A Database of Binary Septics and Applications to Galois Theory and Coding

Jurgen Mezinaj[0009-0008-6531-0727]

Department of Mathematics and Statistics, Oakland University, Rochester, MI 48309, USA
mezinaj@oakland.edu

Abstract. Binary septic forms, taken here as a case study, contain arithmetic and Galois-theoretic information. Systematic computation and storage of their invariants enable computations over large height-bounded families, with applications to Galois groups, arithmetic structures, and coding theory. This paper describes a database of binary septics with fixed arithmetic fields, including invariant computation, classification data, and arithmetic properties. We discuss the data schema, exact-arithmetic computations, and implementation checks, together with applications to Galois theory and the construction of algebraic codes derived from structured polynomial families. As a concrete application, we reduce the stored septics modulo finite fields, form the superelliptic curves and generate CSS quantum codes from their one-point algebraic-geometric codes, using the Galois group to stratify the resulting point-count and designed code data.

Keywords: Binary septics · Galois groups · Polynomial database · Superelliptic curves · Coding theory.

1 Introduction

Polynomial databases are useful when exact arithmetic, invariant theory, and later applications must be linked to the same canonical object. In this paper the object is a binary septic, or equivalently an integral polynomial

$$f(x) = a_7x^7 + a_6x^6 + \cdots + a_1x + a_0.$$

After normalization, each polynomial gives one row in the database of binary septics. The row stores the coefficient tuple, height, discriminant, invariant data, factorization checks, and the certified Galois group over $\mathbb{Q}$. The same row can then be reduced modulo finite fields and used to construct superelliptic curves

$$X_{f,m} : y^m = \overline{f}(x),$$

whose rational points provide evaluation sets for algebraic-geometric codes and CSS quantum codes.

The emphasis here is computational. The standard ingredients–superelliptic curves, Riemann–Roch spaces, AG codes, and the CSS construction–are recalled only in the form needed by the implemented computation. The contribution is the reproducible computation which passes from a canonical polynomial database to arithmetic and coding-theoretic output. The main steps are:

1. enumerate and normalize primitive height-bounded septic coefficient tuples;
2. test irreducibility, separability, discriminants, invariants, and Galois groups using exact arithmetic;
3. reduce admissible rows modulo finite fields and construct superelliptic curves;
4. compute rational point counts, one-point Riemann–Roch bases, AG-code parameters, and CSS quantum-code parameters;
5. store all intermediate fields so that the tables can be regenerated from the same database record.

The underlying database of binary septics, constructed in [6] and publicly available as [7], contains 1,686,353 unique irreducible rows. In several coding scans we focus either on the non-S_7 subdatabase of 126,396 rows or on the full database, depending on the comparison being made. The Galois labels are not treated as unexplained labels for coding, but they stratify the computed code data and show how arithmetic classes influence reductions and point counts.

The guiding question of this paper is the following. Starting from the database of septic polynomials constructed in [6], and using the graded weighted AG/QWAG framework of Shaska [9], we ask whether the arithmetic class of a septic polynomial, as measured by its Galois group over $\mathbb{Q}$, is visible in the superelliptic curves and CSS quantum codes obtained from its reductions modulo finite fields. More precisely, for admissible reductions of

$$X_{f,5}: \qquad y^5 = \overline{f}(x)$$

over $\mathbb{F}_q$, we compute point counts, one-point Riemann–Roch spaces, AG-code parameters, and designed CSS parameters. The Galois group is not assumed to determine the resulting code. Instead, it is used to stratify the computed data and to test whether different arithmetic strata exhibit different distributions of point counts and selected designed code parameters.

The background on superelliptic curves, one-point AG codes, and CSS codes is expository. The contribution of this paper is the implementation connecting the polynomial database of [6] to finite-field reductions, point-count data, one-point AG-code parameters, CSS designed parameters, and Galois-stratified computational summaries.

2 Height Search and Input Records

A polynomial is represented by a primitive integral tuple

$$(a_0, \ldots, a_7) \in \mathbb{Z}^8, \qquad \gcd(a_0, \ldots, a_7) = 1, \qquad a_7 > 0,$$

with

$$f(x) = a_7 x^7 + \cdots + a_0, \qquad H(f) = \max_i |a_i|.$$

The database is obtained by a height-bounded search over such tuples, followed by exact arithmetic filters: the polynomial is required to have degree 7, to be irreducible over $\mathbb{Q}$, and to have nonzero discriminant.

The enumeration, normalization, invariant computation, and Galois-group certification follow [6]; the present paper uses the resulting records as input for the coding-theoretic construction. The records are publicly available as [7].

The database construction is used here only as the input layer. The full record contains arithmetic invariants and Galois-certification data, as described in [6]. For the present coding-theoretic application, the required fields are the primitive tuple $(a_0, \ldots, a_7)$, the leading coefficient a_7, the discriminant Δ_f, the Galois label $\mathrm{Gal}_{\mathbb{Q}}(f)$, and the finite-field output fields

$$q, \quad N, \quad r_1, \quad r_2, \quad k_Q, \quad d_{\mathrm{des}}.$$

3 Weighted Superelliptic Curves from Septics

Let q be a prime power and let $m \geq 2$ with $\mathrm{char}(\mathbb{F}_q) \nmid m$. For a good reduction of a database polynomial, we form

$$X_{f,m}/\mathbb{F}_q: \qquad y^m = \overline{f}(x).$$

Good reduction means $q \nmid a_7 \Delta_f$. Then $\overline{f}$ has degree seven and is separable. The smooth projective model has genus

$$g = \frac{(m-1)(7-1) - (\nu - 1)}{2}, \qquad \nu = \gcd(m, 7), \tag{1}$$

by Riemann–Hurwitz [12]. In the main application we take

$$m = 5, \qquad q \in \{11, 31, 61\}.$$

Then $\gcd(5, 7) = 1$, so $g = 12$. The fields $q = 11, 31, 61$ were chosen because they are prime fields with $5 \mid (q-1)$, so the fifth-root automorphism is defined over the base field, while the computations remain small enough to regenerate and verify. There is a unique point P_∞ above infinity, and

$$\mathrm{ord}_{P_\infty}(x) = -5, \qquad \mathrm{ord}_{P_\infty}(y) = -7.$$

Throughout, N denotes the number of affine rational evaluation points,

$$N = \#\big(X_{f,5}(\mathbb{F}_q) \setminus \{P_\infty\}\big),$$

so the projective point count is $N+1$. The pole semigroup is $\langle 5, 7\rangle$, and a basis of the one-point Riemann–Roch space is

$$L(rP_\infty) = \mathrm{Span}_{\mathbb{F}_q}\{x^i y^j : i \geq 0, 0 \leq j < 5, 5i + 7j \leq r\}. \tag{2}$$

For $r > 2g - 2 = 22$, Riemann–Roch gives $\ell(rP_\infty) = r - g + 1 = r - 11$.

We use the weighted projective plane

$$\mathbb{P}(5, 7, 5) = \mathrm{Proj}\, \mathbb{F}_q[X, Y, Z], \qquad \deg X = \deg Z = 5, \quad \deg Y = 7.$$

Thus a monomial $X^i Y^j Z^k$ has weighted degree

$$5i + 7j + 5k.$$

The homogenized equation

$$Y^5 = F_7(X, Z)$$

is weighted homogeneous of degree 35, since $\deg(Y^5) = 5 \cdot 7 = 35$ and each monomial $X^i Z^{7-i}$ also has weighted degree 35. Weighted projective spaces and their use in graded AG/QWAG constructions are standard in the framework of Dolgachev, Iano-Fletcher, and Shaska [2,5,9]. The weighted model organizes the homogeneous equation and the monomial basis; the evaluation is performed on the smooth projective curve. The present paper does not reprove the weighted AG/QWAG framework of Shaska [9]; it specializes that construction to the superelliptic curves obtained from reductions of the database.

Because $5 \mid (q-1)$ for $q = 11, 31, 61$, the automorphism

$$(x, y) \longmapsto (x, \zeta_5 y)$$

is defined over the base field. Its fixed affine points are exactly those with $y = 0$, i.e. the roots of $\overline{f}$. Writing

$$r_f = \#\{a \in \mathbb{F}_q : \overline{f}(a) = 0\}$$

for the number of roots of $\overline{f}$, we have

$$N = r_f + 5\#\{a \in \mathbb{F}_q : \overline{f}(a) \neq 0, \overline{f}(a) \in (\mathbb{F}_q^\times)^5\}, \qquad N \equiv r_f \pmod 5. \tag{3}$$

This relates the factorization of $\overline{f}$, Frobenius action on roots, and the code length.

4 Weighted AG Codes and CSS Parameters

Let $P_1, \ldots, P_N$ be the affine rational points of $X_{f,5}$, and set

$$D = P_1 + \cdots + P_N.$$

Recall from (2) the graded one-point space $L(rP_\infty)$, spanned over $\mathbb{F}_q$ by the monomials x^iy^j with $i \geq 0$, $0 \leq j < 5$, and $5i + 7j \leq r$. The inequality $5i + 7j \leq r$ is the pole-order condition at P_∞; equivalently, it is the weighted-degree condition induced by the model $\mathbb{P}(5, 7, 5)$. Evaluating these functions at the affine rational points $P_1, \ldots, P_N$ gives the weighted one-point AG code

$$C_w(r) = \{(h(P_1), \ldots, h(P_N)) : \ h \in L(rP_\infty)\} \subseteq \mathbb{F}_q^N.$$

For $r > 2g - 2$, Riemann–Roch gives

$$\dim C_w(r) = \ell(rP_\infty) = r - g + 1,$$

provided the evaluation map is injective, and the Goppa designed-distance bound is

$$d(C_w(r)) \geq N - r.$$

In the scan we only use divisor degrees $r < N$. Hence $\deg(rP_\infty) < \deg D = N$, so the evaluation map is injective. These are the parameter bounds used in the scan; the weighted model supplies the grading and monomial basis, while the parameter bounds are the usual one-point AG-code bounds, cited through the graded framework of Shaska [9] and the standard AG-code references [3,12,15].

For integers r_1, r_2 satisfying the admissibility conditions

$$2g - 2 < r_2 < r_1 < N,$$

we define the nested one-point AG codes

$$C_i = C_L(D, r_iP_\infty) = C_w(r_i), \qquad i = 1, 2.$$

Since $r_2P_\infty \leq r_1P_\infty$, we have $C_2 \subset C_1$. By Riemann–Roch,

$$\dim C_i = r_i - g + 1, \qquad d(C_i) \geq N - r_i.$$

The nested CSS construction applied to $C_2 \subset C_1$ gives designed parameters

$$[\![N,\ k_Q,\ \geq d_{\text{des}}]\!]_q, \qquad k_Q = r_1 - r_2, \tag{4}$$

with

$$d_{\text{des}} = \min\{\, N - r_1,\ r_2 - (2g - 2)\,\}.$$

Following the AG-code and CSS frameworks [3,15,12,1,10,11], these designed parameters are recorded in the tables as designed-distance statements.

In the scan we use the distance-balancing line

$$r_1 + r_2 = N + 2g - 2. \tag{5}$$

This is the distance-balancing line because it makes

$$N - r_1 = r_2 - (2g - 2),$$

so the two terms defining d_{des} coincide. Writing

$$r_1 = N - s, \qquad r_2 = 2g - 2 + s,$$

we obtain

$$[\![N,\ N - 2g + 2 - 2s,\ \geq s]\!]_q. \tag{6}$$

The extremal value of s maximizes the designed distance but gives $k_Q \in \{1, 2\}$. For this reason, to compare rows in a finite scan we use the product

$$S_{\text{abs}} = k_Q d_{\text{des}}.$$

This statistic selects rows with both $k_Q > 0$ and $d_{\text{des}} > 0$; it is only a finite selection criterion. It is not an invariant of the curve; it is only a selection statistic used to choose one representative row from each Galois stratum and field. On the balanced line,

$$S_{\text{abs}}(s) = \big(N - 2g + 2 - 2s\big)s.$$

For each Galois stratum and field, the script evaluates admissible integral values of s and records a row attaining the largest value of S_{abs}.

5 Computational Results and Galois Stratification

The computations in this section should be interpreted as a Galois-stratified finite scan. The input is the septic database of [6]; the geometric construction is the weighted one-point AG/QWAG construction specialized to the curves $X_{f,5} : y^5 = \overline{f}(x)$, following the framework of [9]. For each admissible reduction, the genus is fixed at $g = 12$. Hence, after the choice of q, r_1, and r_2, the main geometric quantity varying from row to row is the affine point count

$$N = \#\big(X_{f,5}(\mathbb{F}_q) \setminus \{P_\infty\}\big).$$

The purpose of the stratification is not to prove that $\mathrm{Gal}_{\mathbb{Q}}(f)$ determines the code, but to measure whether the Galois stratum organizes the point-count data and the designed CSS parameters obtained from the reductions. Within each stratum the main geometric quantity affecting the code is N; once $g = 12$, N, and the divisor degrees r_1, r_2 are fixed, the designed CSS parameters follow from the formulas above.

5.1 Implementation and Data Flow

From the software point of view, the contribution is a SageMath-based workflow which takes normalized septic records as input and produces finite-field reductions, point counts, one-point AG-code data, CSS designed parameters, verification logs, and the summary tables below. The computation is organized as a row-by-row computation. The input database is stored as split CSV files with a common header. Each row is read, parsed as an integral coefficient tuple, and checked against the stored discriminant and Galois label. For each $q \in \{11, 31, 61\}$, the script first tests $q \nmid a_7 \Delta_f$. It then reduces f modulo q, checks squarefreeness, counts affine rational points on $y^5 = \overline{f}(x)$, builds the pole-order monomial basis $5i + 7j \leq r$, searches admissible pairs (r_1, r_2) on the balanced line, and writes the resulting CSS parameters to CSV output.

The main computational checks are:

1. row count and duplicate checks for the input database;
2. good-reduction and squarefreeness checks;
3. point-count congruence $N \equiv r_f \pmod 5$;
4. Riemann–Roch dimension checks for $L(rP_\infty)$;
5. nested-code dimension checks for $C_2 \subset C_1$;
6. regeneration of Tables 1, 2, and 3 from the output files.

5.2 Admissible Rows

The full $m = 5$ scan used the weighted model $\mathbb{P}(5, 7, 5)$ and the fields $q = 11, 31, 61$. A row was retained only when the reduction was squarefree of degree seven, the evaluation set was nonempty, and the one-point divisor degrees satisfied $r_1 > r_2 > 22$ on the distance-balancing line (5). Table 1 is obtained by applying the same SageMath [14] workflow to every admissible database row and retaining only rows passing the good-reduction, squarefreeness, divisor, and balanced-line checks. The retained CSS row counts are shown in Table 1. For every retained

Table 1. Retained CSS rows for the $m = 5$, $g = 12$, $\mathbb{P}(5, 7, 5)$ scan.

Field	retained CSS rows
$\mathbb{F}_{11}$	871,090
$\mathbb{F}_{31}$	1,144,311
$\mathbb{F}_{61}$	1,168,425
Total	3,183,826

row, the genus is fixed at $g = 12$, so the main varying geometric input is the affine point count $N = \#\big(X_{f,5}(\mathbb{F}_q) \setminus \{P_\infty\}\big)$. Once N, r_1, r_2 are fixed, the dimensions and designed distances follow from (4).

5.3 Selected Code Parameters by Galois Group

Table 2 records the selected row in each Galois stratum and field. The selection criterion is the largest value of $S_{\mathrm{abs}} = k_Q d_{\mathrm{des}}$ among the retained product-balanced CSS rows in that stratum. The table is therefore a finite computational summary, not a classification theorem and not a claim of global optimality.

Table 2. Selected designed CSS codes by Galois stratum. The number in parentheses is $S_{\mathrm{abs}} = k_Q d_{\mathrm{des}}$. Each entry is selected among retained product-balanced rows in the corresponding stratum

$\mathrm{Gal}_{\mathbb{Q}}(f)$	$q = 11$	$q = 31$	$q = 61$
S_7	$[\![46, 12, \geq 6]\!]_{11}$ (72)	$[\![95, 37, \geq 18]\!]_{31}$ (666)	$[\![145, 61, \geq 31]\!]_{61}$ (1891)
A_7	$[\![45, 11, \geq 6]\!]_{11}$ (66)	$[\![80, 28, \geq 15]\!]_{31}$ (420)	$[\![131, 55, \geq 27]\!]_{61}$ (1485)
$\mathrm{PSL}(3, 2)$	$[\![45, 11, \geq 6]\!]_{11}$ (66)	$[\![91, 35, \geq 17]\!]_{31}$ (595)	$[\![130, 54, \geq 27]\!]_{61}$ (1458)
$C_7 \rtimes C_3$	$[\![31, 5, \geq 2]\!]_{11}$ (10)	$[\![76, 26, \geq 14]\!]_{31}$ (364)	$[\![121, 49, \geq 25]\!]_{61}$ (1225)
D_7	$[\![36, 6, \geq 4]\!]_{11}$ (24)	$[\![80, 28, \geq 15]\!]_{31}$ (420)	$[\![111, 45, \geq 22]\!]_{61}$ (990)
$C_7 \rtimes C_6$	$[\![31, 5, \geq 2]\!]_{11}$ (10)	$[\![66, 22, \geq 11]\!]_{31}$ (242)	$[\![111, 45, \geq 22]\!]_{61}$ (990)
C_7	$[\![35, 7, \geq 3]\!]_{11}$ (21)	$[\![60, 18, \geq 10]\!]_{31}$ (180)	$[\![105, 41, \geq 21]\!]_{61}$ (861)

The comparison shows that the selected value of S_{abs} increases substantially with the field size in every Galois stratum. For instance, in the S_7-stratum the selected products are

$$72, \qquad 666, \qquad 1891$$

over $\mathbb{F}_{11}$, $\mathbb{F}_{31}$, and $\mathbb{F}_{61}$, respectively. This reflects the role of the affine point count N: since $g = 12$ is fixed, larger values of N allow larger admissible values of r_1, r_2, and hence larger selected values of both k_Q and d_{des}.

The congruence $N \equiv r_f \pmod 5$ of (3) also gives a visible arithmetic check on the computation. For example, over $\mathbb{F}_{61}$ the $C_7 \rtimes C_3$ stratum is concentrated in the class $N \equiv 1 \pmod 5$: in the scan, 24,369 of 24,395 retained rows in this stratum have $N \equiv 1 \pmod 5$. This provides a consistency check for the Galois-stratified point-count data. Here r_f is the number of roots of $\overline{f}$ over $\mathbb{F}_q$, and the congruence follows from the $(x, y) \mapsto (x, \zeta_5 y)$ action of (3).

For a designed quantum parameter set $[\![N, k_Q, \geq d_{\mathrm{des}}]\!]_q$, we define the designed Singleton defect by

$$\sigma_Q^{\mathrm{des}} = N + 2 - k_Q - 2d_{\mathrm{des}}.$$

The largest selected product in this scan occurs over $\mathbb{F}_{61}$ in the S_7 stratum, giving $[\![145, 61, \geq 31]\!]_{61}$ (Table 2), with designed Singleton defect $\sigma_Q^{\mathrm{des}} = (145 + 2) - (61 + 2 \cdot 31) = 24 = 2g$. This illustrates the role of the point count: with the genus fixed, larger N permits larger admissible r_1, r_2, and hence stronger selected parameters within the same one-point family.

5.4 Variance Explained by the Galois Stratum

The selected-row comparison above is qualitative. To quantify how much of the variation in a designed code quantity Y is accounted for by the Galois stratum, we use the correlation ratio

$$\eta^2(G, Y) = \frac{\sum_H \#\mathcal{D}_H \, (\overline{Y}_H - \overline{Y})^2}{\sum_{x \in \mathcal{D}} \big(Y(x) - \overline{Y}\big)^2},$$

where H runs over the Galois strata, $\mathcal{D}_H \subseteq \mathcal{D}$ is the part of the admissible scan with Galois group H, $\overline{Y}_H$ is the mean of Y over $\mathcal{D}_H$, and $\overline{Y}$ is the overall mean. A larger value of $\eta^2(G, Y)$ means that a larger fraction of the variation of Y is explained by the stratum. We apply this to the designed quantities

$$R_{\mathrm{des}} = \frac{k_Q}{N}, \qquad \delta_{\mathrm{des}} = \frac{d_{\mathrm{des}}}{N}, \qquad S_{\mathrm{abs}} = k_Q d_{\mathrm{des}}, \qquad S_{\mathrm{rel}} = R_{\mathrm{des}} \, \delta_{\mathrm{des}},$$

Table 3. Correlation ratio $\eta^2(G,Y)$: the fraction of the variation in a designed code quantity Y explained by the Galois stratum, on the full admissible scan $\mathcal{D}_{\mathrm{full}}$ and the non-S_7 scan $\mathcal{D}_{\neq S_7}$. For each scan and field the displayed Y is the designed quantity with the most pronounced stratum dependence.

q	scan	Y	$\eta^2(G,Y)$
11	$\mathcal{D}_{\mathrm{full}}$	S_{rel}	0.010524
31	$\mathcal{D}_{\mathrm{full}}$	S_{rel}	0.008112
61	$\mathcal{D}_{\mathrm{full}}$	S_{abs}	0.001414
11	$\mathcal{D}_{\neq S_7}$	S_{rel}	0.048970
31	$\mathcal{D}_{\neq S_7}$	S_{rel}	0.052995
61	$\mathcal{D}_{\neq S_7}$	δ_{des}	0.013936

evaluated both on the full admissible scan $\mathcal{D}_{\mathrm{full}}$ (3,183,826 constructions) and on its non-S_7 part $\mathcal{D}_{\neq S_7}$ (333,964 constructions), since the full scan is dominated by the generic stratum S_7.

In the full scan the values are small, reflecting the dominance of the S_7-stratum. After removing S_7, the stratification becomes more visible, especially for S_{rel} over $\mathbb{F}_{11}$ and $\mathbb{F}_{31}$. This is a finite statistical statement about the constructed database; no implication that the Galois group determines the parameters is claimed.

6 Verification, Data, and Software Availability

The verification checks used for the coding scan are: uniqueness of the input row, good reduction $q \nmid a_7 \Delta_f$, squarefreeness of $\overline{f}$, point-count consistency, Riemann–Roch dimension checks for the monomial bases, nested-code dimension checks, and regeneration of Tables 1 and 2 from the stored output files.

Data and software availability. The source code and sample files used to regenerate Tables 1 and 2 are available in the accompanying repository [8]. The underlying database of binary septics is publicly available [7]. The code repository [8] contains input rows, sample QWAG output rows, SageMath scripts, verification logs, and software version information. In particular it records the SageMath [14], GAP [13], Python, and package versions used in the computations.

7 Outlook

The computations show three features. First, the septic database supplies a large family of admissible reductions from which weighted one-point AG codes and CSS designed parameters can be computed uniformly. Second, within the fixed family $m = 5$, $g = 12$, and $q \in \{11, 31, 61\}$, the selected designed parameters improve as the affine point count N increases; in particular, larger fields allow larger admissible divisor degrees and hence larger selected values of k_Q, d_{des}, and $S_{\mathrm{abs}} = k_Q d_{\mathrm{des}}$. Third, the Galois group of the original polynomial provides a meaningful arithmetic stratification of the computed data. The numerical values of $\eta^2(G,Y)$ indicate that this dependence is weak on the full scan, where the S_7-stratum dominates, but becomes more visible after removing the generic S_7 class. Thus the computations support a stratified experimental relation between Galois data, point counts, and designed CSS parameters; they do not assert that the Galois group alone determines the resulting quantum code.

The computed records allow comparisons among arithmetic strata, reductions, and code parameters. The present scan shows that superelliptic curves attached to septic polynomials produce CSS designed parameters with $k_Q > 0$ and $d_{\mathrm{des}} > 0$ inside the selected one-point family, and that Galois strata organize the point counts appearing in the code construction. Future work includes larger prime sets, exact-distance searches for more selected rows, comparison with quantum-code tables such as [4], and extension to higher-degree polynomial databases.

Disclosure of Interests. The author has no competing interests to declare that are relevant to the content of this article.

References

1. Calderbank, A.R., Shor, P.W.: Good quantum error-correcting codes exist. Physical Review A **54**(2), 1098–1105 (1996). https://doi.org/10.1103/PhysRevA.54.1098
2. Dolgachev, I.V.: Weighted projective varieties. In: Group Actions and Vector Fields, Lecture Notes in Mathematics, vol. 956, pp. 34–71. Springer (1982)
3. Goppa, V.D.: Codes on algebraic curves. Soviet Mathematics Doklady **24**, 170–172 (1981)
4. Grassl, M.: Bounds on the minimum distance of linear codes and quantum codes (2007), online tables, http://www.codetables.de
5. Iano-Fletcher, A.R.: Working with weighted complete intersections. In: Explicit Birational Geometry of 3-Folds, London Mathematical Society Lecture Note Series, vol. 281, pp. 101–173. Cambridge University Press (2000)
6. Mezinaj, J.: From polynomials to databases: arithmetic structures in Galois theory. Albanian Journal of Mathematics **19**(1), 3–35 (2025). https://doi.org/10.51286/albjm/1764087998
7. Mezinaj, J.: AIMS7: A database of degree-seven polynomials and Galois groups. GitHub repository (2026), https://github.com/YuriMezini/AIMS-7-data
8. Mezinaj, J.: QWAG-Superelliptic: Code for quantum weighted AG codes from septic reductions. GitHub repository (2026), https://github.com/YuriMezini/QWAG-Superelliptic
9. Shaska, T.: Graded quantum codes from weighted projective spaces (2026), manuscript
10. Shor, P.W.: Scheme for reducing decoherence in quantum computer memory. Physical Review A **52**(4), R2493–R2496 (1995). https://doi.org/10.1103/PhysRevA.52.R2493
11. Steane, A.M.: Error correcting codes in quantum theory. Physical Review Letters **77**(5), 793–797 (1996). https://doi.org/10.1103/PhysRevLett.77.793
12. Stichtenoth, H.: Algebraic Function Fields and Codes, Graduate Texts in Mathematics, vol. 254. Springer, 2 edn. (2009)
13. The GAP Group: GAP – Groups, Algorithms, and Programming (2025), version 4.13.1, https://www.gap-system.org
14. The Sage Developers: SageMath, the sage mathematics software system (2025), version 10.4, https://www.sagemath.org
15. Tsfasman, M.A., Vladut, S.G.: Algebraic-Geometric Codes, Mathematics and its Applications, vol. 58. Kluwer (1991)