On Computing Finite Fields of Z_n, From Brain Sloan Techniques

Anand G Puranik
Assistant Professor, Dept of Mathematics
Government Science College Chitradurga – 577 501 India

Abstract- *In this paper, sets that form finite fields, were extracted from the set of integers modulo n, where n is finite composite number, by using Brain Sloan techniques. Previously Brain Sloan invented unique technique to extract cyclic multiplicative groups from U(n), using isomorphism and number theory results. Similar techniques were used to create computer programs, which construct sets, those form finite fields.*

Keywords- Algebra, Fields, Groups, Computer programming, Scilab, C, Phython,

I. INTRODUCTION

Finite Field Theory from its inception by Galois[8], became a fundamental mathematical activity for many Mathematicians.Finite fields provide lot of impetus and applications in number theory, algebraic geometry, Galois Theory, finite geometry, cryptography and coding theory.It is easy and simple to understand, Zp the set of integers modulo p, where p is a prime number, is a Finite Field or Galois Field[20][8],[10]. These can be verified by the computer software, by constructing additive group of whole set and multiplicative group of non-zero elements of Zp. In this paper we concentrate on the extraction of the sets from the set of integers modulo n, (n is a composite number), which form finite field, with the binary operations $+_n$ and X_n.In my previous article, I used the technique of removal of zero divisors from the cyclic subgroups of additive group Zn, where n is a composite number. As usual theresults mentioned in this research work also, originated from earlier literature mentioned in the references.Computing and verification of algebraic structures using programming languages is also an interesting intellectual and mathematical challenge. In this direction we find very less literatures, but have prime importance in modern world. At the outset, inspired by the article entitled, " All Cyclic Subgroups In Group (Z_mXZ_n , +) Using Python" , by the authors Bobbi Rahman, SamsulArifin, IndrabayuMuktyas[16] I started search for programming techniques for algebraic structures. As mentioned in my earlier paper, most of the material were brought from the works of, Nor MuhainiahMohd Ali, Deborah Lim Shin Fei, Nor HanizaSarmin, ShaharuddinSalleh, entitled, " A VISUAL MODEL FOR COMPUTING SOME PROPERTIES OF U(n) AND Zn"[17]. My interest in finding those subsets of Z_n, (where n is not a prime number), that form multiplicative groups under the binary operation multiplication modulo n (X_n) took shape from the article entitled, "ON THE NUMBER OF CYCLIC SUBGROUPS OF A FINITE GROUP", by the authors Mohammad Hossein Jafari and Ali Reza Madadi, Then the research paper entitled," MULTIPLICATIVE GROUPS IN Z_m", by the aouther, BrianSloan [5], gave a fresh impetus over the subject and provided a clue to write C program.**Initially I framed Scilab programs to compute additive cyclic groups, then those sets forming finite field were obtained by removing zero divisors. Later, using Brain Sloan technique, The multiplicative subgoups of Zn, where n is not prime (composite) number, are computed. Then the additive identity element is added to the same multiplicative group, then the new set is verified for characteristics of Field.** All the initial terms, terminologies and results used are brought from the earlier research papers, on the same lines of researchpapers[17],[16],[20]).

Throughout all groups are assumed to be finite. A group is a non-empty set G with a binary operations *, that is closed, associative, includes an identity element and each element in G, has an inverse. **Identity element** refers to an element 'e' (called the identity) in G such that a*e = e*a=a for all a in G. **Inverses**. For each element a in G, there is an element b in G (called an inverse of a) such that a*b=b*a=e. Gallian [8] has shown that the identity and inverse of any elements on a group are unique, and also cancellation laws holds in the group. If a group G, has another property a*b=b*a, for any a and b in G, then we said that group G is commutative. The basic properties of groups are used as given in [8], [10] and [20]. A cyclic group G is a group in which any element g in G can be written as g^n for n = 1, 2,...O(G). Furthermore, the characteristics of cyclic groups were mentioned in resources[20], [10] and [8]. Subgroup is a non-empty subset H, of a group G which is also a group with the same binary operation as in G. For an element 'a' in the group G (i.e, a∈ G), we can form a subset S that contain all those elements of G which are of the forms, a^n for n =1, 2,... . This subset forms a subgroup in G, and called a cyclic subgroup that generate by a. Recall that any cyclic group is commutative

and subgroups of a cyclic group are also cyclic. The set of all integers modulo n, denoted by Z_n, is a group of modulo addition operations. The group $(Z_n , +_n)$ are constructed using the division algorithm on the set of all integers. This process can be studied in [8][10] and [20]. Furthermore, the formation process of the group $(Z_mXZ_n,\times+)$ can be studied in [8] and [20]. Representing groups and their internal structures, using computers is also one of the present needs. This will help both in understanding group theory and invent more algebraic structures, with less human efforts. Scilab is one of the application package made for the Mathematical and Statistical computations, based on C and Frotran. So initialy, I wrote programs in Scilab. Python is the best suited and current trending programming language used for Mathematical computations. Python is a multipurpose programming language and easy to study (see [18]). Python can also run on various operating system platforms, such as Windows, Linux, Mac OS, Android (see [18]), and the others. Furthermore, study of the C, Python programs which are the focused results of this paper and its output will be discussed. The main technique used here, never computes factors of n, and order of elements, before generating the required groups. This is the major deviation from the earlier results. Following the definitions of cyclic groups and the generator of a group, the additive as well as multiplicative groups were constructed, from Z_n, removing redundant sets or groups. An element a in G is taken and the cyclic group generated by 'a' denoted as <a> is computed, using both additive and multiplicative integer modulo operations.

The research article [5], induces many curious facts regarding the multiple groups of Z_n, where n is not prime. It reveals many multiplicative groups under multiplication modulo m, which are may be cyclic and may not be cyclic. Let us take one example, from the additive group of Z_{40} under +40, as mentioned in the research article[5]. Consider the set {8,16,24,32}which is a group under X_{40}, as revealed by the following composition table,

X_{40}	8	16	24	32
8	24	8	32	16
16	8	16	24	32
24	32	24	16	8
32	16	32	8	24

$+_{40}$	0	8	16	24	32
0	0	8	16	24	32
8	8	16	24	32	0
16	16	24	32	0	8
24	24	32	0	8	16
32	32	0	8	16	24

It is sufficient to comprehend, from the above table that the set S={0,8,16,24,32} form a field under the binary operations +40 and X40, with 0 and 16 as the additive identity elements respectively. A similar example is the with the set T={0,4,8} in the set Z_{12} with 0 and 4 as the identity elements of $+_{12}$ and X_{12}. From the works of Brain Sloan[5], we get only peripheral ideas, not centrally focused on the multiplicative subgroup. A typical example again arises from the set Z_{40}, as mentioned in the same paper[5]. Consider the set S={0,5,10,15,20,25,30,35} which forms a additive group under+40. But the set S1={5,10,15,20,25,30,35} does not form a group under multiplication X40. Again if you consider the subsets of S1, say S2={5,15,25,35} and S3={5,25} both will form a group under the binary operation X_{40}. Such curious facts, leads to high potential thoughts having multiple direction, even consisting of number of groups that can be formed from the set of Z_m, where n is not a prime number. So in this paper, without giving prime importance to the theoretical aspects of group theory, I concentrated only on extraction of those sets, which with zero (say S) form a additive group under $+_m$ and without zero(S–{0}) form a group under X_m.

II. METHODS, RESULTS ANDTECHNIQUES

In this paper, for method 1, initially given n is tested as a prime or composite number, if n is prime, then Z_n itself is a finite Field(Prime field). The program displays the set(Field) members. Displays the multiplicative subgroups of $Z_n – \{0\}$ and terminates. When n is not a prime number, without going deep in the theory of groups, both the additive subgroups in Z_n and multiplicative groups in $Z_n – \{0\}$ are generated by each element of the bigger set Z_n. Later, only distinct additive subgroups of Z_n removing zero divisors are tested for multiplicative groups. Those subgroups and multiplicative groups were considered, to test the field property. Hence the finite Fields of the Z_n are computed. In the next Method 2, the cyclic multiplicative subgroups of Z_n, where n is not prime, were computed by using **Brain Sloan technique**. Then from the cyclic multiplicative subgroups, by adding zero, we verify for the additive subgroup. If all the elements of the set form additive subgroups and non-zero elements form the cyclic multiplicative subgroup, then it will be Field.Computer programs to compute both the number of Fields and Fields in the set Z_n are written in Scilab, later in C, Phython.

III. ALGORITHM

A simple algorithm of both method 1 and method 2, are given before converting the same into high language program. In method 1, the algorithm has following steps.

a. Read n,
b. Test n as a prime, if n is prime then display the elements and print Z_n itself a prime field. Stop.
c. If n is not prime, then find the factors of n,
d. For each factor of n, find the cyclic additive subgroup generated by the factor under $+_n$.

e. From the additive cyclic group generated remove zero(additive identity element), then verify that the non-zero elements form the multiplicative subgroup under Xn.

f. If both d and e steps holds then, the subgroup becomes a field in the set Z_n.

Method 2, algorithm is based on the Brain Sloan Technique of finding multiplicative cyclic subgroups in Z_n, where n is not a prime.

1. Read n
2. Verify n is a prime number or not
3. If n is a prime number, display Z_n is a field
4. Else find the factors of n, as f[i], for i=1,2,..n/2
5. For each factor f[i], find the unit elements U(i) of the ring
6. The number elements (say, L) in the set U(i)+{0} is counted.
7. If L is p, or p^k, for some prime number then the set U(i)+{0} is verified for the field property
8. Else U(i) forms only a multiplicative group, under the binary operation X_n.
9. Stop.

IV. SOURCE CODE

Consider the following source code in Scilab, which produces cyclic additive subgroup generated by each element in the group of Z_m, where m may be prime or composite number. The subgroups generated are redundant as both a and a^{-1} generate the same subgroup.

```
clc();
n=input("n=");
ifmodulo(n,2)==0then
k=n
else
k=n+1
end
flag=1
fori=1:k/2
ifmodulo(n,i)==0then
flag=0
break;
end
end
ifflag==0then
printf("n is not a prime\n")
printf("hence there are additive subgroups\n")
G=[0:n-1];
disp(G);
fori=1:n-1
ifmodulo(n,i)==0then
t(i)=i
H(i,1)=t(i)
printf("%d\t",H(i,1))
k(i)=1
forj=2:n
H(i,j)=modulo(t(i)+H(i,j-1),n)
printf("%d  ",H(i,j))
ifH(i,j)==0then
break
end
k(i)=k(i)+1
end
s(i)=k(i)+1
printf("A subgroup of order %d\n",s(i))
printf("\n")
end
end
fori=1:n-1
ifmodulo(k(i),2)==0then
u=k(i)
else
u=k(i)+1
end
flag=1
forj=1:u/2
ifmodulo(k(i),j)==0then
flag=0
printf("%d is not prime\n",k(i))
printf("there are  multiplicative subgroups\n")
break;
end
end
ifflag==1then
printf("%d is a prime\n",k(i))
printf("there are n0 multiplicative subgroups\n")
end
ifk(i)==2then
forl=1:k(i)
printf("%d\t",H(i,l))
end
end
end
else
printf("nis a prime number\n")
printf("Hence Zn itself is a field \n")
printf("there are no additive subgroups\n")
end
```

The output for Z_{24} is given below.

n=24

0. 1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23.
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 0 A subgroup of order 24

2 4 6 8 10 12 14 16 18 20 22 0 A subgroup of order 12

3 6 9 12 15 18 21 0 A subgroup of order 8

4 8 12 16 20 0 A subgroup of order 6

6 12 18 0 A subgroup of order 4

8 16 0 A subgroup of order 3

12 0 A subgroup of order 2

Note that subgroups of same orders, generated were reoccurred. But properly writing the source code, the reoccurrence can be removed. Another change from the previous literature, introduced was the factor of n and order of the elements were not used here. My main intention is only to concentrate on the finite fields, extracted from the set Z_m, where m is not prime. This intension is achieved only simple facts of group theory and characteristics of Fields. Use simple mathematical open source software was used to compute such finite fields.

V. CONCLUSION

The conclusions that can be obtained from this study are as follows: a)Finite Prime fields from the set of integers modulo n (Z_n) were computed. b) Using the Scilab or Python program, we can determine all cyclic subgroups and Finite fields of the group or set ($Z_n \times_n +_n$) easily. c) From the program that has been created, the maximum number of Finite fields may be verified. d) Further the same technique may be extended to verify, whether the finite prime fields, characteristic fields in Direct product of groups exit or not.

VI. ACKNOWLEDGMENT

Author of this paper, thanks all the researchers mentioned in the introduction, and also thanks for the editor and reviewer of this paper. The suggestions of reviewer and editor were incorporated in this article, to provide excellent material on the subject,

REFERENCES

[1] Adkins, W.A. and Weintraub, S.H., 2012. Algebra: an approach via module theory (Vol. 136). Springer Science & Business Media.
[2] Arifin, S., 2018. Grup Faktordari Sebarang Subgrup Siklikdari Grup (Zn,+). SCIENCE TECH: JurnalIlmiahI lmu Pengetahuandan Teknologi, 4 (2), 53-58.
[3] Arifin, S. and Garminia, H. 2018. Valuation Dimension of Ring n Using Python. International Journal of Engineering & Technology, 7 (4), 6351-6356
[4] Arifin, S. and Garminia, H. 2019. Uniserial Dimension Of Module m n × Over Using Python. International Journal of Scientific & Technology Research, 8(7), 194-199
[5] Brian Sloan, MULTIPLICATIVE GROUPS IN $Z_{m,,}$Semantic Scholar Carpus Id 416419
[6] [Dummit, D.S. and Foote, R.M., 2004. Abstract algebra (Vol. 3). Hoboken: Wiley.
[7] Fraleigh, J.B. 2000. A First Course in Abstract Algebra, Sixth Edition, Addison-Wesley, New York.
[8] [Gallian, J.A. 2017. Contemporary Abstract Algebra, 9th Edition, USA.
[9] Google. (2018, 30 April): available at https://play.google.com/store/apps/details?id=org.qpython .qpy&hl=e n
[10] Herstein I. 1996. Abstract Algebra, 3rd Edition, Prentice Hall, New York.
[11] Huang, H. 2018, 28th July. Algebra Lecture Notes, Auburn University Press, available at http://www.auburn.edu/ huanghu/math5310/
[12] [Isaacs, I.M., 1994. Algebra, a graduate course, Brooks.Cole Publishing Company, Pacific Grove, California.
[13] Malik, D.S., Moderson, J.N., and Sen, M.K. 1997. Fundamentals of Abstract Algebra, USA.
[14] Muktyas, I.B., and Arifin, S. 2018. SebarangPembangunSubgrupSiklik Dari SuatuGrup (Zn,+). JurnalMatematika" MANTIK", 4 (2), 116-121.
[15] Muktyas, I.B., and Arifin, S. 2018. SemuaSubgrupSiklikdariGrup (Zn,+). JurnalTeorema: TeoridanRisetMatematika. Vol 3 No 2, 177-186, September 2018.
Mohammad Hossein Jafari and Ali Reza Madadi, ON THE NUMBER OF CYCLIC SUBGROUPS OF A FINITE GROUP Bull. Korean Math. Soc. 54 (2017), No. 6, pp. 2141–2147 https://doi.org/10.4134/BKMS.b160783 pISSN: 1015-8634 / eISSN: 2234-3016.
[16] Nor MuhainiahMohd Ali, Deborah Lim Shin Fei, Nor HanizaSarmin, ShaharuddinSalleh, entitled, A VISUAL MODEL FOR COMPUTING SOME PROPERTIES OF U(n) AND Zn, Proceedings of the secound IMT-GT

Regional Conference on Mathematics, Statistics and Applications UniversitiSains Malaysia
[17] Python. 2018, 30 April. available at https://www.python.org/.
[18] Rotman, J. J. 2003. Advanced Modern Algebra, Prentice Hall, New York
[19] Surjeetsingh and QuaziZameeruddin, Modern Algebrs S Chand and Company Ltd New Delhi – 110014.