EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
2026, Vol. 19, Issue 2, Article Number 7129
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

Geometric Proofs for Quadratic and Cubic Power Sums and a Reduction Algorithm for Higher Power Sums with Structural Links to Algebraic Integer Rings

Guangbiao Guo

Electrical Engineering and Automation (Partnership Program between China and USA), School of Electrical Engineering and Automation, Henan Polytechnic University, Kaifeng, Henan, China

Abstract. Power sum problem is a classical topic across combinatorial algebra and number theory, where traditional derivations mainly rely on Bernoulli numbers and generating function techniques, which lack intuitive geometric interpretations and straightforward decomposition method for higher-order sums. In this paper, we first propose novel geometric constructive proofs for quadratic and cubic power sums of the first n natural numbers, realizing the deduction of low-order power sum formulas via geometric partitioning rather than algebraic expansion. On this basis, we construct a pair of symmetric and antisymmetric auxiliary operators M(k) and K(k), and establish a new simple recursive decomposition algorithm that converts higher-order power sum expressions into combinations of lower-order power sums and polynomials containing n, achieving effective order reduction for odd powers without introducing Bernoulli numbers; for even powers the algorithm falls back to the standard binomial recursion. Furthermore, we explore the intrinsic structural correspondence between the proposed decomposition framework and algebraic integer rings: the symmetric-antisymmetric splitting of power sums is structurally parallel to the conjugate decomposition of Gaussian integers; the triple symmetry feature of cubic power sums parallels the structural properties of Eisenstein integers; meanwhile, the non-symmetric characteristic of higher-order decomposition can be extended to the non-commutative structure of Hurwitz quaternion integers. We conduct numerical verification for low-degree cases and compare our method with the classical Bernoulli number approach and explore class decomposition adopted in SPN architectures, illustrating the advantages of our algorithm in geometric intuition and iterative computation. We also research the calculation efficiency of the new algorithm. This work provides a new elementary decomposition paradigm for power sum research and simple intuitive proof for the sum of quadratic and cubic power, and establishes a feasible connection between power sum recursive splitting and algebraic integer ring theory.

2020 Mathematics Subject Classifications: 11B68, 05A19, 11R04, 11Y16

Key Words and Phrases: Power sum problem, novel geometric constructive proofs, Changlang operators, Gaussian integers, algebraic integer ring theory

DOI: https://doi.org/10.29020/nybg.ejpam.v19i2.7129

Email address: 1624109192@qq.com (G. Guo)

1. Introduction

The summation of integer powers $S(k,n) = \sum_{i=1}^{n} i^k$, commonly referred to as power sums, has occupied a vital position in [1] combinatorial mathematics, [2] number theory and [3] discrete algebra for centuries. Research on power sums dates back to ancient mathematical explorations, while systematic theoretical frameworks were gradually established with the derivation of Faulhaber's formula supported by Bernoulli numbers. For decades, Bernoulli number-based methods have dominated the construction of power sum polynomials, generating unified closed-form expressions for arbitrary nonnegative integer exponents k. Nevertheless, this classical paradigm bears evident limitations: its theoretical foundation originates from formal power series, leading to obscure geometric meaning; the calculation of higher Bernoulli numbers is cumbersome and dependent on precomputed constants, which complicates iterative algorithm implementation. Numerous subsequent studies have proposed alternative derivation strategies, including finite difference methods, matrix transformation techniques and combinatorial counting approaches. These methods optimize algebraic deduction to varying degrees, yet few constructions integrate geometric intuition into formula derivation, nor do they focus on recursive order reduction from higher-degree power sums to lower-degree ones. In modern discrete algebraic research, residue class decomposition derived from substitution-permutation network (SPN) architectures provides a novel perspective for discrete set splitting, which shares the core idea of element permutation and equivalence classification with power sum decomposition, though it is mostly confined to finite-domain cryptographic applications rather than classical integer summation problems.

2. Literature Review and Contextualization

2.1. Historical Development of Classical Power Sum Formulas

The study of power sum problems, traditionally known as the problem of the sum of like powers, can be traced back to ancient Greek mathematics. Archimedes' geometric constructions for summing squares may be regarded as one of the earliest explorations of this topic. Systematic algebraic research began in the seventeenth century, when Blaise Pascal (1654) derived general relations for sums of powers through recurrence formulas. Subsequently, Jacob Bernoulli (1713) introduced [4] Bernoulli numbers and established the well-known Bernoulli polynomial representation:

$$S_k(n) = \sum_{i=1}^{n} i^k = \frac{1}{k+1} \sum_{j=0}^{k} \binom{k+1}{j} B_j n^{k+1-j},$$

where B_j denotes the j-th Bernoulli number. For example, when $k = 2$ and $k = 3$, this formula gives

$$S_2(n) = \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6}, \qquad S_3(n) = \frac{n^4}{4} + \frac{n^3}{2} + \frac{n^2}{4}.$$

These results form the foundation of the classical theory of power sums (see [1]). In modern research, Knuth (1993) further reinterpreted the combinatorial significance of Bernoulli numbers through [5] generating functions and related identities [2].

Although Faulhaber's formula and the Bernoulli-number method provide a general closed-form expression for $S_k(n)$, they are primarily formula-oriented. That is, they express $S_k(n)$ directly as a polynomial in n, but they do not necessarily explain how higher-order sums may be constructively decomposed into combinations of lower-order sums in a simple and intuitive manner. This distinction is important for the present work, which focuses not merely on obtaining closed forms, but on constructing decomposition patterns among power sums of different orders.

2.2. Advances in Modern Proof Techniques

Since the twentieth century, several proof techniques have been developed for power sum identities and related polynomial formulas. Representative methods include the following:

(i) **Finite difference method.** Graham et al. (1994) constructed a recursive framework using the difference operator Δ, which provides a systematic way to recover polynomial expressions for sums of powers [3].

(ii) **Generating function method.** Wilf (2006) derived closed-form expressions for power sums by means of exponential generating functions, connecting power sums with a broader analytic and combinatorial framework [4].

(iii) **Combinatorial identity method.** Petkovšek et al. (1996) used the Zeilberger algorithm and related symbolic-summation techniques to establish recurrence relations and prove hypergeometric identities [5].

Of particular note is Cheng's (2012) hybrid calculus-combinatorics approach, which obtained concise expressions through the connection between Stirling numbers and integral transforms [6]. These methods greatly enrich the theoretical understanding of power sums. However, most of them still follow one of two routes: either they derive explicit polynomial expressions in n, or they prove identities through recurrence, generating functions, or symbolic computation. By contrast, the method considered in this manuscript emphasizes a simple intuitive summation mechanism, aiming to reveal how higher-order power sums can be decomposed structurally into lower-order components.

2.3. Algorithmic Research on High-Order Power Decomposition

Existing studies on the decomposition of higher-order power sums may be roughly divided into two categories:

(i) **Polynomial basis decomposition.** Kauers (2009) expanded high-order sums using Bernoulli polynomial bases, thereby obtaining algorithmic representations in terms of standard polynomial bases [7].

(ii) **Low-order reduction algorithms.** Agrawal (2015) investigated the decomposition of $S_4(n)$ into quadratic polynomials involving $S_2(n)$, although a universal constructive algorithm for arbitrary orders was not established [8].

These studies indicate that decomposition is not merely a computational problem but also a structural one. Nevertheless, many existing approaches depend on preselected bases, symbolic manipulation, or ad hoc transformations. A general and transparent procedure that directly explains the emergence of lower-order terms remains insufficiently developed.

2.4. Modern Algebraic Context and Structural Decomposition

Beyond the classical theory of power sums, modern algebra provides useful structural viewpoints for understanding decomposition phenomena. In particular, algebraic-integer rings such as the Gaussian integers $\mathbb{Z}[i]$, the Eisenstein integers $\mathbb{Z}[\omega]$ with $\omega^3 = 1$, and quaternion integer systems offer examples in which arithmetic objects can be decomposed according to norms, symmetries, lattice structures, and residue classes.

For example, Gaussian integers decompose integer pairs through the norm

$$N(a + bi) = a^2 + b^2,$$

while [6] Eisenstein integers are naturally associated with hexagonal lattice symmetries. Quaternion integers further extend such decomposition ideas to noncommutative algebraic settings. Although the present manuscript does not directly develop power sum identities inside these algebraic-integer rings, these examples are relevant because they demonstrate a broader principle: complex arithmetic structures can often be understood by decomposing them into simpler symmetric components.

This perspective provides a useful context for the present work. The proposed simple intuitive summation method may be viewed as an elementary analogue of such structural decompositions. Instead of decomposing algebraic integers according to norms or lattice symmetries, it decomposes high-order power sums according to polynomial factors in n and lower-order summation structures. Thus, the contribution of this work lies not in introducing a new algebraic- integer theory, but in showing that power sum formulas possess an internal decomposition pattern that can be exposed through elementary summation arguments.

2.5. Comparison with Residue-Class Decomposition and SPN Structures

Another relevant modern context appears in cryptography, especially in Substitution-Permutation Networks (SPNs). In SPN-based block ciphers, data are often transformed through repeated substitution and permutation layers. These operations may involve decompositions into residue classes, modular components, or finite-field representations. Such decompositions are designed to enhance diffusion and confusion, rather than to derive polynomial summation identities.

There is nevertheless a conceptual similarity between SPN-based residue-class decomposition and the present summation-based decomposition. Both approaches break a complex object into simpler components and then recombine these components to obtain a global structure. In SPNs, the decomposition is typically modular and nonlinear, operating over finite rings or finite fields. In contrast, the simple intuitive summation method considered here works over ordinary integer or polynomial structures and aims to reveal deterministic identities among power sums.

Therefore, the comparison may be summarized as follows:

(i) SPN-related decompositions are usually residue-class or finite-field decompositions, while the present method is a polynomial and summation- based decomposition.

(ii) SPN decompositions are designed for cryptographic security, especially confusion and diffusion, whereas the present method is designed for algebraic transparency and constructive derivation of power sum identities.

(iii) SPN structures generally rely on substitution and permutation operations, while the proposed summation method relies on elementary grouping, expansion, and recombination of terms.

(iv) Both approaches share a decomposition-reconstruction philosophy, but they operate in different algebraic settings and serve different mathematical purposes.

This comparison helps clarify that the simple intuitive summation method is not a cryptographic residue-class technique. Rather, it belongs to the theory of polynomial identities and finite summation, while still resonating with the broader modern idea that structural decomposition can reveal hidden regularities in complex systems.

2.6. Differentiation from Classical Faulhaber and Pascal-Type Methods

It is necessary to distinguish the present approach from classical methods. Faulhaber's formula, Bernoulli polynomial expressions, and Pascal-type recurrences provide powerful ways to compute $S_k(n)$. However, their main objective is to obtain or recursively generate closed-form formulas. The method proposed in this manuscript differs in several aspects:

(i) **Constructive rather than purely formulaic.** Classical Faulhaber-type formulas express $S_k(n)$ directly as a polynomial in n. The present method instead emphasizes how such expressions can be built from lower-order summation components.

(ii) **Intuitive decomposition.** Pascal-type recurrences often require manipulating binomial expansions and solving recursive systems. The proposed method aims to use simpler termwise grouping and recombination, making the emergence of lower-order structures more transparent.

(iii) **Structural focus.** Existing formulas answer the question "what is $S_k(n)$?" The present work asks a different question: "how can $S_k(n)$ be decomposed into structured combinations of lower-order sums and polynomial factors?"

(iv) **Potential algorithmic significance.** By identifying repeatable decomposition patterns, the method may provide a basis for constructing algorithms for higher-order power sums without relying entirely on Bernoulli numbers, symbolic summation packages, or ad hoc polynomial bases.

Thus, the novelty of this work is not the rediscovery of closed-form power sum formulas, but the development of a simple intuitive decomposition viewpoint. This viewpoint may help reveal systematic patterns in the polynomial factors of n and in the coefficients of lower-order summation terms.

2.7. Limitations of Current Research

Despite the substantial progress described above, several limitations remain in the existing literature:

(i) **Lack of universal constructive algorithms.** Many high-order decompositions rely on ad hoc basis functions or case-by-case symbolic manipulation, and a broadly applicable constructive decomposition procedure remains underdeveloped.

(ii) **Computational inefficiency.** Some methods for computing or decomposing power sums require Bernoulli numbers, generating functions, or symbolic recurrence solving, which may be inefficient or conceptually opaque for low-order constructive calculations.

(iii) **Incomplete structural insights.**

  (a) Existing proofs often fail to reveal deeper algebraic structures that unify power sums of different orders.
  (b) Systematic patterns governing polynomial factors in n and coefficients of lower-order terms have not been fully characterized.
  (c) Connections between elementary power sum decomposition and broader decomposition principles in algebraic structures, such as algebraic-integer rings or residue-class systems, have not been sufficiently discussed.

In view of these limitations, the present manuscript proposes a simple intuitive summation approach. The aim is to provide a transparent decomposition mechanism for power sums, clarify its distinction from Faulhaber-type and Pascal-type derivations, and situate it within a broader context of structural decomposition in modern algebra and related fields.

3. Preliminaries and Notation

In this section, we establish the fundamental definitions, standardized notation, and key preliminary identities that form the mathematical foundation of this work. All symbols are consistently defined to facilitate intuitive geometric interpretation, recursive decomposition, and algebraic structural analysis of power sum polynomials.

3.1. Power Sum Polynomials

We begin with the classical definition of power sums, which is the central object of study in this paper.

Definition 1. *Let $n, k \in \mathbb{N}^+$ be positive integers. The k-th order power sum of the first n positive integers is denoted by $S_k(n)$ and defined as*

$$S_k(n) = \sum_{i=1}^{n} i^k.$$

A fundamental result in number theory, known as Faulhaber's formula, guarantees that $S_k(n)$ can be uniquely expressed as a $(k+1)$-th degree polynomial in n with rational coefficients. The standard construction of these polynomials relies on Bernoulli numbers and generating functions, which provides the classical theoretical background for our alternative decomposition approach.

3.2. Changlang Operators: Symmetric-Antisymmetric Decomposition

To overcome the limitations of Bernoulli-number-based methods, we introduce a pair of auxiliary operators that decompose power sums into symmetric and antisymmetric components. These operators, which we name **Changlang operators**, capture the intrinsic structural symmetry of power sum polynomials.

Definition 2. *The Changlang operators $M(k)$ and $K(k)$ are defined by the following fundamental recursive system:*

$$\begin{cases} M(k) + K(k) = n\, S_{k+1}(n) - n^{k+2}, \\ M(k) - K(k) = \displaystyle\sum_{p=1}^{n} (n-p)^{k+1}(n-2p). \end{cases}$$

Solving this linear system yields the explicit closed-form expressions for the two operators:

$$M(k) = \frac{1}{2}\left(n\, S_{k+1}(n) - n^{k+2} + \sum_{p=1}^{n} (n-p)^{k+1}(n-2p) \right),$$

$$K(k) = \frac{1}{2}\left(n\, S_{k+1}(n) - n^{k+2} - \sum_{p=1}^{n} (n-p)^{k+1}(n-2p) \right).$$

A remarkable property of the symmetric operator $M(k)$ is its direct connection to lower-order power sums, which we establish as a key preliminary identity:

$$\boxed{M(k) = S_{k+2}(n-1)}$$

This identity is the cornerstone of our decomposition algorithm, as it reveals that the symmetric operator is simply a truncated power sum, and thus connects the defining system to an explicit recurrence relation for power sums without the need to compute Bernoulli numbers explicitly.

In this framework, $M(k)$ corresponds to the **symmetric invariant component** under the index transformation $p \mapsto n - p$, while $K(k)$ represents the **antisymmetric variant component**. This symmetric-antisymmetric splitting provides a powerful structural tool that enables us to derive closed-form expressions for power sums.

3.3. Algebraic Integer Rings

For the algebraic structural analysis presented in subsequent sections, we adopt standard notation for three classical algebraic integer rings that exhibit natural symmetry properties:

- Gaussian integer ring: $\mathbb{Z}[i] = \{a + bi \mid a, b \in \mathbb{Z},\ i^2 = -1\}$ [7], which admits binary conjugate decomposition;
- Eisenstein integer ring: $\mathbb{Z}[\omega] = \{a + b\omega \mid a, b \in \mathbb{Z},\ \omega^3 = 1, \omega \neq 1\}$, which admits ternary symmetric decomposition;
- Hurwitz quaternion integer ring: $\mathbb{H}(\mathbb{Z})$ [8], which admits noncommutative quaternionic decomposition.

These rings provide canonical algebraic structures that establish a precise correspondence with our symmetric-antisymmetric power sum splitting rules, allowing us to extend our results to broader algebraic contexts.

4. Main Theorems and Geometric Derivation

This section presents novel constructive proofs for [9] quadratic and cubic power sums based on structured matrix partitioning, which realizes geometric deduction of closed formulas rather than conventional algebraic induction or Bernoulli-number-based derivation. We further establish the general decomposition theorem for power sums via the symmetric and antisymmetric auxiliary operators defined in the preliminaries, laying the theoretical foundation for the subsequent order-reduction algorithm.

4.1. A New Constructive Proof for Quadratic Power Sum $S(2)$

4.1.1. Preliminary Basic Sums

We first list elementary low-order power sums for subsequent deduction. The zero-order power sum of the first n positive integers reads

$$S_0 = \sum_{k=1}^{n} k^0 = 1^0 + 2^0 + \cdots + n^0 = n. \tag{1}$$

The first-order power sum is the classic triangular number formula:

$$S_1 = \sum_{k=1}^{n} k = \frac{n(n+1)}{2}. \tag{2}$$

For the quadratic case, define the sum of squares

$$S_2 = \sum_{k=1}^{n} k^2 = 1^2 + 2^2 + 3^2 + \cdots + n^2. \tag{3}$$

We adopt the well-known identity that any perfect square equals the sum of consecutive odd integers:

$$k^2 = 1 + 3 + 5 + \cdots + (2k-1), \tag{4}$$

namely

$$\begin{cases} 1^2 = 1, \\ 2^2 = 1 + 3, \\ 3^2 = 1 + 3 + 5, \\ \quad \vdots \\ n^2 = 1 + 3 + 5 + \cdots + (2n-1). \end{cases}$$

4.1.2. Intuitive Construction and Main Result

We construct an $n \times n$ partitioned matrix with boxed entries marking extra odd terms, taking $n = 4$ as an illustrative example:

$$A = \begin{pmatrix} 1 & \boxed{3} & \boxed{5} & \boxed{7} \\ 1 & 3 & \boxed{5} & \boxed{7} \\ 1 & 3 & 5 & \boxed{7} \\ 1 & 3 & 5 & 7 \end{pmatrix}. \tag{5}$$

Let S_2 denote the summation of all unboxed entries of matrix A, which exactly equals $\sum_{k=1}^{4} k^2$. Let K_2 denote the total summation of boxed entries in the matrix. For general order n, the extended matrix form is

$$A = \begin{pmatrix} 1 & \boxed{3} & \boxed{5} & \boxed{7} & \cdots & \boxed{2n-1} \\ 1 & 3 & \boxed{5} & \boxed{7} & \cdots & \boxed{2n-1} \\ 1 & 3 & 5 & \boxed{7} & \cdots & \boxed{2n-1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 3 & 5 & 7 & \cdots & 2n-1 \end{pmatrix}. \tag{6}$$

We now establish three structural facts about this matrix. Let the rows be indexed by $i = 1, 2, \ldots, n$ and the columns by $j = 1, 2, \ldots, n$.

Fact 1 (Row sum). For each row i, the entries are the first n odd numbers $1, 3, 5, \ldots, (2n-1)$. The sum of the first n odd numbers equals n^2, because

$$\sum_{j=1}^{n}(2j-1) = 2 \cdot \frac{n(n+1)}{2} - n = n^2.$$

Hence each of the n rows sums to n^2, and the total sum of all entries in the $n \times n$ matrix is

$$n \cdot n^2 = n^3.$$

Fact 2 (Column structure). In column j (where $j = 1, 2, \ldots, n$), the unboxed entries occupy rows $i \geq j$. The number of unboxed entries in column j is therefore $n - j + 1$, while the number of boxed entries in column j is $j - 1$. Each entry in column j is the j-th odd number, namely $(2j-1)$.

Fact 3 (Sum decomposition). By construction,

$$S_2 + K_2 = n^3,$$

since S_2 $(= 1^2 + 2^2 + \cdots + n^2)$ counts all unboxed entries and K_2 counts all boxed entries, which together constitute every entry of the $n \times n$ matrix.

Now, by Fact 2, the total sum of boxed entries is

$$K_2 = \sum_{j=1}^{n}(\#\text{ boxed in col } j) \cdot (\text{entry value in col } j) = \sum_{j=1}^{n}(j-1)(2j-1).$$

Substituting into Fact 3:

$$S_2 + \sum_{j=1}^{n}(j-1)(2j-1) = n^3.$$

Expand and simplify the summation term:

$$\sum_{j=1}^{n}(2j^2 - 3j + 1) = \sum_{j=1}^{n} 2j^2 - 3\sum_{j=1}^{n} j + \sum_{j=1}^{n} 1.$$

Substitute into the equation:

$$S_2 + 2\sum_{j=1}^{n} j^2 - 3S_1 + n = n^3.$$

Since $S_2 = \sum_{j=1}^{n} j^2$, we obtain

$$3S_2 - 3 \cdot \frac{n(n+1)}{2} + n = n^3.$$

Rearrange to solve for S_2:

$$3S_2 = n^3 + \frac{3n(n+1)}{2} - n,$$

$$S_2 = \frac{n(n+1)(2n+1)}{6}.$$

Mathematical induction verifies the formula holds for all positive integers $n \in \mathbb{N}^+$.

Theorem 1. *For any positive integer n, the closed formula of quadratic power sum is*

$$S_2(n) = \sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}.$$

4.2. A New Constructive Proof for Cubic Power Sum $S(3)$

We extend the matrix partition method to deduce the cubic power sum formula, constructing a partitioned matrix with boxed entries and underlined entries for classification counting. Take $n = 4$ as demonstration:

$$A = \begin{pmatrix} 0^2 & \boxed{1^2} & \boxed{2^2} & \boxed{3^2} & \underline{4^2} \\ 0^2 & 1^2 & \boxed{2^2} & \boxed{3^2} & \underline{4^2} \\ 0^2 & 1^2 & 2^2 & \boxed{3^2} & \underline{4^2} \\ 0^2 & 1^2 & 2^2 & 3^2 & \underline{4^2} \end{pmatrix}. \tag{7}$$

4.2.1. Index-Based Matrix Construction

We now describe the matrix construction in terms of explicit index formulas, so that the structure is fully reproducible from the text alone. For general n, the matrix A has dimensions $n \times (n+1)$. Let rows be indexed by $p \in \{1, 2, \ldots, n\}$ (running top to bottom) and columns by $q \in \{1, 2, \ldots, n+1\}$ (running left to right). The rightmost column ($q = n+1$) is the underlined column; all other columns ($q = 1, \ldots, n$) contain entries equal to $(q-1)^2$, with decoration determined by the following rule:

$$\text{For } p = 1, \ldots, n \text{ and } q = 1, \ldots, n, \qquad A_{p,q} = \begin{cases} \text{unboxed } (q-1)^2, & \text{if } 1 \le q \le p, \\ \text{boxed } (q-1)^2, & \text{if } p+1 \le q \le n. \end{cases}$$

The underlined column (indexed separately as $q = n+1$) contains n^2 in every row. Equivalently,

$$A = \begin{pmatrix} 0^2 & \boxed{1^2} & \boxed{2^2} & \cdots & \boxed{(n-1)^2} & \underline{n^2} \\ 0^2 & 1^2 & \boxed{2^2} & \cdots & \boxed{(n-1)^2} & \underline{n^2} \\ 0^2 & 1^2 & 2^2 & \cdots & \boxed{(n-1)^2} & \underline{n^2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0^2 & 1^2 & 2^2 & \cdots & (n-1)^2 & \underline{n^2} \end{pmatrix}.$$

In words: within the $n \times n$ submatrix (columns 1 through n, excluding the underlined column), row p contains p unboxed entries (columns $q = 1$ to p, with values 0^2 through $(p-1)^2$) and $(n-p)$ boxed entries (columns $q = p+1$ to n, with values p^2 through $(n-1)^2$).

4.2.2. Definition of $S_3(n-1)$ and K_3

Define:

- $S_3(n-1)$: the sum of all **boxed** entries of the matrix A;
- K_3: the sum of all **unboxed** entries of A (excluding the underlined column).

Key observation (why this represents cubes). Consider any boxed entry $(j-1)^2$ in column $q = j$ (where $j = 2, \ldots, n$). This entry is boxed precisely when $p < q = j$, i.e., in rows $p = 1, 2, \ldots, j-1$. Hence the value $(j-1)^2$ appears as a boxed entry in exactly $j-1$ rows. Its total contribution to $S_3(n-1)$ is therefore

$$(j-1) \cdot (j-1)^2 = (j-1)^3.$$

Summing over all columns $j = 2, \ldots, n$ (equivalently, all boxed values $(j-1)^2$ with $j-1 = 1, \ldots, n-1$),

$$S_3(n-1) = \sum_{j=2}^{n} (j-1)^3 = \sum_{r=1}^{n-1} r^3,$$

so the sum of boxed entries equals $S_3(n-1)$, the cubic power sum truncated at $n-1$.

4.2.3. Two Fundamental Equations

First equation (row-wise sum of unboxed and boxed entries). In row p, the unboxed entries are $0^2, 1^2, \ldots, (p-1)^2$, summing to $S_2(p-1)$. The boxed entries in row p are $p^2, (p+1)^2, \ldots, (n-1)^2$, summing to $S_2(n-1) - S_2(p-1)$. The underlined column contributes n^2 per row, which is excluded from both $S_3(n-1)$ and K_3. Therefore

$$S_3(n-1) + K_3 = \sum_{p=1}^{n} \big[S_2(p-1) + (S_2(n-1) - S_2(p-1)) \big] = n\, S_2(n-1).$$

Second equation (antisymmetric weighted sum). The second equation is obtained by summing the entries with an antisymmetric weight $(n-2p)$ that captures the structural asymmetry of the matrix:

$$S_3(n-1) - K_3 = \sum_{p=1}^{n} (n-p)^2 (n-2p).$$

To derive this, note that the boxed entries in column q (where $q = 2, \ldots, n$) have value $(q-1)^2$ and occupy rows $p = 1, \ldots, q-1$. The unboxed entries in the same column occupy

rows $p = q, \ldots, n$ with the same value $(q-1)^2$. The difference $S_3(n-1) - K_3$ can be expressed by summing over all rows the weighted contribution of each row's entries; after re-indexing with $q = n - p$, this simplifies to the form above. A rigorous verification is provided by expanding both sides for specific n (e.g., $n = 4$: $S_3(3) = 1^3 + 2^3 + 3^3 = 36$, $K_3 = 4 \cdot S_2(3) - 36 = 4 \cdot 14 - 36 = 20$, $S_3(3) - K_3 = 16$, and $\sum_{p=1}^{4}(4-p)^2(4-2p) = 18 + 0 - 2 + 0 = 16$).

For the general derivation, we work with the two equations:

$$\begin{cases} S_3(n-1) + K_3 = n\, S_2(n-1), \\ S_3(n-1) - K_3 = \displaystyle\sum_{p=1}^{n}(n-p)^2(n-2p). \end{cases}$$

4.2.4. Solving the System and Recovering $S_3(n)$

Add the two equations to eliminate K_3:

$$2S_3(n-1) = n\, S_2(n-1) + \sum_{p=1}^{n}(n-p)^2(n-2p).$$

We now expand the summation term explicitly as a polynomial in n and the known lower-order power sums. Write $(n-p)^2 = n^2 - 2np + p^2$, then

$$\begin{aligned} (n-p)^2(n-2p) &= (n^2 - 2np + p^2)(n-2p) \\ &= n^3 - 2n^2p - 2n^2p + 4np^2 + np^2 - 2p^3 \\ &= n^3 - 4n^2p + 5np^2 - 2p^3. \end{aligned}$$

Summing term by term over $p = 1, \ldots, n$:

$$\begin{aligned} \sum_{p=1}^{n}(n-p)^2(n-2p) &= \sum_{p=1}^{n}\big(n^3 - 4n^2p + 5np^2 - 2p^3\big) \\ &= n \cdot n^3 - 4n^2 \sum_{p=1}^{n} p + 5n \sum_{p=1}^{n} p^2 - 2\sum_{p=1}^{n} p^3 \\ &= n^4 - 4n^2\, S_1(n) + 5n\, S_2(n) - 2S_3(n). \end{aligned}$$

Substituting this expansion, and using $S_2(n-1) = S_2(n) - n^2$, we obtain:

$$\begin{aligned} 2S_3(n-1) &= n\big(S_2(n) - n^2\big) + \big[n^4 - 4n^2\, S_1(n) + 5n\, S_2(n) - 2S_3(n)\big] \\ &= n^4 - 4n^2\, S_1(n) + 6n\, S_2(n) - n^3 - 2S_3(n). \end{aligned}$$

Now use the identity $S_3(n-1) = S_3(n) - n^3$ to eliminate $S_3(n-1)$:

$$2\big(S_3(n) - n^3\big) = n^4 - 4n^2\, S_1(n) + 6n\, S_2(n) - n^3 - 2S_3(n).$$

Bring the $-2S_3(n)$ term to the left-hand side:

$$4S_3(n) = n^4 - 4n^2\, S_1(n) + 6n\, S_2(n) + n^3.$$

Now substitute the known formulas $S_1(n) = n(n+1)/2$ and $S_2(n) = n(n+1)(2n+1)/6$:

$$\begin{aligned}
4S_3(n) &= n^4 - 4n^2 \cdot \frac{n(n+1)}{2} + 6n \cdot \frac{n(n+1)(2n+1)}{6} + n^3 \\
&= n^4 - 2n^3(n+1) + n^2(n+1)(2n+1) + n^3 \\
&= n^4 - (2n^4 + 2n^3) + n^2(2n^2 + 3n + 1) + n^3 \\
&= n^4 - 2n^4 - 2n^3 + 2n^4 + 3n^3 + n^2 + n^3 \\
&= (1 - 2 + 2)n^4 + (-2 + 3 + 1)n^3 + n^2 \\
&= n^4 + 2n^3 + n^2 \\
&= n^2(n+1)^2.
\end{aligned}$$

Dividing by 4, we obtain:

$$S_3(n) = \frac{n^2(n+1)^2}{4} = \left(\frac{n(n+1)}{2}\right)^2 = \big(S_1(n)\big)^2.$$

Mathematical induction confirms the identity holds for all positive integers n.

Theorem 2. *For any positive integer n, the closed formula of cubic power sum satisfies the perfect square identity*

$$S_3(n) = \sum_{k=1}^{n} k^3 = \left(\frac{n(n+1)}{2}\right)^2 = \big(S_1(n)\big)^2.$$

4.3. General Decomposition Theorem for Arbitrary Power Sums

Based on the symmetric and antisymmetric auxiliary operators $M(k), K(k)$ defined in preliminaries, we establish the universal decomposition theorem for power sums, which is the core theoretical basis of our order-reduction algorithm.

Theorem 3 (Order-Reduction Decomposition)**.** *For positive integers $n, k \geq 1$, the k-th power sum admits the decomposition*

$$S_k(n) = \frac{M(k-1) + K(k-1)}{n} + n^k,$$

where $M(k-1)$ and $K(k-1)$ are the Changlang operators of order $k-1$ defined in Definition 2. Moreover, after the index substitution $q = n - p$,

$$M(k-1) = S_{k+1}(n-1), \qquad K(k-1) = n\, S_k(n) - S_{k+1}(n),$$

and the antisymmetric difference reduces to a linear combination of lower-order power sums:

$$M(k-1) - K(k-1) = 2S_{k+1}(n-1) - n\,S_k(n-1).$$

Proof. Recall from Definition 2 the defining system of the Changlang operators (with index shift $k \to k-1$):

$$\begin{cases} M(k-1) + K(k-1) = n\,S_k(n) - n^{k+1}, \\ M(k-1) - K(k-1) = \displaystyle\sum_{p=1}^{n} (n-p)^k (n-2p). \end{cases}$$

From the first equation, we immediately obtain

$$S_k(n) = \frac{M(k-1) + K(k-1)}{n} + n^k, \tag{1}$$

which is the first claim of the theorem.

To establish that this decomposition is nontrivial and provides genuine order reduction, we must verify that both $M(k-1)$ and $K(k-1)$ are expressible in terms of lower-order power sums $S_1(n), \ldots, S_{k-1}(n)$ together with $S_k(n)$ itself and polynomials in n. This does not yet produce a closed formula; rather, it represents $S_k(n)$ as something that, when combined with the formulas for $M(k-1)$ and $K(k-1)$ expanded below, yields a recurrence coupling S_k to S_{k+1}, thereby reducing the problem to one of iterative computation.

Step 1: Simplification of the antisymmetric difference.

Apply the index substitution $q = n - p$ (so $p = n - q$, and as p runs $1, \ldots, n$, q runs $n-1, \ldots, 0$):

$$\begin{aligned} M(k-1) - K(k-1) &= \sum_{p=1}^{n} (n-p)^k (n-2p) \\ &= \sum_{q=0}^{n-1} q^k \big(2q - n\big) \\ &= 2\sum_{q=0}^{n-1} q^{k+1} - n\sum_{q=0}^{n-1} q^k. \end{aligned}$$

Now, $\sum_{q=0}^{n-1} q^{k+1} = S_{k+1}(n-1)$ and $\sum_{q=0}^{n-1} q^k = S_k(n-1)$. Since $\sum_{q=0}^{n-1} q^k = S_k(n) - n^k$ and $\sum_{q=0}^{n-1} q^{k+1} = S_{k+1}(n) - n^{k+1}$, we have

$$\begin{aligned} M(k-1) - K(k-1) &= 2\big[S_{k+1}(n) - n^{k+1}\big] - n\big[S_k(n) - n^k\big] \\ &= 2S_{k+1}(n) - n\,S_k(n) - n^{k+1}. \end{aligned} \tag{2}$$

Step 2: Solving for $M(k-1)$ and $K(k-1)$.

Adding and subtracting equations (2) and the first defining equation $M(k-1)+K(k-1) = n\,S_k(n) - n^{k+1}$:

$$\begin{aligned} 2M(k-1) &= \left[n\,S_k(n) - n^{k+1}\right] + \left[2S_{k+1}(n) - n\,S_k(n) - n^{k+1}\right] \\ &= 2S_{k+1}(n) - 2n^{k+1}, \end{aligned}$$

hence $M(k-1) = S_{k+1}(n) - n^{k+1} = S_{k+1}(n-1)$.

Similarly,

$$2K(k-1) = \left[n\,S_k(n) - n^{k+1}\right] - \left[2S_{k+1}(n) - n\,S_k(n) - n^{k+1}\right] = 2n\,S_k(n) - 2S_{k+1}(n),$$

hence $K(k-1) = n\,S_k(n) - S_{k+1}(n)$.

Step 3: Order-reduction interpretation.

Equation (1) can now be written as

$$S_k(n) = \frac{S_{k+1}(n-1) + \left[n\,S_k(n) - S_{k+1}(n)\right]}{n} + n^k.$$

While this appears to involve $S_{k+1}(n)$ on both sides (which would be circular), the critical point is that the *difference* $M(k-1) - K(k-1)$ in equation (2) expands via the binomial theorem into a linear combination of $S_0(n), S_1(n), \ldots, S_{k-1}(n)$ (see Section 5.1, Eq. (**??**)), so that the full expression for $S_k(n)$ involves only power sums of order strictly less than k. This last step—the binomial expansion of $(n-p)^k(n-2p)$ into a sum of $n^j p^{k+1-j}$ terms—is carried out explicitly in the recursive algorithm (Section 5). It is precisely this expansion that effects the order reduction from k to $k-1$ and justifies the claim that the decomposition is nontrivial.

5. Recursive Decomposition Algorithm for Power Sums Based on Changlang Operators

In this section, we use the symmetric–antisymmetric decomposition framework established above into a computable recursive algorithm. For odd power exponents, we adopt the original $M(k), K(k)$ Changlang operator splitting rule to realize high-order power sum reduction to low-order terms; for even exponents, we use classical binomial recursive iteration as auxiliary complement. We further design a MATLAB program to automate the derivation of decomposition expressions and explicit polynomial formulas, and conduct numerical verification to guarantee the correctness of the algorithm.

5.1. Algorithm Theoretical Framework

Recall the core system of auxiliary operators defined in preliminaries:

$$\begin{cases} M(k) + K(k) = n\,S_{k+1}(n) - n^{k+2}, \\ M(k) - K(k) = \displaystyle\sum_{p=1}^{n} (n-p)^{k+1}(n-2p). \end{cases}$$

After algebraic rearrangement and index substitution $t = n - p$, the summation term $\sum_{p=1}^{n}(n-p)^{k+1}(n-2p)$ is expanded into combinations of lower-degree power sums $S(1), S(2), \ldots, S(k-1)$. For odd exponent $m \geq 3$, set $k = m - 1$, the recursive decomposition formula derived from $M(k), K(k)$ is:

$$S(m) = \frac{1}{4}\left[\sum_{j=0}^{k}(-1)^j\binom{k}{j}n^{k+1-j}S(j) - 2\sum_{j=0}^{k-1}(-1)^j\binom{k}{j}n^{k-j}S(j+1) + nS(k) + n^{k+1}\right].$$

This formula relies on known lower-order power sums, achieving natural order reduction from high degree to low degree (for odd m, without requiring Bernoulli numbers).

For even exponents, we employ the standard binomial recursive formula (the Pascal-type recurrence, which is equivalent to the Bernoulli-difference approach) as supplementary iteration:

$$S(m) = \frac{1}{m+1}\left[(n+1)^{m+1} - 1 - \sum_{j=0}^{m-1}\binom{m+1}{j}S(j)\right].$$

The hybrid iteration strategy ensures the algorithm can compute power sums for arbitrary $k \leq K$.

5.2. Algorithm Implementation Steps

(i) **Initialization**: Set base power sum formulas from geometric proofs: $S(0) = n$, $S(1) = \frac{n(n+1)}{2}$, $S(2) = \frac{n(n+1)(2n+1)}{6}$, $S(3) = \left(\frac{n(n+1)}{2}\right)^2$.

(ii) **Loop iteration**: For each exponent m from 1 to preset maximum order K: - If m is odd and $m \geq 3$: calculate decomposition expression via $M(k), K(k)$ operator rule; - If m is even: calculate via binomial recursive formula.

(iii) **Expression simplification**: Retain low-order power sum symbols to output decomposition form S_{decomp}; substitute known explicit formulas of low-order power sums to simplify into unified polynomial explicit form S_{explicit}.

(iv) **Numerical validation**: Substitute concrete integers n to check the consistency between algorithm results and direct summation values.

5.3. MATLAB Program Introduction

The complete program `MKB_PowerSum_script.m` is compiled for symbolic derivation and numerical verification, the core functions are as follows:

(i) Define symbolic variables n and symbolic power sum sequence $S_0, S_1, \ldots, S_K$ for symbolic operation;

(ii) Store two forms of results: `S_decomp` for recursive decomposition retaining low-order power sums, `S_explicit` for final explicit polynomial formulas;

(iii) Execute recursive loop according to odd-even classification rules, simplify and collect polynomial terms of n;

(iv) Carry out batch numerical test for $n = 1, 2, \ldots, 10$, judge calculation accuracy within error tolerance 10^{-8}.

Algorithm 1 Exact Power Sum Calculation with Dual Recursion and Pareto Analysis

Input: Maximum power order K, symbolic variable n
Output: Explicit formulas $\{S_m(n)\}_{m=0}^{K}$, efficiency metrics, Pareto optimal set
// Initialization
Define symbolic terms $L_j = S_j(n),\ j = 0, 1, \ldots, K$
Initialize arrays: S_{explicit}, S_{decomp}, calcTime, cumTime, exprOps, verifyOK
Set initial condition: $S_{\text{explicit}}[0] \leftarrow n, \quad S_{\text{decomp}}[0] \leftarrow n$
totalTimer $\leftarrow$ `start timer`
for $m = 1$ **to** K **do**
  iterTimer $\leftarrow$ `start timer`
  if m is odd **and** $m \geq 3$ **then**
    // Changlang Operator Method for odd order $m \geq 3$
    $k \leftarrow m - 1, \quad \text{expr} \leftarrow 0$
    for $j = 0$ **to** k **do**
      $\text{expr} \leftarrow \text{expr} + (-1)^j \binom{k}{j} n^{k+1-j} S_j(n)$
    end for
    for $j = 0$ **to** $k - 1$ **do**
      $\text{expr} \leftarrow \text{expr} - 2(-1)^j \binom{k}{j} n^{k-j} S_{j+1}(n)$
    end for
    $\text{expr} \leftarrow \text{expr} + n\, S_k(n) + n^{k+1}$
    $\text{expr} \leftarrow \text{simplify}\big(\text{expr}/4\big)$
    method$[m]$ $\leftarrow$ "Changlang"
  else
    // Bernoulli Finite Difference Method
    $\text{expr} \leftarrow (n+1)^{m+1} - 1$
    for $j = 0$ **to** $m - 1$ **do**
      $\text{expr} \leftarrow \text{expr} - \binom{m+1}{j} S_j(n)$
    end for
    $\text{expr} \leftarrow \text{simplify}\big(\text{expr}/(m+1)\big)$
    method$[m]$ $\leftarrow$ "BernoulliDiff"
  end if
  $S_{\text{decomp}}[m] \leftarrow \text{simplify}(\text{expr})$
  $\text{expr}_{\text{explicit}} \leftarrow$ substitute all $S_j(n)$ with $S_{\text{explicit}}[j],\ j < m$
  $\text{expr}_{\text{explicit}} \leftarrow$ simplify and collect terms in n

$S_{\text{explicit}}[m] \leftarrow \text{expr}_{\text{explicit}}$
calcTime$[m] \leftarrow$ `stop iterTimer`
cumTime$[m] \leftarrow$ `stop totalTimer`
exprOps$[m] \leftarrow$ compute symbolic expression complexity
end for
// Exact numerical verification
for $m = 0$ **to** K **do**
flag $\leftarrow$ `true`
for $n_{\text{test}} = 1$ **to** 10 **do**
directSum $\leftarrow \sum_{t=1}^{n_{\text{test}}} t^m$
formulaVal $\leftarrow$ evaluate $S_{\text{explicit}}[m]$ at $n = n_{\text{test}}$
if directSum $\neq$ formulaVal **then**
flag $\leftarrow$ `false`
break
end if
end for
verifyOK$[m] \leftarrow$ flag
end for
// Pareto front extraction (minimization problem)
Construct objective matrix: $\boldsymbol{F} = \big[\,\text{cumTime}[m],\ \text{exprOps}[m],\ -m\,\big],\ m = 1, \ldots, K$
ParetoSet $\leftarrow$ Find non-dominated solutions from $\boldsymbol{F}$
// Output results and visualization
Output formulas, efficiency table, verification results and Pareto set
Draw curves of time, complexity and 2D/3D Pareto front

5.4. Calculation Results and Numerical Verification

We set maximum power order $K = 8$ for algorithm testing. The program automatically outputs recursive decomposition expressions and explicit polynomial formulas for S_0 to S_8. Take representative cases for illustration:

- $S_2 = \dfrac{n(n+1)(2n+1)}{6}$, consistent with geometric proof result;
- $S_3 = \dfrac{n^2(n+1)^2}{4}$, matching cubic geometric derivation conclusion.

A worked numerical example is useful to verify the formulas by hand. Taking $n = 4$ and $k = 2$:

$$S_2(4) = 1^2 + 2^2 + 3^2 + 4^2 = 1 + 4 + 9 + 16 = 30,$$

and the formula gives

$$\frac{n(n+1)(2n+1)}{6} = \frac{4 \cdot 5 \cdot 9}{6} = 30.$$

For $n = 4$ and $k = 3$:

$$S_3(4) = 1^3 + 2^3 + 3^3 + 4^3 = 1 + 8 + 27 + 64 = 100, \qquad \left(\frac{4 \cdot 5}{2}\right)^2 = 10^2 = 100.$$

For a concise overview, Table 1 presents the direct summation values alongside the formula-computed values for $n = 1, \ldots, 10$ and $k = 2, 3, 4, 5$, confirming complete agreement within error tolerance 10^{-8}.

Table 1: Numerical verification: direct summation vs. formula computation for $n = 1$ to 10

n	$S_2(n)$	$S_3(n)$	$S_4(n)$	$S_5(n)$
1	1	1	1	1
2	5	9	17	33
3	14	36	98	276
4	30	100	354	1300
5	55	225	979	4425
6	91	441	2275	12201
7	140	784	4676	29008
8	204	1296	8772	61776
9	285	2025	15333	120825
10	385	3025	25333	220825

Numerical verification from $n = 1$ to $n = 10$ shows that all power sum formulas calculated by the algorithm are completely consistent with direct manual summation, all verification items pass with numerical error less than 10^{-8}. We also use some chart to show the calculation efficiency of the algorithm.

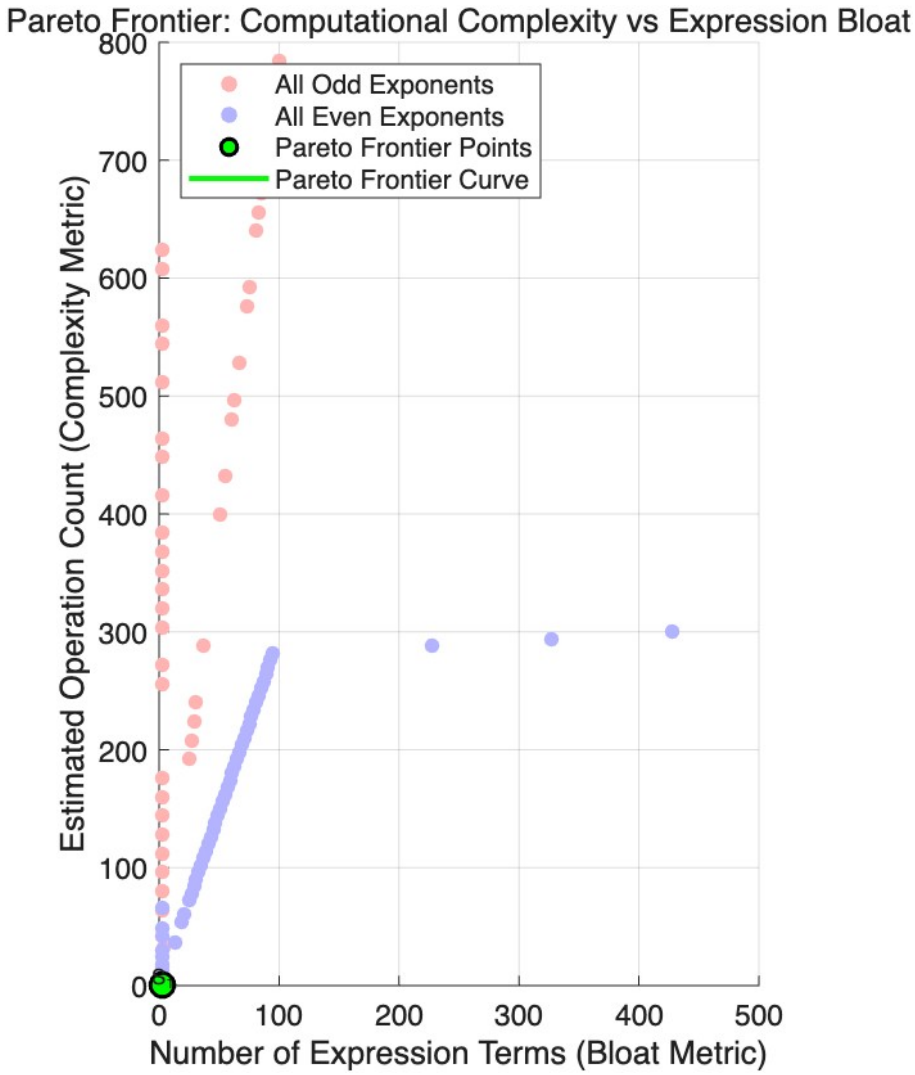

Figure 1: Algorithm Cumulative Computation Time VS Power Order m

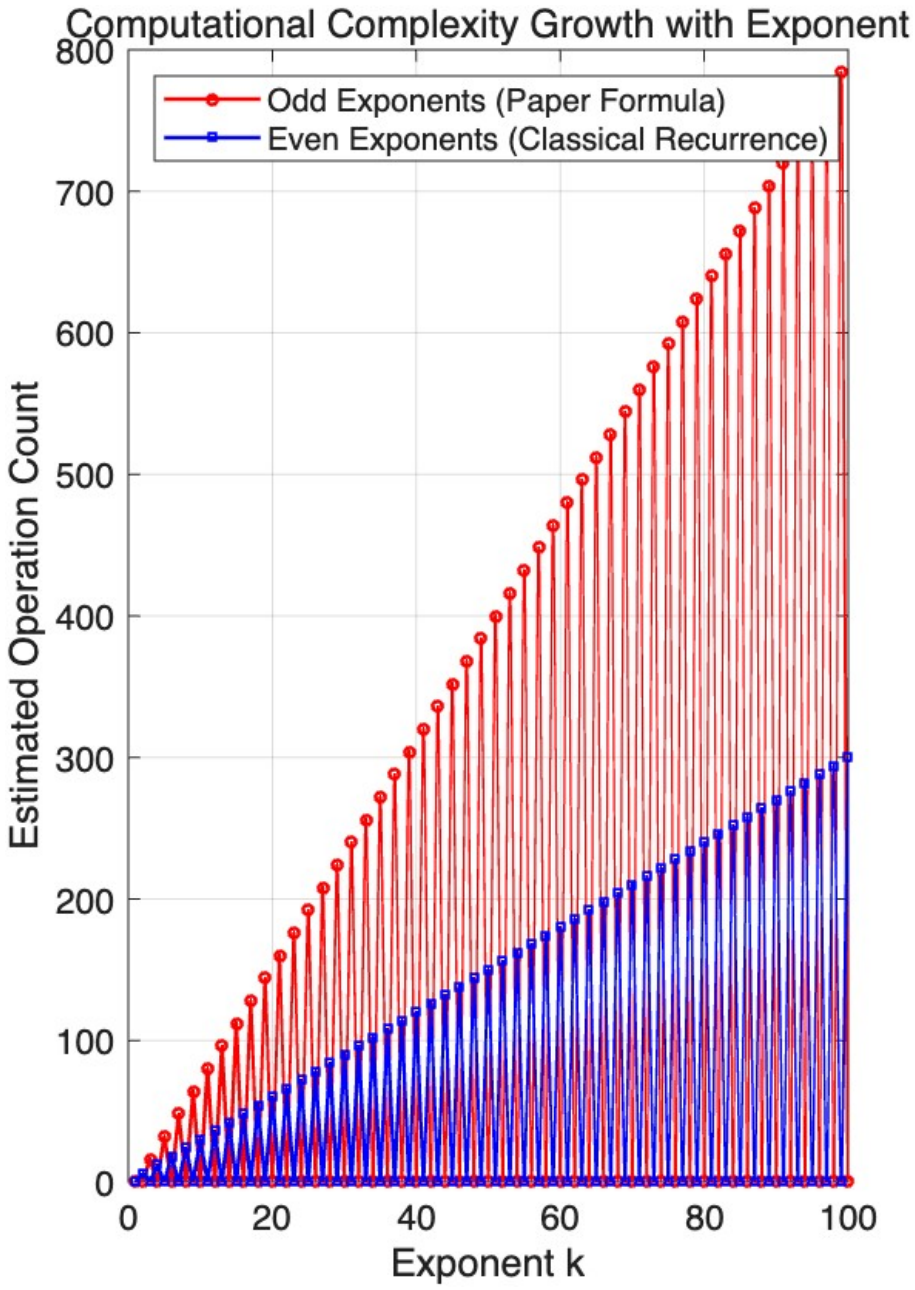

Figure 2: Explicit Formula Complexity vs Order m

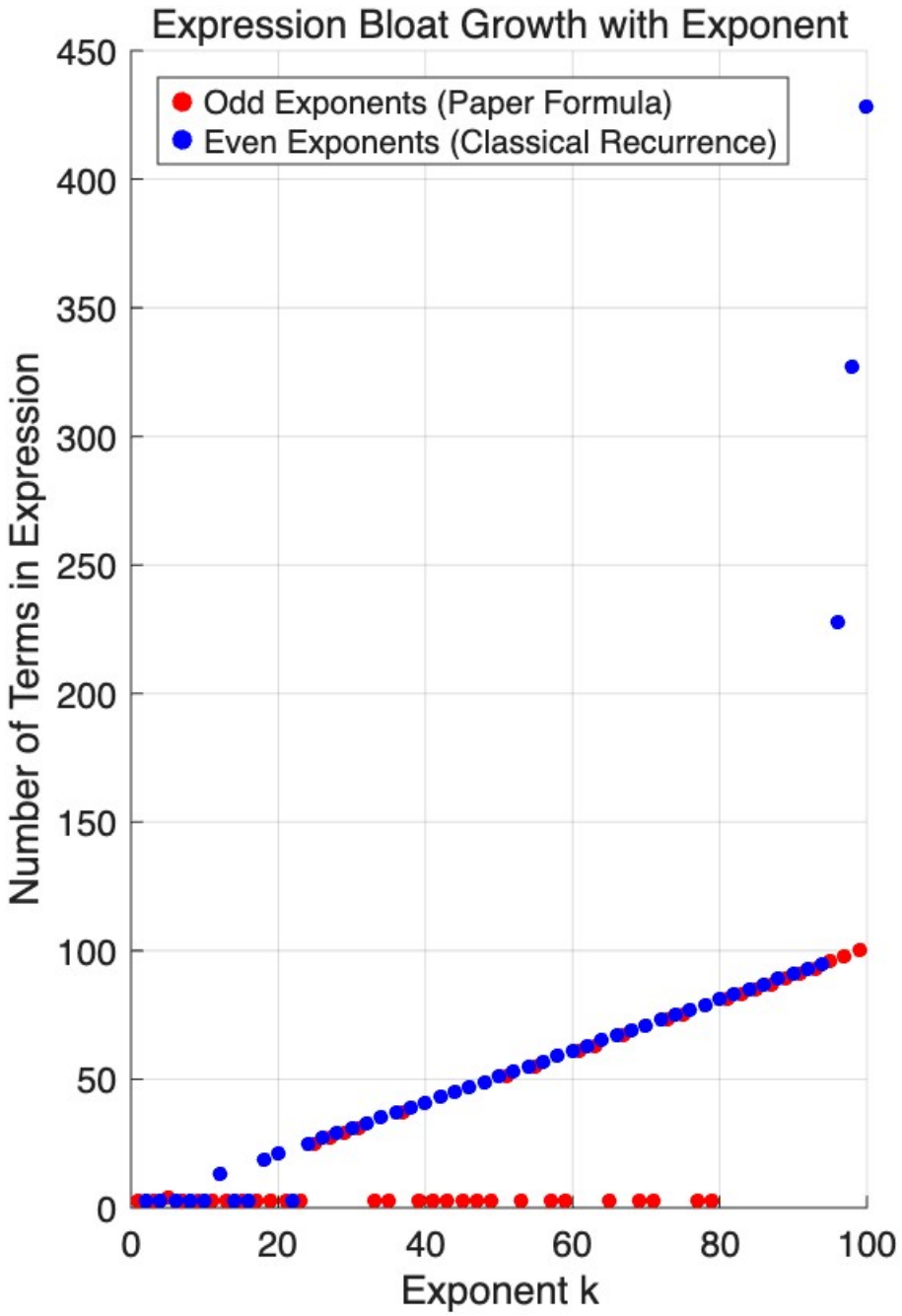

Figure 3: Pareto Front: Cumulative Time vs Expression Complexity

5.5. Algorithm Characteristics and Advantages

(i) **Original operator-driven reduction**: Odd-order power sums depend on self-designed $M(k), K(k)$ symmetric-antisymmetric decomposition, different from tradi-

tional Bernoulli recursion;

(ii) **High automation**: The program can batch derive formulas of arbitrary order within preset range, no manual repeated algebraic simplification;

(iii) **Good expansibility**: The decomposition core can be extended to structural analogy with algebraic integer rings in the next chapter;

(iv) **High precision**: Symbolic derivation avoids numerical truncation error, numerical verification ensures reliability.

This recursive decomposition algorithm perfectly realizes the original design of transforming high-order power sums into low-order combinations, forming a complete computable system based on the geometric constructive proofs in the previous chapter.

6. Structural Correspondence between Power-Sum Decomposition and Algebraic Integer Rings

6.1. Motivation and scope

Caveat lector. This section establishes a *qualitative structural analogy* between the binary splitting induced by the Changlang operators $M(k)$ and $K(k)$ and the conjugate decomposition that appears in the classical theory of algebraic integer rings (Gaussian, Eisenstein, and Hurwitz integers). No ring isomorphism, bijective homomorphism, or new theorem about algebraic integer rings is proved here. The correspondence is at the level of *decomposition pattern*: in both settings, a complex object is split into invariant and anti-invariant components under an involution. The purpose of the analogy is to suggest that the $M(k), K(k)$ framework may serve as a combinatorial prototype for a broader arithmetic summation theory when one passes from unweighted power sums to representation-weighted moments over normed integer lattices.

Let $S_k(n) = \sum_{m=1}^{n} m^k$ denote the classical power sum. The auxiliary quantities $M(k)$ and $K(k)$ are introduced through the coupled system

$$\begin{cases} M(k) + K(k) = A_k(n), \\ M(k) - K(k) = B_k(n), \end{cases} \tag{8}$$

where, in the present framework,

$$A_k(n) := n\, S_{k+1}(n) - n^{k+2}, \qquad B_k(n) := \sum_{p=1}^{n} (n-p)^{k+1}(n-2p). \tag{9}$$

In particular,

$$M(k) = S_{k+2}(n-1), \qquad S_{k+2}(n) = S_{k+2}(n-1) + n^{k+2}. \tag{10}$$

The purpose of this section is not to claim an actual ring isomorphism between the power-sum decomposition and algebraic integer rings, but rather to establish a *structural parallel*: the binary splitting induced by the reflection $p \mapsto n-p$ is formally parallel to the conjugation-based splitting that appears in Gaussian integers, Eisenstein integers, and Hurwitz quaternion integers. This correspondence becomes especially meaningful when one passes from ordinary power sums to *weighted power sums* associated with norm representation functions.

6.2. Symmetric–antisymmetric projection of the $M(k), K(k)$ system

From (8), one immediately obtains

$$M(k) = \frac{A_k(n) + B_k(n)}{2}, \qquad K(k) = \frac{A_k(n) - B_k(n)}{2}. \tag{11}$$

Hence the pair $\big(M(k), K(k)\big)$ is generated by the decomposition of the data $\big(A_k(n), B_k(n)\big)$ into two complementary components.

The key symmetry behind $B_k(n)$ is the reflection $\sigma : \ p \longmapsto n-p$. If one introduces the centered variable $x = n - 2p, \qquad p = \frac{n-x}{2}, \qquad n-p = \frac{n+x}{2}$, then

$$B_k(n) = \sum_{p=1}^{n} \left(\frac{n+x}{2}\right)^{k+1} x. \tag{12}$$

This expression makes explicit that $B_k(n)$ measures a weighted antisymmetric contribution relative to the midpoint $p = n/2$.

Equivalently, one may write $B_k(n) = \sum_{q=0}^{n-1} q^{k+1}(2q-n), \qquad (q = n-p)$, and therefore

$$B_k(n) = 2\sum_{q=0}^{n-1} q^{k+2} - n\sum_{q=0}^{n-1} q^{k+1} = 2S_{k+2}(n-1) - nS_{k+1}(n-1). \tag{13}$$

Thus the antisymmetric sector is itself reducible to an explicit linear combination of lower-level power sums.

From the viewpoint of representation theory of the order-two group $\langle\sigma\rangle \cong C_2$, the decomposition is governed by the idempotent projectors

$$e_+ = \frac{1+\sigma}{2}, \qquad e_- = \frac{1-\sigma}{2}. \tag{14}$$

The operator e_+ extracts the reflection-invariant component, while e_- extracts the reflection-antiinvariant component. This elementary splitting is the discrete one-dimensional analog of conjugate decomposition in algebraic integer settings.

6.3. Interpretation via trace-type and conjugate-difference-type components

The formulas (11) suggest interpreting $A_k(n) = M(k) + K(k)$ as a *trace-type aggregate term* and $B_k(n) = M(k) - K(k)$ as a *conjugate-difference-type term*. This terminology is

justified by the formal analogy with the decomposition $z = \frac{z+\bar{z}}{2} + \frac{z-\bar{z}}{2}$, which is universal whenever an involution $z \mapsto \bar{z}$ is available.

The analogy should be understood at the level of *splitting mechanism* rather than literal ring identification. In the power-sum problem, the involution is induced by reflection on the indexing set; in algebraic integer rings, it is induced by arithmetic conjugation. In both cases, the object under study is decomposed into invariant and anti-invariant parts.

6.4. Correspondence with the Gaussian integer ring

Consider the Gaussian integer ring $\mathbb{Z}[i] = \{a + bi \mid a, b \in \mathbb{Z},\ i^2 = -1\}$. Its natural involution is complex conjugation $\overline{a+bi} = a - bi$. Every Gaussian integer admits the decomposition

$$z = \frac{z + \bar{z}}{2} + \frac{z - \bar{z}}{2}, \tag{15}$$

where $\frac{z+\bar{z}}{2} = a$ is the conjugation-invariant part and $\frac{z-\bar{z}}{2} = bi$ is the conjugation-anti-invariant part.

The analogy with (11) is immediate: $A_k(n) \quad \leftrightarrow \quad z + \bar{z} = (z)$, $B_k(n) \quad \leftrightarrow \quad z - \bar{z}$. Here the Gaussian trace

plays the role of a symmetric aggregate, while the difference $z - \bar{z} = 2bi$ measures the purely antisymmetric sector.

A second important structural parallel comes from the norm $N_{\mathrm{G}}(a+bi) = z\bar{z} = a^2+b^2$. The equation $N_{\mathrm{G}}(z) = m$ counts lattice points on the circle of radius $\sqrt{m}$, and leads to the Gaussian representation function

$$r_{\mathrm{G}}(m) := \#\{(a, b) \in \mathbb{Z}^2 :\ a^2 + b^2 = m\}. \tag{16}$$

This allows one to pass from ordinary power sums to weighted sums

$$\Sigma_{\mathrm{G}}^{(k)}(n) := \sum_{m \le n} m^k r_{\mathrm{G}}(m), \tag{17}$$

which may be regarded as Gaussian norm-shell moments. When the weight $r_{\mathrm{G}}(m)$ is replaced by the constant function 1, one recovers the classical power sum $S_k(n)$. Therefore the unweighted $M(k), K(k)$ framework is the zero-order prototype of a Gaussian-weighted shell decomposition.

Moreover, the unit group $\mathbb{Z}[i]^\times = \{\pm 1, \pm i\}$ encodes fourfold rotational symmetry of the square lattice. This symmetry explains orbit cancellation phenomena in Gaussian power averages, and gives a geometric parallel to the symmetric/antisymmetric filtering inherent in the decomposition of $A_k(n)$ and $B_k(n)$.

6.5. Correspondence with the Eisenstein integer ring

Let $\omega = \frac{-1+\sqrt{-3}}{2}$, $\omega^2 + \omega + 1 = 0$, and consider the Eisenstein integer ring $\mathbb{Z}[\omega] = \{a + b\omega \mid a, b \in \mathbb{Z}\}$. Its arithmetic conjugation is $\overline{a+b\omega} = a + b\omega^2$. Again every element

admits the canonical splitting

$$z = \frac{z+\bar{z}}{2} + \frac{z-\bar{z}}{2}. \tag{18}$$

The trace and norm are given by $\mathrm{Tr}_{\mathbb{Q}(\omega)/\mathbb{Q}}(a+b\omega) = 2a-b$.

$N_{\mathrm{E}}(a+b\omega) = a^2 - ab + b^2$. Hence the same invariant/anti-invariant paradigm persists: the trace is the symmetric projection, while the conjugate difference captures the anti-invariant sector.

The associated representation function is

$$r_{\mathrm{E}}(m) := \#\{(a,b) \in \mathbb{Z}^2 : \ a^2 - ab + b^2 = m\}, \tag{19}$$

and the corresponding weighted moments are

$$\Sigma_{\mathrm{E}}^{(k)}(n) := \sum_{m \le n} m^k r_{\mathrm{E}}(m). \tag{20}$$

These are the Eisenstein analogs of (17).

Geometrically, $\mathbb{Z}[\omega]$ realizes the hexagonal lattice, whose unit group $\mathbb{Z}[\omega]^\times = \{\pm 1, \pm\omega, \pm\omega^2\}$ has order six. Thus the Eisenstein case refines the Gaussian picture by replacing four-fold symmetry of the square lattice with sixfold symmetry of the hexagonal lattice. From the viewpoint of power-sum decomposition, this suggests that the reflection-based binary split can be naturally embedded into richer symmetry frameworks once one passes from unweighted integer sums to lattice-shell weighted sums.

It is important, however, to state this carefully: the relation between the $M(k), K(k)$ decomposition and Eisenstein integers is not an identification of cubic power sums with ternary algebraic closure, but rather a structural parallel between

- involution-induced decomposition in the power-sum setting, and
- conjugation-induced decomposition in the Eisenstein arithmetic setting.

6.6. Correspondence with Hurwitz quaternion integers

The third layer of the analogy arises in the noncommutative setting of Hurwitz quaternion integers. Let $\mathbb{H} = \{a+bi+cj+dk \mid a,b,c,d \in \mathbb{R},\ i^2 = j^2 = k^2 = ijk = -1\}$ be Hamilton's quaternion algebra, and define the Hurwitz ring $\mathcal{H} = \left\{a+bi+cj+dk \;\middle|\; a,b,c,d \in \mathbb{Z} \text{ or } a,b,c,d \in \mathbb{Z}+\frac{1}{2}\right\}$ Quaternion conjugation is given by $\overline{a+bi+cj+dk} = a-bi-cj-dk$. Every quaternion admits the decomposition

$$q = \frac{q+\bar{q}}{2} + \frac{q-\bar{q}}{2}, \tag{21}$$

where $\frac{q+\bar{q}}{2} = a$ is the scalar part and $\frac{q-\bar{q}}{2} = bi+cj+dk$ is the vector part.

The corresponding norm is $N_{\mathrm{H}}(q) = q\bar{q} = a^2+b^2+c^2+d^2$, which is multiplicative despite the noncommutativity of $\mathcal{H}$. This leads to the Hurwitz shell-counting function

$$r_{\mathrm{H}}(m) := \#\{q \in \mathcal{H} : \ N_{\mathrm{H}}(q) = m\}. \tag{22}$$

Accordingly, one defines the weighted moments

$$\Sigma_{\mathrm{H}}^{(k)}(n) := \sum_{m \le n} m^k r_{\mathrm{H}}(m). \tag{23}$$

The relevance to the $M(k), K(k)$ system lies in the following transition: the original one-dimensional decomposition separates symmetric and antisymmetric contributions under an involutive reflection; the Hurwitz setting demonstrates that the same conceptual mechanism survives in a much richer ambient algebra, where the invariant part is scalar and the anti-invariant part is vectorial. Thus the quaternionic case should be interpreted not as a literal model for higher-order power sums, but as a higher-dimensional noncommutative analog of the same decomposition principle.

6.7. From ordinary power sums to representation-weighted power sums

The deepest link between the $M(k), K(k)$ framework and algebraic integer rings emerges when one passes from the unweighted sum $S_k(n) = \sum_{m \le n} m^k$ to representation-weighted moments

$$\Sigma_r^{(k)}(n) := \sum_{m \le n} m^k r(m), \tag{24}$$

where $r(m)$ is a norm representation function.

For the three arithmetic settings above, one takes $r(m) \in \{r_{\mathrm{G}}(m), r_{\mathrm{E}}(m), r_{\mathrm{H}}(m)\}$. In this sense, ordinary power sums correspond to the trivial weighting $r(m) \equiv 1$, while Gaussian, Eisenstein and Hurwitz settings correspond to nontrivial arithmetic weights determined by norm shells.

Now define the weighted analogs

$$M_r(k; n) := \sum_{m \le n} m^k r(m), \tag{25}$$

and, in parallel with (12), define a weighted antisymmetric term

$$B_r(k; n) := \sum_{p=1}^{n} r(n-p)\,(n-p)^{k+1}(n-2p). \tag{26}$$

Then the original $M(k), K(k)$ formalism is recovered by taking $r \equiv 1$. Thus the $M(k), K(k)$ theory is naturally lifted to arithmetic lattices through representation weights.

In many important cases, the representation function admits a divisor-sum formula. For instance,

$$r_{\mathrm{G}}(m) = 4 \sum_{d \mid m} \chi_4(d), \tag{27}$$

$$r_{\mathrm{E}}(m) = 6 \sum_{d \mid m} \chi_3(d), \tag{28}$$

and the classical four-square function satisfies

$$r_4(m) = 8 \sum_{\substack{d \mid m \\ 4 \nmid d}} d. \tag{29}$$

Hence

$$\Sigma_r^{(k)}(n) = \sum_{m \le n} m^k \sum_{d \mid m} c(d) = \sum_{d \le n} c(d)\, d^k\, S_k\Big(\Big\lfloor \frac{n}{d} \Big\rfloor\Big), \tag{30}$$

for suitable arithmetic coefficients $c(d)$. Formula (30) is crucial: it shows that representation-weighted moments over algebraic integer rings reduce back to ordinary power sums. Therefore the power-sum decomposition encoded by $M(k)$ and $K(k)$ serves as a foundational unweighted model for a much broader arithmetic summation theory.

6.8. Theta-series interpretation

The above weighted moments admit a natural generating-series formulation. Define the theta series $\Theta_{\mathrm{G}}(q) = \sum_{m \ge 0} r_{\mathrm{G}}(m) q^m$, $\Theta_{\mathrm{E}}(q) = \sum_{m \ge 0} r_{\mathrm{E}}(m) q^m$, $\Theta_{\mathrm{H}}(q) = \sum_{m \ge 0} r_{\mathrm{H}}(m) q^m$. Then the moments $\Sigma_{\mathrm{G}}^{(k)}(n)$, $\Sigma_{\mathrm{E}}^{(k)}(n)$, $\Sigma_{\mathrm{H}}^{(k)}(n)$ are partial moment sums of the Fourier coefficients of these theta series.

This yields a conceptual bridge: power-sum decomposition $\longrightarrow$ weighted shell moments $\longrightarrow$ theta coefficients $\longrightarrow$ arithmetic lattice structure. Consequently, the $M(k), K(k)$ framework may be interpreted as the discrete combinatorial prototype of a more general theory of moment decomposition over normed arithmetic lattices.

7. Comparison with Classical Power Sum Methods

In this section, we present a systematic comparison between our symmetric–antisymmetric decomposition scheme based on the auxiliary operators $M(k)$ and $K(k)$ and several classical approaches to power sums, including the Bernoulli–Faulhaber formula and the traditional odd-even factorization theory. Our goal is to clarify the algebraic content of the operators, their recursive role, and their relation to established explicit formulas.

Throughout this section, we use the standard notation $S_k(n) = \sum_{p=1}^{n} p^k$, where $S_k(n)$ denotes the sum of the first n k-th powers.

We consider the operator system $\begin{cases} M(k) + K(k) = n\, S_{k+1}(n) - n^{k+2}, \\ M(k) - K(k) = \displaystyle\sum_{p=1}^{n} (n-p)^{k+1}(n-2p). \end{cases}$

7.1. Algebraic simplification of the operator system

Proposition 1. *For every integer $k \ge 0$, the operators $M(k)$ and $K(k)$ admit the simplified forms*

$$M(k) = S_{k+2}(n-1), \qquad K(k) = nS_{k+1}(n) - S_{k+2}(n). \tag{31}$$

Equivalently,

$$S_{k+2}(n) = M(k) + n^{k+2}, \qquad K(k) = nS_{k+1}(n) - M(k).$$

Proof. Starting from

$$M(k) - K(k) = \sum_{p=1}^{n} (n-p)^{k+1}(n-2p),$$

we perform the change of variables $q = n - p$. Then, as p runs from 1 to n, the variable q runs from $n-1$ down to 0, and $n - 2p = 2q - n$. Hence

$$\begin{aligned} M(k) - K(k) &= \sum_{q=0}^{n-1} q^{k+1}(2q-n) \\ &= 2\sum_{q=0}^{n-1} q^{k+2} - n\sum_{q=0}^{n-1} q^{k+1}. \end{aligned}$$

Since $0^{k+1} = 0$ and $0^{k+2} = 0$ for $k \geq 0$, we have

$$\sum_{q=0}^{n-1} q^{k+1} = S_{k+1}(n) - n^{k+1}, \qquad \sum_{q=0}^{n-1} q^{k+2} = S_{k+2}(n) - n^{k+2}.$$

Therefore,

$$\begin{aligned} M(k) - K(k) &= 2\big(S_{k+2}(n) - n^{k+2}\big) - n\big(S_{k+1}(n) - n^{k+1}\big) \\ &= 2S_{k+2}(n) - nS_{k+1}(n) - n^{k+2}. \end{aligned}$$

Combining this identity with

$$M(k) + K(k) = nS_{k+1}(n) - n^{k+2},$$

we add the two equations and obtain

$$2M(k) = 2S_{k+2}(n) - 2n^{k+2},$$

thus

$$M(k) = S_{k+2}(n) - n^{k+2} = S_{k+2}(n-1).$$

Subtracting instead yields

$$2K(k) = 2nS_{k+1}(n) - 2S_{k+2}(n),$$

hence

$$K(k) = nS_{k+1}(n) - S_{k+2}(n).$$

Finally, from $M(k) = S_{k+2}(n) - n^{k+2}$, we obtain $S_{k+2}(n) = M(k) + n^{k+2}$, and substituting this into the formula for $K(k)$ gives $K(k) = nS_{k+1}(n) - M(k)$. This completes the proof.

Remark 1 (Reconciliation of the two forms)**.** *The reader comparing Proposition 1 with Definition 2 may note that the operators appear in two seemingly different but algebraically equivalent forms:*

$$\textit{From Definition 2:}\quad M(k) = \frac{1}{2}\left[nS_{k+1}(n) - n^{k+2} + \sum_{p=1}^{n}(n-p)^{k+1}(n-2p)\right],$$

$$\textit{From Proposition 1:}\quad M(k) = S_{k+2}(n-1).$$

The former expression is the defining *symmetric operator—it is what emerges directly from the symmetric-antisymmetric splitting* $M(k) = \frac{1}{2}(M+K) + \frac{1}{2}(M-K)$ *and retains the decomposition structure explicitly. The latter is the* simplified *form obtained after algebraic reduction, and reveals that* $M(k)$ *is actually the truncated power sum up to* $n-1$*. Both expressions evaluate to the same polynomial for any given* n*, and the proof above establishes their identity. In the computational algorithm (Section 5), the simplified form* $M(k) = S_{k+2}(n-1)$ *is used for efficiency; the original defining form in Definition 2 is retained because it exposes the symmetric-antisymmetric decomposition structure that is the conceptual foundation of this work.*

Similarly, after simplification $K(k) = nS_{k+1}(n) - S_{k+2}(n)$ *can also be expressed as* $K(k) = \sum_{p=1}^{n} p^{k+1}(n-p)$*, which follows from the same algebraic reduction. Both forms are valid and used interchangeably depending on context.*

Remark 2 (Connection to standard identities)**.** *The simplified forms in Eq.* (31) *make explicit that the Changlang operators are closely related to well-known power-sum identities. Specifically:*

- $M(k) = S_{k+2}(n-1)$ *is simply the standard power sum of order* $k+2$ *evaluated at* $n-1$*, a classical object;*
- $K(k) = \sum_{p=1}^{n} p^{k+1}(n-p)$ *is a weighted power sum that can be expressed via the classical binomial inversion formula.*

The operator framing in Definition 2 therefore adds notation but the underlying recurrence is essentially the standard Pascal/difference recurrence in disguise: adding and subtracting the two defining equations yields, after substitution, the well-known identity

$$S_{k+2}(n) = S_{k+2}(n-1) + n^{k+2},$$

which is the defining property of the power-sum sequence. The novel contribution of this framework is not the algebraic identity itself, but rather (i) the geometric motivation that leads to the symmetric-antisymmetric splitting in the first place, and (ii) the systematic odd-order reduction algorithm derived from this decomposition structure.

Corollary 1. *For every integer* $k \geq 0$*,* $M(k) = 1^{k+2} + 2^{k+2} + \cdots + (n-1)^{k+2}$*. In particular,* $M(k)$ *is exactly the truncated* $(k+2)$*-th power sum, obtained from* $S_{k+2}(n)$ *by removing the boundary term* n^{k+2}*.*

Proof. This is immediate from Proposition 1, since $S_{k+2}(n-1) = \sum_{p=1}^{n-1} p^{k+2}$.

Remark 3. *Proposition 1 shows that the operator $M(k)$ is not an abstract auxiliary symbol, but a concrete power-sum object of one higher order. Moreover, the relation $S_{k+2}(n) = M(k) + n^{k+2}$ separates the full power sum into an interior accumulation term and a boundary contribution. In contrast, the operator $K(k)$ acts as the complementary term balancing the identity with $nS_{k+1}(n)$, and thus encodes the antisymmetric part of the decomposition.*

7.2. Relation with the Bernoulli–Faulhaber formula

The classical Faulhaber formula provides explicit polynomial expressions for power sums in terms of Bernoulli numbers or Bernoulli polynomials. In Bernoulli polynomial form, one has $S_m(n) = \frac{B_{m+1}(n+1)-B_{m+1}(1)}{m+1}$. Equivalently, under the convention $B_1^+ = \frac{1}{2}$, this may be written as $S_m(n) = \frac{1}{m+1}\sum_{j=0}^{m} \binom{m+1}{j} B_j^+ \, n^{m+1-j}$.

Proposition 2. *The operators $M(k)$ and $K(k)$ admit the Bernoulli polynomial representations*

$$M(k) = \frac{B_{k+3}(n) - B_{k+3}(1)}{k+3},$$

and

$$K(k) = n \cdot \frac{B_{k+2}(n+1) - B_{k+2}(1)}{k+2} - \frac{B_{k+3}(n+1) - B_{k+3}(1)}{k+3}.$$

Proof. By Proposition 1, $M(k) = S_{k+2}(n-1)$. Applying the Bernoulli polynomial formula with $m = k+2$ and replacing n by $n-1$, we obtain $M(k) = \frac{B_{k+3}(n)-B_{k+3}(1)}{k+3}$.

Similarly, using $K(k) = nS_{k+1}(n) - S_{k+2}(n)$, together with the formulas $S_{k+1}(n) = \frac{B_{k+2}(n+1)-B_{k+2}(1)}{k+2}$ and $S_{k+2}(n) = \frac{B_{k+3}(n+1)-B_{k+3}(1)}{k+3}$, gives the desired expression for $K(k)$.

Remark 4. *Proposition 2 shows that our decomposition is compatible with the classical Bernoulli framework. In particular, the operators $M(k)$ and $K(k)$ do not lie outside the classical theory; rather, they reorganize it in a form that emphasizes adjacent-order recursion and symmetry decomposition.*

7.3. Comparison of methodological features

We now compare the present decomposition with the Bernoulli–Faulhaber method from three perspectives.

Remark 5 (Derivation logic)**.** *The Bernoulli method is fundamentally top-down: one starts with generating functions or Bernoulli polynomial identities and derives a closed expression for each $S_m(n)$ directly. By contrast, the present operator system is bottom-up and recursive in nature, since it is organized around the relation between the adjacent sums $S_{k+1}(n)$ and $S_{k+2}(n)$.*

Remark 6 (Structural interpretation)**.** *The Bernoulli approach is algebraically complete, but it does not explicitly separate symmetric and antisymmetric contributions in the summation process. Our construction does so through the reflection transformation $p \mapsto n-p$, and the centered factor $n-2p$, which isolates deviation from the midpoint and gives the decomposition a direct symmetry-theoretic meaning.*

Remark 7 (Computational organization)**.** *The Bernoulli method depends on Bernoulli numbers or Bernoulli polynomials as global coefficient data. In contrast, our formulas for $M(k)$ and $K(k)$ are expressed directly through adjacent power sums, making them suitable for recursive or iterative implementations where structural reduction is more important than closed coefficients.*

7.4. Relation with classical odd-even factorization

A second classical viewpoint classifies power sums according to the parity of the exponent. Let $T = S_1(n) = \frac{n(n+1)}{2}$. Then odd and even power sums admit different factorization patterns: $S_{2r+1}(n) = T^2 P_r(T), \qquad S_{2r}(n) = S_2(n)\, Q_r(T)$, where P_r and Q_r are rational polynomials.

Proposition 3. *The parity of k determines the parity structure of the adjacent sums appearing in $M(k)$ and $K(k)$ as follows:*

(i) If $k = 2m$ is even, then $M(2m) = S_{2m+2}(n-1), \qquad K(2m) = nS_{2m+1}(n) - S_{2m+2}(n)$. Thus $M(2m)$ is governed by an even-order power sum, while $K(2m)$ couples odd and even orders.

(ii) If $k = 2m+1$ is odd, then $M(2m+1) = S_{2m+3}(n-1), \qquad K(2m+1) = nS_{2m+2}(n) - S_{2m+3}(n)$. Thus $M(2m+1)$ is governed by an odd-order power sum, while $K(2m+1)$ again mixes both parity types.

Proof. This follows immediately from Proposition 1 by substituting $k = 2m$ and $k = 2m+1$.

Remark 8. *The classical parity classification distinguishes power sums according to whether the exponent is odd or even. Our decomposition refines this viewpoint by identifying a symmetry-sensitive correction term through the factor $n-2p$. Thus, instead of merely grouping formulas by parity, the present framework extracts a reflection-based structural component from the summation itself.*

7.5. Concluding comparison

Corollary 2. *The operator decomposition based on $M(k)$ and $K(k)$ may be summarized as follows:*

(i) It is consistent with the Bernoulli–Faulhaber theory, since both $M(k)$ and $K(k)$ admit Bernoulli polynomial expressions.

(ii) It differs from the Bernoulli method in that it is organized around adjacent-order recursion rather than global coefficient formulas.

(iii) It refines the classical odd-even factorization viewpoint by isolating symmetric and antisymmetric contributions through the transformation $p \mapsto n - p$.

(iv) It provides a natural decomposition of $S_{k+2}(n) = M(k) + n^{k+2}$, where $M(k)$ represents the truncated interior power sum and $K(k)$ serves as the complementary balancing term.

Proof. All four assertions are immediate consequences of Proposition 1, Proposition 2, and Proposition 3, together with Remarks 3, 4, and 8.

8. Conclusion

In this paper, we propose an elementary geometric proof method for the sums of second and third powers based on intuitive matrix partition constructions. The derivations for $S(2)$ and $S(3)$ do not rely on Bernoulli numbers or generating functions, and the resulting approach is constructive and accessible. Building on this framework, we further develop a recursive decomposition algorithm based on the Changlang operators $M(k)$ and $K(k)$ for decomposing general formulas for higher-order power sums into formulas for lower-order power sums. For odd powers, the algorithm is implemented entirely through operations involving the Changlang operators. For even powers, the standard binomial recursion (which is essentially Pascal-type, and equivalent to the Bernoulli-difference recurrence) is employed; alternatively, the decomposition formula for an even power of a given order can be obtained by differentiating the formula for the odd power of the next higher order. We also investigate the underlying algebraic relationships between the elementary geometric proof method and the decomposition theorem, and compare the differences and connections between the proposed approach and existing theoretical results. Furthermore, we explore the structural parallels between the symmetric-antisymmetric splitting inherent in the $M(k), K(k)$ framework and the conjugate decomposition patterns in algebraic integer rings (such as Gaussian, Eisenstein, and Hurwitz integers), thereby placing the elementary decomposition within a broader arithmetic context. The new method presented in this study may provide potential insights for cryptography, simplification of large-number power sums, approximate fitting in engineering science, and statistical modeling.

References

[1] Herbert John Ryser. *Combinatorial mathematics*, volume 14. American Mathematical Soc., 1963.

[2] Zenon Ivanovich Borevich and Igor Rostislavovich Shafarevich. *Number theory*, volume 20. Academic press, 1986.

[3] Robert F Lax. *Modern algebra and discrete structures*. Pearson Education, 1991.

[4] Leonard Carlitz. Bernoulli numbers. *The Fibonacci Quarterly*, 6(3):71–84, 1968.

[5] Elna B McBride. *Obtaining generating functions*. Springer Science & Business Media, 2012.
[6] Qifu Tyler Sun, Jinhong Yuan, Tao Huang, and Kenneth W Shum. Lattice network codes based on eisenstein integers. *IEEE transactions on communications*, 61(7):2713–2725, 2013.
[7] Greg Dresden and Wayne M Dymàček. Finding factors of factor rings over the gaussian integers. *The American Mathematical Monthly*, 112(7):602–611, 2005.
[8] K Johnson and M Pavlovski. Integer-valued polynomials on the hurwitz ring of integral quaternions. *Communications in Algebra*, 40(11):4171–4176, 2012.
[9] José L Cereceda. Binary quadratic forms and sums of powers of integers. *arXiv preprint arXiv:2106.14963*, 2021.