Johnson graph codes

Iwan Duursma1 · Xiao Li1

Received: 17 March 2021 / Revised: 3 December 2021 / Accepted: 13 December 2021 / Published online: 28 February 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Array codes are the preferred codes for distributed storage, such that different rows in an array are stored at different nodes. Layered codes use a sparse format for stored arrays with a single parity check per column and no other parity checks. Remarkably, the simple structure of layered codes is optimal when data is collected from all but one node. Codes that collect data from fewer nodes include improved layered codes, determinant codes, cascade codes and moulin codes. As our main result we show that the concatenation of layered codes with suitable outer codes achieves the performance of cascade and moulin codes which is conjectured to be optimal for general regenerating codes. The codes that we use as outer codes are in a new class of codes that we call Johnson graph codes. The codes have properties similar to those of Reed–Muller codes. In both cases the topological structure of the set of coordinates can be used to identify information sets and codewords of small weight.

Keywords Algebraic coding theory · Codes for distributed storage · Regenerating codes · Johnson graph · Multilinear algebra · Determinants

Mathematics Subject Classification Primary 94B25 · Secondary 05E30 · 68P20

1 Introduction

Codes for distributed storage encode data as a collection of $n$ smaller pieces that are then stored at $n$ different nodes. A code is regenerating of type $(n, k, d)$ if the original data can be recovered from any $k$ of the $n$ pieces, i.e., by downloading data from any $k$ of the $n$ nodes, and

This is one of several papers published in Designs, Codes and Cryptography comprising the “Special Issue: On Coding Theory and Combinatorics: In Memory of Vera Pless”.

This work was supported in part by NSF Grant CCF 1619189.

✉ Iwan Duursma
duursma@illinois.edu
Xiao Li
xiaoli17@illinois.edu

1 University of Illinois at Urbana-Champaign, Urbana, USA
if moreover any failed node can be repaired with small amounts of helper data from any \( d \) of the remaining nodes [1]. Locally repairable codes follow a different strategy and prioritize a combination of high minimum distance and for each node a small set of helper nodes that will provide repair data in case the node fails [8]. The need for special codes for distributed storage arises because many of the classical error-correcting codes, including Reed–Solomon codes, do not have the required properties for efficient node repair.

Our first main result, Theorem 37, gives a construction of regenerating codes of type \((n, k, d)\) in the regime \( n = d + 1 \) using concatenation of layered codes. Corollary 38 describes the main parameters of the codes. They are the same as those for cascade and moulin codes that use a different construction. The concatenation of layered codes makes use of outer codes with special properties. As our second main result, Definition 13 gives a construction for a new class of codes and Theorem 16 proves that the codes have the required properties as outer codes. We call the newly defined class of codes Johnson graph codes. While our motivation for the construction of Johnson graph codes comes from their application in distributed storage we believe that the codes are of interest by themselves. Their main properties are similar to those of binary Reed–Muller codes. In detail the comparison is as follows. For a block code of length \( N = \binom{n}{r} \) label each coordinate by a unique \( v \)-subset \( L \subset \{1, 2, \ldots, n\} \). For a pair of integers \( k \leq n \) and \( r \leq \min(v, k) \), the codes in Definition 13 have the following two properties.

\((J1)\) For any \( k \)-subset \( A \subset \{1, 2, \ldots, n\} \), the coordinates \( L : d(L, A) \leq |v - k| + 2r \) are maximally linearly independent.

\((J2)\) For any \( v \)-subset \( L \subset \{1, 2, \ldots, n\} \) with \( d(L, A) > |v - k| + 2r \), the coordinate \( L \) is a linear combination of the coordinates \( L' : L \cap A \subseteq L' \subseteq L \cup A \).

The first property gives us many easily recognizable information sets and the second property gives us many easily recognizable short relations among symbols in a codeword. We compare the two properties with those of the binary Reed–Muller codes. In detail the comparison is as follows. For a block code of length \( N = \binom{n}{r} \) label each coordinate by a unique \( v \)-subset \( L \subset \{1, 2, \ldots, n\} \). For a pair of integers \( k \leq n \) and \( r \leq \min(v, k) \), the codes in Definition 13 have the following two properties.

\((H1)\) For any subset \( A \subset \{1, 2, \ldots, m\} \), the coordinates \( L : d(L, A) \leq r \) are maximally linearly independent.

\((H2)\) For any subset \( L \subset \{1, 2, \ldots, m\} \) with \( d(L, A) > r \), the coordinate \( L \) is a linear combination of the coordinates \( L' : L \cap A \subseteq L' \subseteq L \cup A \).

For \( A = \emptyset \), properties \((H1)\) and \((H2)\) reflect the standard properties that the code \( RM(r, m) \) has dimension \( 1 + \binom{m}{1} + \cdots + \binom{m}{r} \) and dual minimum distance \( d^\perp = 2^r + 1 \).

The outline of the paper is as follows. The next section reviews the definition and basic properties of layered codes. Section 3 introduces Johnson graph codes and defines them axiomatically as codes with property \((J1)\). A general construction for codes with properties \((J1)\) and \((J2)\) is presented in Sect. 4. As a special case of the general construction, in Sect. 5 we construct Johnson graph codes from Reed–Solomon codes. In Sect. 6 we use Johnson graph codes as outer codes in the concatenation of layered codes. The concatenated codes are regenerating codes of type \((n, k, d)\) with \( n = d + 1 \).

2 Layered codes

Layered codes are defined in [13]. They are an important building block for several families of regenerating codes, including improved layered codes [12] and coupled-layer codes [11].
The distances in Table 1 are the distances induced by the Johnson graph. Data recovery proceeds sequentially and starts with layers at \( r = 0 \). As \( r \) increases, missing data in layers of type \( r \) will be recovered from available data in combination with stored parities. In the next section we define a family of codes on Johnson graphs that, for a minimum of stored extra parities and for any combination of \( k \) accessed disks, recover data in under-accessed layers from available data in fully-accessed layers. The recovery process itself is described in Sect. 6.

Layered codes are of type \((n, k, d) = (n, n - 1, n - 1)\). Data is stored redundantly at \( n \) nodes and can be collected from any \( k = n - 1 \) nodes. Data at a failed node is restored

| \#L \ | \( |A \cap L| \) \ | \( |A \setminus L| \) \ | \( |L \setminus A| \) \ | \( r \) |
|---|---|---|---|---|
| (a) Access in \(|A| = 5 \) nodes \ | 5 \ | 1 \ | 4 \ | 0 \ | 0 |
| 10 \ | 2 \ | 3 \ | 1 \ | 1 |
| (b) Access in \(|A| = 3 \) nodes \ | 3 \ | 0 \ | 3 \ | 1 \ | 0 |
| 9 \ | 1 \ | 2 \ | 2 \ | 1 |
| 3 \ | 2 \ | 1 \ | 3 \ | 2 |

These codes are defined with \( d = n - 1 \), i.e., repair of a failed node uses helper data from all remaining nodes. Determinant codes [4], cascade codes [5,6], and moulin codes [3] use a different construction that removes this constraint. For a distributed storage system with \( n \) nodes and for a parameter \( 2 \leq v \leq n \), a layered code is defined by \( \binom{v}{n} \) disjoint checks of length \( v \). The symbols \((c_i : i \in L)\) in a layer \( L \subset \{0, 1, \ldots, n - 1\}\) of size \( v \) satisfy \( \sum_{i \in L} c_i = 0 \). The use of longer checks makes a code more storage efficient while shorter checks require fewer helper nodes to recover erased data.

With one parity per layer, the overall redundancy of a layered code is \( R = \binom{v}{n} \). Let \( \alpha \) be the number of symbols stored by each of the \( n \) nodes. Counting the total number of symbols in the code in two ways, by node and by layer, gives \( n\alpha = Rv \). Let \( \beta \) be the number of symbols that is downloaded from a helper node to repair a failed node. There is one such symbol for each layer that contains both the helper node and the failed node. Counting the total amount of helper information in two ways, by pairs of nodes and by layer, gives \( \binom{v}{2} \beta = R \frac{v}{2} \). So that \( R = \binom{v}{n}, \alpha = \binom{n-1}{v-1}, \beta = \binom{n-2}{v-2} \).

**Example 1** (Motivating example) Consider the layered code with \((n, k, d) = (6, 5, 5)\) and \( v = 4 \). It has \((R, \alpha, \beta) = (\binom{4}{6}, \binom{3}{3}, \binom{3}{5}) = (15, 10, 6)\). The code has \( \binom{4}{6} = 15 \) layers, each containing four symbols that sum to zero. The symbols may be put in a \( 6 \times 15 \) array format such that rows correspond to nodes and columns to layers. Symbols \( c_1, c_2, c_3, c_5 \) for the layer \( \{1, 2, 3, 5\} \) will form a column \((0, c_1, c_2, c_3, 0, c_5)^T\). Apart from the assigned zeros in the array, the only checks for the array are that each column sums to zero. When a layer is accessed in 5-out-of-6 nodes either all symbols in a layer are available or one symbol is missing but can be recovered using the zero check sum for the layer. In general, upon choosing any five nodes, the layers are of two types (Table 1(a)). The \( \binom{5}{3} = 5 \) layers in the first group are fully accessed and all their symbols are available. The \( \binom{5}{2} = 10 \) layers in the second group miss one symbol that is however protected by the parities on the individual layers. Once we lower \( k \), the partitioning of the layers changes (Table 1(b)).

The partitioning of the layers can be captured by a partitioning of vertices in a Johnson graph. The distances in Table 1 are the distances induced by the Johnson graph. Data recovery proceeds sequentially and starts with layers at \( r = 0 \). As \( r \) increases, missing data in layers of type \( r \) will be recovered from available data in combination with stored parities. In the next section we define a family of codes on Johnson graphs that, for a minimum of stored extra parities and for any combination of \( k \) accessed disks, recover data in under-accessed layers from available data in fully-accessed layers. The recovery process itself is described in Sect. 6.
using small amounts of data collected from \(d = n - 1\) healthy nodes. The improved layered code construction [12] stores additional data at each node to make it possible to collect data from \(k < n - 1\) nodes. Cascade codes [5,6] use a combination of regenerating codes of type \((n, d, d)\) called determinant codes [4] that together operate as a code of type \((n, k, d)\) with \(k < d\). Moulin codes [3] form \((n, k, d)\) codes with \(k < d\) using a combination of \((n, k, k)\) regenerating codes. Cascade and moulin codes share the same parameters: to store data of a given size, they store the same amount of data per node and collect the same amounts of data from healthy nodes to restore a failed node. In Sect. 6 we propose a class of concatenated layered codes with the same performance parameters as cascade and moulin codes for the special case that \(d = n - 1\).

3 Johnson graph codes

Definition 2 The Johnson graph \(J(n, v)\) is the undirected graph with vertex set all \(v\)-subsets of a given \(n\)-element set such that two vertices are adjacent if they intersect in \(v - 1\) elements. The default choice for an \(n\)-element set is the set \(\{0, 1, \ldots, n - 1\}\).

Our goal is to construct a linear code of labelings of the vertices in a Johnson graph such that labelings are uniquely determined by their restriction to a neighborhood in the graph. In this section we formulate the requirements for the code. In the following three sections we give a general construction and we establish important properties. In Sect. 6 the codes are used in the construction of exact repair regenerating codes for use in distributed storage.

With the usual definition of distance between two vertices as the length of the shortest path between two vertices, two vertices in \(J(n, v)\) are at distance \(r\) if they intersect in \(v - r\) elements. For any given vertex \(x\), the vertices in \(J(n, v)\) partition into subsets according to their distance to \(x\). We refer to the subsets as the shells around \(x\).

Example 3 The shells around \(x = \{0, 1\} \subset \{0, 1, 2, 3, 4\}\) are (we write \(x = 01\))

\[S_0(01) = \{01\}, \; S_1(01) = \{02, 03, 04, 12, 13, 14\}, \; S_2(01) = \{23, 24, 34\}.

We construct a code of length 10 for the graph \(J(5, 2)\) that recovers the three symbols at the vertices \(S_2(x)\) from the seven symbols at the vertices \(S_0(x)\) and \(S_1(x)\), for any choice of vertex \(x\), by storing three fixed parities. Let \(\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4\) be five distinct elements of the field \(F\). For a general word \(c = (c_{ij}) \in F^{10}\) define a parity vector \(Hc \in F^3\) as

\[Hc = \sum c_{ij} \begin{pmatrix} 1 \\ \alpha_i + \alpha_j \\ \alpha_i \alpha_j \end{pmatrix}.

For each of the ten sets \(S_2(x)\), where \(x\) is a vertex in \(J(5, 2)\), the corresponding columns in the parity matrix \(H\) are linearly independent. For \(S_2(23) = \{23, 24, 34\}\), the three columns form the invertible submatrix

\[
\begin{pmatrix} 1 & 1 & 1 \\ \alpha_2 + \alpha_3 & \alpha_2 + \alpha_4 & \alpha_3 + \alpha_4 \\ \alpha_2 \alpha_3 & \alpha_2 \alpha_4 & \alpha_3 \alpha_4 \end{pmatrix}.
\]

Thus the three symbols \(c_{23}, c_{24}, c_{34}\) can be recovered from the remaining seven symbols in combination with the three fixed parities \(Hc\). Moreover, the three symbols can be recovered efficiently one at a time. The equation \((\alpha_2^3 - \alpha_4, 1)Hc\) expresses \(c_{23}\) as a linear combination of \(c_{01}, c_{02}, c_{03}, c_{12}, c_{13}\). The symbols \(c_{24}, c_{34}\) are recovered similarly.
We encounter Johnson graphs in two different ways. For a storage system with \( n \) nodes and for a parameter \( 2 \leq v \leq n \), layered codes are defined by \( \binom{n}{v} \) disjoint checks of length \( v \), one for each \( v \)-subset of nodes. The \( v \)-subsets are called layers. They form a set of vertices for a Johnson graph \( J(n, v) \). A second graph \( J(n, k) \) occurs as follows. For a regenerating code of type \( (n, k, d) \) data is collected by contacting any \( k \) nodes. The possible data collection scenarios are the \( k \)-subsets of an \( n \)-element set. They form the vertices of a Johnson graph \( J(n, k) \).

For each collection scenario, i.e., for each \( k \)-subset \( A \subset \{0, 1, \ldots, n - 1\} \), we define a partition of the layers in the layered code, i.e., of the vertices in \( J(n, v) \). Table 1 illustrates vertex partitions into shells for the cases \( k > v \) and \( k < v \).

For \( k \geq v \), layers \( L \) with \( L \subset A \) are fully accessible. They form the shell \( S_0(A) = \{ L : L \subset A \} \). Layers at distance \( r \) from \( S_0(A) \) form the shell \( S_r(A) = \{ L : |L \setminus A| = r \} \). For \( k \leq v \), layers are accessible in at most \( k \) symbols. The maximally accessible layers are those with \( A \subset L \). They form the shell \( S(A) = \{ L : L \supset A \} \). The shell at distance \( r \) from these layers is \( S_r(A) = \{ L : |A \setminus L| = r \} \). In general, since \( k - v = |A \setminus L| - |L \setminus A| \),

\[
L \in S_r(A) \iff r = \min(|L \setminus A|, |A \setminus L|). \tag{1}
\]

For \( L \in J(n, v) \), let \( B_r(L) \) be the neighborhood of radius \( r \) around \( L \).

**Definition 4** For the Johnson graph \( J(n, v) \) and for a subset \( A \) of \( \{0, 1, \ldots, n - 1\} \), define a generalized neighborhood as a union of vertex neighborhoods.

\[
B_r(A) = \begin{cases} 
\bigcup_{L_0 \subset A} B_r(L_0) & (|A| \geq v), \\
\bigcup_{L_0 \supset A} B_r(L_0) & (|A| \leq v).
\end{cases}
\]

It is easy to see that membership \( L \in B_r(A) \) depends only on the size of \( L \setminus A \) or \( A \setminus L \).

\[
L \in B_r(A) \iff \begin{cases} 
|L \setminus A| \leq r & (|A| \geq v), \\
|A \setminus L| \leq r & (|A| \leq v)
\end{cases} \iff \min(|L \setminus A|, |A \setminus L|) \leq r. \tag{2}
\]

The shell \( S_r(A) = B_r(A) \setminus B_{r-1}(A) \). With the above notions in mind, we define Johnson graph codes as codes on the vertices of a Johnson graph such that generalized neighborhoods form information sets.

**Definition 5** A Johnson graph code \( JGC(n, v, k, r) \) on the vertices of the Johnson graph \( J(n, v) \) is a code of length \( N = \binom{n}{v} \) and dimension \( K = |B_r(A)| \) such that, for any \( k \)-subset \( A \) of \( \{0, 1, \ldots, n - 1\} \), \( B_r(A) \) forms an information set.

It is worth noting that the definition only refers to matroid properties of the code. In matroid terms it asks that the set of bases for the matroid of the code is large enough that it includes all generalized neighborhoods \( B_r(A) \). The definition includes as trivial examples all MDS codes of length \( N \) and dimension \( K \). Any Johnson graph code, and in particular any length \( N \), dimension \( K \) MDS code, can be used in the construction of concatenated layered codes in Sect. 6. The codes constructed in the next section have the advantage of being defined over a field of size \( n \) instead of \( N = \binom{n}{v} \) and moreover have a parity check structure that allows efficient erasure decoding. We include some lemmas for later use and prove in Proposition 12 that the dual of a Johnson graph code is again a Johnson graph code.

**Lemma 6** Let \( k = |A| \).

\[
|B_r(A)| = \begin{cases} 
\sum_{i=0}^{r} \binom{n-k}{i} \binom{k}{r-i} & (k \geq v), \\
\sum_{i=0}^{r} \binom{v-k}{i} \binom{n-k}{r-i} & (k \leq v).
\end{cases}
\]
The first sum counts \( L \) with \(|L \cap A^c| = i \leq r\) and \(|L \cap A| = v - i\). The second sum counts \( L \) with \(|L^c \cap A| = i \leq r\) and \(|L^c \cap A^c| = n - v - i\).

We introduce the following notation. Let
\[
d_1 = \min(v, n - k), \quad d_2 = \min(k, n - v),
\]
and let \( R = \min(d_1, d_2) \).

**Lemma 7** \( S_r(A) = \emptyset \) for all \( r > R \).

**Proof** Use (1) with \( \min(|L \setminus A|, |A \setminus L|) \leq \min(d_1, d_2) = R \).

**Lemma 8** \( S_r(A) = S_{R-r}(A^c) \).

**Proof** For \( L \in J(n, v) \), let
\[
\begin{pmatrix}
a & b \\
c & d
\end{pmatrix} = \begin{pmatrix} |L \cap A^c| & |L \cap A| \\
|L^c \cap A^c| & |L^c \cap A|
\end{pmatrix}.
\]

Then \( L \in S_r(A) \) for \( r = \min(a, d) \) and \( L \in S_s(A^c) \) for \( s = \min(b, c) \). Where \( r \) and \( s \) are related to \( R \) via \( r + s = \min(a + b, a + c, b + d, c + d) = \min(v, n - k, k, n - v) = R \).

The lemma gives a dual description for the shell \( S_r(A) \) in terms of the complement \( A^c \) of \( A \).

**Example 9** The three layers \( S_0(012) = \{0123, 0124, 0125\} \subset J(6, 4) \) can be described dually as \( S_2(345) \).

A different dual description is obtained by considering the set \( \{L^c : L \in S_r(A)\} \) as a subset of vertices in the Johnson graph \( J(n, n - v) \). Note that this graph is isomorphic to \( J(n, v) \) with only difference that the vertex labels \( L \subset \{0, 1, \ldots, n - 1\} \) are replaced with their complements \( L^c = \{0, 1, \ldots, n - 1\} \setminus L \). Two \( v \)-subsets are adjacent in \( J(n, v) \) if and only if their complements are adjacent in \( J(n, n - v) \).

**Lemma 10** For shells \( S_r(A) \subset J(n, v) \) and \( S_{R-r}(A) \subset J(n, n - v) \),
\[ L \in S_r(A) \iff L^c \in S_{R-r}(A). \]

**Proof** As in the proof of Lemma 8, \( L \in S_r(A) \) for \( r = \min(a, d) \) and \( L^c \in S_s(A) \) for \( s = \min(c, b) \). So that \( r + s = R \).

**Lemma 11**
\[ B_r(A)^c = B_{R-r-1}(A^c). \]

**Proof** Use that \( B_r(A)^c = \cup_{r > r} S_t(A) \) and apply Lemma 8.

**Proposition 12** The dual code of a Johnson graph code \( JGC(n, v, k, r) \) is a code \( JGC(n, v, n - k, R - r - 1) \).

**Proof** The code has the required length and dimension. Lemma 11 and the fact that the complement of an information set is an information set for the dual code, proves that \( B_{R-r-1}(A^c) \) is an information set for any \( k \)-subset \( A \).
4 Construction of Johnson graph codes

We give a general construction for Johnson graph codes of length \( N = \binom{n}{k} \) from MDS codes of length \( n \). Section 5 describes in more detail the special case where the length \( n \) MDS code is of Reed–Solomon type. Codewords of length \( N \) will have their coordinates indexed by the vertices of the Johnson graph \( J(n, v) \). The codes are generated by the coordinatization vectors of suitably chosen matrices \( M \). For a \( v \times n \) matrix \( M \),

\[
\pi(M) = (\det(M_L) : L \in J(n, v)),
\]

where \( M_L \) is the \( v \times v \) minor of \( M \) with columns in the \( v \)-subset \( L \subset \{0, 1, \ldots, n - 1\} \).

**Definition 13** For given \( n \) and \( v \), let \( C_0 \) be a code of length \( n \) and let \( t \geq 0 \). Define a code \( C \) of length \( N \) as the span of the vectors \( \pi(M) \) for all \( v \times n \) matrices \( M \) with at least \( t \) rows in \( C_0 \).

We show in two steps that the code \( C \) is a Johnson graph code \( JGC(n, v, k, r) \) with \( k = \dim C_0 \) and \( r = \min(v, \dim C_0) - t \).

**Lemma 14** The code \( C \) has dimension \( K = |B_r(I_0)| \), where \( r = \min(v, \dim C_0) - t \) and \( I_0 \subset \{0, 1, \ldots, n - 1\} \) is of size \( \dim C_0 \).

**Proof** Let \( g = \{g_0, g_1, \ldots, g_{n-1}\} \) be a basis for \( F^n \) and let \( \{g_i : i \in I_0\} \) be a basis for \( C_0 \). The vectors \( \pi(M) \), for matrices \( M \) with rows \( \{g_i : i \in L\} \) such that \( |L \cap I_0| \geq t \), form a basis for \( C \). Furthermore,

\[
|L \cap I_0| = t \iff \min(|L \setminus I_0|, |I_0 \setminus L|) = \min(v - t, \dim C_0 - t).
\]

And thus, using (2) with \( r = \min(v, \dim C_0) - t \),

\[
\{ L \in J(n, v) \mid |L \cap I_0| \geq t \} = B_r(I_0).
\]

From the lemma it is clear that the dimension of the code \( C \) only depends on \( \dim C_0 \) and not on \( C_0 \). The occurrence of information sets in \( C \) on the other hand depends on the choice of the code \( C_0 \).

**Lemma 15** For \( C \) as in the definition, and for an information set \( A_0 \subset \{0, 1, \ldots, n - 1\} \) for \( C_0 \), the set

\[
\{ L \in J(n, v) \mid |L \cap A_0| \geq t \} = B_r(A_0)
\]

is an information set for \( C \).

**Proof** The size of \( B_r(A_0) \) equals the dimension of \( C \). It therefore suffices to prove the independence of the coordinates in \( B_r(A_0) \). For each \( L \in B_r(A_0) \) construct a codeword \( \pi(M) \) from a \( v \times n \) matrix \( M \) with two types of rows. For \( j \in L \cap A_0 \) include as row in \( M \) the unique codeword in \( C_0 \) that is 1 in \( j \) and 0 in \( A_0 \setminus j \). For \( j \in L \setminus A_0 \) include as row in \( M \) the unit vector that is 1 in \( j \) and 0 elsewhere. Upto a permutation of columns that puts the columns in \( A_0 \setminus L \) in the leading positions, followed by columns in \( L \cap A_0 \) and \( L \setminus A_0 \), \( M \) will be of the form

\[
M = \begin{pmatrix} O & I \mid X & Y \\ O \mid O & I \mid O \end{pmatrix}, \quad M_L = \begin{pmatrix} I \mid X \\ O \mid I \end{pmatrix}.
\]
Clearly, det($M_L$) = 1. And det($M_{L'}$) = 0 for $L' \neq L$ with $|L' \cap A_0| \geq |L \cap A_0|$. The constructed codewords $\pi(M)$, for $L \in S_r(A_0), \ldots, S_1(A_0), S_0(A_0)$, in that order, form an invertible triangular matrix in the positions $B_r(A_0)$. And thus $B_r(A_0)$ is an information set for $C$. \hfill \Box

**Theorem 16** For an MDS code $C_0$, the code $C$ in Definition 13 is a Johnson graph code.

**Proof** Lemmas 14 and 15. \hfill \Box

**Lemma 17** For each $L \in S_r(A_0)$, the proof of Lemma 15 constructs a codeword $\pi(M)$ that depends on $L$. It has as special property that the coordinate $\pi(M)_L$ is the unique nonzero coordinate in the positions $B_r(A_0)$. Moreover the weight of $\pi(M)$ is at most $2t + n - \dim C_0 - v$.

**Proof** For $r = \min(v, \dim C_0) - t$, we have, as in the proof of Lemma 14, that $L' \in B_r(A_0)$ if and only if $|L' \cap A_0| \geq t$. For $L \in S_r(A_0)$, equality holds and $|L \cap A_0| = t$. It follows that $|L' \cap A_0| \geq |L \cap A_0|$ for all $L' \in B_r(A_0)$, and thus, as in the proof of Lemma 15, that det($M_{L'}$) = 0 for all $L' \in B_r(A_0)$, $L' \neq L$. For the weight of $\pi(M)$, note that a full minor $M_{L'}$ in the matrix $M$ is nonsingular only if $M_{L'}$ contains all columns in $L \setminus A_0$ and no columns in $A_0 \setminus L$, i.e., only if

$$L \cap A_0^c \subset L' \subset L \cup A_0^c.$$  

For $|L \cap A_0| = t$, $|L \cap A_0^c| = v - t$ and $|L \cup A_0^c| = n - \dim C_0 + t$. The number of $L'$ such that $M_{L'}$ is nonsingular is therefore at most $2t + n - \dim C_0 - v$. \hfill \Box

**Example 18** For the Johnson graph $J(n = 5, v = 3)$, let

$$g = \begin{pmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$  

Let $C_0 = \text{span}(g_0, g_1), t = 2$. Then $r = 0$, $I_0 = \{0, 1\}$, $B_0(I_0) = \{012, 013, 014\}$, and the code $C$ is spanned by the first block of rows in the matrix

\[
\begin{array}{cccccccc}
012 & 1 & \cdots & 1 & 1 & 1 & 0 & 1 \\
013 & \cdots & 1 & 0 & 1 & 1 & 0 & 1 \\
014 & \cdots & 1 & 0 & 1 & 1 & 1 & 1 \\
023 & \cdots & 1 & \cdots & \cdots & \cdots & 0 & 1 \\
024 & \cdots & 1 & \cdots & \cdots & \cdots & 1 & 1 \\
034 & \cdots & 1 & \cdots & \cdots & \cdots & 1 & 1 \\
123 & \cdots & 1 & \cdots & \cdots & \cdots & 1 & 1 \\
124 & \cdots & 1 & \cdots & \cdots & \cdots & 1 & 1 \\
134 & \cdots & 1 & \cdots & \cdots & \cdots & 1 & 1 \\
234 & \cdots & \cdots & \cdots & \cdots & \cdots & 1 & 1 \\
\end{array}
\]
Matrix entries \( \cdot \) are 0 independent of the choice of the code \( C_0 \). Lemma 17 gives the number of remaining entries in a row: \( \binom{2}{1}, \binom{1}{1} \) or \( \binom{0}{0} \), for rows in the first, second, or third block of rows. The presence of additional zeros in the matrix is due to \( C_0 \) not being MDS. The sets \( A_0 = \{0, 2\} \) and \( A_0 = \{1, 4\} \) are not information sets for \( C_0 \). For each of the remaining \( A_0 \in J(5, 2) \), \( B_0(A_0) \) is an information set for \( C_0 \).

For \( t = 0 \), and given a basis \( g = \{g_0, g_1, \ldots, g_{n-1}\} \) for \( F^n \), the proof of Lemma 15 assigns to each \( L \in J(n, v) \) a unique matrix \( M \) and vector \( \pi(M) \). The vectors \( \pi(M) \) form an invertible \( N \times N \) matrix \( /\Lambda_1(g) \). The assignment of \( /\Lambda_1(g) \) to an ordered list of vectors \( g \) is functorial. That is, if \( g \) and \( h \) are two ordered lists of vectors and both are interpreted as \( n \times n \) matrices, with product \( gh \) as \( n \times n \) matrices, then \( /\Lambda_1(gh) = /\Lambda_1(g) /\Lambda_1(h) \) as product of \( N \times N \) matrices. The functoriality property says that the determinant of a \( v \times v \) minor in \( gh \) can be expressed in terms of determinants of \( v \times v \) minors in \( g \) and \( h \). Both properties are classical and have short proofs.

Lemma 19 (Cauchy–Binet formula) For a \( v \times n \) matrix \( A \) and \( n \times v \) matrix \( B \),
\[
\det(AB) = \pi(A) \cdot \pi(B^T).
\]

**Proof** Compare determinants in the \((v + n) \times (v + n)\) matrix product
\[
\begin{pmatrix}
I_v & A \\
0 & I_n
\end{pmatrix}
\begin{pmatrix}
0 & -A \\
B & I_n
\end{pmatrix}
= \begin{pmatrix}
AB & 0 \\
B & I_n
\end{pmatrix}.
\]

Using a cofactor expansion over the minors \(-A_L\) of \(-A\), for \( L \in J(n, v) \), for the second matrix.
\[
\det\left(\begin{array}{c|c}
0 & -A \\
\hline
B & I_n
\end{array}\right) = \sum_L \det\left(\begin{array}{c|c|c}
0 & -A_L & 0 \\
\hline
B_L & 0 & 0 \\
0 & 0 & I_{n-v}
\end{array}\right)
= \sum_L \det(A_L) \det(B_L) = \pi(A) \cdot \pi(B^T).
\]

In the first equality, the blocks \(-A_L\) and \( B_L \) can be assumed to be in the given positions after replacing the matrix with a conjugate, i.e. by applying the same permutations to both rows and columns.

Proposition 20 Let \( D_0 \) be the dual code of \( C_0 \). The dual code \( D \) of \( C \) is generated by vectors \( \pi(M) \), for all \( v \times n \) matrices \( M \) with at least \( v + 1 - t \) rows in \( D_0 \).

**Proof** For generators \( \pi(A) \in C \) and \( \pi(B) \in D \), the \( v \times v \) matrix \( AB^T \) has a \( t \times (v + 1 - t) \) all zero submatrix. The extended \( t \times v \) submatrix of \( AB^T \) is of rank at most \( t - 1 \) and thus \( \det(AB^T) = 0 \). With the lemma, the generators are orthogonal.

Lemma 21 (Functoriality of exterior algebras) The assignment \( g \mapsto /\Lambda(g) \), that maps a \( n \times n \) matrix \( g \) to a \( N \times N \) matrix \( /\Lambda(g) \), satisfies
\[
/\Lambda(gh) = /\Lambda(g) /\Lambda(h).
\]

**Proof** Evaluate entries of \( /\Lambda(gh) \) using the Cauchy–Binet formula.
The functorial property $\Lambda(gh) = \Lambda(g)\Lambda(h)$ does not depend on the chosen ordering for the rows and columns in $\Lambda(g)$, i.e., on the ordering of the vertices in $J(n, v)$. However, the matrix $\Lambda(g)$, and in particular its shape, depends on the chosen ordering. The ordering that we use to define Johnson graph codes is in general different from the lexicographic ordering and depends on $k$. Let $E_0 = \{0, \ldots, k - 1\}$. A vertex $L'$ precedes $L$ if $|L' \cap E_0| > |L \cap E_0|$. In case of intersections of equal size we order $L'$ and $L$ lexicographically. The partitioning of $J(n, v)$ into shells $S_r(E_0)$, for $r = 0, 1, \ldots, R$, gives the matrix $\Lambda(g)$ a block structure, such that shells appear in the order $S_0(E_0), S_1(E_0), \ldots, S_R(E_0)$, and such that rows and columns within the same shell are ordered lexicographically. We call the modified order the $k$-lexicographic order, with notation $L' \leq_k L$. It is a total order on the vertices of $J(n, v)$.

**Lemma 22** For an upper triangular matrix $g$, the matrix $\Lambda(g)$ is upper triangular whenever the vertex ordering on $J(n, v)$ is a refinement of the Bruhat order, defined by the rule $(\beta_0, \ldots, \beta_{v-1}) \leq (\alpha_0, \ldots, \alpha_{v-1})$ if $\beta_i \leq \alpha_i$, for all $i$. Both the lexicographic and the $k$-lexicographic order on $J(n, v)$ have this property.

**Proof** The entry $\Lambda(g)_{\beta, \alpha} = \det(g_{ij} : i \in \beta, j \in \alpha)$ is nonzero only if $\beta_i \leq \alpha_i$ for all $i$, i.e. only if $\beta \leq \alpha$ in the Bruhat order. Clearly this implies $\beta \leq \alpha$ in the lexicographic order as well as $|\beta \cap E_0| \geq |\alpha \cap E_0|$. And therefore $\beta \leq_k \alpha$. \hfill $\Box$

A square matrix has all its pivots on the main diagonal if it is full rank and if it reduces to echelon form without row permutations. Clearly this is the case if and only if the matrix has a $LU$ factorization.

**Proposition 23** Let $\Lambda(g)$ be defined with an ordering of rows and columns that refines the Bruhat order. If the matrix $g$ has a $LU$ decomposition then so has the matrix $\Lambda(g)$.

**Proof** If $g = LU$ then by functoriality $\Lambda(g) = \Lambda(L)\Lambda(U)$. For the given order, $\Lambda(L)$ is lower triangular and $\Lambda(U)$ is upper triangular by Lemma 22. \hfill $\Box$

Let $g = \{g_0, g_1, \ldots, g_n\}$ be a basis for $F^n$. We say that $g$ is in block form, with respect to a code $C_0$ and a choice of information set $A_0$ for $C_0$, if the coordinates are ordered such that $A_0 = \{0, 1, \ldots, k - 1\}$, if $C_0$ is the span of rows $\{g_0, g_1, \ldots, g_{k-1}\}$, and if the remaining rows are zero in the positions $A_0$.

Block form: $g = \begin{pmatrix} G_0 | G \\ 0 | G_1 \end{pmatrix}$, systematic form: $g = \begin{pmatrix} I | G \\ 0 | -I \end{pmatrix}$.

A matrix in block form can be reduced, using a combination of column permutations within blocks of columns and row operations within blocks of rows, to a block form with $G_0 = I$ and $G_1 = -I$. In the reduced systematic form, $g^2 = I_n$. Thus, by Lemma 21, also $\Lambda(g)^2 = I_N$, independent of the ordering of vertices in $J(n, v)$.

**Lemma 24** For a matrix $g$ in block form, and for a $k$-lexicographic ordering of the vertices in $J(n, v)$, the matrix $\Lambda(g)$ is an upper triangular block matrix. Moreover, if $g$ is in systematic form then the diagonal blocks are identity matrices up to sign.

**Proof** As in the proof of Lemma 22, $\Lambda(g)_{\beta, \alpha}$ is nonzero only if $|\beta \cap E_0| \geq |\alpha \cap E_0|$. The blocks $I$ and $-I$ in $g$ imply that the entry $\Lambda(g)_{\beta, \alpha}$ is nonzero for $|\beta \cap E_0| = |\alpha \cap E_0|$ only if $\beta = \alpha$. \hfill $\Box$
5 Construction using Reed–Solomon codes

We consider the construction of Johnson graph codes $JGC(n, v, k, r)$ in Definition 13 for the special case that the MDS code $C_0$ is a Reed-Solomon code. In general, a code of length $n$ and dimension $k$ is MDS if a codeword is uniquely determined by any $k$ of its coordinates. For distinct field elements $\alpha_0, \alpha_1, \ldots, \alpha_{n-1} \in F$, and for $f \in F[x]$, let

$$ev(f) = (f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{n-1})).$$

For $0 \leq k \leq n$, a Reed–Solomon code of dimension $k$ is defined as the space of vectors $\{ev(f) : \deg f < k\}$. A codeword $ev(f)$ is uniquely determined by any $k$ of its coordinates and thus the code is MDS. The vectors $ev(x^i)$, for $i = 0, 1, \ldots, n - 1$, form a basis for $F^n$. They form a $n \times n$ Vandermonde matrix $g$ with rows $g_i = ev(x^i)$ such that the leading $k$ rows of $g$ span a Reed–Solomon code of dimension $k$, for any $0 \leq k \leq n$.

For the special case of Reed–Solomon codes, the construction in Definition 13 assigns to a subset $I \in J(n, v)$ a coordinatization vector $\pi(M)$, for the matrix $M$ with rows $\{ev(x^i)\}_{i \in I}$. The vector $\pi(M)$ has coordinates, for $L \in J(n, v)$,

$$\pi(M)_L = \det(\alpha_j^i : i \in I, j \in L) =: \det\left(x^i; \alpha_L\right).$$

By Theorem 16, the vectors

$$\left\{ \left(\det\left(x^i; \alpha_L\right) : I \cap \{0, 1, \ldots, k - 1\} \geq t \right) \right\},$$

form a basis for a Johnson graph code $C$ of type $JGC(n, v, k, r)$, where $r = \min(v, k) - t$. By Proposition 20, the vectors

$$\left\{ \left(\det\left(x^i; \alpha_L\right) : I \cap \{0, 1, \ldots, n - k - 1\} \geq v + 1 - t \right) \right\},$$

form a basis for a dual Johnson graph code $D$ of type $JGC(n, v, n - k, R - r - 1)$.

**Example 25** The code $C = JGC(5, 2, 1)$ and its dual $D$, both defined with $J(5, 2)$, are generated by vectors

$$\pi(M) = (\det\left(\begin{array}{c c c c c}
\alpha_{i_0}^{j_0} & \alpha_{i_0}^{j_1} \\
\alpha_{j_0}^{i_0} & \alpha_{j_0}^{i_1}
\end{array}\right) : 0 \leq j_0 < j_1 \leq 4).$$

$$C = JGC(5, 2, 2, 1) : 0 \leq i_0 < i_1 \leq 4, i_0 \leq 1.
\quad\quad\quad D = JGC(5, 2, 3, 0) : 0 \leq i_0 < i_1 \leq 4, i_1 \leq 2.$$

See also Example 3. The codes $C$ and $D$ use the same matrix $g$ when $\{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\} = F$. In general, the dual code $D_0$ of a Reed–Solomon code $C_0$ is equivalent to a Reed–Solomon code. For the general case, when the $n$ elements form a subset $\{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\} \subset F$, let $p(z) = (z - \alpha_0)(z - \alpha_1)\cdots(z - \alpha_{n-1})$. If $C$ is defined with $g$ then $D$ is defined with $g \Delta$, for $\Delta = -\text{diag}(p'(\alpha_0), p'(\alpha_1), \ldots, p'(\alpha_{n-1}))^{-1}$.

Let $E = \{0, 1, \ldots, n - 1\}$. The $n \times n$ Vandermonde matrix $g$ has rows $\{ev(x^i)\}_{i \in E}$. Let $f_i(x) = \prod_{j \neq i}(x - \alpha_j)$, for $i \in E$. The rows $\{ev(f_i)\}_{i \in E}$ describe the matrix $g$ in row reduced upper triangular form. A matrix $g$ in row reduced block form depends on a
choice of information set $A \subseteq E$ for $C_0$. After reordering columns we may assume $A = \{0, 1, \ldots, k - 1\}$. Let
\[
h_i(x) = \begin{cases} 
  x^i, & 0 \leq i \leq k - 1. \\
  f_k(x)x^{i-k}, & k \leq i \leq n - 1.
\end{cases}
\]

The rows $\{ev(h_i)\}_{i \in E}$ describe the matrix $g$ in row reduced block form.

**Remark 26** The two reduced forms for $g$, the upper triangular form and the block form, are obtained with lower triangular row operations and the three matrices share the same row spaces for a set of $k$ leading rows, for any $0 \leq k \leq n$.

For $g = \{ev(f_i)\}_{i \in E}$ in upper triangular form, the matrix $\Lambda(g)$ has entries
\[
\Lambda(g)_{I, L} = \det \left( f_i(\alpha_j) : i \in I, j \in L \right) =: \det(f_I; \alpha_L).
\]

For $g = \{ev(h_i)\}_{i \in E}$ in block form, the entries are
\[
\Lambda(g)_{I, L} = \det \left( h_i(\alpha_j) : i \in I, j \in L \right) =: \det(h_I; \alpha_L).
\]

The matrix $g$ in block form depends on a choice of information set $A \subseteq J(n, k)$. The matrix entry $\det(h_I; \alpha_L)$ further depends on a row index $I \in J(n, v)$ and a column index $L \in J(n, v)$. In [2, Appendix B] we show that as a function of $A$ and $L$, for a given $I$, the entry $\det(h_I; \alpha_L)$ can be interpreted, in general up to a triangular transformation, as a generalized principal subresultant of the two polynomials $p(x) = \prod_{j \in L}(x - \alpha_j)$ and $q(x) = \prod_{j < k}(x - \alpha_j)$. In particular it can be expressed as a linear combination of determinants of Sylvester type matrices for $p(x)$ and $q(x)$.

**Lemma 27** Let
\[
I = \{0, 1, \ldots, t - 1\} \cup \{k, k + 1, \ldots, k + v - t - 1\}.
\]

That is, $I$ is minimal in $J(n, v)$ such that $I \cap \{0, 1, \ldots, k - 1\} = \emptyset$. For all $L \in J(n, v)$, $\det(f_I; \alpha_L) = \det(h_I; \alpha_L)$, and the upper triangular form for $g$ and the block form for $g$ share the same $I$-th row in $\Lambda(g)$.

**Proof**
\[
\langle f_i : i \in I \rangle = \langle f_i : 0 \leq i \leq t - 1 \rangle + \langle f_i : k \leq i \leq k + v - t - 1 \rangle = \langle f_kx^{i-k} : k \leq i \leq k + v - t - 1 \rangle = \langle h_i : i \in I \rangle.
\]

\hfill \Box

For the choice of $I$ in the lemma, $\det(h_I; \alpha_L) = 0$ if and only if the $t$-th principal subresultant is zero for polynomials $q(x) = f_k(x)$ and $p(x) = \prod_{j \in L}(x - \alpha_j)$. We include an example and refer for further details to [2, Appendix B].

**Example 28** Let $I = \{0, 3, 4\}$, i.e., the case $t = 1, k = 3, v = 3$ in the lemma.
\[
\det(h_I; x_0, x_1, x_2) = \det \begin{pmatrix} 1 & 1 & 1 \\
  q(x_0) & q(x_1) & q(x_2) \\
  q(x_0)x_0 & q(x_1)x_1 & q(x_2)x_2 
\end{pmatrix},
\]
where $q(x) = f_3(x) = (x - \alpha_0)(x - \alpha_1)(x - \alpha_2)$. Let $p(x) = (x - x_0)(x - x_1)(x - x_2)$ and let $d(x) = \gcd(p(x), q(x))$. For $\deg d(x) > 1$, e.g., for $q(x_0) = q(x_1) = 0$, the matrix
Proposition 31. Let \( g \) be a Vandermonde matrix. Let rows and columns in \( g \) share the same ordering in which row \( L' \) precedes row \( L \) whenever \( L', L \in J(n, v) \) are adjacent and \( |L'| < |L| \). Then, for every \( L \in J(n, v) \), the determinant of the submatrix \( \Lambda(g)_{L_1, L_2 \leq L} = \Lambda(g)_{L_1, L_2 \leq L} \) factors as a product of binomial differences \( \alpha_i - \alpha_j \), \( i \neq j \).

Proof. We prove by induction to \( L \) that \( d_{\leq L} = \det \Lambda(g)_{\leq L} \) factors completely, as polynomial in \( \alpha_0, \alpha_1, \ldots, \alpha_{n-1} \), into factors \( \alpha_i - \alpha_j \). In general factors appear with multiplicities. For the base case \( L_0 = \{0, \ldots, v-1\} \), the scalar \( \Lambda(g)_{\leq L_0} = \det(x^{L_0}; \mu_{L_0}) \) is a Vandermonde determinant of degree \( d_0 = \sum_{i=0}^{v-1} i \). For the inductive case, observe that each new column \( L \) contributes two types of linear factors to \( d_{\leq L} \). There is a first contribution of \( d_0 \) factors because every entry in column \( L \) is divisible by \( \det(x^{L_0}; \mu_{L_0}) \) of degree \( d_0 \). Secondly, there is a contribution of one linear factor \( \alpha_i - \alpha_j \) for each \( L' < L \) that is adjacent to \( L \). By Lemmas 29 and 30 the number of such \( L' \) is \( |L| \). As \( \Lambda(g)_{\leq L} \) is enlarged to \( \Lambda(g)_{\leq L} \), both the degree of the determinant and its number of linear factors increases by \( |L| + d_0 \). 

The proposition applies to all principal minors of the matrix \( \Lambda(g) \). In particular, for the full matrix \( \Lambda(g) \) it follows that \( \det \Lambda(g) = (\det g)^m \), for \( m = (n-1)_{v-1} \).
We mention three types of codes that share common properties with Johnson graph codes: Grassmann codes [10] and Schubert codes [7], Polynomial Chinese remainder codes, and product matrix MSR codes [9].

For \( v \leq n \), rational points on the Grassmann variety \( G_{v,n} \) over a finite field \( F \) correspond to \( v \) dimensional \( F \)-linear subspaces of \( F^n \). The variety has a smooth embedding, using Plücker coordinates, into projective space \( \mathbb{P}^{N-1} \), for \( N = \binom{n}{v} \). The Grassmann code is the row space of the \( N \times K \) matrix whose columns are the images in \( \mathbb{P}^{N-1} \) of rational points \( P_1, P_2, \ldots, P_K \in G_{v,n} \). Over a field \( F \) of size \( q \), \( K \leq \#G_{v,n}(F) = \binom{n}{v}/q \). Let \( g = \{g_1, \ldots, g_n\} \) be a basis for \( F^n \). For \( \alpha \in J(n, v) \), write the elements in \( \alpha \) in increasing order \( 1 \leq \alpha_1 < \cdots < \alpha_v \leq n \). The Schubert variety \( \Omega_\alpha \subset G_{v,n} \) is the subvariety of subspaces \( W \) whose intersection with the flag generated by \( g \) reaches dimension \( i \) after at most \( \alpha_i \) steps.

\[
\Omega_\alpha = \{ W \in G_{v,n} : \dim(W \cap \langle g_1, \ldots, g_{\alpha_i} \rangle) \geq i, \text{ for } i = 1, \ldots, v \}.
\]

A Schubert code is defined by projecting the \( N \times K \) matrix onto a \( N \times K_\alpha \) submatrix with columns in \( \Omega_\alpha \). In general this reduces the rank of the matrix to \( k_\alpha \), and yields a Schubert code of dimension \( k_\alpha \) and length \( K_\alpha \). With hindsight, our construction of Johnson graph codes matches the column space of the \( N \times K_\alpha \) matrix, for the maximal \( \alpha \) in the Bruhat order with \( \alpha_i = k \). The Johnson graph code, as row space of a \( K_\alpha \times N \) matrix of rank \( k_\alpha \), has dimension \( k_\alpha \) and length \( N \). In our description, we find generators for the code by taking the top \( k_\alpha = \lvert B_r([0, 1, \ldots, k-1]) \rvert \) rows in a \( N \times N \) matrix \( \Lambda(g) \). With the above we can interpret information sets for the Johnson graph code as projections of the Plücker embedding of a Schubert variety that separate points.

We give a different construction for codes \( J(n, v, v, 1) \). The proof for the general case follows after the example.

**Example 32** Example 25 describes a code \( C = JGC(5, 2, 2, 1) \) and the equivalent code \( C' = JGC(5, 3, 3, 1) \). The first code uses an embedding of the pair \( \{x, y\} \subset \{\alpha_0, \alpha_1, \ldots, \alpha_4\} \) with Plücker coordinates \( \det_I(x, y), I \in \{01, 02, 03, 04, 12, 13, 14\} \). The second code uses an embedding of the triplet \( \{x, y, z\} \subset \{\alpha_0, \alpha_1, \ldots, \alpha_4\} \) with Plücker coordinates \( \det_I(x, y, z), I \in \{012, 013, 014, 023, 024, 123, 124\} \). A different embedding that also yields a code \( JGC(5, 2, 2, 1) \) is as vector of coefficients for the polynomial

\[
(1 + xt + x^2 t^2 + x^3 t^3) (1 + yt + y^2 t^2 + y^3 t^3) \quad (n = 5, v = 2).
\]

Similarly the coefficients for the polynomial

\[
(1 + xt + x^2 t^2) (1 + yt + y^2 t^2) (1 + zt + z^2 t^2) \quad (n = 5, v = 3).
\]

give an embedding for \( \{x, y, z\} \) that yields a code \( JGC(5, 3, 3, 1) \).

**Proposition 33** Let \( v \leq n \). Let \( f_0, f_1, \ldots, f_{n-1} \in F[x] \) be \( n \) polynomials of degree \( n - v \), such that any two polynomials are relative prime and any \( n - v + 1 \) polynomials are linearly independent. For each \( L \in J(n, v) \), let \( f_L = \prod_{i \in L} f_i \). Then \( \deg f_L = v(n - v) \). For any given \( A \in J(n, v) \), the collection \( \{ f_L : |L \cap A| \geq v - 1 \} \) forms a basis for \( F[x]_{\leq v(n-v)} \).

**Proof** The size of the collection \( \{ f_L \} \) is \( 1 + v(n-v) \), which is the dimension of \( F[x]_{\leq v(n-v)} \). It suffices to prove that the \( f_L \) in the collection span \( F[x]_{\leq v(n-v)} \). For any \( A \in A \), the collection contains \( n - v + 1 \) multiples \( f_A/f_j \cdot f_j \), for \( j = i \) or \( j \neq A \). Since the \( f_j \) span \( F[x]_{\leq n-v} \), the collection contains all multiples of \( f_A/f_i \), for all \( i \in A \). Since \( f_j \) and \( f_i \) are relative prime, the collection spans all multiples of \( f_A/f_i f_j \), for all pairs \( i, j \in A \). With induction it follows that the collection spans all multiples of 1, i.e., spans \( F[x]_{\leq v(n-v)} \). \( \square \)
Corollary 34 A special case of the proposition is $f_i = 1 + \alpha_i x + \cdots + \alpha_i^{n-v} x^{n-v}$, for distinct $\alpha_0, \alpha_1, \ldots, \alpha_{n-1} \in F$, such that $\gcd(n - v + 1, |F| - 1) = 1$.

A special case of a Johnson graph code is the following. Let $\alpha_0, \alpha_1, \ldots, \alpha_n \in F$ be distinct field elements. Let $S$ be a symmetric matrix of size $k-1 \times k-1$, with associated symmetric form $f(x, y) = (1, x, \ldots, x^{k-2}) S (1, y, \ldots, y^{k-2})^T$. Label vertices $\{i, j\} \in J(n, 2)$ with $f(\alpha_i, \alpha_j)$. Then $S$, and thus $f(\alpha_i, \alpha_j)$ for all $\{i, j\} \in J(n, 2)$, is uniquely determined by the set of values $f(\alpha_i, \alpha_j), i, j \in A$, for any $k$-subset $A$ of $\{0, 1, \ldots, n - 1\}$. Thus the code is of type $JGC(n, 2, k, 0)$. In [9], two copies of the code are combined to form an MSR regenerating code. Let the second copy be defined with a matrix $T$ and symmetric form $g(x, y)$. For each $i \in \{0, 1, \ldots, n - 1\}$, node $i$ stores the polynomial $f(\alpha_i, y) + \alpha_i g(\alpha_i, y)$, which is a polynomial of degree $k - 2$ in the single variable $y$. When data is collected from $k$ nodes $A$, any two accessed nodes $i, j \in A$ can recover $f(\alpha_i, \alpha_j)$ and $g(\alpha_i, \alpha_j)$ from $f(\alpha_i, \alpha_j) + \alpha_i g(\alpha_i, \alpha_j)$ and $f(\alpha_j, \alpha_j) + \alpha_j g(\alpha_j, \alpha_i)$, that is together they can decouple their stored values. With the decoupled values, the matrices $S$ and $T$ can be recovered as before. In this construction, the Johnson graph code is used as inner code and the coupling as outer code. In the next section, Johnson graph codes are used as outer codes and layered codes as inner codes.

6 Concatenated layered codes

In this section we use Johnson graph codes as a tool to concatenate layered codes with different layer sizes into longer codes with improved download properties. In a layered code, data in different layers is independent, and data can in general only be recovered by accessing any $n - 1$ out of $n$ nodes. In a concatenated code, Johnson graph codes provide relations between data in different layers that ensure that data can be recovered from fewer than $n - 1$ nodes. The design of concatenated codes is concerned with (1) The amount of data that remains unrecovered when accessing layered codes in fewer than $n - 1$ nodes, (2) The extra amount of data that can be recovered through the use of Johnson graph codes, and (3) How the latter can compensate for the former. For the most part in this section we only need to refer to basic features of layered codes and Johnson graph codes which will reduce the design problem to a combinatorial tiling problem.

The main feature of layered codes that we use is a partition into shells of all layers of a given size when $k$ of the $n$ nodes are accessed. Table 1 gives a partition into shells for layers of size four when $n = 6$ and $k = 3$. For the same $n$ and $k$, Table 2 gives the shell sizes for the partitions of layers with size four or smaller.

Symbols in a layer $L \subset \{0, 1, \ldots, n - 1\}$ are protected by a single parity $\sum_{i \in L} c_i = 0$. So that symbols in a layer $L$ are known to a data collector $A \subset \{0, 1, \ldots, n - 1\}$ with access to symbols $\{c_i : i \in A\}$ if and only if $|A \cap L| \geq |L| - 1$.
between layers in the same column, i.e. data from $L_i$ will transfer enough helper data to under-accessed layers, in the right order, to recover all data that needs to be transferred to under-accessed layers. The problem of full data recovery reduces to a combinatorial tiling problem of choosing a sequence of rounds.

Table 3 corresponds to a concatenation of layered codes with $w < 4$ can be used to recover data in under-accessed layers of size $v = 4$. Let $A = \{0, 1, 2\}$. When considered as vertices in the Johnson graph $J(6, 4)$, the $\binom{6}{4} = 15$ layers of size 4

Definition 35 Given a layered code with $n$ nodes, and upon accessing a subset of nodes $A \subset \{0, 1, \ldots, n - 1\}$, we call a layer $L$ fully-accessed if all symbols in $L$ are available to the data collector. We call $L$ sufficiently-accessed if $L$ is not fully-accessed but its symbols can be recovered from the collected symbols with the single parity check on the symbols in $L$. A layer that can not be recovered using a combination of collected symbols and parity checks is called under-accessed.

In Table 2, sufficiently-accessed layers, with $|A \cap L| = |L| - 1$, appear on the diagonal. Fully-accessed layers appear below the diagonal and under-accessed layers above the diagonal. A fully-accessed layer $L$ does not need the parity check $\sum_{i \in L} c_i = 0$ to recover its symbols. If we replace the check with a check of the form $\sum_{i \in L} c_i = s$, for a symbol $s$ of our choice, then the fully-accessed layer can pass the symbol $s$ to a different under-accessed layer. The transfer of data proceeds in rounds. Each round requires a choice of three parameters $w$, $v$, and $r$. Here $w$ is the size of a fully-accessed layer providing helper data, $v$ is the size of an under-accessed layer receiving helper data, and $r$ restricts the layers receiving helper data to those with $w \leq |A \cap L| \leq v - 1 - r$. It is assumed that at the time of transfer, data has been recovered from all layers of size $w < v$ and from all layers of size $v$ with $|A \cap L| > v - 1 - r$. A Johnson graph code $JGC(n - w, v - w, k - w, r)$ is used to transfer one symbol of helper data to each layer $L$ with $|L| = v$ and $w \leq |A \cap L| \leq v - 1 - r$ using a combination of helper data from sublayers of size $w$ and data available from layers with $|L| = v$ and $|A \cap L| > v - 1 - r$. Table 2 can be used to keep track of the amount of data that needs to be transferred to under-accessed layers. The problem of full data recovery reduces to a combinatorial tiling problem of choosing a sequence of rounds $(w_i, v_i, r_i)$ that will transfer enough helper data to under-accessed layers, in the right order, to recover all their data.

There is a wide range of choices for the sequence $(w_i, v_i, r_i)$ to achieve full data recovery. There is a unique sequence if we restrict the choice to $w_i = 1$ in each round. This results in concatenated codes with the same storage and bandwidth parameters as the improved layered codes in [12]. Another choice is to set $r = v - 1 - w$ in each round, so that helper data from $L_w$ is transferred only to layers $L$ with $A \cap L = L_w$. This results in codes with the same storage and bandwidth as cascade codes in [6]. We include examples of each type that recover data in the layers of size $|L| = 5$ in Table 2 under the assumption that data in smaller layers has been recovered.

Example 36 Table 3 corresponds to a concatenation of layered codes with $v = 4$ (1×), $v = 2$ (3×), and $v = 1$ (2×). The multiplicities are unique if we restrict data transfer to transfer between layers in the same column, i.e. data from $L_w$ is transferred only to layers $L$ with $A \cap L = L_w$. We describe in detail how for the given multiplicities data stored in layers of size $w < 4$ can be used to recover data in under-accessed layers of size $v = 4$. Let $A = \{0, 1, 2\}$.

| $|A \cap L|$ | 3 | 2 | 1 | 0 | Multiplicity |
|----------------|---|---|---|---|-------------|
| $|L| = 4$       | -3 · 1 | -1 · 2 | 1 x |
| 3              | +1 · 1 | -3 · 1 | -1 · 2 | 0 x |
| 2              | +1 · 1 | -3 · 1 | 3 x |
| 1              | +1 · 1 | 2 x |
| $\Sigma$       | 0 | 0 | 0 | -9 |
the full vector describes the parameters for the concatenated code.

under-accessed layer gives the unique multiplicities of layered codes in a concatenated layered code such that each
three accessed nodes. The fully-accessed layers of size $w$ fully-accessed layers of size $w$
shells are under-accessed.

$\begin{array}{cccc}
J(4, 2) = & (L : L \supseteq \{1, 2\}) & J(5, 3) = & (L : L \supseteq \{2\}) \\
\# & 0 & 1, 2 & 3, 4, 5 & r \\
3 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
3 - * & * & * & * & - & 0 & 0 & 0 & 0 & 0 \\
3 - * & * & * & * & - & 1 & 1 & 1 & 1 & 1 \\
\end{array}$

divide over shells $S_0(A), S_1(A), S_2(A)$ of sizes 3, 9, 3, respectively (Table 1(b), or the line $|L| = 4$ in Table 2). Layers in $S_0(A)$ are sufficiently-accessed and layers in the remaining shells are under-accessed.

We recover the data stored in $S_1(A) = \{L : |L| = 4, |A \cap L| = 2\}$ and $S_2(A) = \{L : |L| = 4, |A \cap L| = 1\}$ recursively in two rounds. For the recovery, helper data will be transferred from sublayers of size $w = 2$ and $w = 1$, respectively. Let $A = \{0, 1, 2\}$ be the three accessed nodes. The fully-accessed layers of size $w = 2$ are 01, 02, 12 and the fully-accessed layers of size $w = 1$ are 0, 1, 2. Table 4 shows the subgraphs $\{L : L \supseteq \{1, 2\}\}$ and $\{|L : L \supseteq \{2\}\}$ of $J(6, 4)$.

(Round $1 - w = 2, v = 4, r = 1$) The vertices of the subgraph $J(4, 2) = \{L : L \supseteq \{1, 2\}\} \subset J(6, 4)$ are labeled with a vector $(c_L)$ of length 6. The layer $\{1, 2\}$ stores a vector $s = H_{1c}$ of length 3. The six layers in $J(4, 2)$ belong to $S_0(A)(3 \times)$ and $S_1(A)(3 \times)$ (Table 4). The vector $s$ together with three available $c_L$, for $L \in S_0(A)$, determine the full vector $(c_L)$. For $L \in S_1(A)$, after receiving the symbol $c_L$, the data in $L$ can be recovered.

(Round $2 - w = 1, v = 4, r = 2$) The vertices of the subgraph $J(5, 3) = \{L : L \supseteq \{1, 2\}\} \subset J(6, 4)$ are labeled with a vector $(c_L)$ of length 10. The layer $\{2\}$ stores a vector $s = H_{2c}$ of length 1. The ten layers in $J(5, 3)$ belong to $S_0(A)(3 \times), S_1(A)(6 \times)$ and $S_2(A)(1 \times)$ (Table 4). The vector $s$ together with nine available $c_L$, for $L \in S_0(A) \cup S_1(A)$, determine the full vector $(c_L)$. For $L \in S_2(A)$, recovery of its data requires two additional symbols, and we store two vectors $s$ of length 1 at each layer $L_w$ of size 1. With the two additional symbols, the data in $L \in S_2(A)$ can be recovered.

The parameters for the code in Example 36 appear in Table 6. The following proposition gives the unique multiplicities of layered codes in a concatenated layered code such that each under-accessed layer $L$ receives all its missing data from the sublayer $A \cap L$. The corollary describes the parameters for the concatenated code.

**Theorem 37** Let

$$f_{\ell}(t) = \frac{1}{(1 - \ell t)(1 + t)^{\ell}} = \sum_{i \geq 0} a_i t^i.$$  

The concatenation of a single layered code with layers of size $v$ together with $a_i$ copies of layered codes with layers of size $v - i$, for $0 < i < v$, results in a concatenated code whose data is recoverable from any $k = n - 1 - \ell$ nodes.

**Proof** For general number of nodes $n$, top layer size $|L| = v$, and number of accessed nodes $|A| = k$, a sublayer of size $v - i$ provides $a_i$ helper symbols to $a_{i - j}(n - k)^{\binom{j}{2}}$ layers of size $v - i + j$, each of which receives $j - 1$ symbols for full data recovery, for $0 < j \leq i$. This gives the recursion $a_i = \sum_{0 < j \leq i} a_{i - j}(n - k)^{\binom{j}{2}}(j - 1)$, with $f_{n - k - 1}(t)$ as generating function.  

\[\square\]
Table 5  Different concatenated (7,4,6) codes

|   | 5 | 4 | 3 | 2 | 1 | M | α | β |
|---|---|---|---|---|---|---|---|---|
| 3 − 2 | 1 | − | 3 | 2 | 9 | 324 | 81 | 27 |
| 3 − 1 | 1 | − | 3 | − | 12 | 288 | 72 | 25 |
| 2 − 2 | 3 | − | − | 12 | − | 468 | 117 | 42 |
| 2 − 1 | 6 | − | − | 14 | 15 | 756 | 189 | 74 |
| 1 − 1 | 4 | − | − | − | 24 | 336 | 84 | 40 |

Table 6  Parameters

(M = M_1 − M_0, α, β) = (81, 27, 9) = (3^4, 3^3, 3^2) for a (n, k, d) = (6, 3, 5) concatenated layered code

| #L | M_1 | α | β | M_0 | M_1 − M_0 | multiplicity |
|----|-----|---|---|-----|-----------|-------------|
| | 4   | 15 | 10 | 6  | 45        | 1 ×         |
| 3 | 20  | 40 | 10 | 4  | 40        | 0 ×         |
| 2 | 15  | 15 | 5  | 1  | 12        | 3 ×         |
| 1 | 6   | 0  | 1  | 0  | 0         | 2 ×         |
| Σ | 90  | 27 | 9  | 9  | 81        |             |

Corollary 38  For the concatenated code in Theorem 37,

α = [t^{n-1}] f_ℓ(t)(1 + t)^{n-1},  β = [t^{n-2}] f_ℓ(t)(1 + t)^{n-2}.

For the special case v = n − ℓ = k + 1,

M = kα,  α = (n − k)^k,  β = (n − k)^{k−1}.

The latter parameters satisfy the MSR (for Minimal Storage Regenerating code) conditions M = kα and α = (d − k + 1)β.

The parameters for the codes in the corollary match those of cascade codes and moulin codes. Concatenated layered codes are of type (n, k, d) with k < d = n − 1. Cascade codes and moulin codes extend to arbitrary n in which case the parameters remain the same as for codes with d = n − 1. An unrestricted n means that nodes can be added to an existing (n, k, d) configuration such that the new configuration (n′, k, d), with n′ > n, maintains the property that data can be collected from any k nodes and any d nodes can be contacted to repair a single failed node.

Example 39  Table 5 shows different concatenated codes that are constructed without the assumption of the proposition that each under-accessed layer L receives all its missing data from the unique sublayer A ∩ L. The codes are constructed with top layers of size v = 5 and smaller layered codes are added to provide missing data. When accessed in four nodes A, the layers of size 5 partition into S_0(A), S_1(A), S_2(A). The code in row i − j of the table uses layers of size i to provide missing data to layers in S_1(A) and layers of size j to provide missing data to layers in S_2(A). When the multiplicities in a row have a common divisor it is to avoid fractional amounts in data transfer.

References

1. Dimakis A.G., Godfrey P.B., Wu Y., Wainwright M.J., Ramchandran K.: Network coding for distributed storage systems. IEEE Trans. Inf. Theory 56(9), 4539–4551 (2010).
2. Duursma I., Li X.: Johnson graph codes. Preprint at http://arxiv.org/abs/1912.10388
3. Duursma I., Li X., Wang H.-P.: Multilinear algebra for distributed storage. SIAM J. Appl. Algebra Geom. 5(3), 552–587 (2021).
4. Elyasi M., Mohajer S.: Determinant coding: a novel framework for exact-repair regenerating codes. IEEE Trans. Inf. Theory 62(12), 6683–6697 (2016).
5. Elyasi M., Mohajer S.: A cascade code construction for (n, k, d) distributed storage systems. In: 2018 IEEE ISIT Proceedings, pp. 1241–1245 (2018).
6. Elyasi M., Mohajer S.: Cascade codes for distributed storage systems. IEEE Trans. Inf. Theory 66(12), 7490–7527 (2020).
7. Ghorpade S.R., Tsfasman M.A.: Schubert varieties, linear codes and enumerative combinatorics. Finite Fields Appl. 11(4), 684–699 (2005).
8. Gopalan P., Huang C., Simitci H., Yekhanin S.: On the locality of codeword symbols. IEEE Trans. Inf. Theory 58(11), 6925–6934 (2012).
9. Rashmi K.V., Shah N.B., Kumar P.V.: Optimal exact-regenerating codes for distributed storage at the mrs and mbr points via a product-matrix construction. IEEE Trans. Inf. Theory 57(8), 5227–5239 (2011).
10. Ryan C.T., Ryan K.M.: The minimum weight of the Grassmann codes C(k, n). Discret. Appl. Math. 28(2), 149–156 (1990).
11. Sasidharan B., Vajha M., Kumar P.V.: An explicit, coupled-layer construction of a high-rate MSR code with low sub-packetization level, small field size and d<(n − 1). In: 2017 IEEE ISIT Proceedings, pp. 2048–2052 (2017).
12. Senthoor K., Sasidharan B., Kumar P.V.: Improved layered regenerating codes characterizing the exact-repair storage-repair bandwidth tradeoff for certain parameter sets. In: 2015 IEEE ISIT Proceedings, pp. 2051–2055 (2015).
13. Tian C., Sasidharan B., Aggarwal V., Vaishampayan V., Kumar P.V.: Layered exact-repair regenerating codes via embedded error correction and block designs. IEEE Trans. Inf. Theory 61(4), 1933–1947 (2014).

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.