Isometry Classes of Codes via Hall Polynomials

Sanchayan Dutta* and Greg Kuperberg†
University of California, Davis

We count binary linear codes up to isometry and linear codes over finite fields up to monomial equivalence. We also count Calderbank–Shor–Steane (CSS) quantum stabilizer spaces up to coordinate permutations. Burnside's lemma reduces these problems to counting invariant subspaces for the classical codes and invariant flags for the CSS spaces. We evaluate the counts both by Birkhoff's submodule formula and by explicit recurrences for coarsened Hall polynomials. For binary projective codes, species inversion removes repeated columns. Our Python and Sage programs reproduce all seven of Harald Fripertinger's binary tables and match the complete later monomial tables for $q = 2,3,4,5,8,9,11$. The binary program extends the published complete rows from length 40 to 64 in under an hour. For quantum CSS spaces, a two-stage Hall recurrence gives complete class counts by ordered check ranks through length 46 in under an hour. We retain the X and Z labels and count classes under coordinate permutations without constructing explicit representatives.

1. INTRODUCTION

A binary linear code of length n is a subspace of $\mathbb{F}_2^n$. The $2^n - 1$ codes of dimension one fall into n classes under coordinate permutations, one for each possible weight of the unique nonzero word. In higher dimensions, a single permutation must carry one entire subspace to another. Over a general finite field, monomial equivalence also permits each coordinate to be multiplied by a nonzero scalar. Fripertinger [1] used Burnside's lemma to count these classes by averaging the numbers of subspaces fixed by each monomial transformation.

These fixed-subspace counts can be computed using module theory. For a fixed transformation T, let x act on $\mathbb{F}_q^n$ as T. The invariant subspaces are then exactly the $\mathbb{F}_q[x]$-submodules. Since $\mathbb{F}_q[x]$ is a principal ideal domain, the structure theorem decomposes this module into components described by partitions. Hall polynomials count submodules with prescribed submodule and quotient types [2]. Summing over quotient types gives the count in Birkhoff's formula [3, 4]. We develop recurrences for these sums to obtain the fixed-subspace counts needed for Burnside's lemma.

The same method applies to binary CSS stabilizer codes, specified by two orthogonal subspaces S_X and S_Z. Writing orthogonality as $S_X \subseteq S_Z^\perp$ turns the problem into one of counting nested subspaces. For each coordinate permutation, the fixed count is obtained by first choosing an invariant $S_Z^\perp$ and then an invariant S_X inside it. The resulting class counts are symmetric in the two check ranks and the number of logical qubits. Koh et al. [5] classified CSS codes with $k \geq 1$ through length 14 under qubit permutations and global Hadamard duality. Our calculation gives complete rank tables through length 46 under coordinate permutations alone, with the X and Z spaces distinguished. These counts require no explicit representatives.

* dutta@ucdavis.edu
† greg@math.ucdavis.edu

2. BINARY LINEAR CODES

A binary linear $[n,k]$ code is a k-dimensional subspace of $\mathbb{F}_2^n$, considered up to coordinate permutation. Let $A_2(n,k)$ count these classes, including the zero code and full space. We set $A_2(0,0) = 1$ and extend $A_2(n,k)$ by zero outside $0 \leq k \leq n$. Unless stated otherwise, coordinates that vanish identically and codewords of weight one are allowed.

To count classes without constructing representatives, we use Burnside's lemma as in Fripertinger [1]. Birkhoff's submodule formula [3, 4] and the Hall polynomial framework [2] give the required fixed counts. We develop the binary recurrence first. Sections 3 and 4 extend it to general finite fields and CSS spaces.

2.1. Permutation modules

Burnside's lemma requires counting the codes fixed by each coordinate permutation. This count depends only on the cycle lengths, since relabeling coordinates does not change the number of fixed codes. Write $\lambda = 1^{m_1}2^{m_2}\cdots \vdash n$, where m_r is the number of cycles of length r. There are $n!/z_\lambda$ permutations of this type, where

$$z_\lambda = \prod_{r\geq 1} r^{m_r} m_r!. \tag{2.1}$$

Let $F_\lambda(k)$ count the k-dimensional codes fixed by one such permutation, and set $F_\lambda(t) = \sum_{k=0}^n F_\lambda(k)t^k$. Burnside's lemma becomes

$$A_2(n,k) = \sum_{\lambda\vdash n} \frac{F_\lambda(k)}{z_\lambda}. \tag{2.2}$$

To compute $F_\lambda(k)$, fix a permutation π and let x act on $V = \mathbb{F}_2^n$ as π. A code is invariant under π exactly when it is an $\mathbb{F}_2[x]$-submodule. On a cycle of length r, the vectors $e, \pi e, \ldots, \pi^{r-1}e$ form a basis, and $\pi^r e = e$. This gives the cyclic module $\mathbb{F}_2[x]/(x^r - 1)$. Since different cycles act independently,

$$V_\lambda \cong \bigoplus_{r\geq 1} (\mathbb{F}_2[x]/(x^r-1))^{m_r}. \tag{2.3}$$

We can decompose these modules further. Write $r = 2^a r_0$, where r_0 is odd. In characteristic two,

$$x^r - 1 = (x^{r_0} - 1)^{2^a}. \tag{2.4}$$

Since $x^{r_0} - 1$ has no repeated irreducible factors, the Chinese remainder theorem gives one summand $\mathbb{F}_2[x]/(f^{2^a})$ for each of its monic irreducible factors f. For example, a transposition on three coordinates has cycles of lengths 2 and 1, giving

$$\frac{\mathbb{F}_2[x]}{((x+1)^2)} \oplus \frac{\mathbb{F}_2[x]}{(x+1)},$$

whose partition type is $(2,1)$.

In general, grouping summands with the same f gives the f-primary components of V_λ. The projections onto these components are polynomials in π, so every invariant code splits into submodules of these components. Our counting problem therefore reduces to counting submodules of each primary component.

2.2. The local Hall sum

It remains to count the submodules of a primary component. The structure theorem describes their possible types, but does not give their number. Birkhoff's formula supplies the count for each type.

Let $P = \mathbb{F}_q[x]$ and fix a monic irreducible polynomial f. The field $K = P/(f)$ has $Q = q^{\deg f}$ elements. An f-primary module has the form

$$M_\eta = \bigoplus_j P/(f^{\eta_j}), \qquad \eta_1 \geq \eta_2 \geq \cdots > 0.$$

The partition η records the lengths of its cyclic summands. It is convenient to use the profile $N = \eta'$, where N_i counts the summands of length at least i. For example, $\eta = (3,2)$ gives $N = (2,2,1)$. The module has dimension $(\deg f)\sum_i N_i$ over $\mathbb{F}_q$.

Every submodule U has a similar partition μ and profile $B = \mu'$, satisfying

$$B_i = \dim_K(f^{i-1}U/f^iU), \qquad N_i \geq B_i \geq B_{i+1} \geq 0. \tag{2.5}$$

Here the powers of f act on U itself, rather than giving the intersections $U \cap f^i M_\eta$. All profiles are extended by zeros.

To count submodules with a prescribed profile B, we use the Gaussian coefficient

$$\begin{bmatrix} a \\ b \end{bmatrix}_Q = \prod_{j=0}^{b-1} \frac{Q^{a-j}-1}{Q^{b-j}-1}$$

which counts the b-dimensional subspaces of K^a. It is one for $b = 0$ and zero when $b < 0$ or $b > a$.

Theorem 2.1 (Birkhoff's submodule formula)**.** *The number of submodules of M_η with profile B is*

$$h_N(B;Q) = \prod_{i\geq 1} Q^{B_{i+1}(N_i - B_i)} \begin{bmatrix} N_i - B_{i+1} \\ B_i - B_{i+1} \end{bmatrix}_Q. \tag{2.6}$$

The idea behind the formula is to count injective module maps from a module of type μ into M_η, then divide by the number of automorphisms of the source. At level i, the vectors killed by f and belonging to $f^{i-1}M_\eta$ form a space of dimension N_i over K. The B_{i+1} independent directions already chosen for longer summands leave room for $B_i - B_{i+1}$ new independent choices. These vectors must then be lifted to generators of the cyclic summands. After dividing by the automorphism count, the Gaussian coefficients record the choices at each level, while the powers of Q account for the lifts. We use Butler's form [4], also implemented in SageMath [6].

For example, take $N = (2,1)$. The profile (1) gives $Q+1$ submodules of length one. At length two, the profiles $(1,1)$ and (2) give Q and 1 submodules respectively. Writing $H_N(t;Q)$ for the generating polynomial by length, we include the zero and full submodules to obtain

$$H_{(2,1)}(t;Q) = 1 + (Q+1)t + (Q+1)t^2 + t^3.$$

Birkhoff's formula keeps track of the submodule type. For Burnside's lemma we only need the dimension. Hall polynomials relate these two counts. The polynomial $g^\eta_{\alpha,\mu}(Q)$ counts submodules of type μ whose quotient has type α. Summing over quotient types gives Birkhoff's count, and summing over submodule types gives the generating polynomial

$$\begin{aligned} h_{\eta'}(\mu';Q) &= \sum_\alpha g^\eta_{\alpha,\mu}(Q), \\ H_N(t;Q) &= \sum_B h_N(B;Q)t^{|B|}. \end{aligned} \tag{2.7}$$

Here $|B| = \sum_i B_i$, and $H_\varnothing(t;Q) = 1$. Thus $[t^s]H_N(t;Q)$ counts all submodules of length s.

Finally, let $N^{(f)}$ be the profile of the f-primary component of the permutation module V_λ, and put $d_f = \deg f$. A submodule of length s has binary dimension $d_f s$. Since the primary components can be counted independently, we multiply their generating polynomials to get

$$F_\lambda(t) = \prod_f H_{N^{(f)}}(t^{d_f}; 2^{d_f}). \tag{2.8}$$

The coefficient of t^k is the number of k-dimensional codes fixed by the permutation, as required by Burnside's lemma.

2.3. Computing the Hall sum

Birkhoff's formula counts submodules of a given profile B, but computing $H_N(t;Q)$ requires summing over all possible profiles. We can do this efficiently because each factor in (2.6) depends only on consecutive entries B_i and B_{i+1}.

For nonempty N, let L be the largest index with $N_L > 0$. We build profiles backward, starting with $B_{L+1} = 0$. Let $D_{i+1}(a,s)$ denote the total weight accumulated from layers $i+1,\ldots,L$, where $B_{i+1} = a$ and the entries sum to s. Initially, $D_{L+1}(0,0) = 1$, with all other entries zero.

At layer i, choose $B_i = b$ with $a \leq b \leq N_i$. The corresponding factor of Birkhoff's formula gives the update

$$D_i(b, s+b) \mathrel{+}= D_{i+1}(a,s) Q^{a(N_i - b)} \begin{bmatrix} N_i - a \\ b - a \end{bmatrix}_Q. \tag{2.9}$$

Starting each layer at zero, we add the contributions of all possible choices. Every profile is counted exactly once, so the final coefficients are

$$[t^s]H_N(t;Q)=\sum_b D_1(b,s). \tag{2.10}$$

Writing $m=\sum_i N_i$ and $c=N_1$, this takes $O(L(m+1)(c+1)^2)$ arithmetic operations and $O((m+1)(c+1))$ stored integers, assuming precomputed Gaussian coefficients and powers of Q and excluding integer bit costs.

2.4. Aggregation and restrictions

We can compute $F_\lambda(t)$ without factoring every x^r-1 by grouping irreducible polynomials according to the multiplicative orders of their roots. For odd $m>1$, let $d_m=\mathrm{ord}_m(2)$ and $e_m=\varphi(m)/d_m$, with $d_1=e_1=1$. The $\varphi(m)$ roots of order m form e_m irreducible polynomials of degree d_m.

Recall that a cycle of length $r=2^a r_0$, with r_0 odd, contributes a summand of length 2^a for each irreducible factor of $x^{r_0}-1$. A factor whose roots have order m occurs exactly when $m\mid r_0$. Thus all irreducibles of order m share the profile

$$N_i^{(m)}=\sum_{r=2^a r_0} m_r\,\mathbf{1}_{m\mid r_0}\mathbf{1}_{i\le 2^a}. \tag{2.11}$$

Since there are e_m such irreducibles, their contributions to (2.8) combine to give

$$F_\lambda(t)=\prod_{\substack{1\le m\le n\\ m\ \mathrm{odd}}} H_{N^{(m)}}(t^{d_m};2^{d_m})^{e_m}. \tag{2.12}$$

Empty profiles contribute one. We cache each local polynomial by (Q,N) to avoid repeated calculations.

The same counts give two restrictions without repeating the enumeration. A code has full support if no coordinate vanishes on every codeword. Deleting a zero coordinate preserves the dimension, while deleting a coordinate supporting a weight-one vector lowers both length and dimension by one. In each case, adjoining the deleted coordinate reverses the operation on permutation classes. Hence

$$\begin{aligned} A_2^{\mathrm{fs}}(n,k)&=A_2(n,k)-A_2(n-1,k),\\ A_2^{d\ge 2}(n,k)&=A_2(n,k)-A_2(n-1,k-1). \end{aligned} \tag{2.13}$$

The zero code is included in the second count, since it has no nonzero codewords.

2.5. Projective codes by species inversion

A binary code is projective if its generator columns are nonzero and pairwise distinct. Equivalently, its dual has no words of weight one or two. Let $P_2(n,k)$ count projective classes, with $P_2(0,0)=1$.

We first count codes whose generator columns are distinct, but may include one zero column. Let $\mathcal{C}_k$ denote all k-dimensional binary codes and $\mathcal{D}_k$ those with distinct columns. The fixed-code counts from the previous sections give the cycle index

$$Z_k(\mathbf{p})=\sum_{m\ge 0}\sum_{\lambda\vdash m}\frac{F_\lambda(k)}{z_\lambda}p_\lambda,\qquad p_\lambda=\prod_{r\ge 1}p_r^{m_r}. \tag{2.14}$$

Grouping equal columns and keeping one from each group preserves the dimension. Conversely, every code can be obtained by repeating the columns of a code with distinct columns. To count the latter codes, we therefore need to undo this repetition.

Proposition 2.2. *The number of projective binary code classes is*

$$P_2(n,k)=[x^n]\frac{Z_k(x-x^2,x^2-x^4,\dots)}{1+x}. \tag{2.15}$$

Proof. Repeating a coordinate amounts to replacing it by a nonempty set of identical coordinates. In the language of combinatorial species, this gives $\mathcal{C}_k=\mathcal{D}_k\circ\mathcal{E}_+$, where $\mathcal{E}_+$ describes nonempty sets.

Species inversion reverses this operation. The compositional inverse of $\mathcal{E}_+$ is a virtual species with type series $x-x^2$ [7, Lemma 1.1]. The composition rule [8] therefore gives

$$\widetilde{\mathcal{D}}_k(x)=Z_k(x-x^2,x^2-x^4,\dots).$$

A code with distinct columns has at most one zero column. Removing it gives a projective code, and conversely every projective code admits one additional zero column. Hence

$$\widetilde{\mathcal{D}}_k(x)=(1+x)\sum_{n\ge 0}P_2(n,k)x^n.$$

Dividing by $1+x$ proves the formula. □

To compute through degree n, we only need cycle types $\lambda\vdash m$ with $m\le n$, since substituting $p_r=x^r-x^{2r}$ produces no terms below degree m. We expand these products, truncate at degree n, and divide by $1+x$.

We use the duality $F_\lambda(k)=F_\lambda(m-k)$ for $\lambda\vdash m$. Projective counts need not satisfy this symmetry, since the dual of a projective code need not be projective.

2.6. Direct-sum components

In Fripertinger's notation, $W_{nk2}=A_2(n,k)$ counts all binary classes, $S_{nk2}=A_2^{\mathrm{fs}}(n,k)$ counts those with full support, and $T_{nk2}=\sum_{j=1}^k S_{nj2}$.

We can also count codes that cannot be split into smaller codes on disjoint sets of coordinates. Such codes are called indecomposable. To see why every full-support code has a unique decomposition into these components, call a nonzero codeword minimal if its support contains no strictly smaller support of another nonzero codeword.

Minimal codewords span the code. Indeed, if v is not minimal, we can write $v=w+(v+w)$, where both terms have smaller support, and apply induction on weight. Now group

coordinates connected by the supports of minimal codewords. Since these words span the code, the groups give a direct-sum decomposition. Every minimal word must lie entirely in one component of any such decomposition. Thus the components are uniquely determined.

Let R_{nk2} count indecomposable full-support classes. Every full-support code is determined by a multiset of its components, each of which may occur any number of times. This gives the generating function

$$1+\sum_{n,k\ge 1} S_{nk2}x^n y^k = \prod_{n,k\ge 1}(1-x^n y^k)^{-R_{nk2}}. \tag{2.16}$$

We recover R_{nk2} in increasing order of length. After removing all factors of smaller length, the coefficient of $x^n y^k$ gives R_{nk2}. Multiplying by $(1-x^n y^k)^{R_{nk2}}$ removes its contribution. The calculation can be truncated to $n \le N$ and $k \le K$.

The same formula holds for projective codes, replacing S_{nk2} and R_{nk2} by their projective counterparts. Indeed, a direct sum is projective exactly when each component is projective. The argument also extends to general finite fields under monomial equivalence. In the spanning argument, subtracting a suitable scalar multiple of a word with smaller support cancels a coordinate. Unlike species inversion, which groups identical columns, this decomposition separates independent coordinate blocks.

Table I gives selected counts from the accompanying program. An independent enumeration of all subspaces checks the fixed counts in every dimension and for every permutation type through $n = 7$. Orthogonal complementation commutes with coordinate permutations, so $F_\lambda(k) = F_\lambda(n-k)$ and $A_2(n,k) = A_2(n,n-k)$. For the identity permutation, this fixed count is the Gaussian coefficient $F_{1^n}(k) = \begin{bmatrix} n \\ k \end{bmatrix}_2$.

3. LINEAR CODES OVER FINITE FIELDS

The binary calculation extends to monomial equivalence over $\mathbb{F}_q$, where $q = p^h$ is a prime power. The acting group is

$$\mathrm{Mon}(n,q) = (\mathbb{F}_q^\times)^n \rtimes S_n. \tag{3.1}$$

Its elements permute the coordinates and multiply each coordinate by a nonzero scalar. Field automorphisms are not included. Write $A_q(n,k)$ for the number of its orbits on $\mathrm{Gr}(k,n)$. For $q = 2$, this agrees with Section 2. For larger q, the Burnside average must include coordinate scalings as well as permutations.

3.1. Cycle products and repeated roots

On a cycle of length r, the diagonal multipliers can be moved by a diagonal change of basis to a single position. The remaining scalar is their product $\beta \in \mathbb{F}_q^\times$. The corresponding module is

$$\mathbb{F}_q[x]/(x^r - \beta). \tag{3.2}$$

A monomial conjugacy type is described by nonnegative integers $\mathbf{m} = (m_{r,\beta})$ satisfying $\sum_{r,\beta} r m_{r,\beta} = n$. Here $m_{r,\beta}$ is the number of length-r cycles with product β. Put

$$Z_q(\mathbf{m}) = \prod_{r,\beta} \big(r(q-1)\big)^{m_{r,\beta}} m_{r,\beta}!. \tag{3.3}$$

For a specified product β, the repeated-root reduction requires its Frobenius root. Write $r = p^a r_0$ with $(r_0, p) = 1$, and let γ be the unique element of $\mathbb{F}_q$ satisfying $\gamma^{p^a} = \beta$. Then

$$x^r - \beta = (x^{r_0} - \gamma)^{p^a}. \tag{3.4}$$

The factors of $x^{r_0} - \gamma$ are distinct. Each such irreducible f receives $m_{r,\beta}$ primary summands of length p^a. This determines $N^{(f)}(\mathbf{m})$. In the sum over cycle products, γ may instead be denoted β after reindexing through the Frobenius bijection.

Theorem 3.1 (Monomial enumeration)**.** *For every prime power q,*

$$\begin{aligned} A_q(n,k) &= [t^k] \sum_{\mathbf{m}} \frac{F_{\mathbf{m}}(t)}{Z_q(\mathbf{m})}, \\ F_{\mathbf{m}}(t) &= \prod_f H_{N^{(f)}(\mathbf{m})}(t^{\deg f}; q^{\deg f}), \end{aligned} \tag{3.5}$$

where the sum is over the types of total length n.

Proof. Fix a permutation with c cycles. For a cycle of length r, exactly $(q-1)^{r-1}$ assignments of the diagonal entries have a prescribed product. The cycle products are therefore uniform and independent in $\mathbb{F}_q^\times$. The average over products contributes $(q-1)^{-c}$. If a permutation has m_r length-r cycles, assigning their products with multiplicities $m_{r,\beta}$ can be done in $m_r!/\prod_\beta m_{r,\beta}!$ ways. Combining these choices with the product average and the permutation weight $1/z_\lambda$ gives the weight $1/Z_q(\mathbf{m})$. For a resulting type, fixed codes are submodules of the direct sum of (3.2). Primary decomposition and Theorem 2.1 give the product in (3.5). Burnside's lemma completes the proof. □

This is the monomial invariant-subspace framework of Fripertinger [1], evaluated here by the recurrence (2.9). The group has a scalar kernel on subspaces, but this does not change Burnside's formula. Every scalar element fixes every code and occurs with its proper group weight.

3.2. Normalization of the cycle weights

We check the cycle weights independently of the fixed-subspace counts. For every $n \ge 0$,

$$\sum_{\substack{\mathbf{m} \\ \sum_{r,\beta} r m_{r,\beta} = n}} \frac{1}{Z_q(\mathbf{m})} = 1. \tag{3.6}$$

Indeed, the ordinary generating function of the left side is

$$\begin{aligned} &\prod_{r\ge 1} \prod_{\beta \in \mathbb{F}_q^\times} \sum_{j \ge 0} \frac{z^{rj}}{(r(q-1))^j\, j!} \\ &\qquad = \exp\left(\sum_{r \ge 1} \frac{z^r}{r} \right) = \frac{1}{1-z}. \end{aligned} \tag{3.7}$$

TABLE I. Selected binary counts from complete rank rows through $n = 64$. The rows $n > 40$ extend beyond the newer monomial table at $n = 40$. The complete rows are supplied with the source.

n	$A_2(n,3)$	$A_2(n,4)$	$A_2(n,7)$
40	415767	614843228	373355389974113117565300
60	5451293	114250906542	349096297133338537936573941306757950
61	6067621	142263232527	1075903304685688569404532618623428003
62	6742858	176586904346	327976966553229329437872990893634909 7
63	7481636	218522121361	989136863658202966088208017329690634 7
64	8288869	269612583108	295195528310000633678270000521936969 07

This identity also checks the constant coefficient of the class polynomial, since every operator fixes the zero code. Direct orbit comparisons check the assignment of cycle products to primary profiles.

The number of decorated types has a different generating function. Writing this number as $p_{q-1}(n)$, we have

$$\sum_{n\geq 0} p_{q-1}(n) z^n = \prod_{r\geq 1} (1-z^r)^{-(q-1)}. \tag{3.8}$$

For $q = 2$ this is the ordinary partition count. For larger fields, every cycle length has $q-1$ possible products. This explains why direct enumeration of decorated types can be expensive even when each local Hall sum is small. The state aggregation below combines types with the same primary profiles while preserving their total weight.

3.3. Exact truncation and state aggregation

Suppose only dimensions $k \leq K$ are wanted. All polynomial products may be reduced modulo t^{K+1}. For a primary factor of degree d, retain only local lengths $s \leq R = \lfloor K/d \rfloor$. Factors of degree greater than K contribute only their constant term. Moreover, if $B_i > 0$, then $B_1, \ldots, B_i$ are all positive and $|B| \geq i$. Thus layers beyond R can be dropped from (2.9) without changing the requested coefficients.

To evaluate the outer sum, we can enumerate the types $\mathbf{m}$. Alternatively, for each cycle-length type we process its cycles successively. At each stage we keep a dictionary of the primary profiles obtained. Assignments giving the same profiles at that stage can be merged by adding their weights. For repeated cycles, distributions of the products can be grouped with their multinomial coefficients. The key must retain the irreducible factor associated with each profile until all cycles have been processed. Equal degrees alone do not justify merging factors, since later cycles can update them differently.

Local polynomials can be cached after this aggregation. Profiles with the same (Q,N) have identical local counts, even if they belong to different irreducible polynomials. This separates the field arithmetic from the integer arithmetic in the Hall recurrence.

Monomial duality gives $A_q(n,k) = A_q(n,n-k)$. Indeed, the orthogonal complement transforms under the inverse transpose of the monomial matrix, which is again monomial. For $n \geq 1$, full support is recovered by $A_q^{\mathrm{fs}}(n,k) = A_q(n,k) - A_q(n-1,k)$. No factor of $q-1$ appears in this difference, since a zero coordinate has no additional structure.

There are simple checks independent of the Hall calculation. For $n \geq 1$,

$$\begin{aligned} A_q(n,0) &= A_q(n,n) = 1, \\ A_q(n,1) &= A_q(n,n-1) = n. \end{aligned} \tag{3.9}$$

For a one-dimensional code, diagonal multipliers make all nonzero coordinates of a generator equal to one. Permutations then identify generators with the same support size. There is one class for each size from one to n. Duality gives the last equality. In particular there is one full-support class of dimension one at every positive length.

Table II gives the resulting fixed-dimension extension over $\mathbb{F}_8$. By duality, the calculation determines every dimension through $n = 15$. For $16 \leq n \leq 23$ it provides $k \leq 7$ and the dual dimensions. Independent orbit enumeration checks every dimension through $n = 5$ for $q = 3,4$ and through $n = 4$ for $q = 8,9,11$. Extension-field arithmetic is therefore checked directly. Section 5 gives the complete-row comparison over all eight fields in the newer monomial tables.

4. CSS STABILIZER SPACES

We work over $\mathbb{F}_2$. A CSS space has separate pure X and pure Z check spaces. Commutation is expressed by orthogonality, so we specify an ordered pair (S_X, S_Z) with

$$\begin{aligned} &S_X \perp S_Z, \qquad k = n - r_X - r_Z, \\ &r_X = \dim S_X, \qquad r_Z = \dim S_Z. \end{aligned} \tag{4.1}$$

The CSS construction gives a quantum code with k logical qubits [10, 11]. We count binary stabilizer spaces with generator phases suppressed, equivalently choosing the standard positive pure X and pure Z stabilizers. Other sign sectors are not counted separately.

Equivalence means applying the same coordinate permutation to both spaces. Write $C_n(a,b)$ for the number of classes with $(r_X, r_Z) = (a,b)$, where $a,b \geq 0$ and $a+b \leq n$. We retain the X and Z labels. We do not identify pairs under arbitrary local Clifford transformations or impose a distance threshold.

TABLE II. Monomial classes over $\mathbb{F}_8$ from our SageMath implementation. In one hour, the program computes all counts with $k \le 7$ through length 23. Fripertinger's monomial table ends at $n = 15$ [9].

n	$A_8(n,3)$	$A_8(n,4)$	$A_8(n,7)$
15	1925775246	8721550451527687	491089979129105242944092936
19	1387806455127	12354432696289159739420	425375676007016061624056063769001455340952O3
20	6453237234422	37372158676601409675912O	63724040144650189570324790326180205714946235762S
21	28885777573888	10784537201640923199139613	90917211998783016112119728816191975315754940095061 68
22	124733726605372	29755518540332539327767476 2	12381891120147320394247317744205732616917549903472145554 6
23	520627048662547	786580663944929487728796902 4	16129620389962899228280155170196804862460812495839861743288 59

4.1. From orthogonal pairs to flags

Orthogonal complementation turns the constraint into an inclusion. Put $A = S_X$ and $D = S_Z^{\perp}$. Then (4.1) is equivalent to the flag $A \subseteq D \subseteq \mathbb{F}_2^n$, with $\dim A = a$ and $\dim D = n - b$. The inverse is $(S_X, S_Z) = (A, D^{\perp})$. Since permutations preserve the dot product, this is an equivariant bijection. A pair is fixed by π exactly when both A and D are submodules of V_λ.

For a primary component with profile N, first choose D with profile B, then A with profile E inside D. Every module D of a given type has the same number of such submodules, so we multiply the two Hall counts. Define the local flag polynomial

$$G_N(u,v;Q) = \sum_{B,E} h_N(B;Q) h_B(E;Q) u^{|E|} v^{|N|-|B|}, \tag{4.3}$$

where B and E are nonincreasing profiles satisfying $0 \le E_i \le B_i \le N_i$. The exponents record the local length of A and the local codimension of D. We set $G_\varnothing(u,v;Q) = 1$.

Theorem 4.1 (CSS enumeration)**.** *For the phase-free ordered CSS spaces defined above,*

$$\begin{aligned} C_n(a,b) &= [u^a v^b] \sum_{\lambda \vdash n} \frac{F_\lambda^{\mathrm{CSS}}(u,v)}{z_\lambda}, \\ F_\lambda^{\mathrm{CSS}}(u,v) &= \prod_f G_{N^{(f)}}(u^{d_f}, v^{d_f}; 2^{d_f}), \end{aligned} \tag{4.4}$$

where $d_f = \deg f$.

Proof. By (4.2), it suffices to count invariant flags. Projection onto the primary components decomposes both A and D. The choices in distinct components are independent. In a component of degree d_f, binary dimensions are d_f times local lengths. Thus the product in (4.4) counts fixed pairs by their two check ranks. Averaging over permutation types gives the number of classes. □

Only ordinary submodule counts occur in (4.3). The dot product need not be nondegenerate on D, and reciprocal irreducible pairs need not be tracked. Orthogonal complementation has already been used in the global bijection (4.2).

4.2. A direct flag recurrence

We can evaluate (4.3) without listing all profiles of D by choosing the layers of B and E together. For nonempty N, let L be the largest index with $N_L > 0$ and process $i = L, L-1, \dots, 1$. At layer i, write $a = B_{i+1}$, $e = E_{i+1}$, $b = B_i$, and $f = E_i$. The two Hall factors give the transition weight

$$\begin{aligned} W_i(b,f;a,e) = Q^{a(N_i-b)+e(b-f)} \\ \cdot \begin{bmatrix} N_i - a \\ b - a \end{bmatrix}_Q \begin{bmatrix} b - e \\ f - e \end{bmatrix}_Q . \end{aligned} \tag{4.5}$$

The allowed choices satisfy $a \le b \le N_i$ and $e \le f \le b$. Initialize $T_{L+1}(0,0,0,0) = 1$ and all other states to zero. In $T_{i+1}(a,e,s,t)$, the indices s and t record the sums of the B and E heights in the layers already chosen. Update

$$T_i(b,f,s+b,t+f) \mathrel{+}= T_{i+1}(a,e,s,t) W_i(b,f;a,e). \tag{4.6}$$

After layer 1,

$$G_N(u,v;Q) = \sum_{b,f,s,t} T_1(b,f,s,t) u^t v^{|N|-s}. \tag{4.7}$$

Each path chooses profiles B and E, and its weight is the product of their Hall factors, as in (2.9). Both current layer heights must be retained. Dimensions alone do not determine the next transition.

For $N = (m)$, the operator on the primary component is scalar over $\mathbb{F}_Q$ and the coefficient with local check ranks (a,b) is

$$[u^a v^b] G_{(m)}(u,v;Q) = \begin{bmatrix} m \\ b \end{bmatrix}_Q \begin{bmatrix} m-b \\ a \end{bmatrix}_Q . \tag{4.8}$$

For the identity permutation this gives

$$\begin{bmatrix} n \\ b \end{bmatrix}_2 \begin{bmatrix} n-b \\ a \end{bmatrix}_2 .$$

This counts ordered orthogonal pairs on labeled coordinates before taking orbits.

4.3. Symmetry of the three dimensions

The successive quotients of the flag have dimensions a, k, and b. The fixed-flag counts are symmetric in all three, including the number of logical qubits.

Lemma 4.2. *For every linear operator T on an m-dimensional finite vector space, the number of invariant j-dimensional subspaces equals the number of invariant $(m-j)$-dimensional subspaces.*

TABLE III. CSS permutation classes $C_6(r_X, r_Z)$. The X and Z spaces are distinguished, and generator phases are suppressed. A dash means $r_X + r_Z > 6$.

$r_X \backslash r_Z$	0	1	2	3	4	5	6
0	1	6	16	22	16	6	1
1	6	37	87	87	37	6	–
2	16	87	149	87	16	–	–
3	22	87	87	22	–	–	–
4	16	37	16	–	–	–	–
5	6	6	–	–	–	–	–
6	1	–	–	–	–	–	–

Proof. Annihilators send T-invariant subspaces to T^{T}-invariant subspaces of complementary dimension. A matrix and its transpose are similar, as follows from their common rational canonical form. Thus T and T^{T} have the same invariant-subspace counts. □

Corollary 4.3. *For every fixed permutation, the CSS fixed-pair count is invariant under all permutations of (a,k,b), where $k = n-a-b$. The same holds for $C_n(a,b)$. In particular,*

$$C_n(a,b) = C_n(b,a) = C_n(n-a-b,b). \tag{4.9}$$

Proof. Fix an invariant D of dimension $a+k$. By Lemma 4.2, the numbers of invariant subspaces $A \subseteq D$ of dimensions a and k are equal. Summing over D interchanges the first two entries of (a,k,b). Alternatively fix A and apply the lemma to the induced operator on V/A. This interchanges the last two entries. These two transpositions generate all permutations of the triple. The equality survives the Burnside average. □

This numerical symmetry does not identify physical codes with different values of k. Two boundary checks are $C_n(0,b) = A_2(n,b)$ and $C_n(a,n-a) = A_2(n,a)$. The first has no X checks. In the second, the flag has $A = D$.

4.4. An independent rank-one count

For the smallest nontrivial pair of check ranks, we can count column types directly instead of invariant subspaces.

Proposition 4.4. *For every $n \geq 2$,*

$$C_n(1,1) = \binom{n}{2} + \sum_{j=1}^{\lfloor n/2 \rfloor} \binom{n-2j+2}{2}. \tag{4.11}$$

Proof. Each check space has a unique nonzero vector. Place these vectors in the rows of a two-row binary matrix. A permutation class is determined by the multiplicities of its four column types $00, 10, 01, 11$. Orthogonality requires an even number of 11 columns.

If there are no 11 columns, both 10 and 01 must occur because the rows are nonzero. After reserving one column of each type, the remaining $n-2$ columns are distributed among three types. This gives $\binom{n}{2}$ classes. If there are $2j > 0$ columns of type 11, both rows are already nonzero. The remaining $n-2j$ columns have three possible types, giving $\binom{n-2j+2}{2}$ classes. Summing over j proves the formula. □

At $n = 6$ this gives 37, as in Table III. The formula provides an independent check at every completed length, even when direct enumeration of all CSS pairs is impractical.

4.5. Global exchange

We may also allow a global Hadamard operation, which exchanges S_X and S_Z. The group becomes $S_n \times C_2$. For unequal ranks, retain one of (a,b) and (b,a). At equal ranks, Burnside's lemma also requires the fixed points of exchange followed by a permutation. Let $J_\pi(a)$ count the subspaces $U \in \mathrm{Gr}(a,n)$ satisfying $\pi^2 U = U$ and $U \perp \pi^{-1}U$. Burnside's lemma gives $\frac{1}{2}\left(C_n(a,a) + \frac{1}{n!}\sum_{\pi \in S_n} J_\pi(a)\right)$. A pair fixed by permutation followed by exchange is $(U, \pi^{-1}U)$ with the stated conditions. The additional bilinear condition requires more than ordinary Hall flags, so the small-length programs evaluate it directly. At $n = 6$, summing over all unordered check-rank pairs gives 512 global-exchange classes. All CSS counts here suppress phases and impose no distance threshold.

5. COMPUTATION AND VERIFICATION

Table IV compares our monomial counts with Fripertinger's tables, with agreement in every computed entry. Table V reports the largest lengths completed by each implementation after 30 minutes and one hour.

CONCLUSION

Burnside's lemma expresses the class counts for classical codes and ordered CSS spaces as averages of invariant-subspace and flag counts. Primary decomposition and the Hall formulas reduce these counts to local calculations. We obtain complete binary class counts through length 64 and complete ordered CSS rank tables through length 46. The monomial calculations over $\mathbb{F}_8$ reach length 23 for $k \leq 7$ and the dual dimensions.

ACKNOWLEDGMENTS

We are grateful to Harald Fripertinger for his extensive tables of isometry classes of codes, which served as valuable benchmarks for our computations. We also benefited from Bergeron, Labelle, and Leroux's textbook on *Combinatorial Species and Tree-like Structures*, which inspired our species inversion algorithm for counting projective binary codes.

TABLE IV. Comparison with Fripertinger's monomial tables [9]. Listed lengths require counts in every dimension, including those obtained by duality. The comparison columns use the one-hour outputs and count each duality pair once. The ten uncomputed entries form the length-18 row over $\mathbb{F}_7$. An asterisk marks the binary input limit $n = 64$.

q	Published n	30 min n	1 h n	Verified	Uncomputed	Mismatches
2	40	64^*	64^*	440	0	0
3	27	31	32	209	0	0
4	25	26	26	181	0	0
5	22	21	22	143	0	0
7	18	16	17	89	10	0
8	15	15	16	71	0	0
9	12	14	15	48	0	0
11	11	12	13	41	0	0

TABLE V. Largest completed lengths after 30 minutes and one hour. B3 and B4 retain $k \le 12$. "Colored" collapses equal nonzero columns. B5 computes collapse coefficients separately for each pair of cycle types, while B6 constructs them by a dynamic program for each source cycle type. "Quotient" enumerates module types V_λ/C and counts codes as kernels of surjections. B8 also reports the full-support and projective counts by dual dimension $n-k$. SageMath pairs (q,K) specify the field order and dimension bound $k \le K$. Other binary runs and all independent Python runs cover every dimension. CSS outputs give either both check ranks or counts grouped by k. An asterisk marks completion of the input range and a dagger a memory stop or program abort.

Binary Python				SageMath				CSS Python			
ID	Output	30 min	1 h	Cell	(q,K)	30 min	1 h	ID	Output	30 min	1 h
B1	All	64^*	64^*	08	$(8,4)$	22	24	C1	Ranks	$38^\dagger$	$38^\dagger$
B2	All	64^*	64^*	09	$(8,4)$	$10^\dagger$	$10^\dagger$	C2	Ranks	43	46
B3	Projective	44	46	11	$(8,4)$	$11^\dagger$	$11^\dagger$	C3	Ranks	$44^\dagger$	$44^\dagger$
B4	Indec. proj.	40	45	12	$(8,7)$	22	23	C4	By k	36	38
B5	Colored full	$21^\dagger$	$21^\dagger$	13	$(8,4)$	18	19	C5	Exchange	6	6
B6	Colored row	35	37	14	$(8,4)$	$14^\dagger$	$14^\dagger$	C6	Exchange	8^*	8^*
B7	Quotient	$22^\dagger$	$22^\dagger$	52	$(8,4)$	14	$14^\dagger$				
B8	Quotient, dual	$22^\dagger$	$22^\dagger$	60	$(8,4)$	14	$15^\dagger$				
				62	$(5,4),(7,6)$	$4^*,6^*$	$4^*,6^*$				
				80	$(2,n)$	23	25				

Independent Python	Pq3	Pq4	Pq5	Pq7	Pq8	Pq9	Pq11
$(n_{30\,\mathrm{min}}, n_{1\,\mathrm{h}})$	$(31,32)$	$(26,26)$	$(21,22)$	$(16,17)$	$(15,16)$	$(14,15)$	$(12,13)$

The B and C labels distinguish program variants. "All" denotes unrestricted binary code counts. "Projective" requires nonzero, pairwise distinct generator columns, and "Indec. proj." further requires indecomposability. For CSS spaces, "Ranks" reports counts by the ordered pair (r_X, r_Z), while "By k" groups counts by the number of logical qubits $k = n - r_X - r_Z$. "Exchange" also identifies spaces under the global interchange of X and Z. SageMath cell numbers identify the corresponding notebook implementations. The label Pqq denotes the independent Python implementation over $\mathbb{F}_q$.

[1] H. Fripertinger. The number of invariant subspaces under a linear operator on finite vector spaces. *Adv. Math. Commun.*, 5: 407–416, 2011. doi:10.3934/amc.2011.5.407. 1, 4

[2] I. G. Macdonald. *Symmetric Functions and Hall Polynomials*. Clarendon Press, Oxford, 2nd edition, 1995. 1

[3] G. Birkhoff. Subgroups of Abelian groups. *Proc. London Math. Soc. (2)*, 38:385–401, 1935. doi:10.1112/plms/s2-38.1.385. 1

[4] L. M. Butler. A unimodality result in the enumeration of subgroups of a finite abelian group. *Proc. Amer. Math. Soc.*, 101: 771–775, 1987. doi:10.1090/S0002-9939-1987-0911049-8. 1, 2

[5] Jin Ming Koh, Anqi Gong, Andrei C. Diaconu, Daniel Bochen Tan, Alexandra A. Geim, Michael J. Gullans, Norman Y. Yao, Mikhail D. Lukin, and Shayan Majidy. Entangling logical qubits without physical operations. arXiv:2601.20927, 2026. URL https://arxiv.org/abs/2601.20927. 1

[6] The Sage Developers. SageMath Reference Manual, combinatorics, q-analogues. URL https://doc.sagemath.org/html/en/reference/combinat/sage/combinat/q_analogues.html. Birkhoff subgroup-counting routine. Accessed October 2026. 2

[7] I. M. Gessel and J. Li. Enumeration of point-determining graphs. arXiv:0705.0042v2, 2009. URL https://arxiv.org/abs/0705.0042v2. 3

[8] F. Bergeron, G. Labelle, and P. Leroux. *Combinatorial Species and Tree-like Structures*, volume 67 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1998. 3

[9] H. Fripertinger. Isometry classes of codes. Table index and monomial subspace tables. Accessed October 20, 2026. 6, 8

[10] A. R. Calderbank and P. W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, 1996. doi:10.1103/PhysRevA.54.1098. 5

[11] A. M. Steane. Multiple-particle interference and quantum error correction. *Proc. Roy. Soc. London Ser. A*, 452:2551–2577, 1996. doi:10.1098/rspa.1996.0136. 5