ON EVEN K-GROUPS OF RINGS OF INTEGERS OF REAL ABELIAN FIELDS

MENG FAI LIM AND CHAO QIN

Abstract. We present approaches for calculating the precise orders of the algebraic K-groups $K_{4n-2}(\mathcal{O}_K)$ for a totally real abelian K. Along the way, we also establish a formula connecting the order of $K_{4n-2}(\mathcal{O}_E)$ of a totally real p-elementary field E to its intermediate cyclic p-degree fields. Additionally, we have compiled a list of values pertaining to these K-groups. Given the substantial space that these data would occupy, the list has not been incorporated into the paper. We encourage interested readers to consult the supplement document or the arXiv version of the paper, where these compiled list of values can be found.

1. Introduction

For a ring R, we let $K_i(R)$ denote the algebraic K-groups of R in the sense of Quillen [15]. Thanks to the pioneering work of Quillen [15], Garland [7], and Borel [2], we now know that the even K-groups of the ring of integers of a number field are finite. However, these foundational results provide limited insight into the precise orders of these groups. It was the conjecture proposed by Birch and Tate that initially provided an approach to understanding the order of $K_2(\mathcal{O}_F)$ by evaluating the Dedekind zeta function of the field F at $s = -1$. This conjecture was later generalized by Lichtenbaum [11, 12] to include the higher even K-groups. Coates [6] then provided a crucial link by suggesting that this conjecture could be attacked using the main conjecture of Iwasawa theory [9], an insight that has been pivotal in establishing the conjecture for totally real abelian fields and forms the backbone of our computational approaches. Since then, significant progress has been made towards establishing this conjecture, particularly notable in the case of a totally real abelian field.

Building on these conjectures, Browkin and his collaborators [3, 4, 5] have gone a long way into computing the precise order of $K_2(\mathcal{O}_F)$ for quadratic fields and specific classes of cubic fields (also see the work of Zhou [26]). The primary objective of this paper is to continue this line of study by exploring the computation of the order of higher even K-groups, specifically $K_{4k-2}(\mathcal{O}_F)$. More precisely, we discuss three different approaches for calculating the order of higher even K-groups, with each being particularly suited to certain types of number fields. The primary difference between these methods lies in the technique used to compute the special values of the Dedekind zeta function. Although most software tools offer built-in functions for numerical approximations of these values, our objective is to compute them exactly as fractions.

Our first approach applies in principle to all totally real abelian fields F. Leveraging the Artin formalism for L-functions, we establish a connection between the value $\zeta_F(1-2k)$ and a suitable product involving generalized

1991 *Mathematics Subject Classification.* 11R42, 11R70, 19F27.

Key words and phrases. Even K-groups, real multi-quadratic fields, totally real p-elementary fields.

M. F. Lim's research is partially supported by the Fundamental Research Funds for the Central Universities No. CCNU25JCPT031.

Chao Qin's research is supported by the National Natural Science Foundation of China under Grant No. 12001546 and Heilongjiang Province under Grant No. 3236330122.

Bernoulli numbers. Subsequently, we proceed to compute each of these generalized Bernoulli numbers individually. For this step, the associated Dirichlet character must be manually inputted from the LMFDB to ensure a precise fractional computation. We demonstrate this entire procedure in the body of the paper through a concrete example (see Appendix B). The second method is specifically for the real quadratic fields and is based on a formula of Siegel-Zagier. For this, we provide a simplified approach using Siegel-Zagier's formula and also offer a faster method to calculate $w_j(F)$, improving efficiency without compromising on obtaining exact values. This will be elaborated upon in Section 3.

The third approach is specifically tailored for a p-elementary totally real abelian field E, in which the Galois group $\mathrm{Gal}(E/\mathbb{Q})$ is isomorphic to $(\mathbb{Z}/p\mathbb{Z})^{\oplus n}$ for some prime p and positive integer n. The key to this approach is the following theorem on which it relies.

Theorem 1.1 (Theorem 4.1)**.** *Let p be a prime and n an integer ≥ 2. Suppose that E is a totally real abelian extension of $\mathbb{Q}$ with Galois group $G = \mathrm{Gal}(E/\mathbb{Q}) \cong (\mathbb{Z}/p\mathbb{Z})^{\oplus n}$. Let $K_1, K_2, \ldots, K_{\frac{p^n-1}{p-1}}$ denote all the p-degree extensions of $\mathbb{Q}$ contained in E. Then we have the following equality*

$$|K_{4k-2}(\mathcal{O}_E)| = \frac{1}{|K_{4k-2}(\mathbb{Z})|^{\frac{p^n-p}{p-1}}} \prod_{j=1}^{\frac{p^n-1}{p-1}} |K_{4k-2}(\mathcal{O}_{K_j})|.$$

The proof of the theorem will be given in Section 4. From a computational point of view, this result is rather advantageous to have. For instance, if we want to compute the size of K_{4n-2}-groups of the ring of integers of $\mathbb{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5})$, it suffices to compute those for the intermediate quadratic fields $\mathbb{Q}(\sqrt{2})$, $\mathbb{Q}(\sqrt{3})$, $\mathbb{Q}(\sqrt{5})$, $\mathbb{Q}(\sqrt{6})$, $\mathbb{Q}(\sqrt{10})$, $\mathbb{Q}(\sqrt{15}), \mathbb{Q}(\sqrt{30})$, as well as for $\mathbb{Q}$. After obtaining these values, we can simply plug them into the formula provided by the aforementioned theorem.

1.1. **Potential direction for future research.**

- Our results are promising, showing feasibility for selected classes of fields of larger degrees, as evidenced by our comprehensive compilation of K-group values, which extends over nearly 90 pages (omitted from the main text but available in the arXiv version). We hope to continue this line of study and extend our calculations to more extensive classes of number fields in a future work.
- It is natural to ask what we can do with the comprehensive collection of K-groups values obtained. One direction we hope to study in the near future is to examine the Galois module structure of the K-groups as module over the Galois group of the number field relative to $\mathbb{Q}$ building on our present numerical data. To the best knowledge of the authors, while the existing literature already contains a substantial amount of theoretical research (with on-going advancement made even now) on this topic, when it comes to numerical computation or specific methods determination in this regard, we have yet come across any relevant studies.

Acknowledgement. The collaboration began at a workshop at Harbin Institute of Technology in 2022. The authors extend their gratitude to Jun Wang and Yichao Zhang for organizing the workshop and for the insightful discussions that took place during the paper's preparation. Part of this research was conducted during the first author's multiple visits to Harbin Engineering University and the second author's visits to Central China Normal University. We would like to express our appreciation to both universities for their warm hospitality. We are

particularly grateful to Daniel Delbourgo for his valuable insights, keen interest, and ongoing encouragement, all of which have been instrumental in advancing this work.

2. K-groups and Dedekind zeta function

We begin with a brief and quick review of the definition of the higher K-groups. Let R be a ring with identity. For each integer $m \geq 1$, denote by $\mathrm{GL}_m(R)$ the group of invertible $m \times m$ matrices with entries in R. We then set $\mathrm{GL}(R) = \varinjlim_m \mathrm{GL}_m(R)$, where the transition map $\mathrm{GL}_m(R) \longrightarrow \mathrm{GL}_{m+1}(R)$ is given by

$$A \mapsto \begin{pmatrix} A & 0 \\ 0 & 1 \end{pmatrix}.$$

Let $\mathrm{BGL}(R)$ be the classifying space of the group $\mathrm{GL}(R)$, i.e., $\mathrm{BGL}(R)$ is an Eilenberg-MacLane space of type $(\mathrm{GL}(R), 1)$ in the sense of [18, Section 8.1]. Up to homotopy equivalence, this space is characterized by the property that it is path-connected with homotopy groups

$$\pi_n\big(\mathrm{BGL}(R)\big) \cong \begin{cases} \mathrm{GL}(R), & \text{for } n = 1, \\ 0, & \text{for } n \geq 2. \end{cases}$$

From the space $\mathrm{BGL}(R)$, one obtains a new space, denoted by $\mathrm{BGL}(R)^+$, via the $+$-construction of Quillen (see [14, 15]). The higher K-groups $K_n(R)$ are then defined by

$$K_n(R) := \pi_n\big(\mathrm{BGL}(R)^+\big).$$

It's well-known that Quillen's construction recovers the classical K_1-groups of Bass and K_2-groups of Milnor (for instance, see [23, Chapter IV]).

In this paper, we are interested in the K-groups of the ring $\mathcal{O}_F$, where $\mathcal{O}_F$ is the ring of integers of a number field F. As a start, we recall the following fundamental results of Quillen and Borel.

Theorem 2.1. *The groups $K_n(\mathcal{O}_F)$ are finitely generated for all $n \geq 1$. Furthermore, one has*

$$\mathrm{rank}_{\mathbb{Z}}\left(K_n(\mathcal{O}_F)\right) = \begin{cases} r_1(F) + r_2(F), & \text{if } n \equiv 1 \pmod 4, \\ r_2(F), & \text{if } n \equiv 3 \pmod 4, \\ 0, & \text{if } n \text{ is even.} \end{cases}$$

Here $r_1(F)$ (resp., $r_2(F)$) is the number of real embeddings (resp., number of pairs of complex embeddings) of the number field F.

Proof. Quillen [15] was the first to establish that these K-groups are finitely generated. Subsequently, calculations of Borel [2] confirmed the ranks of the K-groups as stated in the theorem. We should also mention that prior to the works of Qullen and Borel, the finiteness of $K_2(\mathcal{O}_F)$ has been proven by Garland [7]. □

Unfortunately, the results of Quillen and Borel do not provide a means to determine the exact order of the even K-groups $K_{2i}(\mathcal{O}_F)$. It was only due to the remarkable insight of Birch, Tate and Lichtenbaum [11, 12] that one can hope to understand these orders via special values of Dedekind zeta function, a concept we will now outline briefly. Let $\zeta_F(s)$ be the Dedekind zeta function of F. This function, $\zeta_F(s)$, admits an analytic continuation to the whole complex plane, with the exception of a simple pole at $s = 1$. Consequently, it makes

sense to speak of $\zeta_F(1-2k)$ for a positive integer k. Thanks to the collective gallant efforts of numerous mathematicians, we have the following.

Theorem 2.2. *Let F be a totally real abelian number field of degree r $(= r_1(F))$. Then for every integer $k \geq 1$, we have*

$$\zeta_F(1-2k) = (-1)^{kr} 2^r \frac{|K_{4k-2}(\mathcal{O}_F)|}{|K_{4k-1}(\mathcal{O}_F)|}.$$

Proof. Lichtenbaum [11] first formulated this conjecture up to a power of 2. Subsequently, the work of Coates [6] suggested that one might possibly attack this conjecture via the main conjecture of Iwasawa [9]. Building on this insight, Bayer and Neukirch [1] showed that the main conjecture of Iwasawa implies a cohomological version of Lichtenbaum's conjecture (for a detailed exposition of this cohomological version, readers are referred to [1]). Notably, this cohomological formulation is equivalent to the K-theoretical version, a connection established by the Quillen-Lichtenbaum conjecture. This conjecture is now a theorem, being a consequence of the groundbreaking work of Rost-Voevodsky ([20]; see also Rognes-Weibel [16]). Prior to these developments, the main conjecture of Iwasawa has already been proven by by Mazur-Wiles [13] and Wiles [24]. □

The value $|K_{4k-1}(\mathcal{O}_F)|$ can be described rather easily. Let μ_∞ be the group of all the roots of unity of $\bar{F}$, where $\bar{F}$ is the algebraic closure of F. For an integer $j \geq 1$, we write $\mu_\infty^{\otimes j}$ for the j-fold tensor product of μ_∞ with $\mathrm{Gal}(\bar{F}/F)$ acting diagonally. Set $w_j(F)$ to be the order of $(\mu_\infty^{\otimes j})^{\mathrm{Gal}(\bar{F}/F)}$. The following gives a relation of $|K_{4k-1}(\mathcal{O}_F)|$ in terms of these values.

Theorem 2.3. *Let F be a totally real abelian number field of degree r. Then for every integer $k \geq 1$, we have*

$$\left|K_{4k-1}(\mathcal{O}_F)\right| = \begin{cases} 2^r w_{2k}(F), & \text{if } k \text{ is odd,} \\ w_{2k}(F), & \text{if } k \text{ is even.} \end{cases}$$

Proof. See [23, Chap. VI, Theorem 9.5]. □

Combining the above theorems, we have the following observation.

Corollary 2.4. *Let F be a totally real abelian number field of degree r. Then for every integer $k \geq 1$, we have*

$$\left|K_{4k-2}(\mathcal{O}_F)\right| = \begin{cases} (-1)^r w_{2k}(F)\zeta_F(1-2k), & \text{if } k \text{ is odd,} \\ \dfrac{1}{2^r} w_{2k}(F)\zeta_F(1-2k), & \text{if } k \text{ is even.} \end{cases}$$

In principle, the values of $w_{2k}(F)$ can be determined rather easily (for instances, see [23, Chap. VI, Propositions 2.2 and 2.3]). Consequently, the main challenge lies in calculating the values of $\zeta_F(1-2k)$. For number fields of small degree, these can be computed via bulit-in functions in mathematical software programs. However, even in the case of a real quadratic field, the computed values are not exact when the discriminant of the real quadratic number field F becomes large. In this context, computing $\zeta_F(-19)$ often leads to a loss of significant digits. In the subsequent discussion of the paper, we will outline strategies to circumvent this issue. To begin with, we introduce the following approach that utilizes generalized Bernoulli numbers.

Let F be an abelian totally real number field with Galois group $G = \mathrm{Gal}(F/\mathbb{Q})$. By Artin formalism, we have

$$\zeta_F(1-2k) = \zeta(1-2k) \prod_{\chi \neq \chi_0} L(\chi, 1-2k)$$

where χ runs through all the nontrivial characters of G. It's well-known that

$$\zeta(1-2k) = -\frac{B_{2k}}{2k} \quad \text{and} \quad L(\chi, 1-2k) = -\frac{B_{2k,\chi}}{2k}$$

(for instance, see [21, Theorem 4.2]). The above therefore gives a way to compute the $\zeta_F(1-2k)$ via generalized Bernoulli numbers. To see a specific example, we refer readers to the first listing in Appendix B.

3. Quadratic field

In this section, we shall describe a method (due to Siegel-Zagier) of computing the L-values for a totally real quadratic field. To prepare for this, we need to introduce some further notations. For a given integer $j \geq 0$ and an ideal $\mathfrak{a}$ of $\mathcal{O}_F$, we define

$$\sigma_j(\mathfrak{a}) = \sum_{\mathfrak{b} | \mathfrak{a}} |\mathcal{O}/\mathfrak{b}|^j,$$

where the sum is taken over all nonzero ideals $\mathfrak{b}$ of $\mathcal{O}_F$ that divide $\mathfrak{a}$. In the special case where $\mathcal{O}_F$ coincides with $\mathbb{Z}$, we shall simplify the notation to $\sigma(m) = \sigma(m\mathbb{Z})$. Note that in this context, we have

$$\sigma_j(m) = \sum_{d|m} d^j,$$

where d runs through all the positive divisors of m. Furthermore, for integers $j, k \geq 1$, we set

$$s_j^F(2k) = \sum_{\substack{\nu \in \mathfrak{d}^{-1} \\ \nu \gg 0 \\ \mathrm{tr}(\nu) = j}} \sigma_{2k-1}\big((\nu)\mathfrak{d}\big).$$

Here $\mathfrak{d} = \mathfrak{d}_F$ is the different of F and the sum is taken over all totally positive elements in $\mathfrak{d}$ with trace j. With these notation in hand, we can now state the following formula of Siegel.

Theorem 3.1 (Siegel)**.** *Let F be a totally real number field. Then for every integer $k \geq 1$, we have*

$$\zeta_F(1-2k) = 2^{|F:\mathbb{Q}|} \sum_{j=1}^{r} b_j(2k|F:\mathbb{Q}|) s_j^F(2k),$$

where the numbers $b_j(2k|F:\mathbb{Q}|)$ are rational and depend only on $2k|F:\mathbb{Q}|$, and the integer r is given by

$$r = \begin{cases} \left[\dfrac{k|F:\mathbb{Q}|}{6}\right], & \text{if } k|F:\mathbb{Q}| \equiv 1 \ (mod\ 6), \\ \left[\dfrac{k|F:\mathbb{Q}|}{6}\right] + 1, & \text{otherwise.} \end{cases}$$

Remark 3.2. Some of the values of Siegel coefficients $b_j(m)$ for $4 \leq m \leq 40$ can be found in [25, Table 1]. For the convenience of the readers, we provide an approach to computing these terms following [17].

For $k = 4, 6, ...$, we set

$$G_k(z) = 1 - \frac{2k}{B_k}\sum_{n=1}^{\infty}\sigma_{k-1}(n)q^n$$

and

$$\Delta = q\prod_{n=1}^{\infty}(1-q^n)^{24}.$$

We then define

$$T_h = G_{12r-h+2}\Delta^{-r},$$

where

$$r = \begin{cases} [\frac{h}{12}]+1, & h \not\equiv 2 \bmod 12; \\ [\frac{h}{12}], & h \equiv 2 \bmod 12. \end{cases}$$

By [17, P252, (11)], we have

$$T_h = q^{-r} + c_{h,r-1}q^{-(r-1)} + \cdots + c_{h,1}q^{-1} + c_{h,0} + \cdots$$

for some $c_{h,i} \in \mathbb{Q}$. Furthermore, [17, P253, Theorem 2] tells us that $c_{h,0} \neq 0$. Hence it makes sense to write

$$b_j(h) := -c_{h,j}/c_{h,0}$$

for $j = 1, 2, ..., r$. These are the Siegel coefficients (see [17, P254-255] or [25, P61, (26)]) that appears in Siegel's formula.

For a real quadratic number field K, Zagier has expressed Siegel's formulas in terms of certain elementary functions which we now describe. For integers $j, m \geq 1$, define

$$e_j(m) = \sum_{\substack{b^2+4ac=m \\ a,c>0}} a^j.$$

We also denote by $\chi := \chi_K$ the nontrivial character of $\mathrm{Gal}(K/\mathbb{Q})$, and extend it to a function of $\mathbb{Z}$ in the usual way. Zagier's formula is then as follow (cf. [25, (14),(16)]).

Theorem 3.3 (Zagier)**.** *Let K be a real quadratic field with discriminant D. Then for every integer $k \geq 1$, we have*

$$\zeta_K(1-2k) = 4\sum_{j=1}^{[k/3]+1} b_j(4k)\sum_{m|j}\chi(m)m^{2k-1}e_{2k-1}\big((j/m)^2D\big).$$

For a list of the formula for some values of k, we refer readers to [25, Table 2]. A notable advantage of this formula lies in its ability to yield exact values of $\zeta_K(1-2k)$ for real quadratic number fields with large discriminant. Moreover, by circumventing the need for L-value calculations in software like Pari or Magma, this formula potentially eliminates the dependency on GRH.

We shall utilize the aforementioned formula to calculate the special values of the Dedekind zeta function of a real quadratic field. Furthermore, we introduce a faster method to compute $w_j(F)$, optimizing the process while maintaining the accuracy required to determine the sizes of the K groups. This approach significantly improves computational efficiency in the quadratic case (see Listing 2 in Appendix B).

4. p-ELEMENTARY EXTENSIONS

We are now in a position to prove the following theorem as mentioned in our introductory section.

Theorem 4.1. *Let p be a fixed prime. Let E be a totally real Galois extension of $\mathbb{Q}$ with Galois group $G = \mathrm{Gal}(E/\mathbb{Q}) \cong (\mathbb{Z}/p\mathbb{Z})^{\oplus n}$ with $n \geq 2$. Denote by $K_1, K_2, ..., K_{\frac{p^n-1}{p-1}}$ all the p-degree extensions of $\mathbb{Q}$ contained in E. Then we have*

$$|K_{4k-2}(\mathcal{O}_E)| = \frac{1}{|K_{4k-2}(\mathbb{Z})|^{\frac{p^n-p}{p-1}}} \prod_{j=1}^{\frac{p^n-1}{p-1}} |K_{4k-2}(\mathcal{O}_{K_j})|$$

for every positive integer k.

Proof. For each $j = 1, ..., \frac{p^n-1}{p-1}$, we let $\chi_{j,r}$ be all the nontrivial character of $\mathrm{Gal}(K_j/\mathbb{Q})$, where $r = 1, ..., p-1$. When viewed as characters of $\mathrm{Gal}(E/\mathbb{Q})$, they are exactly all nontrivial characters of $\mathrm{Gal}(E/\mathbb{Q})$. By Artin formalism of L-functions, we have

$$\zeta_E(s) = \zeta(s) \prod_{j=1}^{\frac{p^n-1}{p-1}} \prod_{r=1}^{p-1} L(\chi_{j,r}, s).$$

On the other hand, we also have

$$\zeta_{K_j}(s) = \zeta(s) \prod_{r=1}^{p-1} L(\chi_{j,r}, s)$$

for each j. Hence we have

$$\zeta(s)^{\frac{p^n-p}{p-1}} \zeta_E(s) = \prod_{j=1}^{\frac{p^n-1}{p-1}} \zeta_{K_j}(s).$$

In view of the above equality and Corollary 2.4, the proposition is reduced to proving the equality

$$w_{2k}(\mathbb{Q})^{\frac{p^n-p}{p-1}} w_{2k}(E) = \prod_{j=1}^{\frac{p^n-1}{p-1}} w_{2k}(K_j). \tag{1}$$

If ℓ is a prime, we write $w_j^{(\ell)}(F)$ for the order of $(\mu_{\ell^\infty}^{\otimes j})^{\mathrm{Gal}(\bar{F}/F)}$, where μ_{ℓ^∞} is the group of all the ℓ-power roots of unity of $\bar{F}$. Plainly, one has $w_j(F) = \prod_\ell w_j^{(\ell)}(F)$. It therefore remains to show that

$$w_{2k}^{(\ell)}(\mathbb{Q})^{\frac{p^n-p}{p-1}} w_{2k}^{(\ell)}(E) = \prod_{j=1}^{\frac{p^n-1}{p-1}} w_{2k}^{(\ell)}(K_j) \tag{2}$$

for every prime ℓ. We first consider the case when the prime ℓ is odd. Since $\mathrm{Gal}(E/\mathbb{Q})$ is not cyclic, we have either $E \cap \mathbb{Q}(\mu_\ell) = \mathbb{Q}$ or $E \cap \mathbb{Q}(\mu_\ell) = K_i$ for some unique i.

Suppose that $E \cap \mathbb{Q}(\mu_\ell) = \mathbb{Q}$. Then we have $|L(\mu_\ell) : L| = \ell - 1$ for $L = E, K_j, \mathbb{Q}$. If $2k$ is not divisible by $\ell - 1$, it follows from [23, Chap. VI, Proposition 2.2(c)] that $w_{2k}^{(\ell)}(L) = 1$ for $L = E, K_j, \mathbb{Q}$, and so equality (2) is immediate in this case. If $2k$ is divisible by $\ell - 1$, then [23, Chap. VI, Proposition 2.2(c)] tells us that $w_{2k}^{(\ell)}(L) = \ell^{1+b}$ for $L = E, K_j, \mathbb{Q}$, where b is the highest power of ℓ dividing $2k$. This again verifies the equality (2).

Now, without loss of generality, suppose that $E \cap \mathbb{Q}(\mu_\ell) = K_1$. In other words, K_1 is contained in $\mathbb{Q}(\mu_\ell)$ with $\ell \equiv 1 \pmod{2p}$. If $2k$ is not divisible by $(\ell-1)/p$, it follows from [23, Chap. VI, Proposition 2.2(c)] that $w_{2k}^{(\ell)}(L) = 1$ for $L = E, K_j, \mathbb{Q}$, thus verifying the equality (2). In the event that $2k$ is divisible by $\ell - 1$, a similar argument as in the previous paragraph yields the required equality (2). Therefore, it remains to consider the case where $2k$ is divisible by $(\ell-1)/p$ but not divisible by $\ell - 1$. In this case, one can directly verify that $w_{2k}^{(\ell)}(E) = w_{2k}^{(\ell)}(K_1) = \ell^{1+b}$ and $w_{2k}^{(\ell)}(\mathbb{Q}) = w_{2k}^{(\ell)}(K_j) = 1$ for $j \geq 2$. From this, we see that the equality (2) is satisfied.

We now come to the situation when $\ell = 2$. We first consider the case $\sqrt{2} \notin E$. Under this said assumption, the field $L(\sqrt{-1})$ does not contain any primitive 8th root of unity. Thus, it follows from [23, Chap. VI, Proposition 2.3(c)] that $w_{2k}^{(2)}(L) = 2^{2+d}$ for $L = E, K_j, \mathbb{Q}$, where d is the highest power of 2 dividing $2k$. Plainly, the equality (2) is satisfied in this case. Now suppose that $\sqrt{2} \in E$. In particular, we must then have $p = 2$. Upon relabeling, we may assume $K_1 = \mathbb{Q}(\sqrt{2})$. Note that the fields $K_1(\sqrt{-1})$ and $E(\sqrt{-1})$ will now contain a primitive 8th root of unity but the fields $\mathbb{Q}(\sqrt{-1})$ and $K_j(\sqrt{-1})$ (for $j \geq 2$) do not. Hence, from [23, Chap. VI, Proposition 2.2(c,d)] it follows that we have $w_{2k}^{(2)}(E) = w_{2k}^{(2)}(K_1) = 2^{3+d}$ and $w_{2k}^{(2)}(\mathbb{Q}) = w_{2k}^{(2)}(K_j) = 2^{2+d}$ for $j \geq 2$. Plugging these values into the equality (2), we see that the said equality holds.

The proof of the theorem is therefore complete. □

Note that the asserted equality in the preceding proposition is not true if one remove the "totally real" hypothesis. Indeed, for an imaginary biquadratic field, Guo and Qin has shown that there might be an extra factor of a power of 2 (see [8, Theorem 3.5, Example 3.10]).

From a computational point of view, Theorem 4.1 is rather advantageous to have, as it streamlines the process of computing the K-group for an elementary p-extension of $\mathbb{Q}$ by reducing it to the computation of the K-group for a cyclic p-degree extension of $\mathbb{Q}$. Regarding the latter task, and in light of Corollary 2.4, our focus shifts to determining the values of w_{2k} and the relevant special values. For this, we present the following useful proposition.

Proposition 4.2. *Let p be a prime and $k \geq 1$. Suppose that K is a totally real number field which is a cyclic extension of $\mathbb{Q}$ of degree p. Then the following statements are valid.*

(*i*) $w_{2k}(\mathbb{Q}(\sqrt{2})) = 2^{4+v_2(k)} \displaystyle\prod_{\substack{l \text{ odd prime} \\ l-1 \mid 2k}} l^{1+v_l(k)} = \prod_{\substack{l \text{ prime} \\ l-1 \mid 2k}} l^{1+v_l(8k)}$.

(*ii*) *If the prime p is odd and $K \subseteq \mathbb{Q}(\zeta_{p^2})$, then*

$$w_{2k}(K) = 2^{3+v_2(k)} p^{(2+v_p(k))\delta} \prod_{\substack{l \neq 2,p \\ l-1 \mid 2k}} l^{1+v_l(k)} = \prod_{\substack{l \text{ prime} \\ l-1 \mid 2k}} l^{1+v_l(4pk)},$$

where $\delta = \begin{cases} 1, & \text{if } p-1 \mid 2k; \\ 0, & \text{otherwise.} \end{cases}$

(*iii*) *If K is contained in $\mathbb{Q}(\zeta_q)$ for some odd prime q, then*

$$w_{2k}(K) = 2^{3+v_2(k)} q^{(1+v_q(k))\varepsilon} \prod_{\substack{l \neq 2,q \\ l-1 \mid 2k}} l^{1+v_l(k)},$$

where $\varepsilon = \begin{cases} 1, & \text{if } \frac{q-1}{p} \mid 2k; \\ 0, & \text{otherwise.} \end{cases}$

(*iv*) *For other* K*'s not covered in* $(i)-(iii)$*, one always has*

$$w_{2k}(K) = 2^{3+v_2(k)} \prod_{\substack{l \neq 2 \\ l-1 \mid 2k}} l^{1+v_l(k)} = \prod_{\substack{l \text{ prime} \\ l-1 \mid 2k}} l^{1+v_l(4k)}.$$

Proof. Indeed, for an odd prime p, it follows from [23, Chap. VI, Proposition 2.2] that

$$\mu_{p^\infty}^{\otimes i}(K) = \begin{cases} \mu_{p^{a(K)+v_p(i)}}^{\otimes i}, & \text{if } i \equiv 0 \bmod |K(\mu_p) : K|, \\ 1, & \text{if } i \not\equiv 0 \bmod |K(\mu_p) : K|, \end{cases}$$

where $a(K)$ is the largest integer such that $K(\mu_p)$ contains a primitive $p^{a(K)}$th root of unity. For $p = 2$ and even i, an application of [23, Chap. VI, Proposition 2.3(c)] tells us that

$$\mu_{2^\infty}^{\otimes i}(K) = \mu_{2^{c(K)+v_2(i)}}^{\otimes i}$$

where $c(K)$ is the largest integer such that $K(\sqrt{-1})$ contains a primitive $2^{c(K)}$th root of unity. (Note that our number field K is totally real and so is an exceptional one in the sense of the proposition loc. cit.) The conclusions of the proposition now follow from the above two observations and a case-by-case analysis. □

For the code designed for the computation of even K-groups within the context of a p-elementary extension, see Listing 3 in Appendix B.

5. Periodicity of p-rank

Let K be an real quadratic field. Recall that for integers $j, m \geq 1$, we have defined

$$e_j(m) = \sum_{\substack{b^2+4ac=m \\ a,c>0}} a^j.$$

Plainly, one has $e_1(D) \equiv e_3(D) \pmod 3$. On the other hand, we have

$$\left|K_2(\mathcal{O}_K)\right| = \begin{cases} \frac{4}{5}e_1(8), & K = \mathbb{Q}(\sqrt{2}), \\ 2e_1(5), & K = \mathbb{Q}(\sqrt{5}), \\ \frac{2}{5}e_1(D), & \text{otherwise.} \end{cases} \qquad \left|K_6(\mathcal{O}_K)\right| = \begin{cases} e_3(8), & K = \mathbb{Q}(\sqrt{2}), \\ \frac{1}{2}e_3(D), & \text{otherwise.} \end{cases}$$

Therefore, it follows that $\left|K_2(\mathcal{O}_K)\right|$ is divisible by 3 if and only if $\left|K_6(\mathcal{O}_K)\right|$ is divisible by 3. Indeed, we shall see that there is a general reasoning underlying this observation which will be elucidated in the subsequent theorem.

Theorem 5.1. *Let* F *be a number field. Then we have*

$$\operatorname{rank}_{\mathbb{Z}/p\mathbb{Z}}\left(K_{2i}(\mathcal{O}_F)\right) = \operatorname{rank}_{\mathbb{Z}/p\mathbb{Z}}\left(K_{2i'}(\mathcal{O}_F)\right),$$

whenever $i \equiv i' \ (mod\ |F(\mu_p) : F|)$.

Proof. Although this fact might be well-known among experts but for the convenience of the readers, we shall provide a brief outline of the proof here. By the work of Rost and Voevodsky [20], there is an identification

$$K_{2k}(\mathcal{O}_F)/p \cong H^2\big(\operatorname{Gal}(F_{S_p}/F)), \mu_p^{\otimes(k+1)}\big),$$

where F_{S_p} is the maximal algebraic extension of F unramified outside the set of primes of F above p.

If $j \equiv 0 \bmod [F(\mu_p) : F]$, then the Galois group $\operatorname{Gal}(F_{S_p}/F)$ acts trivially on $\mu_p^{\otimes j}$. Therefore, it follows that

$$\begin{aligned} K_{2k}(\mathcal{O}_F)/p &\cong H^2\big(\operatorname{Gal}(F_{S_p}/F)), \mu_p^{\otimes(k+1)}\big) \\ &\cong H^2\big(\operatorname{Gal}(F_{S_p}/F)), \mu_p^{\otimes(k'+1)}\big) \otimes \mu_p^{\otimes(k-k')} \\ &\cong K_{2k'}(\mathcal{O}_F)/p \otimes \mu_p^{\otimes(k-k')}, \end{aligned}$$

whenever $k \equiv k' \bmod [F(\zeta_p) : F]$. Consequently, the groups $K_{2k}(\mathcal{O}_F)/p$ and $K_{2k'}(\mathcal{O}_F)/p$ have the same rank over $\mathbb{Z}/p\mathbb{Z}$. □

We return to the context of a real quadratic field K. Proposition 5.1 then tells us that the 3-rank

$$r_3\big(K_{4k-2}(\mathcal{O}_K)\big)$$

is a constant function in term of k. A consequence of this is the following.

Corollary 5.2. *Let K be a real quadratic field with discriminant D. Denote by χ the notrivial character of* $\operatorname{Gal}(K/\mathbb{Q})$. *Then the following statements are equivalent.*

(1) $e_1(D)$ *is divisible by* 3.
(2) $e_3(D)$ *is divisible by* 3.
(3) $e_5(4D) + \big(5\chi(2) + 6\big)e_5(D)$ *is divisible by* 9.
(4) $e_7(4D) + 19\chi(2)e_7(D)$ *is divisible by* 27.
(5) $e_9(4D) + \big(8\chi(2) + 3\big)e_9(D)$ *is divisible by* 9.
(6) $e_{11}(9D)$ *is divisible by* 3.
(7) $e_{13}(9D)$ *is divisible by* 3.
(8) $e_{15}(9D)$ *is divisible by* 3.

Remark 5.3. We remark that the list in the preceding corollary is far from exhaustive and goes on. One can of course apply the above discussion for other primes. For instances, for the case $p = 5$, then the following divisibility statements are equivalent.

(1) $e_1(D)$ is divisible by 25.
(2) $e_5(4D) + \big(7\chi(2) - 1\big)e_5(D)$ is divisible by 25.
(3) $e_9(4D) + \big(8\chi(2) + 3\big)e_9(D)$ is divisible by 25.
(4) $e_{13}(9D)$ is divisible by 5.

One might naturally wonder whether these divisibility implications can be directly explained through the lens of these power sums, although we will not explore this particular subject in the current paper.

We end with another possible application of Theorem 5.1. It is a natural question to ask whether $K_{2i}(\mathcal{O}_F)[p^\infty]$ is cyclic for a given prime p. The following corollary gives a sufficient condition for verifying this.

Corollary 5.4. *Let F be a number field. Suppose that $\left|K_{2i_0}(\mathcal{O}_F)[p^\infty]\right| = p$ for some i_0. Then we have*

$$r_p\big(K_{2i}(\mathcal{O}_F)\big) = 1,$$

whenever $i \equiv i_0 \pmod{|F(\mu_p) : F|}$. In other words, $K_{2i}(\mathcal{O}_F)[p^\infty]$ is cyclic for $i \equiv i_0 \pmod{|F(\mu_p) : F|}$.

References

[1] P. Báyer and J. Neukirch, On values of zeta functions and l-adic Euler characteristics. Invent. Math. 50 (1978/79), no. 1, 35-64.

[2] A. Borel, Stable real cohomology of arithmetic groups. Ann. Sci. École Norm. Sup. (4) 7 (1974), 235-272.

[3] J. Browkin, Computing the tame kernel of quadratic imaginary fields (with an appendix by Karim Belabas and Herbert Gangl). Math. Comp. 69 (2000), no. 232, 1667-1683.

[4] J. Browkin, Tame kernels of cubic cyclic fields. Math. Comp. 74 (2005), no. 250, 967-999.

[5] J. Browkin and H. Gangl, Tame and wild kernels of quadratic imaginary number fields. Math. Comp. 68 (1999), no. 225, 291-305.

[6] J. Coates, On K_2 and some classical conjectures in algebraic number theory. Ann. of Math. (2) 95 (1972), 99-116.

[7] H, Garland, A finiteness theorem for K_2 of a number field. Ann. of Math. (2) 94 (1971), 534-548.

[8] X. Guo and H. Qin, The extended Bloch groups of biquadratic and dihedral number fields, J. Pure Appl. Algebra 222 (2018) 3968-3981.

[9] K. Iwasawa, On $\mathbf{Z}_l$-extensions of algebraic number fields. Ann. of Math. (2) 98 (1973), 246-326.

[10] M. Kolster, K-theory and arithmetic. Contemporary developments in algebraic K-theory, 191–258, ICTP Lect. Notes, XV, Abdus Salam Int. Cent. Theoret. Phys., Trieste, 2004.

[11] S. Lichtenbaum, On the values of zeta and L-functions. I. Ann. of Math. (2) 96 (1972), 338-360.

[12] S. Lichtenbaum, Values of zeta-functions, étale cohomology, and algebraic K-theory. Algebraic K-theory, II: "Classical" algebraic K-theory and connections with arithmetic (Proc. Conf., Battelle Memorial Inst., Seattle, Wash., 1972), pp. 489-501. Lecture Notes in Math., Vol. 342, Springer, Berlin, 1973.

[13] B. Mazur and A. Wiles, Class fields of abelian extensions of $\mathbb{Q}$. Invent. Math. 76 (1984), no. 2, 179-330.

[14] D. Quillen, Higher algebraic K-theory. I. Algebraic K-theory, I: Higher K-theories (Proc. Conf., Battelle Memorial Inst., Seattle, Wash., 1972), pp. 85-147. Lecture Notes in Math., Vol. 341, Springer, Berlin 1973.

[15] D. Quillen, Finite generation of the groups K_i of rings of algebraic integers. Algebraic K-theory, I: Higher K-theories (Proc. Conf., Battelle Memorial Inst., Seattle, Wash., 1972), pp. 179-198. Lecture Notes in Math., Vol. 341, Springer, Berlin, 1973.

[16] J. Rognes and C. Weibel, Two-primary algebraic K-theory of rings of integers in number fields (Appendix A by M. Kolster). J. Amer. Math. Soc. 13 (2000), no. 1, 1-54.

[17] C. L. Siegel, Advanced analytic number theory. Second edition. Tata Institute of Fundamental Research Studies in Mathematics, 9. Tata Institute of Fundamental Research, Bombay, 1980.

[18] E. H. Spanier, Algebraic topology. reprint of the 1966 original. Springer-Verlag, New York, 1989.

[19] C. Soulé, K-théorie des anneaux d'entiers de corps de nombres et cohomologie étale. Invent. Math. 55 (1979), no. 3, 251-295.

[20] V. Voevodsky, On motivic cohomology with $\mathbf{Z}/l$-coefficients. Ann. of Math. (2) 174 (2011), no. 1, 401-438.

[21] L. C. Washington, Introduction to cyclotomic fields. Second edition. Graduate Texts in Mathematics, 83. Springer-Verlag, New York, 1997.

[22] C. Weibel, The norm residue isomorphism theorem, J. Topol. 2 (2009), no. 2, 346-372.

[23] C. Weibel, The K-book. An introduction to algebraic K-theory. Graduate Studies in Mathematics, 145. American Mathematical Society, Providence, RI, 2013. xii+618 pp.

[24] A. Wiles, The Iwasawa conjecture for totally real fields. Ann. of Math. (2) 131 (1990), no. 3, 493-540.

[25] D. Zagier, On the values at negative integers of the zeta-function of a real quadratic field. Enseign. Math. (2) 22 (1976), no. 1-2, 55-95.

[26] H. Zhou, Tame kernels of cubic cyclic fields. Acta Arith. 124 (2006), no. 4, 293-313.

[27] H. Zhou, The tame kernel of multiquadratic number fields, Commun. Alg. 37 (2009), no. 2, 630-638.

Appendix A. Algorithms

This appendix provides algorithms for computing the size of higher even K-groups of ring of integers of totally real number fields. The algorithms are implemented in SageMath and are divided into two categories: those for quadratic fields, including Siegel formula-related computations and multi-quadratic field cases, and those for general number fields with Galois group isomorphic to $(\mathbb{Z}/p\mathbb{Z})^n$.

A.1. **Quadratic Field Case.** This section presents algorithms tailored for quadratic fields, including foundational computations, Siegel formula-related functions, and a method for combining K-group sizes across multiple quadratic fields. These algorithms compute the size of higher even K-groups and related quantities using modular forms, zeta functions, and prime products.

Algorithm 1 Computes the size of $K_n(\mathbb{Z})$

Input: Integer n
Output: Absolute value of constant
function KZ(n)
    if $n \mod 8 = 2$ **then**
                                                ▷ Case: $n \equiv 2 \pmod 8$
        Compute $k \leftarrow (n-2)/8$
        Compute $s \leftarrow 2k+1$
        Compute $c \leftarrow \text{bernoulli}(2s) \cdot w(\mathbb{Q}, 1, 2s)/(4s)$
        return $|2c|$
    else if $n \mod 8 = 6$ **then**
                                                ▷ Case: $n \equiv 6 \pmod 8$
        Compute $k \leftarrow (n-6)/8$
        Compute $s \leftarrow 2k+2$
        Compute $c \leftarrow \text{bernoulli}(2s) \cdot w(\mathbb{Q}, 1, 2s)/(4s)$
        return $|c|$
    end if
end function

Algorithm 2 Compute $w_j(F)$

Input: Quadratic field K, integer i
Output: Product of primes
function $\mathrm{w}(K, i)$
    Initialize empty list list1
    ▷ Collect primes where $i/(\ell-1) \in \mathbb{Z}$
    for ℓ in primes$(2, i+2)$ **do**
        if $i/(\ell-1) \in \mathbb{Z}$ **then**
            Append ℓ to list1
        end if
    end for
    Filter list1 $\leftarrow \{\ell \in \text{list1} \mid \ell \neq 2\}$
    Compute $k \leftarrow 2i$
    Initialize empty list list2
    ▷ Collect additional primes
    for ℓ in primes$(2, 2i+2)$ **do**
        if $k/(\ell-1) \in \mathbb{Z}$ and $i/(\ell-1) \notin \mathbb{Z}$ **then**
            Append ℓ to list2
        end if
    end for
    Initialize prod $\leftarrow 1$
    ▷ Compute product for list1 primes
    for $\ell \in$ list1 **do**
        if $i/\ell \notin \mathbb{Z}$ **then**
            prod $\leftarrow$ prod $\cdot\, \ell$
        else
            prod $\leftarrow$ prod $\cdot\, \ell^{\text{valuation}(i,\ell)+1}$
        end if
    end for
    prod $\leftarrow$ prod $\cdot\, 2^{\text{valuation}(i,2)+2}$
    ▷ Adjust for specific quadratic fields
    for $\ell \in$ list2 **do**
        if $K = \mathbb{Q}(\sqrt{\ell})$ **then**
            prod $\leftarrow$ prod $\cdot\, \ell^{\text{valuation}(i,\ell)+1}$
        end if
    end for
    if $K = \mathbb{Q}(\sqrt{2})$ or $K = \mathbb{Q}(\sqrt{8})$ **then**
        prod $\leftarrow 2 \cdot$ prod
    end if
    return prod
end function

Algorithm 3 Compute $T_h = G_{12r-h+2}\Delta^{-r}$

Input: Integer h
Output: Modular form T
function T(h)
    if $h \mod 12 = 2$ **then**
        Compute $r \leftarrow \lfloor h/12 \rfloor$
    else
        Compute $r \leftarrow \lfloor h/12 \rfloor + 1$
    end if
    Compute $k \leftarrow 12r - h + 2$
    if $k = 0$ **then**
        Compute $T \leftarrow \Delta^{-r}$
    else
        Compute Eisenstein series $G \leftarrow$ eisenstein_series_qexp$(k, r+1, \text{normalization='constant'})$
        Compute $T \leftarrow G \cdot \Delta^{-r}$
    end if
    return $T + O(q)$
end function

Algorithm 4 Compute Siegel coefficients $b_j(h)$

Input: Integers l, h
Output: Ratio of modular form coefficients
function B(l, h)
    Compute modular form $T \leftarrow T(h)$
    Extract coefficients $C \leftarrow T$.coefficients()
    Compute length $L \leftarrow |C|$
    return $-C[L-l-1]/C[L-1]$
end function

Algorithm 5 Compute Siegel Sum

Input: Integer D, exponent j
Output: Sum of divisors raised to power j
function eSiegel(D, j)
    Initialize haha $\leftarrow 0$
    Initialize bla $\leftarrow 0$
    if $D/4 \in \mathbb{Z}$ **then**
                                                ▷ Sum over divisors of $D/4$
        for each $i \in$ divisors($\lfloor D/4 \rfloor$) **do**
            haha $\leftarrow$ haha $+ i^j$
        end for
    end if
                                                ▷ Sum over divisors for quadratic residues
    for b in $[1, \lfloor\sqrt{D}\rfloor]$ **do**
        if $(D - b^2)/4 \in \mathbb{Z}$ **then**
            for each $i \in$ divisors($\lfloor (D - b^2)/4 \rfloor$) **do**
                bla $\leftarrow$ bla $+ i^j$
            end for
        end if
    end for
    return haha $+ 2 \cdot$ bla
end function

Algorithm 6 Compute Weighted Sum with Kronecker Symbol

Input: Quadratic field F, integers l, m
Output: Weighted sum
function S(F, l, m)
    Compute discriminant $D \leftarrow F$.discriminant()
    Initialize sum $\leftarrow 0$
                                                ▷ Sum over divisors of l
    for each $j \in$ divisors(l) **do**
        Compute sum $\leftarrow$ sum + kronecker_symbol(D, j) $\cdot\, j^{2m-1} \cdot$ eSiegel($(l/j)^2 \cdot D, 2m - 1$)
    end for
    return sum
end function

Algorithm 7 Compute Zeta Function Value for Quadratic Field

Input: Discriminant D, complex number s
Output: Zeta function value
function ZETA(D, s)
    Create quadratic field $F \leftarrow \mathbb{Q}(\sqrt{D})$
    Set $r \leftarrow 2$
    Compute $k \leftarrow (1-s)/2$
    if $kr \mod 6 = 1$ **then**
        Compute $c \leftarrow \lfloor kr/6 \rfloor$
    else
        Compute $c \leftarrow \lfloor kr/6 \rfloor + 1$
    end if
    Initialize sum $\leftarrow 0$
    ▷ Sum over coefficients
    for j in $[1, c]$ **do**
        Compute sum $\leftarrow$ sum $+ b(j, 2kr) \cdot S(F, j, k)$
    end for
    return $2^r \cdot$ sum
end function

Algorithm 8 Compute size of K-group of Quadratic Field

Input: Quadratic field K, integer n
Output: Rounded K-group size
function KSize(K, n)
    Compute degree $r \leftarrow K$.degree()
    Compute discriminant $D \leftarrow K$.discriminant()
    if $n \mod 2 = 0$ **then**
        Compute $k \leftarrow (n+2)/4$
        if $k \mod 2 = 0$ **then**
                                                                ▷ Even k
            Compute $SS \leftarrow (1/2^r) \cdot w(K, 2k) \cdot \text{Zeta}(D, 1-2k)$
        else
                                                                ▷ Odd k
            Compute $SS \leftarrow (-1)^r \cdot w(K, 2k) \cdot \text{Zeta}(D, 1-2k)$
        end if
    else
        Compute $k \leftarrow (n+1)/4$
        if $k \mod 2 = 0$ **then**
                                                                ▷ Even k
            Compute $SS \leftarrow w(K, 2k)$
        else
                                                                ▷ Odd k
            Compute $SS \leftarrow 2^r \cdot w(K, 2k)$
        end if
    end if
    return round(SS)
end function

Algorithm 9 Compute the size of K-group Size for multi-quadratic Fields

Input: List of quadratic fields list1, list of conductors list2, integer n
Output: Combined K-group size
function MultiSize(list1, list2, n)
    Initialize prod $\leftarrow 1$
    Compute degree $p \leftarrow$ list1[0].degree()
    Set $k \leftarrow 3$ ▷ Assumes Galois group degree
                                                                ▷ Compute product of K-group sizes
    for i in $[0, |\text{list1}| - 1]$ **do**
        Compute prod $\leftarrow$ prod $\cdot$ KSize(list1[i], list2[i], n)
    end for
    Compute denominator denominator $\leftarrow \text{KZ}(n)^{(p^k-p)/(p-1)}$
    return prod/denominator
end function

Purpose: Computes the combined K-group size for a list of quadratic fields.

A.2. **General Number Field Case.** This section presents algorithms for computing K-group sizes for general number fields, particularly those with Galois group $(\mathbb{Z}/p\mathbb{Z})^{\oplus n}$. These algorithms handle subfield checks, prime products, zeta functions, and K-group computations for arbitrary degrees.

Algorithm 10 Check if K is a Subfield of L

Input: Number fields K, L
Output: Boolean indicating if $K \subseteq L$
function IS_SUBFIELD(K, L)
    Compute embeddings $\mathrm{Emb}(K, L)$ of K into L
    return $|\mathrm{Emb}(K, L)| > 0$
end function

Algorithm 11 Compute $w_j(F)$ for a random totally real field F, based on Proposition 4.2

Input: Number field K, conductor cond, integer i
Output: Product of prime powers
function $\mathrm{w}(K, \mathrm{cond}, i)$
    Compute degree $r \leftarrow \deg(K)$
    Initialize empty list list1
    ▷ Collect primes where $i \mod (\ell - 1) = 0$
    for ℓ in prime_range$(2, i + 2)$ **do**
        if $i \mod (\ell - 1) = 0$ **then**
            Append ℓ to list1
        end if
    end for
    Initialize prod $\leftarrow 1$
    if $K = \mathbb{Q}(\sqrt{2})$ **then**
        ▷ Case: $K = \mathbb{Q}(\sqrt{2})$
        for $\ell \in$ list1 **do**
            prod $\leftarrow$ prod $\cdot\ \ell^{1+\mathrm{valuation}(4i,\ell)}$
        end for
    else if r is odd and $K \subseteq$ CyclotomicField(r^2) **then**
        ▷ Case: Odd degree and cyclotomic subfield
        for $\ell \in$ list1 **do**
            prod $\leftarrow$ prod $\cdot\ \ell^{1+\mathrm{valuation}(2ri,\ell)}$
        end for
    else if cond is prime and $K \subseteq$ CyclotomicField(cond) **then**
        ▷ Case: Cyclotomic field with prime conductor
        list2 $\leftarrow \{\ell \in \text{list1} \mid \ell \neq 2, \ell \neq \mathrm{cond}\}$
        for $\ell \in$ list2 **do**
            prod $\leftarrow$ prod $\cdot\ \ell^{1+\mathrm{valuation}(i/2,\ell)}$
        end for
        if $(i \cdot r) \mod (\mathrm{cond} - 1) = 0$ **then**
            prod $\leftarrow$ prod $\cdot\ 2^{3+\mathrm{valuation}(i/2,2)} \cdot \mathrm{cond}^{1+\mathrm{valuation}(i/2,\mathrm{cond})}$
        else
            prod $\leftarrow$ prod $\cdot\ 2^{3+\mathrm{valuation}(i/2,2)}$
        end if
    else
        ▷ Default case
        for $\ell \in$ list1 **do**
            prod $\leftarrow$ prod $\cdot\ \ell^{1+\mathrm{valuation}(2i,\ell)}$
        end for
    end if
    return prod
end function

Algorithm 12 Compute Zeta Function Value for Number Field with $Gal(K/\mathbb{Q}) \cong \mathbb{Z}/p\mathbb{Z}$

Input: Number field K, conductor cond, complex number s
Output: Zeta function value
function ZETA(K, cond, s)
    Compute $n \leftarrow 1 - s$
    Compute degree deg $\leftarrow K$.degree()
    Create Dirichlet group $G \leftarrow$ DirichletGroup(cond)
    Initialize Lval $\leftarrow 1$
    if deg $= 2$ **then**
                                                                ▷ Quadratic field case
        Compute Lval $\leftarrow -$QuadraticBernoulliNumber(n, cond)$/n$
    else
                                                                ▷ General degree case
        Compute character $\chi \leftarrow G.0^{(G.\text{order}()/\ \text{deg})}$
        for i in $[1, \text{deg} - 1]$ **do**
            Compute Lval $\leftarrow$ Lval $\cdot\ (-(\chi^i).\text{bernoulli}(n)/n)$
        end for
    end if
    return $\zeta(s) \cdot$ Lval
end function

Algorithm 13 Compute the size of K-groups for Number Field with $Gal(K/\mathbb{Q}) \cong \mathbb{Z}/p\mathbb{Z}$

Input: Number field K, conductor cond, integer n
Output: Rounded K-group size
function KSize(K, cond, n)
    Compute degree $r \leftarrow K$.degree()
    if $n \mod 2 = 0$ **then**
        Compute $k \leftarrow (n+2)/4$
        if $k \mod 2 = 0$ **then**
                                                                ▷ Even k
            Compute $SS \leftarrow (1/2^r) \cdot w(K, \text{cond}, 2k) \cdot \text{Zeta}(K, \text{cond}, 1-2k)$
        else
                                                                ▷ Odd k
            Compute $SS \leftarrow (-1)^r \cdot w(K, \text{cond}, 2k) \cdot \text{Zeta}(K, \text{cond}, 1-2k)$
        end if
    else
        Compute $k \leftarrow (n+1)/4$
        if $k \mod 2 = 0$ **then**
                                                                ▷ Even k
            Compute $SS \leftarrow w(K, \text{cond}, 2k)$
        else
                                                                ▷ Odd k
            Compute $SS \leftarrow 2^r \cdot w(K, \text{cond}, 2k)$
        end if
    end if
    return round(SS)
end function

Algorithm 14 Compute Scaled Value for K-group

Input: Number field K, conductor cond, integer n
Output: Scaled rational value
function OMG(K, cond, n)
    Compute degree $r \leftarrow \deg(K)$
    Compute $k \leftarrow \lfloor (n+2)/4 \rfloor$
    if k is even **then**
                                                                ▷ Even case
        Compute $SS \leftarrow w(K, \text{cond}, 2k)/2^r$
    else
                                                                ▷ Odd case
        Compute $SS \leftarrow (-1)^r \cdot w(K, \text{cond}, 2k)$
    end if
    return $\mathbb{Q}(\mathbb{R}(SS))$
end function

Algorithm 15 Compute Size of Higher Even K-groups

Input: Number field K, conductor cond, prime p, integers k, n
Output: Size of K-group
function KSizeppower(K, cond, p, k, n)
    Compute degree $r \leftarrow \deg(K)$
    Compute $a \leftarrow \lfloor (n+2)/4 \rfloor$
    Create Dirichlet group $G \leftarrow$ DirichletGroup(cond)
    Collect characters $\leftarrow \{\chi \in G \mid \chi.\text{multiplicative\_order}() \in \{1, p\}\}$
    Compute subfields Subfields $\leftarrow K$.subfields()
    Initialize empty list inter
▷ Collect subfields of degree p
    for each subfield $S \in$ Subfields **do**
        if $\deg(S) = p$ **then**
            Append S to inter
        end if
    end for
    Initialize prod $\leftarrow 1$
▷ Compute Zeta value
    for each $\chi \in$ characters **do**
        prod $\leftarrow$ prod $\cdot$ $(-\chi.\text{primitive\_character}().\text{bernoulli}(2a)/(2a))$
    end for
    prod $\leftarrow$ prod $\cdot\, \zeta(1-2a)^{|\text{inter}|-1}$
    Initialize $ww \leftarrow 1$
▷ Compute product of $w_j(F)$
    for each $i \in$ inter **do**
        $ww \leftarrow ww \cdot \text{OMG}(i, i.\text{conductor}(), n)$
    end for
▷ Compute denominator
    denominator $\leftarrow \text{KZ}(n)^{(p^k-p)/(p-1)}$
    return prod $\cdot\, ww$/denominator
end function

A.3. `KSizeppower(K, cond, p, k, n)`.

Appendix B. Examples

This appendix introduces three methods for computing the size of higher even K-groups, each tailored to specific types of number fields. The major distinction among these methods lies in how the special values of the Dedekind zeta function are computed. While software like SageMath and Magma provide built-in functions for numerical approximations of these values, we aim to compute them exactly as fractions. The first method is designed for fields whose degrees are p-powers, leveraging their specific arithmetic structures. The second method applies to fields of higher degree or those whose degrees are not p-powers, where the associated Dirichlet character must be manually input from the LMFDB to ensure precise fractional computation. Finally, for the quadratic case, we provide a simplified approach using Siegel's formula and also offer a faster method to calculate $w_j(F)$, enhancing efficiency while maintaining exactness. Together, these methods form a comprehensive framework for rigorously determining K-group sizes across various field types.

The first method is the most general and theoretically works for every totally real abelian number field. The primary difference from the first method lies in the computation of the Dedekind zeta function, where the associated Dirichlet character must be manually input from the LMFDB to obtain the special values. While this generality ensures broad applicability, the computation of $w_j(F)$ can become significantly slower as the degree of the field increases, due to the amplified complexity and computational effort required. To illustrate this method, we include an example of computing the Dedekind zeta value for a degree 21 field, highlighting both its flexibility and the challenges posed by high-degree fields.

Listing 1. Computing special value of Dedekind Zeta function for $\mathbb{Q}(x^{21} - 7x^{20} - 70x^{19} + 462x^{18} + 2135x^{17} - 12411x^{16} - 36610x^{15} + 175044x^{14} + 373940x^{13} - 1403661x^{12} - 2218069x^{11} + 6566007x^{10} + 6982234x^9 - 17907827x^8 - 9448729x^7 + 26548844x^6 + 686581x^5 - 16732429x^4 + 5281647x^3 + 1717044x^2 - 573440x - 71167)$

```
from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(2))

M = H._module

chi1 = DirichletCharacter(H, M([0,0]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(42))

M = H._module

chi2 = DirichletCharacter(H, M([6,28]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(14))

M = H._module

chi3 = DirichletCharacter(H, M([10,0]))

```

```
from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(6))

M = H._module

chi4 = DirichletCharacter(H, M([0,2]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(42))

M = H._module

chi5 = DirichletCharacter(H, M([36,28]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(42))

M = H._module

chi6 = DirichletCharacter(H, M([24,14]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(14))

M = H._module

chi7 = DirichletCharacter(H, M([6,0]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(42))

M = H._module

chi8 = DirichletCharacter(H, M([30,14]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(42))

```

```
M = H._module

chi9 = DirichletCharacter(H, M([24,28]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(14))

M = H._module

chi10 = DirichletCharacter(H, M([2,0]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(42))

M = H._module

chi11 = DirichletCharacter(H, M([12,28]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(14))

M = H._module

chi12 = DirichletCharacter(H, M([12,0]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(42))

M = H._module

chi13 = DirichletCharacter(H, M([6,14]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(6))

M = H._module

chi14 = DirichletCharacter(H, M([0,4]))

```

```
from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(14))

M = H._module

chi15 = DirichletCharacter(H, M([8,0]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(42))

M = H._module

chi16 = DirichletCharacter(H, M([36,14]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(42))

M = H._module

chi17 = DirichletCharacter(H, M([18,14]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(42))

M = H._module

chi18 = DirichletCharacter(H, M([30,28]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(14))

M = H._module

chi19 = DirichletCharacter(H, M([4,0]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(42))

```

```
M = H._module

chi20 = DirichletCharacter(H, M([18,28]))

from sage.modular.dirichlet import DirichletCharacter

H = DirichletGroup(637, base_ring=CyclotomicField(42))

M = H._module

chi21 = DirichletCharacter(H, M([12,14]))




CHI = [chi1,chi2,chi3, chi4, chi5, chi6, chi7,chi8,chi9, chi10, chi11, chi12, chi13, chi14,
    chi15, chi16, chi17,chi18,chi19, chi20, chi21]

CChi=[]
for i in CHI:
  CChi.append(i.primitive_character())

prod =1

for i in CChi:
  prod*=-i.bernoulli(2)/2
```

For quadratic fields, we employ Siegel's formula to compute the special values of the Dedekind zeta function. Furthermore, we introduce a faster method to compute $w_j(F)$, optimizing the process while maintaining the accuracy required to determine the sizes of the K groups. This specialized approach significantly improves computational efficiency in the quadratic case.

LISTING 2. Computing special value of Dedekind Zeta function using Siegel's formula

```
# Define a power series ring over rational numbers
R.<q> = QQ[['q']]

# Compute Eisenstein series q-expansions with given normalization and precision
F4 = eisenstein_series_qexp(4, 4000, normalization='constant')  # Eisenstein series of weight
    4
F6 = eisenstein_series_qexp(6, 4000, normalization='constant')  # Eisenstein series of weight
    6

# Define the discriminant modular form Delta as a combination of F4 and F6
Delta = (F4^3 - F6^2) / 1728

def T(h):
    """
```

```
    Computes a modular form T(h) based on Delta and Eisenstein series of various weights.
    :param h: Integer, typically the weight of the modular form.
    :return: A q-expansion of the modular form T(h).
    """
    if Mod(h, 12) == 2:
        r = floor(h / 12)
    else:
        r = floor(h / 12) + 1

    k = 12 * r - h + 2  # Compute the required weight for the Eisenstein series
    if k == 0:
        T = Delta^(-r)  # Case where k = 0, use only powers of Delta
    else:
        G = eisenstein_series_qexp(k, r + 1, normalization='constant')  # Eisenstein series of
     weight k
        T = G * Delta^(-r)  # Combine with Delta

    return T + O(q)  # Return the truncated q-expansion

def b(l, h):
    """
    Compute the normalized coefficient b_j(h).
    :param l: Index of the coefficient to retrieve.
    :param h: Weight parameter for T(h).
    :return: thew value of b_j(h).
    """
    C = T(h).coefficients()  # Get coefficients of T(h)
    L = len(C)  # Total number of coefficients
    return -C[L - l - 1] / C[L - 1]  # Return normalized coefficient

def eSiegel(D, j):
    """
    Computes the Siegel-type divisor sum for a given discriminant D and exponent j.
    :param D: Discriminant.
    :param j: Exponent for the divisor function.
    :return: Siegel sum value.
    """
    haha = 0
    bla = 0

    # Compute first term where D/4 is an integer
    if D / 4 in ZZ:
        for i in divisors(round(D / 4)):
            haha += i^j
```

```
56
57      # Compute second term for values b where (D - b^2)/4 is an integer
58      for b in range(1, floor(sqrt(D)) + 1):
59          if (D - b^2) / 4 in ZZ:
60              for i in divisors(round((D - b^2) / 4)):
61                  bla += i^j
62
63      return haha + 2 * bla
64
65  def S(F, l, m):
66      """
67      Computes a sum involving Kronecker symbols and Siegel sums for a field F.
68      :param F: A quadratic field.
69      :param l: Integer parameter.
70      :param m: Integer parameter.
71      :return: The computed sum.
72      """
73      sum = 0
74      D = F.discriminant()  # Discriminant of the quadratic field
75
76      # Iterate over divisors of l
77      for j in divisors(l):
78          sum += kronecker_symbol(D, j) * j^(2 * m - 1) * eSiegel((l / j)^2 * D, 2 * m - 1)
79
80      return sum
81
82  def Zeta(D, s):
83      """
84      Computes a zeta-like function for a totally real quadratic field with discriminant D.
85      :param D: Discriminant of the quadratic field.
86      :param s: An integer parameter.
87      :return: Zeta value.
88      """
89      F = QuadraticField(D)  # Create the quadratic field
90      r = 2  # Degree of the field (always 2 for quadratic fields)
91      k = (1 - s) / 2  # Parameter derived from s
92
93      # Determine the range for summation based on k*r modulo 6
94      if Mod(k * r, 6) == 1:
95          c = floor(k * r / 6)
96      else:
97          c = floor(k * r / 6) + 1
98
99      sum = 0
```

```
100
101     # Compute Dedekind zeta function as in Theorem 4.1
102     for j in range(1, c + 1):
103         sum += b(j, 2 * k * r) * S(F, j, k)
104
105     return 2^r * sum  # Multiply by 2^r and return the result
106
107
108
109
110 def w(K, i):
111     """
112     Computes w_j(F) for real quadratic fields.
113     :param K: A number field, typically quadratic.
114     :param i: An integer parameter used to determine specific properties of the primes.
115     :return: w_j(F).
116     """
117     list1 = []
118
119     # Collect primes satisfying (i / (ell - 1)) in Z
120     for ell in list(primes(2, i + 2)):
121         if (i / (ell - 1)) in ZZ:
122             list1.append(ell)
123
124     # Remove 2 from list1 to handle it separately
125     list1 = [ell for ell in list1 if ell != 2]
126
127     # Determine additional primes based on k = 2 * i and specific divisibility criteria
128     k = 2 * i
129     list2 = []
130     for ell in list(primes(2, 2 * i + 2)):
131         if (k / (ell - 1)) in ZZ and (i / (ell - 1)) not in ZZ:
132             list2.append(ell)
133
134     # Compute the product based on the properties of list1 and list2
135     prod = 1
136     for ell in list1:
137         if (i / ell) not in ZZ:
138             prod *= ell  # Multiply by prime
139         else:
140             prod *= ell**(i.valuation(ell) + 1)  # Adjust power based on valuation
141
142     # Handle the contribution from 2 separately
143     prod *= 2**(i.valuation(2) + 2)
```

```
144
145     # Additional adjustment for primes in list2
146     for ell in list2:
147         if K == QuadraticField(ell):  # Check if K matches a specific quadratic field
148             prod *= ell**(i.valuation(ell) + 1)
149
150     # Adjust for special cases where K is QuadraticField(2) or QuadraticField(8)
151     if K == QuadraticField(2) or K == QuadraticField(8):
152         prod = 2 * prod
153
154     return prod
155
156
157 def KSize(K, n):
158     """
159     Computes the size of higher even K-groups for the field K and parameter n.
160     :param K: A number field.
161     :param n: An integer parameter related to the K-group.
162     :return: The computed size as an integer.
163     """
164     r = K.degree()  # Degree of the field
165     D = K.discriminant()  # Discriminant of the field
166
167     # Case when n is even
168     if n % 2 == 0:
169         k = (n + 2) // 4
170         if k % 2 == 0:  # k is even
171             SS = (1 / (2^r)) * w(K, 2 * k) * Zeta(D, 1 - 2 * k)
172         else:  # k is odd
173             SS = (-1)^r * w(K, 2 * k) * Zeta(D, 1 - 2 * k)
174
175     # Case when n is odd
176     else:
177         k = (n + 1) // 4
178         if k % 2 == 0:  # k is even
179             SS = w(K, 2 * k)
180         else:  # k is odd
181             SS = 2^r * w(K, 2 * k)
182
183     return round(SS)  # Return the rounded result
```

For p-elementary abelian totally real fields, our code leverages the specific arithmetic properties of these fields to compute the size of higher even K-groups efficiently, directly utilizing the structured relationship between the field and its Dedekind zeta function.

LISTING 3. Code for computing size of higher even K-groups

```
def is_subfield(K, L):
    """
    Checks if K is a subfield of L by verifying the existence of embeddings.
    :param K: A number field or similar algebraic structure.
    :param L: A number field into which embeddings are checked.
    :return: True if K embeds into L, otherwise False.
    """
    return len(K.embeddings(L)) > 0


#Computes w_i(K) based on Propsition 4.2
def w(K, cond, i):
    """
    Computes a weighted product of prime powers for a given number field and conditions.
    :param K: A number field.
    :param cond: An integer condition, e.g., conductor of a field.
    :param i: An integer parameter used in the calculations.
    :return: A product based on certain arithmetic properties of K and i.
    """
    r = K.degree()  # Degree of the number field K.
    list1 = []

    # Collecting primes that satisfy specific congruence properties.
    for ell in prime_range(2, i + 2):
        if i % (ell - 1) == 0:
            list1.append(ell)

    prod = 1

    # Case 1: K is a subfield of QuadraticField(2) and QuadraticField(8)
    if is_subfield(QuadraticField(2), K) and is_subfield(K, QuadraticField(8)):
        for ell in list1:
            prod *= ell ** (1 + (4 * i).valuation(ell))

    # Case 2: K is related to a CyclotomicField of degree r^2
    elif r % 2 == 1 and cond == r ** 2:
        for ell in list1:
            prod *= ell ** (1 + (2 * r * i).valuation(ell))

    # Case 3: K is related to a CyclotomicField of a prime conductor
    elif s_prime(cond) and cond % 2 == 1:
        list2 = [ell for ell in list1 if ell != 2 and ell != cond]
```

```
        for ell in list2:
            prod *= ell ** (1 + (i / 2).valuation(ell))
        if (i * r) % (cond - 1) == 0:
            prod = 2 ** (3 + (i / 2).valuation(2)) * cond ** (1 + (i / 2).valuation(cond)) *
    prod
        else:
            prod = 2 ** (3 + (i / 2).valuation(2)) * prod

    # Default case
    else:
        for ell in list1:
            prod *= ell ** (1 + (2 * i).valuation(ell))

    return prod


#The following function computes the special value of the Dedekind Zeta function for number
    field K.
#But in this case, it only deals with the fields with p degrees.
#We also have a general version which deals with any fields. (see appendix xxx)
def Zeta(K, cond, s):
    """
    Computes a modified zeta function for the number field K.
    :param K: A number field.
    :param cond: An integer condition, typically related to the field's conductor.
    :param s: A negative integer.
    :return: The special value of Dedekind Zeta fucntion of K.
    """
    n = 1 - s
    deg = K.degree()  # Degree of the field.
    G = DirichletGroup(cond)
    Lval = 1

    if deg == 2:
        Lval = -QuadraticBernoulliNumber(n, cond) / n
    else:
        chi = G.0^(ZZ(G.order() / deg))
        for i in (1..deg - 1):
            Lval *= (-(chi^i).bernoulli(n) / n)

    return zeta(s) * Lval
```

```
#Computes the size of even K group of ring of integers of the field K using Theorem 2.3 and Corollary 2.4.
def KSize(K, cond, n):
    """
    Computes the size of even K group of ring of integers of the field K.
    :param K: A number field K.
    :param cond: An integer condition, typically a conductor.
    :param n: An integer parameter.
    :return: The computed size.
    """
    r = K.degree()

    if n % 2 == 0:
        k = (n + 2) // 4
        if k % 2 == 0:
            SS = 1 / (2**r) * w(K, cond, 2 * k) * Zeta(K, cond, 1 - 2 * k)
        else:
            SS = (-1)**r * w(K, cond, 2 * k) * Zeta(K, cond, 1 - 2 * k)
    else:
        k = (n + 1) // 4
        if k % 2 == 0:
            SS = w(K, cond, 2 * k)
        else:
            SS = 2**r * w(K, cond, 2 * k)

    return round(SS)


#This function computes size of even K groups of Z, which appeared in Theorem 4.1.
def KZ(n):
    """
    Computes size of even K groups of Z, which appeared in Theorem 4.1.
    :param n: An integer.
    :return: A scaling factor based on Bernoulli numbers and prime valuations.
    """
    ZZ = IntegerRing()

    if n % 8 == 2:
        k = ZZ((n - 2) / 8)
        s = 2 * k + 1
        c = bernoulli(2 * s) * w(QQ, 1, 2 * s) / (4 * s)
        return abs(2 * c)
    elif n % 8 == 6:
        k = ZZ((n - 6) / 8)
```

```
        s = 2 * k + 2
        c = bernoulli(2 * s) * w(QQ, 1, 2 * s) / (4 * s)
        return abs(c)


#If E is a totally real Galois extension of Q and Galois group is p-elementary,
#then given a list of intermediate field of E and a list of associated conductors of these intermediate fields.
#We can compute the size of even K groups of ring of integers of E by the formula in Theorem 4.1.
def MultiSize(list1, list2, n):
    """
    Computes a combined size metric over multiple fields and conditions.
    :param list1: A list of intermediate fields.
    :param list2: A list of conductors corresponding to the intermediate fields.
    :param n: An integer n=2k.
    :return: Size of the K-group for number field E.
    """
    prod = 1
    p = list1[0].degree()  # Degree of the first field.
    k = 2  # Degree from Galois group.

    # Compute the product of sizes for each field in the list.
    for i in range(len(list1)):
        prod *= KSize(list1[i], list2[i], n)

    # Normalize by a scaling factor.
    return prod / (KZ(n)**((p**k - p) // (p - 1)))
```

School of Mathematics and Statistics, Hubei Key Laboratory of Mathematical Sciences, Central China Normal University, Wuhan, 430079, P.R.China.

Email address: limmf@ccnu.edu.cn

College of Mathematical Sciences, Harbin Engineering University, Harbin, 150001, P.R.China.

Email address: qinchao@hrbeu.edu.cn