International Journal of Education and Science Research Review

E-ISSN 2348-6457

Volume-3, Issue-2 March-April- 2016

P-ISSN 2349-1817

www.ijesrr.org

Email- editor@ijesrr.org

ALGEBRAIC NUMBER THEORY AND CLASS FIELD THEORY

Anupam Tandon

Assistant Professor, Mathematics, SIRDA Engineering College

Abstract

Algebraic number theory mainly investigates algebraic structures related to number fields. Among the principal topics of investigation in this subfield of mathematics are the characteristics of integers and the extensions of those integers. This field of study investigates a variety of topics, including the behaviour of prime factorisation in number rings, the distribution of ideals, and the arithmetic of algebraic integers. As a result of the fact that it offers profound insights into the characteristics of number fields, the study of zeta functions, class groups, and unit groups is one of the primary issues addressed by this area. "Class Field Theory," sometimes known as "CFT," is one of the most significant findings in the field of algebraic number theory. This theory provides a comprehensive description of the abelian extensions of number fields by making use of reciprocity principles and ideal class groups. Through the use of Artin reciprocity and Takagi's theory, a robust relationship is made between Galois theory and number theory. This connection is formed when the structure of abelian extensions is shown. In addition to having a significant impact on contemporary cryptography and arithmetic geometry, CFT also allows for the development of sophisticated tools that may be used to investigate prime behaviour in field extensions. This article begins with a review of some of the most important concepts in algebraic number theory, and then moves on to discuss the fundamental discoveries made in class field theory. An investigation of the influence that concepts such as local and global reciprocity laws, ideal class groups, and ramification theory have had on the teaching of mathematics in the modern era is presented.

Keywords: *Algebraic, Number, Theory*

Introduction

The arithmetic properties of number fields, which are extensions of the rational numbers, are the subject of study in the discipline of algebraic number theory, which is one of the most important branches of mathematics. Q is a mathematical expression. Within the scope of the study, the concept of integers is broadened to include algebraic integer rings, and concepts such as prime factorisation, ideal class groups, and unit groups are investigated. The subject matter has a significant impact on the distribution of prime numbers, modular forms, and Diophantine equations, among other significant effects. Due to the fact that it provides a taxonomy of abelian extensions of number fields, Class subject Theory (CFT) is considered to be a key work in the subject of algebraic number theory. Kronecker's concept of relating field extensions to ideal class groups and Gauss's work on quadratic reciprocity were the foundations upon which the Quantum Field Theory (CFT) was first built. The theory, which was meticulously developed by Takagi, Artin, and Chevalley, establishes not only a strong connection between Galois theory and the mathematics of number fields, but also a strong connection between the two. By providing an outline of the connection between abelian Galois extensions and generalised ideal class groups, it offers a comprehensive basis for understanding field extensions. One of the most significant theoretical achievements, Class Field Theory has several practical applications in a wide range of domains, including modular forms, encryption, and elliptic curves, among others. The fundamental concepts of quantum mechanics are obtained from an analysis of algebraic number theory, which is presented in this article. A variety of topics are included in this discussion, including the structure of number fields, the qualities of ideals, and the reciprocity laws, which serve as the foundation for the present algebraic number theory.

The ring of integers

Let K be an algebraic number field. Each element α of K satisfies an equation

$$\alpha^n + a_1\alpha^{n-1} + \cdots + a_n = 0$$

with coefficient $a_1,\ldots,a_n$ in $\mathbb{Q}$, and α is said to be an algebraic integer if it satisfies suc an equation with coefficient $a_1,\ldots,a_n$ in $\mathbb{Z}$. We shall see (2.1) that the algebraic integers form a subrin $\mathcal{O}_K$ of K.

An algebraic number is an algebraic integer if and only if its minimal polynomial over $\mathbb{Q}$ has coefficients in $\mathbb{Z}$ (see 2.11). Consider, for example, the field $K = \mathbb{Q}[\sqrt{d}]$, where d is a square-free integer. The minimal polynomial of $\alpha = a + b\sqrt{d}, b \neq 0, a, b \in \mathbb{Q}$, is

$$(X-(a+b\sqrt{d}))(X-(a-b\sqrt{d})) = X^2 - 2aX + (a^2 - b^2d),$$

and so α is an algebraic integer if and only on

$$2a \in \mathbb{Z}, \quad a^2 - b^2d \in \mathbb{Z}.$$

2 Z: From this it follows easily that, when d =2; 3 mod 4, α is an algebraic integer if and only if a and b are integers,

$$\mathcal{O}_K = \mathbb{Z}[\sqrt{d}] = \left\{a + b\sqrt{d} \mid a, b \in \mathbb{Z}\right\},$$

o ; and, if d = 1 mod 4, α is an algebraic integer if and only if a and b are either both integers or both half-integers,

$$\mathcal{O}_K = \mathbb{Z}[\tfrac{1+\sqrt{d}}{2}] = \left\{a + b\tfrac{1+\sqrt{d}}{2} \;\middle|\; a, b \in \mathbb{Z}\right\}.$$

For example,

$$\mathcal{O}_{\mathbb{Q}[\sqrt{-5}]} = \mathbb{Z}[\sqrt{-5}]$$
$$\mathcal{O}_{\mathbb{Q}[\sqrt{5}]} = \mathbb{Z}[(1+\sqrt{5})/2].$$

Note that $(1+\sqrt{5})/2$ satisfies $X^2 - X - 1 = 0$ and so it is an algebraic integer in $\mathbb{Q}[\sqrt{5}]$.
Let ζ_d be a primitive dth root of 1, for example, $\zeta_d = \exp(2\pi i/d)$, and let $K = \mathbb{Q}[\zeta_d]$. Then we shall see (6.2) that

$$\mathcal{O}_K = \mathbb{Z}[\zeta_d] = \left\{\sum m_i \zeta_d^i \mid m_i \in \mathbb{Z}\right\}.$$

as one would hope.

Preliminaries

Complexity and algorithmic processes. The reader is assumed to have an inherent understanding of algorithms, which are defined as a set of instructions for creating an output from a specified input data set, which encompasses a finite sequence of nonnegative integers. This section is based on the assumption that the reader has this understanding. While it is true that algorithms may be referred to informally as Turing machines, we have discovered that many of our results are enhanced by using a more realistic "machine model" in terms of running time, which is another idea that is intuitively evident yet unclear. This is in contrast to the use of a computer that is theoretically flawless. See and the literature that is mentioned there for more information on these subjects.

The length of a finite sequence of nonnegative integers n1, n2, . . . , nt is defined to be $\sum_{i=1}^{t} \log(n_i + 2)$ Consider the possibility that it is exactly proportional to the binary representation of the. This would be a rough approximation. Obtaining a very sharp upper limit on the algorithm's running time expressed as a function of the length of the input data is what we mean when we speak about studying the complexity of an algorithm in this research. This is what we mean when we say that we are analysing the complexity of an algorithm. In order to separate it from the difficulty of space, this is more correctly referred to as the complexity of time. In order to be considered a decent algorithm, its execution time should be polynomial-time is $(l + 2)^{O(1)}$, This is a representation of the length of the input denoted by l. The objective of complexity research is to identify the algorithm that is the least complicated for a certain problem. During the course of this investigation, we will not be concerned with the value of the O-constant; rather, we will consider the complexity analysis to be finished once a suitable solution has been found for a problem. When it comes to solving a problem, an algorithm is deemed to be outstanding if it is nearly as simple to solve a single instance of the problem as it is to formulate the algorithm itself. An algorithm that can draw bits at random by using a random number generator is referred to as a probabilistic algorithm. We will sometimes discuss probabilistic algorithms. For instance, a Turing computer that does not guarantee any particular conclusion is an illustration of this. The use of random number generators (RNGs) is expressly prohibited, unless the word "probabilistic" is utilised; in order to emphasise this point, we refer to these approaches as deterministic. In a probabilistic algorithm, each fixed value of the input data has a corresponding distribution. This implies that the output and the amount of time it takes to execute are not the only factors that are impacted by the distribution of the input. What is known as the expected running time of a probabilistic algorithm is the amount of time that is estimated to be required for the algorithm to process an input. One of the most significant aspects of analysing the difficulty of a probabilistic algorithm is determining a maximum upper limit for the expected amount of time it takes to execute the algorithm as a function of the length of the input sequence. We consult in order to get a few recommendations that are applicable and might be used for this purpose. A probabilistic algorithm is called good if its expected running time is $(l+2)^{O(1)}$ where l is the length of the input. For the time being, we are going to disregard the fact that algorithmic number theory is still concerned with parallel algorithms, which are still mostly unimportant. According to a number of the conclusions presented in the study, "there exists" an algorithm that has certain qualities. An algorithm of this kind may be shown to work in every circumstance, at least in principle. If it is not specifically specified differently, every single O-constant is a real number that can be effectively calculated.

Encoding data

As was said before, each and every algorithm accepts finite sequences of nonnegative integers as input and delivers them as output. Working with sequences of nonnegative integers that encompass mathematical ideas is beneficial; but, working with mathematical concepts is preferable in the mathematical discipline of thinking and writing about algorithms. This is because dealing with sequences requires more mental effort. It is more easier to state that an algebraic number field is utilised as the input of an algorithm rather than referring to the sequence of coefficients of a polynomial that defines a field. This is because the polynomial is the function that defines the field. An additional example would be the computation of the kernel of a certain endomorphism of a vector space. This method is not only more concise but also simpler to comprehend in comparison to the process of finding a matrix in which each column represents a basis for the kernel in reference to a particular basis of the vector space. A agreement on an encoding of number fields, vector spaces, and mappings between them in terms of finite sequences of nonnegative integers is required in order to allow such a compact means of expressing things. This is because the amount of space required to represent things is limited. Because of this, the remaining portion of this composition is included here. In situations when there is more than one apparent way to carry out the encoding, it is essential to take into consideration whether or not there is a technique that transitions between encodings that is of sufficient quality. In situations like this, we often fail to discern between the encodings, even if it is not necessarily necessary for them to be completely identical for reasons of practicality. We will make the observation that the issue of encoding mathematical objects does, despite the fact that we will not be doing a systematic investigation into the many basic challenges that are generated by the topic. We will not be doing much more than what is necessary in the succeeding sections of this project.

Elementary arithmetic

When referring to the ring of integers, the sign Z is used. When a sign bit is included into the equation, it becomes clear that all numbers may be represented by integers that are not negative. For arithmetic operations, the time complexity of classical algorithms is denoted by the notation O(l), where l represents the length of the input. The Euclidean technique for determining greatest common divisors and the typical algorithms for multiplication and division with remainder both have a running time that is O(l 2); this means that they both take the same amount of time to complete. With the help of more sophisticated methods this can be improved to $l^{1+o(1)}$ for $l \to \infty$ Although establishing whether or not an integer is prime is a problem that is quite similar to this one, dissecting a positive integer into prime numbers is not known to be advantageously algorithmic. However, there is a good probabilistic algorithm that may be used for this particular subject matter. On the other hand, there are no well-known algorithms that are capable of solving the tasks of finding squarefree integers and calculating the biggest square that can be split by a given positive integer. This is true even when the word "good" is used in a less formal sense.

An input parameter for some algorithms is a prime integer denoted by the letter p. For the sake of this scenario, it is assumed that the prime is encoded independently of any other value; for example, the number n represents the nth prime. Considering that we do not possess a robust deterministic method for identifying primes, it is logical to question what happens to the process when p is neither prime nor known to be prime. This is because primes are not guaranteed to be primes. Primality tests may be helpful in identifying algorithms that result in the conclusion that p is not prime. This may be the case, for instance, when the technique takes an excessive amount of time to finish the computations or when a known prime characteristic is contested. On account of the fact that they generate nontrivial p factors, some approaches could even be helpful as procedures for integer factoring. With regard to both types of algorithms, one can ponder the questions of what implications can be derived if the program seems to have successfully exited. In order to further establish that p is prime, is it possible to utilise this? What conclusions may we draw from the result, supposing that p is not a prime number? Schoof's method does not effectively answer the question of how many points an elliptic curve over a finite field may hold. This is a question that has been raised. All field operations may be performed on rational numbers in polynomial time, and it is simple to describe rational numbers as pairs of integers.

Positiveness is required for the number n. The encoding of the ring Z/nZ elements is assumed to be as nonnegative integers that are less than n. This is the assumption that is made. It is possible to carry out the ring operations in a time that is polynomial. An ideal I ⊂ Z/nZ may be encoded in a number of ways. One of these ways is by using its index d = [Z/nZ: I], which completely describes the ideal and can be any divisor of n. Using a single generator or a limited sequence of components that forms I is another method. Both of these methods are viable options. When describing an element of I, one method to do so is to say that it is a Z/nZ element that is divisible by d. It is also possible to describe it as an explicit Z/nZ linear combination of the generators of I that are given, or as an explicit multiple of a single generator that is supplied. When utilising the extended Euclidean approach, it is simple to demonstrate that any of these ideal and element encodings may be passed on to any other in polynomial time, and that it is also possible to check the equality and inclusion of given ideals. As an example, if one is provided with a nonzero element of Z/nZ, it is possible to determine in polynomial time whether or not it is a unit, identify its inverse, and, if it is not a unit, locate a nontrivial divisor of n. In the event when n= p is a prime number, we discover that all field operations in Fp = Z/pZ may be completed in a time that is polynomial.

Linear algebra.

In light of the fact that an encoding of the elements of a field F has been decided upon, let us imagine that F is the field of rational numbers Q or the field Fp for some prime integer p (for more information, see to section 2.3). In order to describe a vector space over F that has finite dimensions, all that is required is the availability of an integer n that is not negative. The value of n must be specified in unary notation, which means that it must be represented as a sequence of n ones ranging from 1 to n. This is done to guarantee that the encoding is at least n bits long. This is as a result of the fact that the amount of time that is needed by almost any method that makes use of n-dimensional vector spaces begins with n. When it comes to this vector space, every single member is represented by a sequence of n elements that belong to the field F. When representing homomorphisms across vector spaces, matrix notation is the notation that is used. Encapsulating a subspace of a vector space may be accomplished by the use of a number of different element sequences. These include an element sequence that spans a subspace, an element sequence that forms its basis, and the kernel of a homomorphism from one vector space to another. Polynomial-time algorithms are the typical linear algebraic algorithms that are based on Gaussian elimination. These algorithms are applicable to all fields F that we shall investigate. Converting between different subspace representations, identifying whether a subspace is invertible and, if it is, producing its inverse, constructing quotient spaces, direct sums, and tensor products, and generating intersections and sums of subspaces are some of the things that these algorithms are capable of doing. Proofs are straightforward; the most important thing is to establish upper bounds on the size of the integers that are used in the computations. For example, in the scenario when F equals Q, this is the case. When division by a nonzero element fails, a nontrivial divisor of p is identified. Alternatively, the approach works as if F were a field if any of these techniques are used to F = Z/pZ without the knowledge that p is prime. In the second case, it is possible to circumvent the requirement that p is prime by just interpreting the output of the algorithm in terms of free Z/pZ-modules (you may refer to [14] for more information).

Finitely generated abelian groups.

Specifying a finitely generated abelian group is done by giving a sequence of nonnegative integers d1, d2, . . . , dt; the group is then $\bigoplus_{i=1}^{t} \mathbf{Z}/d_i\mathbf{Z},$ which enables us to represent the elements of the group by means of sequences of t integers. In our applications the group is usually either finite $(\text{all } d_i > 0)$ or free abelian $(\text{all } d_i = 0).$ To make the d_i unique one may require that $d_i \text{ divides } d_{i+1} \text{ for } 1 \leq i < t;$ The amount of time required to do this assignment is polynomial. Due to the fact that it could be difficult to execute algorithmically, it is not required to need the di to be prime powers. With this description of finitely created abelian groups and a reference to 2.4, it is up to the reader to devise their own ways of encoding maps and subgroups. They may take inspiration from this description. It is also possible for him to make advantage of the Hermite and Smith reduction of integer matrices (refer to [29]) in order to develop efficient techniques for the counterparts of the challenges that were addressed in 2.4. One of the most important challenges is to keep the intermediate numbers to a minimum.

Number fields

By a number field or an algebraic number field we mean in this paper a field extension K of finite degree of the field Q of rational numbers. For the basic theory of algebraic number fields.

An algebraic number field K is encoded as its underlying Q-vector space together with the multiplication map $K \otimes_{\mathbf{Q}} K \to K,$ as in 2.7; in other words, giving K amounts to giving a positive integer n and a system of n ^{3} rational numbers aijk that describe the multiplication in K on a vector space basis of K over Q (cf. 2.8 above). As in, one shows that the field operations in a number field can be performed in polynomial time. Using standard arguments from field theory one shows that there are good algorithms for determining the irreducible polynomial of a given element of K over a given subfield and for finding a primitive element of K, i.e., an element $\alpha \in K$ for which $K = \mathbf{Q}(\alpha).$ It follows that giving a number field is equivalent to giving an irreducible polynomial - $f \in \mathbf{Q}[X]$ and letting the field be $\mathbf{Q}[X]/f\mathbf{Q}[X].$

When dealing with one-variable polynomials that have coefficients in an algebraic number field, it is feasible to factor irreducible polynomials into polynomial time. This may be accomplished by the use of basis reduction; for references, have a look at [42, 35, 39, 40]. The following two results are worthy of notice. One of the most effective methods for establishing whether or not a certain set of n three rational numbers comprises a number field is shown in the argument that is offered in section 2.8. The first point that has to be stated is this one. Second, it is feasible to discover if two number fields K = Q(α) and K′ are isomorphic in polynomial time, and if they are, it is also possible to find all any isomorphisms that exist between them. For this purpose, the irreducible polynomial f of over Q is reduced to irreducible components in the ring K′ [X]. This allows for the achievement of the desired result. Following that, it is shown that these linear components have a bijective correlation with the field homomorphisms K K′. If and only if the degree of the two fields is identical over Q, then a field homomorphism is regarded to be an isomorphism by the mathematical community.

Considering that K is equal to K′, it can be deduced from the preceding section that it is feasible to discover all automorphisms of K. Furthermore, by combining these automorphisms, it is possible to create a complete multiplication table of field automorphisms of K for the group Aut K in a time that is polynomial. The proof of 3.5 demonstrates that it is feasible to locate all maximal acceptable subfields of a number field of degree n in polynomial time. This might be accomplished by using polynomial time. It is an excessive amount of effort to search for all of the subfields, given that the number of subfields does not have a polynomially bounded number. One of the things that I do not know for certain is whether or not it is possible to compute the number of unique minimal subfields from Q in polynomial time (1). It is possible for linear algebra to determine the intersections and composites of any two subfields that are defined. We would want for our predicted running times to be consistent in K, and we would like to stress that our algorithms handle the number field K as a variable rather than a fixed variable wherever possible.

Conclusion

A collaborative effort between Algebraic Number Theory and Class Field Theory has laid the groundwork for modern number theory. This study gives light on the structure of number fields and their extensions, which is the basis of contemporary number theory. By exploring rings of algebraic integers, ideal class groups, and factorisation characteristics, algebraic number theory offers a basis for understanding the arithmetic behaviour of numbers other than rational integers. This foundation is provided by the study of algebraic number theory. The Class discipline Theory (CFT) is a remarkable achievement in this discipline since it offers a taxonomy of abelian extensions of number fields via the use of reciprocity laws and ideal class groups. Local and global class field theory is a theory that builds links between number fields and their local counterparts. Additionally, the Hilbert Class Field and Artin Reciprocity Law are powerful tools that may be used for the study of field extensions and prime ideal behaviour. These notions have repercussions that extend beyond the realm of mathematics. Ellipstic curve algorithms and pairing-based protocols are two examples of modern encryption methods that make use of the laws of quantum mechanics to ensure the confidentiality of communications sent and received. Not only is class field theory essential in algebraic geometry, but it also plays a significant role in the Langlands program, where it influences the direction that research is taking. Despite the fact that the discipline of number theory is constantly evolving, the ideas of algebraic number theory and class field theory continue to be very important. It is possible to see their influence in the fields of computational mathematics and cryptography, where they provide answers to significant theoretical problems that are applicable in the actual world. As a result of research that expands upon these principles, our understanding of number fields and the profound connections that they have with other areas of mathematics will undoubtedly be extended in the years to come.

Reference

1. L. M. Adleman and M. A. Huang, Recognizing primes in random polynomialtime, Research report, Dept. of Computer Science, Univ. of Southern California, 1988; Lecture Notes in Math., Springer, Heidelberg (to appear). Extended abstract: Proc. 19th Ann. ACM Sympos. on Theory of Computing (STOC), ACM, New York 1987, pp. 462–469.
2. L. M. Adleman and H. W. Lenstra, Jr., Finding irreducible polynomials over finite fields, Proc. 18th Ann. ACM Sympos. on Theory of Computing (STOC), ACM, New York (1986, pp. 350–355.
3. L. M. Adleman, C. Pomerance, and R. S. Rumely, On distinguishing prime numbers from composite numbers, Ann. of Math. (2) 117 (1983), 173–206.
4. Archimedes, The sand-reckoner, in: Opera quae quidem extant, J. Hervagius, Basel, 1544. (Greek and Latin)
5. A. O. L. Atkin and F. Morain, Elliptic curves and primality proving (to appear).
6. E. Bach, Explicit bounds for primality testing and related problems, Math. Comp. 55 (1990), 355–380.
7. E. Bach and J. O. Shallit, Factor refinement, J. Algorithms (to appear).
8. W. E. H. Berwick, Integral bases, Cambridge Univ. Press, Cambridge, 1927.
9. Z. I. Boreviˇc and I. R. Safareviˇc, ˇ Teorija ˇcisel, Izdat. “Nauka”, Moscow, 1964; English transl.: Number theory, Academic Press, New York, 1966.
10. W. Bosma and M. P. M. van der Hulst, Primality proving with cyclotomy, Academisch proefschrift, Universiteit van Amsterdam, 1990.
11. E. Brieskorn and H. Kn¨orrer, Ebene algebraische Kurven, Birkh¨auser, Basel, 198
12. J. Buchmann, Complexity of algorithms in algebraic number theory, Proceedings of the first conference of the Canadian Number Theory Association (R. A. Mollin, ed.), De Gruyter, Berlin, 1990, pp. 37–53.
13. , A subexponential algorithm for the determination of class groups and regulators of algebraic number fields, S´eminaire de Th´eorie des Nombres, Paris 1988–1989 (C. Goldstein, ed.), Birkh¨auser, Boston, 1990, pp. 27–41.
14. J. Buchmann and H. W. Lenstra, Jr., Manuscript in preparation.
15. J. Buchmann and V. Shoup, Constructing nonresidues in finite fields and the extended Riemann hypothesis, in preparation. Extended abstract: Proc. 23rd Ann. ACM Sympos. on Theory of Computing (STOC), ACM, New York 1991, pp. 72–79.
16. J. Buchmann and H. C. Williams, On the computation of the class number of an algebraic number field, Math. Comp. 53 (1989), 679–688.
17. J. P. Buhler, H. W. Lenstra, Jr., and C. Pomerance, Factoring integers with the number field sieve, in preparation.
18. P. J. Cameron, Finite permutation groups and finite simple groups, Bull. London Math. Soc. 13 (1981), 1–22.