Suppose that $f(x) = x^{q-1} - 1 \in F_p[x]$. We may write $f(x) = x^{3k} - 1$. By Lemma 2, every nonzero element $a \in F_q$ is a root of f, that is, $a^{q-1} - 1 = 0$. Now, consider the polynomial $h(x) = x^k - 1 \in F_q[x]$. Suppose that $a \in F_q$ is a cubic residue. Thus, there exists element $y \in F_q$ such that $a = y^3$. It follows that

$$h(a) = a^k - 1 = (y^3)^{3k} - 1 = y^{3k} - 1 = 0.$$

It follows that a is a root of h. But from theorem 3, the polynomial $h(x)$ has at most k different roots. We conclude that $|CR(p)| \leq k = \frac{p-1}{3}$. This completes the proof. □

We summarize the results regarding cubic residues of finite fields.

Corollary 1. *Let F_q be a finite field with q elements.*

1. *If $q \not\equiv 1(\mod 3)$, then $CR(q) = F_q^*$,*
2. *if $q \equiv 1(\mod 3)$, then $CR(q) = \{g^3, g^6, \ldots, g^{3k}\}$ with $q = 3k + 1$ and g is a primitive element of F_q.*

4.2. Cubic roots of Cubic Residues in Finite Fields

In this section we will discuss the cube roots of a cubic residue in finite fields. We propose an algorithm that will help us find the cube roots of a cubic residue in finite fields. First, we give a result on the uniqueness of the cube root of any cubic residue of F_q for $q \not\equiv 2(\mod 3)$.

Lemma 3. *Let F_q be a finite field with $q \not\equiv 1(\mod 3)$. If $a \in F_q$ is a cubic residue, then there exists a unique element $x \in F_q$ such that $x^3 = a$.*

Proof. If $q \equiv 0(\mod 3)$, then $a^{3k} = a$ or equivalently $(a^k)^3 = a$. Hence, $x = a^k$ is a solution of $x^3 = a$. Since every nonzero element is a cubic residue, we have one-to-one correspondence $F_q^* \to F_q^*$ defined by $a \mapsto a^k$. This proves the uniqueness statement. Similarly, if $q \equiv 2(\mod 3)$, then $a = (a^{-\frac{q-2}{3}})^3$. Setting $x = (a^{\frac{q-2}{3}})^{-1}$, it can be checked that x is a solution of $x^3 = a$. Again, there exists one-to-one correspondence $F_q^* \to F_q^*$ defined by $a \mapsto (a^{\frac{q-2}{3}})^{-1}$. □

For the case $q \equiv 1(\mod 3)$ there are exactly three cube roots for every cubic residue element in F_q.

Lemma 4. *Let F_q be a finite field with $q \not\equiv 1(\mod 3)$. If $a \in F_q$ is a cubic residue, then the equation $x^3 = a$ has exactly three solution in F_q.*

Proof. Let $f(x) = x^3 - a$ be a polynomial in $F_q[x]$. Since a is a cubic residue, the polynomial f has at least one root in F_q. By Theorem 3, the polynomial f has at most three distinct root in F_q. Define a subset $CR_a = \{x \in F_q | x^3 = a\}$ for any $a \in CR(q)$. Assume that there exist $b \in CR(q)$ such that $|CR_b| < 3$. Let $k = \frac{q-1}{3}$, by pigeonhole principle there exist an element $c \in CR(q)$ such that $|CR_c| \geq 4$. Indeed, $|F_q^*| - |CR_b|$ is equal to $3k-1$ or $3k-2$ and $|CR(q)| - |\{b\}| = k-1$. If $|CR_c| \geq 4$ then the equation $x^3 = c$ has more than 3 distinct solution x in F_q. It is a contradiction, since the polynomial $x^3 - c$ has at most three distinct root in F_q. We conclude that $|CR_a = \{x \in F_q | x^3 = a\}| = 3$ for any $a \in CR(q)$. This completes the proof. □

4.3. A Cube Root Algorithm

For the case of q with $q \equiv 1(\mod 3)$ cubic residues of F_q can be determined completely using primitive elements. This gives an idea for finding the cubic roots of a cubic reside element of F_q using primitive element. For the case of $q \equiv 0(\mod 3)$, it is easy to see that the cubic root of non-zero element a of F_q is a^k with $k = \frac{q}{3}$. For the case of q with $q \equiv 2(\mod 3)$, based on the proof of Theorem 7 the cubic root of non-zero element a of F_q can be computed using the equation $a = (a^{-\frac{q-2}{3}})^3$. Next, we give the main result of this paper, that is an algorithm to find the cube roots of cubic residues in general finite field F_q using primitive elements. Note that step 3 in the Table 1 is the process

Table 1. Cube root algorithm in finite fields

Algorithm cubic root for a cubic residue of finite field F_q

Input : $a \in F_q^*$
Output A cube root x such that $x^3 = a$
1: if $q \equiv 0(\mod 3)$ then $x = a^{\frac{q}{3}}$
2: if $q \equiv 2(\mod 3)$ then
  $y \leftarrow a^{\frac{q-2}{3}}$
  $x \leftarrow y^{-1}$
3: else
  Pick a primitive element g in F_q
  $g_0 \leftarrow g^3$
  if $a = g_0$ then $x \leftarrow g$
  else
    $k \leftarrow 1$
    while $g_0^k \neq a$ do
      $k \leftarrow k + 1$
      $t \leftarrow (g_0)^k$
      $x \leftarrow g^k$
      if $k = \frac{q-1}{3}$
        break
  if $t \neq a$ then a is a cubic nonresidue
  else
    x is a cube root of a

of finding the root cube in the case finite field F_q with $q \equiv 1(\mod 3)$. This algorithm only gives one root of three roots of a cubic residue in the case $q \equiv 1(\mod 3)$. To get all three roots we can take another primitive element when choosing g_0. We implement the above algorithm in the Python program by taking the case of the prime finite field $\mathbb{Z}_q$ with q is a prime number. The programme was run in Python 2.7.10 with the running time is $O(q^2)$. The source code of our cubic root programme can be seen in Figure 1.

5. Conclusion and Future Works

We present an algorithm to find cube roots in general finite F_q with q element using the properties of finite fields. A cube root of $a \in F_q$ is an element $x \in F_q$ which satisfies the equation $x^3 = a$. The algorithm is divided into three cases based on the value of $q(\mod 3)$. The first case is $q \equiv 0(\mod 3)$ where the solution of the equation $x^3 = a$ is $x = a^{\frac{q}{3}}$. The second case is $q \equiv 2(\mod 3)$, while the cube root of a is the inverse of $a^{\frac{q-2}{3}}$. The last case is when $q \equiv 1(\mod 3)$, we obtain that a cube root of a is g^{3t} for some integer $1 \leq t \leq \frac{q-1}{3}$ where g is a primitive element. The time complexity of our algorithm is $O(q^2)$. However, the implementation can be further optimized by improving the process for finding primitive element (for more detail, see the link https://cp-algorithms.com/algebra/primitive-root.html#: :text=First%2C%20find%20%CF%95(n),g%20is%20a%20primitive%20root). Next, we expect to generalize our algorithm to find the n-th roots of an element in finite field.

Acknowledgements

We most thankful for the Bina Nusantara university that has provided research grants Penelitian Internasional Binus (PIB) 2020 for our research. We are grateful for the insightful comments and the expertise offered by all colleagues to improve this paper in countless ways. We would like also to thank Rafael Herman Yosef for helping to calculate the complexity of the programme in Python.

References

1. Boneh, D., Franklin, M.. Identity based encryption from the weil pairing. *Crypto 2001, Lecture Notes in Computer Science* 2001;**2139**:213–229.
2. Duursma, I., Lee, H.. Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$. *Asiacrypt 2003, Lecture Notes in Computer Science* 2003;**2894**:111–123.
3. A. Dudeanu, G.O., Iftene, S.. An x-coordinate point compression method for elliptic curves over $\mathbb{F}_p$. *Proc of 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2010)* 2010;:65–71.
4. A. J. Menezes, P.C.v.O., Vanstone, S.A.. *Handbook of Applied Cryptography*. CRC Press; 1996.
5. Cox, N.J.. Precipitation statistics for geomorphologists: Variations on a theme by frank ahnert. *Catena 23 (Suppl)* 1992;:189–212.
6. Rosen, K.. *Elementary Number Theory and Its Applications*. Addison-Wesley; 2011. ISBN 9780321500311.
7. Gauss, C.F.. *Disquisitiones Arithmeticae Disquisitiones Arithmeticae, 1801; English translation by A. A. Clarke*. New york: Springer-Verlag; 1986.
8. Shamir, A.. Identity-based cryptosystems and signature schemes, in g. r. blakely and d. chaum, eds. *Advances in Cryptology* 1985;:47–53.
9. Namli, D.. Cubic residue characters. *Int Math Forum 8* 2013;(1-4):67–72.
10. Sun, Z.. On the theory of cubic residues and nonresidues. *Acta Arith* 1998;**84**(4):291–335.
11. Sun, Z.. On the theory of cubic residues and nonresiduescubic residues and binary quadratic forms. *J Number Theory* 2007;**124**(1):62–104. URL `http://dx.doi.org/10.1016/j.jnt.2006.08.001`.
12. Xing, D.S., Cao, Z.F., Dong, X.L.. Identity based signature scheme based on cubic residues. *Sci China Inf Sci* 2011;**54**(10):2001–2012. URL `http://dx.doi.org/10.1007/s11432-011-4413-6`.
13. Ireland, K., Rosen, M.A.. *A classical introduction to modern number theory, Second edition. Graduate Texts in Mathematics*. New York: Springer-Verlag; 1990. ISBN 0-387-97329-X.
14. Euler, L.. Adnotationum ad calculum integralem euleri g. kowalewski (ed.). *Opera Omnia Ser 1; opera mat* 1914;**12**:493–538.
15. Namli, D.. Some results on cubic residues. *International Journal of Algebra* 2015;**9**(5):245–249. URL `http://dx.doi.org/10.12988/ija.2015.5525`.
16. L. Adleman, K.M., Miller, G.. On taking roots in finite fields. *Proc 18th IEEE Symposium on Foundations on Computer Science (FOCS)* 1977;:175–177.
17. Cipolla, M.. Un metodo per la risoluzione della congruenza di secondo grando. *Rend Accad Sci Fis Mat* 1903;**9**:154–163.
18. Lehmer, D.H.. Computer technology applied to the theory of numbers. *Studies in Number Theory* 1969;:117–151.
19. Tonelli, A.. Bemerkung uber die auflosung quadratischer congruenzen. *Göttinger Nachrichten* 1891;:344–346.
20. Shanks, D.. Five number-theoretic algorithms. *Proc 2nd Manitoba Conference on Numberical Mathathematics* 1972;:51–70.
21. Faisal, , Gazali, W.. An algorithm to find square root of quadratic residues over finite fields using primitive elements. *Procedia Computer Science* 2017;**116**:198–205.
22. Irving, R.S.. *Integers, Polynomials and Rings*. New York: Springer; 2004.
23. Howie, J.M.. *Field and Galois Theory*. Springer Undergraduate Mathematic Series. London: Springer-Verlag; 2006.
24. Ling, S., Xing, C.. *Coding Theory A first Course*. New york: Cambridge University Press; 2004.