EDUSJ, Vol, 35, No: 3, 2026 (40 - 49)

Journal of Education and Science (ISSN 1812-125X)

Incidence Matrix Based Construction of Linear Error-Correcting Codes from PG(9,q) and PG(10,q)

A. N. M. S. Ibrahim [1*], N. Y. K. Yahya [2]

[1,2] Department of Mathematics, College of Education for Pure Science, University of Mosul, Iraq

Article information

Article history:
Received: February 08, 2026
Revised: April 06, 2026
Accepted: April 29, 2026
Available online: July 01, 2026

Keywords:
Projective Linear Codes
Finite Fields
Non-Perfect Codes
Error-Correcting Codes

Correspondence:
Abdulmuttalib N. M. S Ibrahim
abdulmutalib.24esp20@student.uomosul.edu.iq

Abstract

Error-correcting codes over finite projective geometries are of significant value in ensuring data reliability. However, because higher-dimensional spaces are far too computationally complex to tinker with, most research has been concentrated on lower dimensions. To help bridge this gap, this paper constructs new projective linear codes directly out of the high-dimensional spaces $PG(9,3)$ and $PG(10,2)$. We generated the companion matrices of primitive polynomials to construct points and subspaces systematically. Then, we constructed point-hyperplane incidence matrices and processed them computationally in MATLAB to determine the basic parameters of our new codes. Through this process, we obtained two specific codes: a ternary code from $PG(9,3)$ with parameters. $[29524, k < 14317, 9841]_3$ that can correct up to 4920 errors, and a binary code from $PG(10,2)$ with parameters $[2047, k < 393, 1023]_2$ that can correct up to 511 errors. By applying the Sphere Packing Bound, it turned out that although these codes are non-perfect, they nevertheless possess an extremely high error-correcting capacity. Overall, we were able to assemble a workable algebraic scheme that transforms the complex structure of higher-dimensional spaces into reliable error-correcting codes, offering useful theoretical insights beyond the familiar low-dimensional models.

DOI:10.33899/jes.v35i3.61424,

1. Introduction

Coding theory is an important area of modern mathematics. Researchers have shown a lot of interest in its relationship with projective spaces over finite fields. Over the last few decades, several studies have demonstrated that finite projective geometry is a great construction and analysis tool to apply to linear codes.

For example, Hirschfeld [1], [2] provided the necessary theorems and definitions of this link. The fundamental tools and ideas required in coding theory were systematized by Hill [3]. Al-Seraji [4], [5] discussed the projective plane of order 17, relating it to error-correcting codes and finding interesting results. Finite geometries were discussed by Greferath and Szonyi [6] in 2020 at a general level. Recently, Al-Zangana [7] used the geometry of the plane of order 27 to produce certain NMDS codes. Also, Yahya [8], [9], [10], [11], [12], [13] explored different uses of finite projective spaces in coding theory, providing new theoretical perspectives.

Despite this useful work, the bulk of the available literature concentrates on the case of low-dimensional spaces, such as 5 or 6. High-dimensional geometries (e.g., 9 or 10) are rarely researched, especially over finite fields F_3 and F_2.We are mostly interested in these higher dimensions as they have an enormous count of points and subspaces. They thus enable us to generate codes of far larger lengths (m) and error-correcting capacity (e) to serve current system demands. Moreover, we did not study a general arbitrary dimension n because it is too abstract and computationally impossible to compute matrices explicitly. Instead, we focused on dimensions 9 and 10. Such a decision allows us to use realistic computations and provide definite algorithms. It is this particular gap that this paper is trying to fill.

In order to build our linear codes, we shall consider the geometries of $PG(9,3)$ and $PG(10,2)$. We are concerned with exploring how large spaces may help to develop reliable codes, and how it is possible to define their precise parameters. We do this by using the companion matrices of primitive polynomials over. F_3 and F_2. Incidence matrices are then built. This approach is highly effective because the companion matrix provides a systematic construction of the entire space, and complex geometrical objects are converted into simple numerical information.

By processing these matrices, we can get the core code parameters: length (m), dimension (k), minimum distance (d), and number of correctable errors (e). Lastly, we apply the Sphere Packing Bound (Hamming bound) to estimate the highest limit of k and to check whether our codes are perfect. Hopefully, this paper presents viable methods for extracting strong codes based on high-dimensional systems.

The remaining part of this paper is organized in the following way: Section 2 overviews the mathematical background of projective spaces and coding theory. The description of our algorithm and the step-by-step processes of building codes from $PG(9,3)$ and $PG(10,2)$ can be found in Section 3. Section 4 discusses the obtained results. Lastly, the conclusion of our findings is presented in Section 5.

2. Mathematical Preliminaries

The following section gives the basic definitions needed for our study. We principally focus on projective geometry concepts and coding theory concepts. More detailed information is available to the readers in [1], [2].

Definition 2.1 [1]**:** We assume F_q to be a finite field of order q. It is possible to obtain the vector space $V(n+1,q)$ based on this. The n-dimensional projective space, which we denote by $PG(n,q)$, is formed by taking all the one-dimensional vector subspaces of $V(n+1,q)$. The number of points in this projective space is then determined by simply using the formula $\theta(n) = \frac{q^{n+1}-1}{q-1}$.

Definition 2.2 (Projective Subspaces) [1]**:** A projective subspace of dimension k in $PG(n,q)$ corresponds geometrically to a $(k+1)$-dimensional vector subspace of $V(n+1,q)$. If we take the points of $PG(n,q)$ and force their last $n-k$ coordinates to be equal to zero, we are naturally restricting the vectors to a $(k+1)$-dimensional subspace. Geometrically, this operation formally defines a k-dimensional projective subspace (such as a line when $k=1$, a plane when $k=2$, or a hyperplane when $k=n-1$).

Remark 2.3: Given a primitive polynomial, its companion matrix serves as a Singer cycle in $PG(n,q)$. It naturally produces all the points and hyperplanes in one single orbit of length $\theta(n)$. The number of intermediate k-subspaces (where $0 < k < n-1$) is, however, far more numerous than $\theta(n)$, and they fall into several individual orbits [1], [2]. In our construction, we have not enumerated all of the possible intermediate subspaces. Rather, we were looking at creating exactly one orbit for each k-subspace. The length of this chosen orbit is also $\theta(n)$, and we constructed it by just taking a base subspace—where the final $n-k$ coordinates are zeros—and successively multiplying it by the companion matrix.

Definition 2.4 [2]**:** When we have a subspace of F_q^m, we call it a linear code C. The normal way to write its parameters is $[m,k,d]_q$. In this notation, m is how long the code is, k is the dimension, and d is the minimum distance. Also, the total count of codewords is exactly $M = q^k$.

Definition 2.5 [2]**:** Given two vectors u and v in F_q^m, the Hamming distance between them, denoted as $d(u,v)$, is simply the number of places they disagree. The minimum distance d is the closest Hamming distance, which you can see when you compare pairs of different codewords of a linear code C. So, we write this as: $d = mim\,\{d(u,v) : u,v \in C, u \neq v\}$.

Theorem 2.6 (The Sphere Packing Bound) [2]**:** If a q-ary code has parameters (m,M,d) and it can fix up to $e = \left\lfloor\frac{d-1}{2}\right\rfloor$ errors, then the number of codewords M cannot pass a certain limit. This limit is:

$$M\sum_{i=0}^{e}\binom{m}{i}(q-1)^i \le q^m$$

Corollary 2.7 [2]**:** A code C is known as a Perfect Code only when the inequality in Theorem 2.6 is actually an equal sign.

3. Construction of Projective Geometry Codes and Algorithms

In this section, we apply our theoretical concepts to construct linear codes from the high-dimensional projective spaces $PG(9,3)$ and $PG(10,2)$. To handle the massive geometric generation and matrix computations, we utilized MATLAB.

3.1 Computational Algorithm

To ensure the reproducibility of our results, the following brief algorithm summarizes the computational steps implemented in MATLAB:

Algorithm: Code Generation and Parameter Calculation

1. **Input:** Dimension n, Field order q, and Primitive polynomial f.
2. **Initialization:** Construct the companion matrix $C(f)$. Set the first point $\mathcal{P}_1 = [1, 0, 0, 0, \dots, 0, 0]$.

3. **Space Generation:** Iteratively compute all points $\mathrm{P}_i = \mathrm{P}_1 \left(C(f)\right)^{i-1}$ or $\mathrm{P}_i = \mathrm{P}_{i-1} \left(C(f)\right)$.
4. **Subspace Orbit:** Identify the base subspace (setting the last coordinates to zero) and multiply it successively by $C(f)$ to generate its specific orbit.
5. **Incidence Matrix:** Construct the matrix M based on the intersection between points and the generated hyperplanes.
6. **Code Extraction:** Shift M by scalar vectors to define all possible codewords, then compute the minimum Hamming distance d.
7. **Output:** Calculate errors $e = \left\lfloor\frac{d-1}{2}\right\rfloor$, estimate k via Sphere Packing Bound, and return the final parameters $[m,k,d]_q$.

3.2 Construction of Codes from $PG(9,3)$

The Polynomial $\phi(v) = v^{10} - v^6 - v^5 - v^4 - 2v - 1$ is a primitive polynomial over F_3. Let $C(\phi)$ be its companion matrix in F_3, which can be represented in block form as follows:

$$C(\phi) = \begin{pmatrix} 0 & & & & & & & & & \\ \vdots & & & & & I_9 & & & & \\ 0 & & & & & & & & & \\ 1 & 2 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

where I_9 is the 9×9 identity matrix. Let the initial point be $\mathrm{P}_1 = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$. Using this companion matrix, we systematically generate all the points of the projective space $PG(9,3)$ using the following formula: $\mathrm{P}_i = \mathrm{P}_1 \left(C(\phi)\right)^{i-1}$ for i =1, 2, ... , 29524.

Table 1. The points of $PG(9,3)$

I	P_i	i	P_i
1	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]	19686	[0 , 1, 2, 2, 0, 2, 2, 2, 2, 1]
2	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0]	19687	[2 , 1, 2, 1, 0, 2, 0, 1, 1, 1]
3	[0, 0, 1, 0, 0, 0, 0, 0, 0, 0]	19688	[1 , 1, 1, 2, 2, 1, 0, 0, 1, 1]
4	[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]	19689	[1, 0, 1, 1, 0, 0, 2, 0, 0, 1]
5	[0, 0, 0, 0, 1, 0, 0, 0, 0, 0]	19690	[1, 0, 0, 1, 2, 1, 1, 2, 0, 0]
6	[0, 0, 0, 0, 0, 1, 0, 0, 0, 0]	19691	[0, 1, 0, 0, 1, 2, 1, 1, 2, 0]
7	[0, 0, 0, 0, 0, 0, 1, 0, 0, 0]	19692	[0, 0, 2, 0, 0, 2, 1, 2, 2, 1]
8	[0, 0, 0, 0, 0, 0, 0, 1, 0, 0]	19693	[2, 1, 0, 1, 2, 2, 0, 2, 1, 1]
9	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]	19694	[1, 1, 1, 0, 2, 0, 0, 0, 2, 1]
10	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1]	19695	[2, 0, 2, 2, 2, 0, 2, 0, 0, 1]
11	[1, 2, 0, 0, 1, 1, 1, 0, 0, 0]	19696	[1, 1, 0, 2, 0, 0, 1, 2, 0, 0]
⋮	⋮	⋮	⋮
19682	[1 , 2, 2, 1, 0, 1, 0, 2, 0, 1]	29521	[0 , 1, 0, 0, 0, 2, 2, 2, 0, 1]
19683	[1 , 0, 2, 2, 2, 1, 2, 0, 2, 0]	29522	[1 , 2, 1, 0, 1, 2, 0, 1, 1, 1]
19684	[0 , 2, 0, 1, 1, 1, 2, 1, 0, 1]	29523	[1 , 0, 2, 1, 1, 2, 0, 0, 1, 1]
19685	[1 , 2, 2, 0, 2, 2, 2, 2, 1, 0]	29524	[1 , 0, 0, 2, 2, 2, 0, 0, 0, 1]

From the generated points of $PG(9,3)$, we select the points where the last $9-\alpha$ coordinates are exactly zero. We denote this base collection as α-$subspace_1$, where α =1, 2, …, 8 represents the projective dimension. This definition is formulated as follows:

α-$subspace_1 = \{\mathrm{P}[x_1, x_2, \dots, x_{10}] \in PG(9,3) : x_s = 0, s = \alpha+2, \dots, 10\}$

To illustrate this, we explicitly write the base representations for the line, the plane, and the hyperplane:

1-$subspace_1(line_1) = \{\mathrm{P}[x_1, x_2, 0,0,0,0,0,0,0,0] \in PG(9,3)\}$

2-$subspace_1(plane_1) = \{\mathrm{P}[x_1, x_2, x_3, 0,0,0,0,0,0,0] \in PG(9,3)\}$

⋮ ⋮

8-$subspace_1(hyperplane_1) = \{\mathrm{P}[x_1, x_2, \dots, x_9, 0] \in PG(9,3)\}$

The subsequent subspaces in the orbit, denoted by α-$subspace_i$, are generated systematically by multiplying the base α-$subspace_1$ by the companion matrix $C(\phi)$:

α-$subspace_i = \alpha$-$subspace_1\left(C(\phi)\right)^{i-1}$ where α =1, 2, …, 8 and i =1, 2, …, 29524.

Table 2. The $\alpha\text{-}subspace_{\mathrm{i}}$ Generated by $\alpha\text{-}subspace_1$ in $PG(9,3)$.

$\boldsymbol{\alpha}$**-subspace**$_{\mathbf{i}}$	$\boldsymbol{\lambda} = \text{Ƥ}_{\boldsymbol{\lambda}}[\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_{10}}]$
$1\text{-}subspace_1$	1, 2, 5013, 8099
$1\text{-}subspace_2$	2, 3, 5014, 8100
$1\text{-}subspace_3$	3, 4, 5015, 8101
⋮	⋮
$1\text{-}subspace_{29524}$	29524, 1, 5012, 8098
$2\text{-}subspace_1$	1, 2, 3, 5013, 5014, 8099, 8100, 10025, 10517, ... , 16197
$2\text{-}subspace_2$	2, 3, 4, 5014, 5015, 8100, 8101, 10026, 10518, ... , 16198
$2\text{-}subspace_3$	3, 4, 5, 5015, 5016, 8101, 8102, 10027, 10519, ... , 16199
⋮	⋮
$2\text{-}subspace_{29524}$	29524, 1, 2, 5012, 5013, 8098, 8099, 10024,13110, ... , 16196
$3\text{-}subspace_1$	1, 2, 3, 4, 5013, 5014, 5015, 8099, 8100, ... , 25340
$3\text{-}subspace_2$	2, 3, 4, 5, 5014, 5015, 5016, 8100, 8101, ... , 25341
$3\text{-}subspace_3$	3, 4, 5, 6, 5015, 5016, 5017, 8101, 8102, ... , 25342
⋮	⋮
$3\text{-}subspace_{29524}$	29524, 1, 2, 3, 5012, 5013, 5014, 8098, 8100, ... , 25339
$4\text{-}subspace_1$	1, 2, 3, 4, 5, 387, 828, 921, 1082, 2278, ..., 29307
$4\text{-}subspace_2$	2, 3, 4, 5, 6, 388, 829, 922, 1083, 2279, ..., 29308
$4\text{-}subspace_3$	3, 4, 5, 6, 7, 389, 830, 923, 1084, 2280, ..., 29309
⋮	⋮
$4\text{-}subspace_{29524}$	29524, 1, 2, 3, 4, 386, 827, 920, 1081, 2277, ..., 29306
$5\text{-}subspace_1$	1, 2, 3, 4, 5, 6, 262, 287, 387, 388, ..., 29491
$5\text{-}subspace_2$	2, 3, 4, 5, 6, 7, 263, 288, 388, 389, ..., 29492
$5\text{-}subspace_3$	3, 4, 5, 6, 7, 8, 264, 289, 389, 390, ..., 29493
⋮	⋮
$5\text{-}subspace_{29524}$	29524, 1, 2, 3, 4, 5, 261, 286, 386, 387, ..., 29490
$6\text{-}subspace_1$	1, 2, 3, 4, 5, 6, 7, 11, 50, 63, ... , 29492
$6\text{-}subspace_2$	2, 3, 4, 5, 6, 7, 8, 12, 51, 64, ... , 29493
$6\text{-}subspace_3$	3, 4, 5, 6, 7, 8, 9, 13, 52, 65, ... , 29494
⋮	⋮
$6\text{-}subspace_{29524}$	29524, 1, 2, 3, 4, 5, 6, 10, 49, 62, ... , 29491
$7\text{-}subspace_1$	1, 2, 3, 4, 5, 6, 7, 8, 11, 12, ... , 29512
$7\text{-}subspace_2$	2, 3, 4, 5, 6, 7, 8, 9, 12, 13, ... , 29513
$7\text{-}subspace_3$	3, 4, 5, 6, 7, 8, 9, 10, 13, 14, ... , 29514
⋮	⋮
$7\text{-}subspace_{29524}$	29524, 1, 2, 3, 4, 5, 6, 7, 10, 11, ... , 29511
$8\text{-}subspace_1$	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, ... , 29520
$8\text{-}subspace_2$	2, 3, 4, 5, 6, 7, 8, 9, 10, 12, ... , 29521
$8\text{-}subspace_3$	3, 4, 5, 6, 7, 8, 9, 10, 11, 13, ... , 29522
⋮	⋮
$8\text{-}subspace_{29524}$	29524, 1, 2, 3, 4, 5, 6, 7, 8, 10, ... , 29519

Theorem 3.1: The projective space $PG(9,3)$ constructs a non-perfect linear code with parameters $[m = 29524, k < 14317, d = 9841]_3$.

Proof: Let $\mathcal{A}_0 = (a_{\mathrm{ij}}) = \begin{cases} 1 & if \quad \text{Ƥ}_{\mathrm{j}} \in \mathcal{A}_{0,\mathrm{i}} \\ 0 & if \quad \text{Ƥ}_{\mathrm{j}} \notin \mathcal{A}_{0,\mathrm{i}} \end{cases}$ here, $\mathcal{A}_0$ is the incidence matrix between points (Ƥ_{j}) and hyperplanes ($\mathcal{A}_{0,\mathrm{i}}$), with dimension 29524×29524, where $\mathcal{A}_{0,\mathrm{i}} = hyperplane_{\mathrm{i}} = 8\text{-}subspace_{\mathrm{i}}$ for i = 1, 2, ... , 29524, j = 1, 2, ... , 29524.

Table 3. Incidence matrix $\mathcal{A}_0$

$\mathcal{A}_{0,\mathrm{i}}$ \ P_{j}	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	…	P_{29524}
$\mathcal{A}_{0,1}$	1	1	1	1	1	1	1	1	1	0	...	0
$\mathcal{A}_{0,2}$	0	1	1	1	1	1	1	1	1	1	...	0
$\mathcal{A}_{0,3}$	0	0	1	1	1	1	1	1	1	1	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\mathcal{A}_{0,29524}$	1	1	1	1	1	1	1	1	0	1	...	1

To generate the shifted matrices, we use the base rows of our incidence matrix $\mathcal{A}_0$, which we denote by $\mathcal{A}_{0,\mathrm{i}}$. Then, we introduce three specific shift vectors of length 29524: $\mathcal{R}_0$ =[0, 0, 0, 0, ..., 0, 0], $\mathcal{R}_1$ =[1, 1, … , 1], and $\mathcal{R}_2$ =[2 , 2 , ... , 2]. We systematically generate the new shifted rows by applying modulo 3 addition. The mathematical formula for this step is: $\mathcal{A}_t = \{\mathcal{A}_{t,\mathrm{i}}: \mathcal{A}_{t,\mathrm{i}} = (\mathcal{A}_{0,\mathrm{i}} + \mathcal{R}_t)(mod 3)\}$ where $t \in \{0, 1, 2\}$ and i =1, 2, ... , 29524. Finally, the collection of these shifted rows forms our completely shifted matrices $\mathcal{A}_1$ and $\mathcal{A}_2$.

Table 4. Shifted matrix $\mathcal{A}_1$

$\mathcal{A}_{1,1}$	2	2	2	2	2	2	2	2	2	1	...	1
$\mathcal{A}_{1,2}$	1	2	2	2	2	2	2	2	2	2	...	1
$\mathcal{A}_{1,3}$	1	1	2	2	2	2	2	2	2	2	...	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\mathcal{A}_{1,29524}$	2	2	2	2	2	2	2	2	1	2	...	2

Table 5. Shifted matrix $\mathcal{A}_2$

$\mathcal{A}_{2,1}$	0	0	0	0	0	0	0	0	0	2	...	2
$\mathcal{A}_{2,2}$	2	0	0	0	0	0	0	0	0	0	...	2
$\mathcal{A}_{2,3}$	2	2	0	0	0	0	0	0	0	0	...	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\mathcal{A}_{2,29524}$	0	0	0	0	0	0	0	0	2	0	...	0

Let C denote the linear code generated by all possible linear combinations of the matrices $\mathcal{A}_t$ and the corresponding vectors $\mathcal{R}_t$ over F_3 where $t \in \{0, 1, 2\}$. The Hamming distance between any two distinct codewords $v, u \in C$ is mathematically defined as: $d(v, u) = |\{\mathrm{s}: v_\mathrm{s} \neq u_\mathrm{s}\}|$, where s =1, 2, ... , 29524 represents the coordinate index.

Table 6. Hamming distance between two codewords in C.

$d(\mathcal{R}_t, \mathcal{R}_r) = 29524, t \neq r$	$d(\mathcal{A}_{t,\mathrm{i}}, \mathcal{A}_{r,\mathrm{i}}) = 29524, t \neq r$
$d(\mathcal{R}_t, \mathcal{A}_{t,\mathrm{i}}) = 9841$	$d(\mathcal{A}_{t,\mathrm{i}}, \mathcal{A}_{t,\mathrm{j}}) = 13122$, i ≠ j
$d(\mathcal{R}_1, \mathcal{A}_{2,\mathrm{i}}) = 19683$	$d(\mathcal{A}_{0,\mathrm{i}}, \mathcal{A}_{1,\mathrm{j}}) = 22963$, i ≠ j
$d(\mathcal{R}_0, \mathcal{A}_{2,\mathrm{i}}) = 19683$	$d(\mathcal{A}_{0,\mathrm{i}}, \mathcal{A}_{2,\mathrm{j}}) = 22963$, i ≠ j
$d(\mathcal{R}_0, \mathcal{A}_{1,\mathrm{i}}) = 29524$	$d(\mathcal{A}_{1,\mathrm{i}}, \mathcal{A}_{2,\mathrm{j}}) = 22963$, i ≠ j

From the results above, the minimum distance of the code C is determined as:
$d = d(C) = min\{29524, 9841, 19683, 13122, 22963\} = 9841$

Consequently, the error-correcting capability e is calculated as: $e = \left\lfloor \frac{d-1}{2} \right\rfloor = \left\lfloor \frac{9841-1}{2} \right\rfloor = 4920$

The length of the code is $m = \frac{3^{10}-1}{3-1} = 29524$, and the total number of codewords is $M = 3^k$. By substituting the values of m, M, and e into the inequality of the Sphere Packing Bound (Theorem 2.6), we obtain: $M\left[\binom{m}{0} + \binom{m}{1}(q-1) + \cdots + \binom{m}{e}(q-1)^e\right] \leq q^m$

Let c represent the volume of the Hamming sphere, defined as:
$c = \binom{m}{0} + \binom{m}{1}(q-1) + \cdots + \binom{m}{e}(q-1)^e \Rightarrow c = 1 + 29524(2) + \cdots + \binom{29524}{4920}(2)^{4920}$

Substituting c back into the main inequality yields: $3^k\ c \leq 3^{29524}$

Solving for the dimension k, we get: $k \leq \frac{\log \frac{3^{29524}}{c}}{\log 3} \Rightarrow k \leq 14316.990745996345$

The calculated upper bound for k is approximately 14316.99, which implies that $k < 14317$. Since the calculated value is not an integer, the equality condition required for a perfect code cannot be satisfied (as the dimension k must be an integer). Therefore, by Corollary 2.7, the code C is not perfect. ■

3.3 Construction of Codes from $PG(10,2)$

The polynomial $\mho(v) = v^{11} - v^{2} - 1$ is a primitive polynomial over F_2. Let $C(\mho)$ be the companion matrix in F_2, which can be represented in block form as follows:

$$C(\mho) = \begin{pmatrix} 0 & & & & & & & & & & \\ \vdots & & & & & I_{10} & & & & & \\ 0 & & & & & & & & & & \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

where I_{10} is the 10×10 identity matrix. Let the initial point be $\mathcal{P}_1 = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$. Using this companion matrix, we systematically generate all the points of the projective space $PG(10,2)$ using the following formula: $\mathcal{P}_{\mathrm{i}} = \mathcal{P}_1 \left(C(\mho)\right)^{\mathrm{i}-1}$ for i =1, 2, ... , 2047

Table 7. The points of $PG(10,2)$

I	$\mathcal{P}_{\mathrm{i}}$	i	$\mathcal{P}_{\mathrm{i}}$
1	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	1368	[1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1]
2	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]	1369	[1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0]
3	[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]	1370	[0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0]
4	[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]	1371	[0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0]
5	[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]	1372	[0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1]
6	[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]	1373	[1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0]
7	[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]	1374	[0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0]
8	[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]	1375	[0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0]
9	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]	1376	[0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1]
10	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]	1377	[1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1]
11	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]	1378	[1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1]
12	[1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]	1379	[1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1]
13	[0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0]	1380	[1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0]
14	[0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0]	1381	[0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1]
15	[0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0]	1382	[1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0]
16	[0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0]	1383	[0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1]
17	[0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0]	1384	[1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0]
18	[0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0]	1385	[0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1]
⋮	⋮	⋮	⋮
1364	[0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0]	2044	[1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0]
1365	[0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0]	2045	[0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1]
1366	[0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0]	2046	[1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
1367	[0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1]	2047	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1]

From the generated points of $PG(10,2)$, we select the points where the last $10-\alpha$ coordinates are exactly zero. We denote this base collection as $\alpha\text{-}subspace_1$, where α =1, 2, …, 9 represents the projective dimension. This definition is formulated as follows:

$\alpha\text{-}subspace_1 = \{\mathcal{P}[x_1, x_2, \ldots, x_{11}] \in PG(10,2) : x_s = 0, s = \alpha+2, \ldots, 11\}$

To illustrate this, we explicitly write the base representations for the line, the plane, and the hyperplane:

$1\text{-}subspace_1(line_1) = \{\mathcal{P}[x_1, x_2, 0,0,0,0,0,0,0,0,0] \in PG(10,2)\}$

$2\text{-}subspace_1(plane_1) = \{\mathcal{P}[x_1, x_2, x_3, 0,0,0,0,0,0,0,0] \in PG(10,2)\}$

$\vdots \qquad\qquad\qquad \vdots$

$9\text{-}subspace_1(hyperplane_{1)} = \{\mathcal{P}[x_1, x_2, \ldots, x_{10}, 0] \in PG(10,2)\}$

The subsequent subspaces in the orbit, denoted by $\alpha\text{-}subspace_{\mathrm{i}}$, are generated systematically by multiplying the base $\alpha\text{-}subspace_1$ by the companion matrix $C(\mho)$:

$\alpha\text{-}subspace_{\mathrm{i}} = \alpha\text{-}subspace_1\left(C(\mho)\right)^{\mathrm{i}-1}$ where α =1, 2, …, 9 and i =1, 2, …, 2047.

Table 8. The α-$subspace_i$ Generated by α-$subspace_1$ in $PG(10,2)$.

α-$subspace_i$	$\lambda = \mathrm{P}_\lambda[x_1, x_2, \dots, x_{11}]$
1-$subspace_1$	1, 2, 1030
1-$subspace_2$	2, 3, 1031
1-$subspace_3$	3, 4, 1032
⋮	⋮
1-$subspace_{2047}$	2047, 1, 1029
2-$subspace_1$	1, 2, 3, 12, 1030, 1031, 1497
2-$subspace_2$	2, 3, 4, 13, 1031, 1032, 1498
2-$subspace_3$	3, 4, 5, 14, 1032, 1033, 1499
⋮	⋮
2-$subspace_{2047}$	2047, 1, 2, 11, 1029, 1030, 1496
3-$subspace_1$	1, 2, 3, 4, 12, 13, 48, 479, 1030, 1031, ... , 1913
3-$subspace_2$	2, 3, 4, 5, 13, 14, 49, 480, 1031, 1032, ... , 1914
3-$subspace_3$	3, 4, 5, 6, 14, 15, 50, 481, 1032, 1033, ... , 1915
⋮	⋮
3-$subspace_{2047}$	2047, 1, 2, 3, 11, 12, 47, 478, 1029, 1030, ... , 1912
4-$subspace_1$	1, 2, 3, 4, 5, 12, 13, 14, 23, 48, ... , 1914
4-$subspace_2$	2, 3, 4, 5, 6, 13, 14, 15, 24, 49, ... , 1915
4-$subspace_3$	3, 4, 5, 6, 7, 14, 15, 16, 25, 50, ... , 1916
⋮	⋮
4-$subspace_{2047}$	2047, 1, 2, 3, 4, 11, 12, 13, 22, 47, ... , 1913
5-$subspace_1$	1, 2, 3, 4, 5, 6, 12, 13, 14, 15, ... , 1975
5-$subspace_2$	2, 3, 4, 5, 6, 7, 13, 14, 15, 16, ... , 1976
5-$subspace_3$	3, 4, 5, 6, 7, 8, 14, 15, 16, 17, ... , 1977
⋮	⋮
5-$subspace_{2047}$	2047, 1, 2, 3, 4, 5, 11, 12, 13, 14, ... , 1974
6-$subspace_1$	1, 2, 3, 4, 5, 6, 7, 12, 13, 14, ... , 1991
6-$subspace_2$	2, 3, 4, 5, 6, 7, 8, 13, 14, 15, ... , 1992
6-$subspace_3$	3, 4, 5, 6, 7, 8, 9, 14, 15, 16, ... , 1993
⋮	⋮
6-$subspace_{2047}$	2047, 1, 2, 3, 4, 5, 6, 11, 12, 13, ... , 1990
7-$subspace_1$	1, 2, 3, 4, 5, 6, 7, 8, 12, 13, ... , 1992
7-$subspace_2$	2, 3, 4, 5, 6, 7, 8, 9, 13, 14, ... , 1993
7-$subspace_3$	3, 4, 5, 6, 7, 8, 9, 10, 14, 15, ... , 1994
⋮	⋮
7-$subspace_{2047}$	2047, 1, 2, 3, 4, 5, 6, 7, 11, 12, ... , 1991
8-$subspace_1$	1, 2, 3, 4, 5, 6, 7, 8, 9, 12, ... , 2025
8-$subspace_2$	2, 3, 4, 5, 6, 7, 8, 9, 10, 13, ... , 2026
8-$subspace_3$	3, 4, 5, 6, 7, 8, 9, 10, 11, 14, ... , 2027
⋮	⋮
8-$subspace_{2047}$	2047, 1, 2, 3, 4, 5, 6, 7, 8, 11, ... , 2024
9-$subspace_1$	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ... , 2046
9-$subspace_2$	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ... , 2047
9-$subspace_3$	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, ... , 1
⋮	⋮
9-$subspace_{2047}$	2047, 1, 2, 3, 4, 5, 6, 7, 8, 9, ... , 2045

Theorem 3.2: The projective space $PG(10,2)$ constructs a non-perfect linear code with parameters $[m = 2047, k < 393, d = 1023]_2$.

Proof: Let $\mathcal{N}_0 = (a_{ij}) = \begin{cases} 1 & if \quad \mathrm{P_j} \in \mathcal{N}_{0,\mathrm{i}} \\ 0 & if \quad \mathrm{P_j} \notin \mathcal{N}_{0,\mathrm{i}} \end{cases}$ here, $\mathcal{N}_0$ is the incidence matrix between points ($\mathrm{P_j}$) and hyperplanes ($\mathcal{N}_{0,\mathrm{i}}$), with dimension 2047×2047, where $\mathcal{N}_{0,\mathrm{i}} = hyperplane_{\mathrm{i}} = 8\text{-}subspace_{\mathrm{i}}$ for i = 1, 2, ... , 2047, j = 1, 2, ... , 2047.

Table 9. Incidence matrix $\mathcal{N}_0$

$\mathrm{P_j}$ / $\mathcal{N}_{0,\mathrm{i}}$	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	...	P_{2047}
$\mathcal{N}_{0,1}$	1	1	1	1	1	1	1	1	1	1	...	0
$\mathcal{N}_{0,2}$	0	1	1	1	1	1	1	1	1	1	...	1
$\mathcal{N}_{0,3}$	1	0	1	1	1	1	1	1	1	1	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\mathcal{N}_{0,2047}$	1	1	1	1	1	1	1	1	1	0	...	1

To generate the shifted matrices, we use the base rows of our incidence matrix $\mathcal{N}_0$, which we denote by $\mathcal{N}_{0,\mathrm{i}}$. Then, we introduce two specific shift vectors of length 2047: $\mathcal{B}_1$ =[0, 0, 0, 0, ... , 0, 0] and $\mathcal{B}_1$ =[1, 1, 1, 1, … , 1, 1]. We systematically generate the new shifted rows by applying modulo 2 addition. The mathematical formula for this step is: $\mathcal{N}_t = \{\mathcal{N}_{1,\mathrm{i}}: \mathcal{N}_{t,\mathrm{i}} = (\mathcal{N}_{0,\mathrm{i}} + \mathcal{B}_t)(mod2)\}$ where $t \in \{0, 1,\}$ and i =1, 2, ... , 2047. Finally, the collection of these shifted rows forms our completely shifted matrix $\mathcal{N}_1$.

Table 10. Shifted matrix $\mathcal{N}_1$

$\mathcal{N}_{1,1}$	0	0	0	0	0	0	0	0	0	0	...	1
$\mathcal{N}_{1,2}$	1	0	0	0	0	0	0	0	0	0	...	0
$\mathcal{N}_{1,3}$	0	1	0	0	0	0	0	0	0	0	...	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\mathcal{N}_{1,2047}$	0	0	0	0	0	0	0	0	0	1	...	0

Let C denote the linear code generated by all possible linear combinations of the matrices $\mathcal{N}_t$ and the corresponding vectors $\mathcal{N}_t$ over F_2 where $t \in \{0, 1\}$. The Hamming distance between any two distinct codewords $v, u \in C$ is mathematically defined as: $d(v, u) = |\{\mathrm{s}: v_\mathrm{s} \neq u_\mathrm{s}\}|$, where s =1, 2, ... , 2047 represents the coordinate index.

Table 11. Hamming distance between two code words in C.

$d(\mathcal{B}_0, \mathcal{B}_1) = 2047$	$d(\mathcal{B}_1, \mathcal{N}_{1,\mathrm{i}}) = 1023$
$d(\mathcal{B}_0, \mathcal{N}_{0,\mathrm{i}}) = 1023$	$d(\mathcal{N}_{0,\mathrm{i}}, \mathcal{N}_{1,\mathrm{i}}) = 2047$
$d(\mathcal{B}_0, \mathcal{N}_{1,\mathrm{i}}) = 1024$	$d(\mathcal{N}_{0,\mathrm{i}}, \mathcal{N}_{1,\mathrm{j}}) = 1023$, i ≠ j

From the results above, the minimum distance of the code C is determined as:
$d = d(C) = min\{2047, 1023, 1024\} = 1023$

Consequently, the error-correcting capability $\boldsymbol{e}$ is calculated as: $e = \left\lfloor \frac{d-1}{2} \right\rfloor = \left\lfloor \frac{1023-1}{2} \right\rfloor = 511$

The length of the code is $m = \frac{2^{11}-1}{2-1} = 2047$, and the total number of codewords is $M = 2^k$. By substituting the values of m, M, and e into the inequality of the Sphere Packing Bound (Theorem 2.6), we obtain: $M\left[\binom{m}{0} + \binom{m}{1}(q-1) + \cdots + \binom{m}{e}(q-1)^e\right] \leq q^m$

Let c represent the volume of the Hamming sphere, defined as:
$c = \binom{m}{0} + \binom{m}{1}(q-1) + \cdots + \binom{m}{e}(q-1)^e \Rightarrow c = 1 + 2047 + \cdots + \binom{2047}{511}$

Substituting c back into the main inequality yields: $2^k\ c \leq 2^{2047}$

Solving for the dimension k, we get: $k \leq \frac{\log \frac{2^{2047}}{c}}{\log 2} \Rightarrow k \leq 392.538726434149$

The calculated upper bound for k is approximately 392.53, which implies that $k < 393$. Since the calculated value is not an integer, the equality condition required for a perfect code cannot be satisfied (as the dimension k must be an integer). Therefore, by Corollary 2.7, the code C is not perfect. ■

4. Results and Discussion

Here we discuss the final parameters of the new linear codes that we discovered. Our key contributions are also clearly stated, and we explain why the findings are important mathematically.

4.1 The New Linear Codes

On the basis of the computation steps and incidence matrices we constructed in Section 3, we obtained the following precise results:

Theorem 3.1: The incidence matrix of the points and hyperplanes in $PG(9,3)$ generates a ternary linear code with parameters $[29524, k < 14317, 9841]_3$. This code has a massive capacity and can correct up to $e = 4920$ errors.
Theorem 3.2: The incidence matrix of the points and hyperplanes in $PG(10,2)$ generates a binary linear code with parameters $[2047, k < 393, 1023]_2$. This code can successfully correct up to $e = 511$ errors.

4.2 Discussion and Mathematical Significance

After finding these parameters, we tested both codes against the Sphere Packing Bound (Theorem 2.6). The calculations show that these codes do not satisfy the exact equality of the bound. Therefore, they are classified as non-perfect codes.

But why are these results mathematically significant? Our main contribution is to show how the algebraic companion matrix technique can be successfully scaled to high dimensions such as 9 and 10. Typically, these very large projective spaces are computationally very difficult to work with, and this is why past research did not proceed past space dimensions 5 or 6. We were able to avoid this geometrical intricacy and reduce it to an operational numerical algorithm.
Thus, we were able to acquire codes whose error-correcting capabilities were incredibly high (correcting 4920 and 511 errors, respectively). This is really what modern systems require to protect large blocks of data because of their high capacity. Finally, this work gives a good mathematical demonstration that higher-dimensional projective geometries are not merely hypothetical entities, but useful tools in the construction of highly reliable error-correcting codes.

Even though we managed to generate single orbits of the intermediate subspaces, we just utilized the hyperplanes to construct the codes in this paper. You may ask: Why did we discover these intermediate orbits in the first place? This was done to make the geometric structure clear and also to prepare the way for future work. Since the length of the selected orbit is equal to the number of points, it is quite likely that new incidence matrices between the points and these particular intermediate subspaces can be built. This implies that our method can be used by researchers in the future to find even more linear codes with totally different parameters.

5. Conclusion

In this paper, we constructed projective linear codes using the incidence matrices of points and hyperplanes in the spaces $PG(9,3)$ and $PG(10,2)$. The main results obtained from our calculations are:

1. For $PG(9,3)$: The generated code has a length of $m = 29524$ and a minimum distance of $d = 9841$. Based on the Sphere Packing Bound, the dimension is $k < 14317$, and the code can correct up to $e = 4920$ errors.
2. For $PG(10,2)$: The code has a length of $m = 2047$ and a minimum distance of $d = 1023$. The dimension is bounded by $k < 393$, and the error-correcting capability is $e = 511$.
3. Code Quality: Even though these codes are non-perfect, they still have a high ability to correct errors, which makes them useful.
4. Subspaces: We also found at least one orbit for subspaces between points and hyperplanes. This shows that we can construct more linear codes using these subspaces in the future.

Finally, this work proves that projective geometry is a good method for building reliable error-correcting codes.

6. Acknowledgements

The authors are grateful to the University of Mosul (https://uomosul.edu.iq) in Mosul, Iraq, for their assistance with this work.

7. Declarations

7.1 Ethics approval and consent to participate: Not applicable.
7.2 Consent for publication: Not applicable.
7.3 Availability of Data and Materials: Data will be provided upon receiving a valid request.
7.4 Conflicts of interest: The authors declare that there is no conflict of interest.
7.5 Funding: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

8. References

[1] J. W. P. Hirschfeld, *Projective geometries over finite fields*, 2nd ed. Oxford: Clarendon Press, 1998. doi: 10.1093/oso/9780198502951.001.0001.
[2] J. W. P. Hirschfeld, *Coding theory: lectures*. Brighton: Sussex University, 2014.
[3] R. Hill, *A first course in coding theory*. in Oxford applied mathematics and computing science series. Oxford: Clarendon Press, 1986.
[4] N. A. M. Al-Seraji, "The geometry of the plane of order seventeen and its application to error-correcting codes," PhD thesis, University of Sussex, Brighton, 2010.
[5] N. A. M. Al-Seraji and Z. S. Jafar, "Applications of the projective plane in coding theory," *Journal of Southwest Jiaotong University*, vol. 55, no. 1, p. 2, 2020, doi: 10.35741/issn.0258-2724.55.1.2.
[6] G. Kiss and T. Szonyi, *Finite geometries*, 1st ed. Boca Raton: Chapman and Hall/CRC, 2019. doi: 10.1201/9781315120072.

[7] S. H. Naji and E. B. Al-Zangana, "Special NMDS codes of dimension three over the finite field of order 27," in *AIP Conference Proceedings*, AIP Publishing LLC, 2023, p. 040068. doi: 10.1063/5.0120529.

[8] A. M. Khalaf and N. Y. K. Yahya, "New Examples in Coding Theory for Construction Optimal Linear Codes Related With Weight Distribution," in *AIP Conf. Proc.*, American Institute of Physics Inc., 2023. doi: 10.1063/5.0170590.

[9] E. B. Al-Zangana and N. Y. K. Yahya, "Subgroups and Orbits by Companion Matrix in Three Dimensional Projective Space," *Baghdad Sci. J.*, vol. 19, no. 4, pp. 805–810, 2022, doi: 10.21123/bsj.2022.19.4.0805.

[10] F. N. Abdullah and N. Y. K. Yahya, "Bounds on minimum distance for linear codes over GF(q)," *Ital. J. Pure Appl. Math.*, no. 45, pp. 894–903, 2021.

[11] N. Y. K. Yahya and E. B. Al-Zangana, "The Non-existence of [1864, 3, 1828]53 Linear Code by Combinatorial Technique," *Int. J. Math. Comput. Sci.*, vol. 16, no. 4, pp. 1575–1581, 2021.

[12] Y. A. F. Yahya and N. Y. K. Yahya, "Applications of the PG(5,q) in coding theory," *Iraqi Journal of Science*, pp. 5002–5014, Nov. 2025, doi: 10.24996/ijs.2025.66.11.25.

[13] H. M. Ahmad and N. Y. K. Yahya, "Applications of the PG(6,2) in Coding Theory," vol. 1, no. 10, 2025.

بناء رموز خطية مصححة للأخطاء بالاعتماد على مصفوفات الوقوع من الفضاءين PG(9,q) و PG(10,q)

عبدالمطلب نوح محمد سعيد ابراهيم[1] ، ندى ياسين قاسم يحيى[2]

[1,2] قسم الرياضيات، كلية التربية للعلوم الصرفة، جامعة الموصل، العراق

المستخلص:

تمتلك رموز تصحيح الأخطاء المُعرَّفة على الهندسات الإسقاطية المنتهية قيمة كبيرة في ضمان موثوقية البيانات. ومع ذلك، ونظراً لأن التعامل مع الفضاءات ذات الأبعاد العالية معقد جداً من الناحية الحسابية، فقد تركزت معظم الأبحاث السابقة على الأبعاد المنخفضة. للمساهمة في سد هذه الفجوة، يقوم هذا البحث ببناء رموز خطية إسقاطية جديدة مباشرة من الفضاءات ذات الأبعاد العالية $PG(9,3)$ و $PG(10,2)$. قمنا بتوليد المصفوفات المرافقة لمتعددات الحدود الأولية لبناء النقاط والفضاءات الجزئية بشكل منهجي. بعد ذلك، قمنا ببناء مصفوفات الوقوع بين النقاط والمستويات الفائقة، ومعالجتها حسابياً في برنامج MATLAB لتحديد المعلمات الأساسية لرموزنا الجديدة. من خلال هذه العملية، حصلنا على رمزين محددين: رمز ثلاثي من الفضاء $PG(9,3)$ بالمعلمات $[29524, k < 14317, 9841]_3$ يمكنه تصحيح أخطاء تصل إلى 4920 خطأ، ورمز ثنائي من الفضاء $PG(10,2)$ بالمعلمات $[2047, k < 393, 1023]_2$ يمكنه تصحيح أخطاء تصل إلى 511 خطأ. وبتطبيق قيد التعبئة الكروية، تبيّن أنه على الرغم من أن هذه الرموز تُصنف كرموز غير تامة، إلا أنها تمتلك قدرة فائقة على تصحيح الأخطاء. في المحصلة، تمكنا من صياغة مخطط جبري عملي يُحوّل البنية المعقدة للفضاءات ذات الأبعاد العالية إلى رموز موثوقة لتصحيح الأخطاء، مما يقدم رؤى نظرية مفيدة تتجاوز النماذج المألوفة ذات الأبعاد المنخفضة.