

Asymptotically Optimal Regenerating Codes Over Any Field

Netanel Raviv

*Computer Science Department,
Technion – Israel Institute of Technology,
Haifa 3200003, Israel*

Abstract

The study of regenerating codes has advanced tremendously in recent years. However, most known constructions require large field size, and hence may be hard to implement in practice. By using notions from the theory of extension fields, we obtain two explicit constructions of regenerating codes. These codes approach the cut-set bound as the reconstruction degree increases, and may be realized over any given field if the file size is large enough. Since distributed storage systems are the main purpose of regenerating codes, this file size restriction is trivially satisfied in most conceivable scenarios. The first construction attains the cut-set bound at the MBR point asymptotically for all parameters, whereas the second one attains the cut-set bound at the MSR point asymptotically for low-rate parameters.

1 Introduction

Since the emergence of cloud storage platforms, distributed storage systems are ubiquitous. As classic erasure correction codes fail to scale with the exponential growth of data, regenerating codes were proposed [27].

A regenerating code is described by the parameters $(n, k, d, B, q, \alpha, \beta)$, where $k \leq d \leq n - 1$ and $\beta \leq \alpha$. The file $x \in \mathbb{F}_q^B$ is to be stored on n storage nodes. The *reconstruction degree* k is the number of nodes required to restore x , a process which is called *reconstruction*, and carried out by a *data collector*. The *repair degree* d is the number of *helper nodes* which are required to restore a lost node, a process which is called *repair*, and carried out by a *newcomer node* (abbrv. newcomer). The parameter α denotes the number of field elements per storage node, and the parameter β denotes the number of field elements which are to be downloaded from each helper node during repair. Further requirements are the ability to reconstruct from *any* set of k nodes, and to repair from *any* set of d nodes.

In [27], the parameters of any regenerating code were shown to satisfy the so called *cut-set* bound

$$B \leq \sum_{i=0}^{k-1} \min\{\alpha, (d-i)\beta\}, \quad (1)$$

from which a tradeoff between α and β is apparent. One point of this tradeoff, in which α is minimized, attains $\alpha = \frac{B}{k}$, whereas the second point, in which β is minimized, attains $\alpha = \beta d$. Codes

This work was done while Netanel Raviv was a visiting student at the University of Toronto, under the supervision of Prof. Frank Kschischang. It is a part of his Ph.D. thesis performed at the Technion, under the supervision of Prof. Tuvia Etzion. *e-mail*: netanel.raviv@gmail.com.

which attain (1) with equality and have $\alpha = \frac{B}{k}$ are called Minimum Storage Regenerating (MSR) codes. Codes which attain (1) with equality and have $\alpha = d\beta$ are called Minimum Bandwidth Regenerating (MBR) codes.

In the first part of this paper, regenerating codes with $\alpha = d\beta$ are constructed. These codes have B that asymptotically attains (1) with equality as k increases, and is close to attaining equality even for small values of k . In addition, as long as the file size is large enough, these codes may be realized over any given field, and in particular, the binary field. This restriction on the file size is usually satisfied in typical distributed storage systems. The second part of the paper contains a construction of regenerating codes with $d \geq 2k - 2$ that have B which approaches αk as k increases. As in the first part, these codes may also be realized over any given field if the file size is large enough.

Conceptually, this construction serves both as a mathematical proof of concept that almost optimal regenerating codes exist over any field, and as a means to reduce the complexity of the involved encoding, reconstruction, and repair algorithms by using smaller finite field arithmetics. Note that by employing the structure of an extension field as a vector space over its base field, one may implement any code over an extension field by operations over the base field. However, this approach requires implementation of sophisticated circuits for multiplication over the extension field, while employing the base field itself enables implementation using matrix multiplication only.

Our techniques are inspired by the code construction in [20], which attains MBR codes for all parameters n, k , and d , as well as MSR codes for all parameters n, k , and d such that $d \geq 2k - 2$, where both constructions have $\beta = 1$. It is noted in [20, Sec. I.C] that only the case $\beta = 1$ is discussed since *striping of data* is possible, and larger β may be obtained by code concatenation. It will be shown in the sequel that allowing a larger β , *not* through concatenation, enables a significant reduction in field size with a small and often negligible loss of code rate.

According to [20, Sec. I.B.], regenerating codes which do not attain (1) with equality are not MBR codes, even if they satisfy $\alpha = d\beta$ and attain (1) asymptotically. Similarly, regenerating codes which attain $B = \alpha k$ asymptotically are not considered MSR codes. To the best of our knowledge, such codes were not previously studied, and hence, we coin the following terms.

Definition 1. *A regenerating code is called a nearly MBR (NMBR) code if it satisfies $\alpha = \beta d$, and B approaches the cut-set bound (1) as k increases. Similarly, a regenerating code is called a nearly MSR (NMSR) code if B approaches αk as k increases.*

This paper is organized as follows. Previous work is discussed in Section 2. Mathematical background on several notions from field theory, number theory, and matrix analysis is given in Section 3. NMBR codes are given in Section 4, and NMSR codes in Section 5, each of which contains a subsection with a detailed asymptotic analysis and numerical examples. Finally, concluding remarks with future research directions are given in Section 6.

2 Previous Work

MBR codes for all parameters n, k , and d were constructed in [20], where the underlying field size q must be at least n , and $B = \binom{k+1}{2} + k(d - k)$. MSR codes for all parameters n, k , and d such that $d \geq 2k - 2$ were also constructed in [20], where the underlying field size must be at least $n(d - k + 1)$, and $B = (d - k + 1)(d - k + 2)$. These codes are given under a powerful framework called *product matrix codes*, and are the main objects of comparison in this paper. Henceforth, these codes are denoted by PM-MBR and PM-MSR, respectively.

Broadly speaking, the construction of PM-MBR codes associates a distinct field element γ to each storage node, which stores $(1, \gamma, \gamma^2, \dots, \gamma^d) \cdot M$, where M is a symmetric matrix that contains the

file x . In our NMBR construction we replace the vector $(1, \gamma, \gamma^2, \dots, \gamma^d)$ by a properly chosen matrix. This matrix is associated with an element of an *extension field* of the field \mathbb{F}_q , an approach which enables a reduction in field size. Our NMSR construction uses similar notions, where the proofs are a bit more involved, and require tools from basic number theory and matrix analysis.

Product matrix codes were recently improved in [5]. The improvement is obtained by operating over a ring \mathcal{R}_m in which addition and multiplication may be implemented by cyclic shifts and binary additions. The size of \mathcal{R}_m is 2^m , where m must not be divisible by $2, \dots, n - 1$ [5, Th. 10]. Using our techniques, it is possible to employ the binary field itself (or any other given field), rather than the aforementioned ring \mathcal{R}_m , with minor loss of rate. PM-MSR codes were also recently improved in [15], which reduced the required field size to $q > n$, whenever q is a power of two. This result follows from a special case of our construction (see Remark 3 in Section 5).

A closely related family of MBR codes called *repair-by-transfer* codes is discussed in [11]. In repair-by-transfer codes a node which participates in a repair must transfer its data without any additional computations. In [11], the field size required for repair-by-transfer MBR codes is reduced to $O(n)$ instead of $O(n^2)$ in previous constructions [25]. A similar result is obtained by [14], which also studied the repair and reconstruction complexities. In addition, [11] obtain *binary* repair-by-transfer codes for the special cases $k = d = n - 2$ and $k + 1 = d = n - 2$.

Further aspects of regenerating codes were thoroughly studied in recent years [2, 7, 8, 9, 18, 21, 26]. In particular, the problem of constructing high rate MSR codes, i.e., with a constant number of parity nodes, has received a great deal of attention [4, 22, 24, 29, 30]. Implementing our techniques for high rate MSR codes is one of our future research directions.

3 Preliminaries

This section lists several notions from field theory, linear algebra, number theory and matrix analysis, which are required for the constructions that follow. To this end, the following notations are introduced. For an integer m , the notation \mathbb{Z}_m stands for the ring of integers modulo m , and $[m] \triangleq \{1, \dots, m\}$. The ring of univariate polynomials over \mathbb{F}_q is denoted by $\mathbb{F}_q[x]$. For integers s and t , the notations $\mathbb{F}_q^{s \times t}$ stands for the ring of $s \times t$ matrices over \mathbb{F}_q . For a matrix A , let $A_{i,j}$ be its (i, j) -th entry, and let A_i be its i -th row or column, where ambiguity is resolved if unclear from context. If A is a matrix in $\mathbb{F}_q^{ms \times mt}$ which consists of $s \cdot t$ blocks of size $m \times m$ each, we denote its (i, j) -th block by $\llbracket A \rrbracket_{i,j}^{(m)}$, and omit the notation (m) if it is clear from the context. The notations I_m and $\mathbf{0}_m$ are used to denote the identity and zero matrix of order m , respectively.

3.1 Companion matrices and representation of extension fields

Definition 2. *The companion matrix of a monic univariate polynomial $P(x) = p_0 + p_1x + \dots + p_{e-1}x^{e-1} + x^e \in \mathbb{F}_q[x]$ is the $e \times e$ matrix*

$$\begin{pmatrix} 0 & 0 & \cdots & -p_0 \\ 1 & 0 & \cdots & -p_1 \\ 0 & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & -p_{e-1} \end{pmatrix}.$$

It is an easy exercise to show that the minimal and characteristic polynomials of a companion matrix are its corresponding polynomial, and the eigenvalues are the roots of that polynomial (which may reside in an extension field of the field of coefficients).

The following lemma, which is well-known, provides a convenient yet redundant representation of extension fields as matrices over the base field. Unlike other representations, this representation encapsulates both the additive and the multiplicative operations in the extension field, both as the respective operations between matrices.

Lemma 1. [13, Ch. 2, Sec. 5] *If $P \in \mathbb{F}_q[x]$ is monic and irreducible of degree m with companion matrix M_P , then the linear span over \mathbb{F}_q of the set $\{M_P^i\}_{i=0}^{m-1}$ is isomorphic to \mathbb{F}_{q^m} . If P is also primitive, then $\{M_P^i\}_{i=0}^{q^m-2} \cup \{0\}$ is isomorphic to \mathbb{F}_{q^m} .*

Lemma 1 also has an inverse [28]. That is, given the field \mathbb{F}_{q^m} , it is possible to represent its elements as all powers of the companion matrix P which corresponds to an irreducible polynomial of degree m over \mathbb{F}_q . Hence, for any m and any such matrix P , let $\theta_P : \mathbb{F}_{q^m} \rightarrow \mathbb{F}_q^{m \times m}$ be the function which maps an element in the extension field \mathbb{F}_{q^m} to its matrix representation in $\mathbb{F}_q^{m \times m}$ as a linear combination of powers of P , and since our results are oblivious to the choice of P , we denote θ_P by θ . Notice that θ is a *field isomorphism*, that is, every y_1 and y_2 in \mathbb{F}_{q^m} satisfy that $\theta(y_1 \cdot y_2) = \theta(y_1) \cdot \theta(y_2)$ and $\theta(y_1 + y_2) = \theta(y_1) + \theta(y_2)$. The function θ can be naturally extended to matrices, where $A \in \mathbb{F}_{q^m}^{s \times t}$ is mapped to

$$\Theta(A) \triangleq \begin{pmatrix} \theta(A_{1,1}) & \theta(A_{1,2}) & \cdots & \theta(A_{1,t}) \\ \theta(A_{2,1}) & \theta(A_{2,2}) & \cdots & \theta(A_{2,t}) \\ \vdots & \vdots & \ddots & \vdots \\ \theta(A_{s,1}) & \theta(A_{s,2}) & \cdots & \theta(A_{s,t}) \end{pmatrix} \in \mathbb{F}_q^{ms \times mt}. \quad (2)$$

Lemma 2. *For any integers m, s, t and ℓ , if $A \in \mathbb{F}_{q^m}^{s \times t}$ and $B \in \mathbb{F}_{q^m}^{t \times \ell}$ then $\Theta(AB) = \Theta(A) \cdot \Theta(B)$.*

Proof. By the definition of Θ , and by using the fact that θ is a field isomorphism, for all $i \in [s]$ and $j \in [\ell]$ we have that

$$\begin{aligned} \llbracket \Theta(AB) \rrbracket_{i,j}^{(m)} &= \theta \left(\sum_{k=1}^t A_{i,k} B_{k,j} \right) = \sum_{k=1}^t \theta(A_{i,k}) \theta(B_{k,j}) \\ &= \sum_{k=1}^t \llbracket \Theta(A) \rrbracket_{i,k}^{(m)} \llbracket \Theta(B) \rrbracket_{k,j}^{(m)} = \llbracket \Theta(A) \cdot \Theta(B) \rrbracket_{i,j}^{(m)}. \end{aligned}$$

□

Lemma 3. *For any integers m and t , if $A \in \mathbb{F}_{q^m}^{t \times t}$ is invertible then $\Theta(A) \in \mathbb{F}_q^{mt \times mt}$ is invertible.*

Proof. According to Lemma 2, since A^{-1} exists it follows that

$$I_{mt} = \Theta(I_t) = \Theta(A \cdot A^{-1}) = \Theta(A) \cdot \Theta(A^{-1}),$$

and hence $\Theta(A^{-1})$ is the inverse of $\Theta(A)$. □

3.2 Kronecker products and cyclotomic cosets

The proofs of the construction of NMSR codes in Subsection 5.1 are slightly more involved than those given in other sections. The main tools in those proofs are cyclotomic cosets and Kronecker products, which are discussed in this subsection.

Definition 3. [6, Sec. 3.7], [23, Sec. 7.5] For an integer m , a prime power q such that $\gcd(q, m) = 1$, and $s \in \mathbb{Z}_m$, a subset of \mathbb{Z}_m of the form $\{s, sq, sq^2, sq^3, \dots\}$ is called a q -cyclotomic coset modulo m .

It is well known (e.g., [6, 23]) that for any m such that $\gcd(q, m) = 1$, the size of any q -cyclotomic coset modulo m divides the order of q in \mathbb{Z}_m (that is, the smallest integer t such that $q^t = 1 \pmod{m}$).

Definition 4. For a matrix $A \in \mathbb{F}_q^{s \times t}$ and a matrix $B \in \mathbb{F}_q^{n \times m}$, the Kronecker product $A \otimes B$ is the matrix

$$\begin{pmatrix} A_{1,1}B & A_{1,2}B & \cdots & A_{1,t}B \\ A_{2,1}B & A_{2,2}B & \cdots & A_{2,t}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{s,1}B & A_{s,2}B & \cdots & A_{s,t}B \end{pmatrix} \in \mathbb{F}_q^{sn \times tm}.$$

The Kronecker product is useful when solving equations in which the unknown variable is a matrix. This application is enabled through an operator called vec , defined as follows.

Definition 5. [17, Def. 1] For a matrix $A \in \mathbb{F}_q^{s \times t}$ with columns A_1, \dots, A_t , let

$$\text{vec}(A) \triangleq \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_t \end{pmatrix} \in \mathbb{F}_q^{st}.$$

The following two lemmas present several properties of the Kronecker product and the vec operator. These lemmas are well known, and their respective proofs may be found, e.g., in [1, 12, 17]. In particular, Lemma 4 which follows discusses a close variant of the so called *Sylvester equation* $AX + XB = C$, where A, B , and C are known matrices, and X is an unknown matrix. For completeness, full proofs are detailed below.

Lemma 4. For an integer m , if A, X , and B are $m \times m$ matrices over \mathbb{F}_q , then $\text{vec}(AXB - X) = (B^\top \otimes A - I_{m^2}) \cdot \text{vec}(X)$.

Proof. Clearly, if X_1, \dots, X_m are the columns of X and B_1, \dots, B_m are the columns of B , then the i -th column of $(AXB - X)$ is

$$\begin{aligned} AXB_i - X_i &= \sum_{j=1}^m B_{j,i}(AX)_j - X_i \\ &= \sum_{j=1}^m B_{j,i}AX_j - X_i \\ &= (B_{1,i}A \quad B_{2,i}A \quad \cdots \quad B_{i-1,i}A \quad B_{i,i}A - I_m \quad B_{i+1,i}A \quad \cdots \quad B_{m,i}A) \cdot \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{pmatrix}, \end{aligned}$$

and hence, according to Definition 4, it follows that

$$\begin{aligned} \text{vec}(AXB - X) &= \begin{pmatrix} B_{1,1}A - I_m & B_{2,1}A & \cdots & B_{m,1}A \\ B_{1,2}A & B_{2,2}A - I_m & \cdots & B_{m,2}A \\ \vdots & \vdots & \ddots & \vdots \\ B_{1,m}A & B_{2,m}A & \cdots & B_{m,m}A - I_m \end{pmatrix} \cdot \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{pmatrix} \\ &= (B^\top \otimes A - I_{m^2}) \cdot \text{vec}(X). \end{aligned}$$

□

Lemma 5. *If A and B are two $m \times m$ matrices over \mathbb{F}_q , and \mathbb{F}_{q^ℓ} is a field which contains all eigenvalues $\lambda_1, \dots, \lambda_m$ of A and μ_1, \dots, μ_m of B , then the eigenvalues of $A \otimes B$ are $\{\lambda_i \mu_j | i, j \in [m]\}$.*

Proof. For any i and j in $[m]$, let v_i and u_j be (column) eigenvectors in $\mathbb{F}_{q^\ell}^m$ such that $Av_i = \lambda_i v_i$ and $Bu_j = \mu_j u_j$. By Definition 4, it follows that

$$\begin{aligned}
(A \otimes B) \cdot (u_j \otimes v_i) &= \begin{pmatrix} B_{1,1}A & B_{1,2}A & \dots & B_{1,m}A \\ B_{2,1}A & B_{2,2}A & \dots & B_{2,m}A \\ \vdots & \vdots & \ddots & \vdots \\ B_{m,1}A & B_{m,2}A & \dots & B_{m,m}A \end{pmatrix} \cdot \begin{pmatrix} u_{j,1}v_i \\ \vdots \\ u_{j,n}v_i \end{pmatrix} \\
&= \begin{pmatrix} u_{j,1}B_{1,1}Av_i + u_{j,2}B_{1,2}Av_i + \dots + u_{j,n}B_{1,m}Av_i \\ u_{j,1}B_{2,1}Av_i + u_{j,2}B_{2,2}Av_i + \dots + u_{j,n}B_{2,m}Av_i \\ \vdots \\ u_{j,1}B_{m,1}Av_i + u_{j,2}B_{m,2}Av_i + \dots + u_{j,n}B_{m,m}Av_i \end{pmatrix} \\
&= \lambda_i \cdot \begin{pmatrix} B_{1,1}u_{j,1}v_i + B_{1,2}u_{j,2}v_i + \dots + B_{1,m}u_{j,n}v_i \\ B_{2,1}u_{j,1}v_i + B_{2,2}u_{j,2}v_i + \dots + B_{2,m}u_{j,n}v_i \\ \vdots \\ B_{m,1}u_{j,1}v_i + B_{m,2}u_{j,2}v_i + \dots + B_{m,m}u_{j,n}v_i \end{pmatrix} \\
&= \lambda_i \cdot \begin{pmatrix} (Bu_j)_1 v_i \\ (Bu_j)_2 v_i \\ \vdots \\ (Bu_j)_m v_i \end{pmatrix} = \lambda_i (Bu_j) \otimes (v_i) = \lambda_i \mu_j (u_j \otimes v_i).
\end{aligned}$$

□

The following technical lemma will be required in the application of Lemma 4. Although it follows immediately from one of the common equivalent definitions of eigenvalues, a full proof is given.

Lemma 6. *If A is an $m \times m$ matrix over \mathbb{F}_q , then $A - I$ is invertible if and only if 1 is not an eigenvalue of A .*

Proof. Assume $A - I$ is invertible. If 1 is an eigenvalue of A then there exists a nonzero vector $v \in \mathbb{F}_q^m$ such that $Av = v$, and hence, $(A - I)v = v - v = 0$, and hence $\ker(A - I) \neq \{0\}$, which implies that $A - I$ is *not* invertible, a contradiction.

Conversely, assume that 1 is not an eigenvalue of A . If $A - I$ is not invertible then there exists a nonzero vector $v \in \mathbb{F}_q^m$ such that $(A - I)v = 0$, which implies that $Av = v$, and hence 1 is an eigenvalue of A , a contradiction. □

4 Nearly MBR codes

For any given n, k, d, q , and a sufficiently large file size B , this section presents regenerating codes with $\alpha = d\beta$, and B which approaches the cut-set bound as k increases. For any such n, k, d and q let b be an integer such that

$$A1. \quad b \geq k \log_q n,$$

A2. $k \mid b$,

and let $B \triangleq \frac{b(b+1)}{2} + b^2 \left(\frac{d}{k} - 1\right)$. Notice that Condition A1 implies that $B = \Omega(kd \log_q(n)^2)$. Since usually, the file size B is in the order of magnitude of billions, and the number of nodes is in the order of magnitude of dozens, Condition A1 is trivially satisfied in many distributed storage systems (see Subsection 4.2 for explicit examples).

4.1 Construction

Given a file $x \in \mathbb{F}_q^B$, define the following data matrix, which resembles the corresponding one in [20]:

$$X = \begin{pmatrix} S & T \\ T^\top & 0 \end{pmatrix} \in \mathbb{F}_q^{\frac{db}{k} \times \frac{db}{k}}, \text{ where}$$

$$S \triangleq \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_b \\ x_2 & x_{b+1} & x_{b+2} & \dots & x_{2b-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_b & \dots & \dots & \dots & x_{\frac{b(b+1)}{2}} \end{pmatrix} \in \mathbb{F}_q^{b \times b}, \quad (3)$$

and $T \in \mathbb{F}_q^{b \times b(d/k-1)}$ contains the remaining $b^2 \left(\frac{d}{k} - 1\right)$ elements of x in some arbitrary order.

Let P be a companion matrix of any primitive polynomial of degree $\frac{b}{k}$ over \mathbb{F}_q , and let i_1, \dots, i_n be distinct integers in the range $\{0, \dots, q^{b/k} - 1\}$, which exist by A1. Using P and i_1, \dots, i_n , define the following encoding matrix,

$$M = \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_n \end{pmatrix} \in \mathbb{F}_q^{\frac{nb}{k} \times \frac{db}{k}}, \text{ where}$$

$$M_j \triangleq (I \quad P^{i_j} \quad P^{2i_j} \quad \dots \quad P^{(d-1)i_j}) \in \mathbb{F}_q^{\frac{b}{k} \times \frac{db}{k}}, \quad (4)$$

and store $M_j \cdot X$ in storage node j . Notice that by the definition of the matrix P , we have that $\alpha = \frac{b^2}{k^2} \cdot d$.

Remark 1. *It is possible to replace P by the companion matrix of an irreducible polynomial which is not necessarily primitive, in which case, let $\{A_j(P)\}_{j=1}^n$ be distinct nonzero linear combinations of $\{P^i\}_{i=0}^{b/k-1}$, and define $M_j \triangleq (I \quad A_j(P) \quad A_j(P)^2 \quad \dots \quad A_j(P)^{d-1})$. However, we choose a primitive polynomial for convenience.*

Remark 2. *For $k = b$ this code is a PM-MBR code [20, Sec. IV], and in which case Condition A1 implies that $q \geq n$. Therefore, the advantage of our techniques exists only for $b > k$.*

Theorem 1. *In the above code, exact repair of any failed node may be achieved by downloading $\beta \triangleq \frac{b^2}{k^2}$ field elements from any d of the remaining nodes.*

Proof. Assume that node i failed, and $D = \{j_1, \dots, j_d\}$ is a subset of $[n]$ of size d such that $i \notin D$. To repair node i , every node $j_t \in D$ computes $M_{j_t} X M_i^\top$, which is a $\frac{b}{k} \times \frac{b}{k}$ matrix over \mathbb{F}_q , and sends

it to the newcomer. The newcomer obtains

$$\begin{pmatrix} M_{j_1} X M_i^\top \\ M_{j_2} X M_i^\top \\ \vdots \\ M_{j_d} X M_i^\top \end{pmatrix} = \begin{pmatrix} M_{j_1} \\ M_{j_2} \\ \vdots \\ M_{j_d} \end{pmatrix} \cdot X \cdot M_i^\top \triangleq M_D X M_i^\top.$$

According to (4), the matrix M_D is of the form

$$M_D = \begin{pmatrix} I & P^{i_{j_1}} & P^{2i_{j_1}} & \dots & P^{(d-1)i_{j_1}} \\ I & P^{i_{j_2}} & P^{2i_{j_2}} & \dots & P^{(d-1)i_{j_2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ I & P^{i_{j_d}} & P^{2i_{j_d}} & \dots & P^{(d-1)i_{j_d}} \end{pmatrix}.$$

Since M_D can be written as $\Theta(M'_D)$ for some invertible Vandermonde matrix M'_D in $\mathbb{F}_{q^{b/k}}^{d \times d}$, it follows by Lemma 3 that M_D is invertible. Thus, the newcomer may multiply from the left by M_D^{-1} and obtain $X M_i^\top$. Since X is a symmetric matrix, exact repair is obtained by transposing. \square

Theorem 2. *In the above code, reconstruction may be achieved by downloading $\alpha = \frac{b^2}{k^2} \cdot d$ field elements per node from any k nodes.*

Proof. Let $K = \{j_1, \dots, j_k\}$ be a subset of $[n]$ of size k , and download $M_{j_i} X$ from node j_i for each $j_i \in K$. The data collector thus obtains

$$\begin{aligned} M_K \cdot X &\triangleq \begin{pmatrix} I & P^{i_{j_1}} & P^{2i_{j_1}} & \dots & P^{(d-1)i_{j_1}} \\ I & P^{i_{j_2}} & P^{2i_{j_2}} & \dots & P^{(d-1)i_{j_2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ I & P^{i_{j_k}} & P^{2i_{j_k}} & \dots & P^{(d-1)i_{j_k}} \end{pmatrix} \cdot \begin{pmatrix} S & T \\ T^\top & 0 \end{pmatrix} \\ &\triangleq (M'_K S + M''_K T^\top \quad M'_K T), \end{aligned}$$

where

$$M'_K \triangleq \begin{pmatrix} I & P^{i_{j_1}} & P^{2i_{j_1}} & \dots & P^{(k-1)i_{j_1}} \\ I & P^{i_{j_2}} & P^{2i_{j_2}} & \dots & P^{(k-1)i_{j_2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ I & P^{i_{j_k}} & P^{2i_{j_k}} & \dots & P^{(k-1)i_{j_k}} \end{pmatrix}, \text{ and } M''_K \triangleq \begin{pmatrix} P^{ki_{j_1}} & P^{2i_{j_1}} & \dots & P^{(d-1)i_{j_1}} \\ P^{ki_{j_2}} & P^{2i_{j_2}} & \dots & P^{(d-1)i_{j_2}} \\ \vdots & \vdots & \vdots & \vdots \\ P^{ki_{j_k}} & P^{2i_{j_k}} & \dots & P^{(d-1)i_{j_k}} \end{pmatrix}.$$

As in the proof of Theorem 1, we have that M'_K is invertible. Hence, if $d > k$, the matrix T may be restored by extracting the $b^2 \left(\frac{d}{k} - 1\right)$ rightmost columns of $M_K X$ and multiplying by $(M'_K)^{-1}$. Having T , it can be used to reduce $M''_K T^\top$ from the remaining columns of $M_K X$, and then extracting S is similar. If $d = k$, then $X = S$, and multiplication by $M_K = M'_K$ suffices for reconstruction. \square

By Theorem 1 and Theorem 2 it is evident that $\alpha = d\beta$, and hence this construction attains minimum bandwidth repair. In Subsection 4.2 it will be shown that although the cut-set bound is not attained with equality, B approaches the cut-set bound (1) as k increases. Moreover, it will be evident that a small and often negligible loss of rate is obtained already for small values of k .

4.2 The proximity of NMBR codes to MBR codes

In this subsection it is shown that the codes constructed in Subsection 4.1 do not attain the cut-set bound (1), and hence cannot be considered MBR codes even though they attain $\alpha = d\beta$ (see Definition 1, and its preceding discussion). However, it is also shown that the cut-set bound is nearly achieved for large enough k , together with few specific examples which demonstrate a small loss of rate.

Let $C \triangleq \sum_{i=0}^{k-1} \min\{\alpha, (d-i)\beta\}$, and recall that by (1) we have that $B \leq C$ for all regenerating codes. Clearly, for codes which attain $\alpha = d\beta$ we have that

$$\begin{aligned} C &= \sum_{i=0}^{k-1} \min\{d\beta, (d-i)\beta\} \\ &= \beta \sum_{i=0}^{k-1} (d-i) = \beta \left(dk - \frac{k(k-1)}{2} \right). \end{aligned} \quad (5)$$

Hence, for the codes which are presented in Subsection 4.1 we have that $C = \frac{b^2}{k} \left(d - \frac{k-1}{2} \right)$. It is readily verified that indeed, $C > B$, thus (1) is not attained, and hence these are not MBR codes. However, we have that

$$\frac{B}{C} = \frac{2d - k \left(2 - \frac{b+1}{b} \right)}{2d - k + 1} = \frac{2 - \frac{k}{d} \cdot \frac{b-1}{b}}{2 - \frac{k}{d} + \frac{1}{d}}$$

and hence the cut-set bound is achieved in the asymptotic regime. That is, since a large k implies a large b (since $k|b$ in Condition A2) and a large d (since $d \geq k$), by following the outline of Subsection 4.1 and choosing a large enough k , one may obtain a code in which B is arbitrarily close to C , regardless of the relation between k and d .

In the remainder of this section, a detailed comparison of parameters between the PM-MBR codes and our NMBR codes is given. From these examples it will be evident that the decrease in file size (in comparison with the cut-set bound), and hence the decrease in the code rate, is a small price to pay for a considerable reduction in field size.

The curious reader might suggest that the extension field representation which is given in Lemma 1, can be applied directly to PM-MBR codes over an extension field, obtaining regenerating codes over the respective base field. This intuition is formalized in the following definition. For this definition, recall that PM-MBR codes may be obtained by choosing $k = b$ in the construction in Subsection 4.1.

Definition 6. *Given a PM-MBR code over an extension field \mathbb{F}_{q^m} with an encoding matrix M and data matrix X , let EPM-MBR be the code over \mathbb{F}_q which results from applying the function Θ from (2) on the encoding matrix M and multiplying it by a data matrix X' . The data matrix X' is given by applying θ on the upper triangular part of the data matrix X , and completing the lower triangular part to obtain symmetry.*

In order to apply the repair and reconstruction algorithms from Theorem 1 and Theorem 2 to EPM-MBR codes, the data matrix X' must be symmetric. Hence, it follows that the only data matrices $X \in \mathbb{F}_{q^m}^{d \times d}$ on which EPM-MBR codes maintain their repair and reconstruction capabilities are those in which all diagonal submatrices $\theta(X_{i,i})$ of X' are symmetric. Since companion matrices are in general not symmetric, this usually induces a further loss of rate. For simplicity, we shall ignore this detail in the comparison which follows, since NMBR codes will be shown to supersede EPM-MBR codes even without this additional rate loss. In the remainder of this section we compare between

EPM-MBR codes, NMBR codes, and PM-MBR codes with the concise vector space representation of extension field elements¹.

In PM-MBR codes the file size B is a function of k and d , and in addition, $\beta = 1$. Further, all parameters are measured in field elements rather than in bits. Therefore, to achieve a fair comparison, one must concatenate a PM-MBR code to itself in order to obtain the same parameters n, k, d, α , and β when measured in bits, and only then compare the resulting B, q , and the rate $\frac{B}{\alpha n}$. In addition, since fields of even characteristic are essential for hardware implementation, we restrict our attention to $q = 2$ in our codes, and to q which is an integer power of 2 for PM-MBR codes. Hence, the PM-MBR code is concatenated with itself $\frac{b^2}{\lceil \log n \rceil k^2}$ times, and considered with $q = 2^{\lceil \log n \rceil}$ (the smallest integer power of two that is at least n), where each element in this field is represented by a vector in $\mathbb{F}_2^{\lceil \log n \rceil}$. Similarly, the EPM-MBR code is concatenated with itself $\frac{b^2}{\lceil \log n \rceil^2 k^2}$ times, and considered with the same $q = 2^{\lceil \log n \rceil}$, where each element in this field is represented by a square matrix in $\mathbb{F}_2^{\lceil \log n \rceil \times \lceil \log n \rceil}$.

Notice that MBR codes have $B = \beta (dk - k(k - 1)/2)$ (see (5)), where B is measured in elements over \mathbb{F}_q . Therefore, by setting $\beta = 1, q = 2^{\lceil \log n \rceil}$, and concatenating a PM-MBR code $\frac{b^2}{\lceil \log n \rceil k^2}$ times with itself, we have that the number of information bits in the file is $C = \frac{b^2}{k} (d - \frac{k-1}{2})$. Similarly, by concatenating an EPM-MBR code $\frac{b^2}{\lceil \log n \rceil^2 k^2}$ times with itself, since each field element is represented by a $\lceil \log n \rceil \times \lceil \log n \rceil$ binary matrix that contains $\lceil \log n \rceil$ information bits, it follows that the number of information bits in the file is $\beta (dk - k(k - 1)/2) \cdot \frac{b^2}{\lceil \log n \rceil^2 k^2} \cdot \lceil \log n \rceil = \frac{C}{\lceil \log n \rceil}$. As a result, by fixing any n, k , and d such that $k \leq d \leq n - 1$, we have Table 1, in which the values of β, α , and B are given in bits.

	NMBR	PM-MBR concatenated $\frac{b^2}{\lceil \log n \rceil k^2}$ times	EPM-MBR concatenated $\frac{b^2}{\lceil \log n \rceil^2 k^2}$ times
q	2	$2^{\lceil \log n \rceil}$	2
β	$\frac{b^2}{k^2}$	$\frac{b^2}{\lceil \log n \rceil k^2}$ field elements in vector form, i.e., $\frac{b^2}{k^2}$ bits.	$\frac{b^2}{\lceil \log n \rceil^2 k^2}$ field elements in matrix form, i.e., $\frac{b^2}{k^2}$ bits.
α	$\frac{b^2}{k^2} \cdot d$	$\frac{b^2}{k^2} \cdot d$	$\frac{b^2}{k^2} \cdot d$
B	$\frac{b(b+1)}{2} + b^2 \cdot \left(\frac{d}{k} - 1\right)$	$\frac{b^2}{k} \left(d - \frac{k-1}{2}\right)$	$\frac{b^2}{k \lceil \log n \rceil} \left(d - \frac{k-1}{2}\right)$
Rate	$\frac{k^2}{dn} \cdot \left(\frac{d}{k} - \frac{1}{2} + \frac{1}{2b}\right)$	$\frac{k^2}{dn} \cdot \left(\frac{d}{k} - \frac{k-1}{2k}\right)$	$\frac{k^2}{dn \lceil \log n \rceil} \cdot \left(\frac{d}{k} - \frac{k-1}{2k}\right)$

Table 1: A comparison of parameters between our NMBR codes (Subsection 4.1) and the PM-MBR codes [20, Sec. IV] for general n, k, d .

Table 2 contains specific examples of the comparison given in Table 1. The parameter b is chosen such that $\frac{b^2}{\lceil \log n \rceil k^2}$ and $\frac{b^2}{\lceil \log n \rceil^2 k^2}$ are integers, and such that the resulting file size is within one of several common use cases. Notice that much smaller values of b may be chosen, for example, if one wishes to increase concurrency by code concatenation. For convenience, some values are given in either MegaBytes (MB), GigaBytes (GB), or TeraBytes (TB) rather than in bits.

From Table 2 it is evident that in comparison with PM-MBR codes, a considerable reduction in field size is obtained by our codes, even for rather small values of k . Furthermore, our techniques

¹I.e., each extension field element is represented by a vector over the base field.

	n	k	d	α	β	b	q	B	Rate
NMBR	30	20	20	250MB	12.5MB	$10000 \cdot k$	2	$\approx 2.5\text{GB}$	≈ 0.33
PM-MBR							32	2.625GB	0.35
EPM-MBR							2	0.525GB	0.07
NMBR	26	22	24	$\approx 841.7\text{MB}$	$\approx 35.07\text{MB}$	$16750 \cdot k$	2	$\approx 10.03\text{GB}$	≈ 0.4583
PM-MBR							32	$\approx 10.41\text{GB}$	≈ 0.4759
EPM-MBR							2	$\approx 2.8\text{GB}$	≈ 0.095
NMBR	260	220	240	$\approx 8.416\text{GB}$	$\approx 35.06\text{MB}$	$16749 \cdot k$	2	$\approx 1.002\text{TB}$	≈ 0.4583
PM-MBR							512	$\approx 1.006\text{TB}$	≈ 0.4601
EPM-MBR							2	$\approx 0.11\text{TB}$	≈ 0.05
NMBR	2600	2200	2400	$\approx 84.18\text{GB}$	$\approx 35.07\text{MB}$	$16752 \cdot k$	2	$\approx 100.32\text{TB}$	≈ 0.4583
PM-MBR							4096	$\approx 100.36\text{TB}$	≈ 0.4585
EPM-MBR							2	$\approx 8.36\text{TB}$	≈ 0.038

Table 2: A comparison of parameters between our NMBR codes (Subsection 4.1) and the PM-MBR codes [20, Sec. IV] for several common parameters n, k, d .

obtain a larger rate in comparison with EPM-MBR codes, which are implemented over the binary field as well.

In many practical applications [19, Slide 38], multiplication in a finite field \mathbb{F}_{2^w} is implemented by table look-ups for $w \leq 8$, and sometimes considered infeasible in large systems with $w > 8$, since it requires numerous table look-ups and expensive arithmetic. Hence, for $n > 2^8 = 256$, our techniques improve the feasibility of storage codes without compromising the code rate significantly.

5 Nearly MSR codes

In this section, for any given n, k, d, q such that $d \leq n - 1$ and $d = 2k - 2$, and for a sufficiently large file size B , regenerating codes in which B approaches αk as k increases are provided. Codes for $d > 2k - 2$ with similar properties are obtained in the sequel from this construction. For any such n, k, d and q , let b be an integer such that

$$\text{B1. } n \leq \frac{q^{b/k} - 1}{g^{b/k}}, \text{ where } g \triangleq \gcd(k - 1, q^{b/k} - 1),$$

$$\text{B2. } k \mid b,$$

and let

$$B \triangleq \frac{b(k-1)}{k} \cdot \left(\frac{b(k-1)}{k} + 1 \right) = \frac{b^2(k-1)}{k} \left(1 - \frac{1}{k} + \frac{1}{b} \right).$$

Condition B1 implies that $\frac{ng}{k} \leq \frac{q^{b/k} - 1}{b}$, and thus, since $g \leq k - 1$, it follows that any integer b such that $b \geq k(\log_q n + \log_q b)$ suffices. Further, Condition B1 implies that

$$B = \Omega \left(k^2 (\log_q(n) + \log_q(b))^2 \right),$$

and hence it is trivially satisfied in many distributed storage systems.

5.1 Construction

Similar to [20], given a file $x \in \mathbb{F}_q^B$, arrange its symbols in the upper triangle of two square matrices S_1, S_2 of dimensions $\frac{b(k-1)}{k} \times \frac{b(k-1)}{k}$ over \mathbb{F}_q , complete the lower triangle of S_1, S_2 to obtain symmetry, and define

$$X \triangleq \begin{pmatrix} S_1 \\ S_2 \end{pmatrix}.$$

Next, a set of integers i_1, \dots, i_n in the range $\{0, \dots, \frac{q^{b/k}-1}{g} - 1\}$ is chosen such that no two reside in the same q -cyclotomic coset modulo $\frac{q^{b/k}-1}{g}$. This choice is enabled by the following lemma.

Lemma 7. *The size of q -cyclotomic cosets modulo $\frac{q^{b/k}-1}{g}$ is at most b/k .*

Proof. According to [6, Th. 4.1.4, p. 123], for any m , the size of any q -cyclotomic coset modulo m is a divisor of $\text{ord}_m(q)$, where $\text{ord}_m(q)$ is the smallest integer t such that $q^t = 1 \pmod{m}$. Since clearly, $\frac{q^{b/k}-1}{g} | q^{b/k} - 1$, it follows that $q^{b/k} = 1 \pmod{\frac{q^{b/k}-1}{g}}$, which implies that $\text{ord}_{(q^{b/k}-1)/g}(q)$ is at most b/k , and the claim follows. \square

Lemma 7 implies that there are at least $\frac{q^{b/k}-1}{g \cdot (b/k)}$ different q -cyclotomic cosets modulo $\frac{q^{b/k}-1}{g}$, which enables the choice of i_1, \dots, i_n by Condition B1. Notice that the choice of i_1, \dots, i_n is possible using a simple algorithm, which maintains a list of feasible elements, iteratively picks an arbitrary element as the next i_j , and removes its coset from the list.

Let P be a companion matrix of any primitive polynomial of degree $\frac{b}{k}$ over \mathbb{F}_q , and let

$$\Phi \triangleq \begin{pmatrix} I & P^{i_1} & \dots & P^{i_{1(k-2)}} \\ I & P^{i_2} & \dots & P^{i_{2(k-2)}} \\ \vdots & \vdots & \ddots & \vdots \\ I & P^{i_n} & \dots & P^{i_{n(k-2)}} \end{pmatrix} \in \mathbb{F}_q^{\frac{bn}{k} \times \frac{b(k-1)}{k}} \quad \Lambda \triangleq \begin{pmatrix} P^{i_{1(k-1)}} & & & \\ & P^{i_{2(k-1)}} & & \\ & & \ddots & \\ & & & P^{i_{n(k-1)}} \end{pmatrix} \in \mathbb{F}_q^{\frac{bn}{k} \times \frac{bn}{k}}.$$

Define the $\frac{bn}{k} \times \frac{bd}{k}$ encoding matrix over \mathbb{F}_q as $M \triangleq (\Phi \ \Lambda\Phi)$ and notice that M is a block-Vandermonde matrix. Moreover, according to Lemma 1, Lemma 3, and the choice of i_1, \dots, i_n , it follows from the properties of Vandermonde matrices in $\mathbb{F}_q^{n \times d}$ that any $\frac{bd}{k} \times \frac{bd}{k}$ block submatrix² of M is invertible. Similarly, every $\frac{b(k-1)}{k} \times \frac{b(k-1)}{k}$ block submatrix of Φ is invertible. Let M_i be the i -th block-row of M , and store $M_i \cdot X$ in node i . By the definition of the corresponding matrices, we have that $\alpha = \frac{b^2(k-1)}{k^2}$.

Remark 3. *For $k = b$ this code is a special case of an PM-MSR code [20, Sec. V], and in which case condition B1 implies that $q \geq n \cdot \gcd(k-1, q-1) + 1$. Hence, the advantage of our techniques exists not only for $b > k$, unlike Remark 2. This improvement also follows from [15, Eq. (37)].*

Theorem 3. *In the above code, exact repair of any failed node may be achieved by downloading $\beta \triangleq \frac{b^2}{k^2}$ field elements from any d of the remaining nodes.*

Proof. Assume that node ℓ failed and $D = \{j_1, \dots, j_d\}$ is a subset of $[n]$ of size d such that $\ell \notin D$. Let Φ_ℓ be the ℓ -th block-row of Φ , and notice that node ℓ stored

$$M_\ell X = (\Phi_\ell \ P^{i_\ell(k-1)}\Phi_\ell) \cdot X = \Phi_\ell S_1 + P^{i_\ell(k-1)}\Phi_\ell S_2. \quad (6)$$

²That is, a submatrix which consists of complete blocks.

To repair node ℓ , every node $j_t \in D$ computes $M_{j_t} X \Phi_\ell^\top$, which is a $\frac{b}{k} \times \frac{b}{k}$ matrix over \mathbb{F}_q , and sends it to the newcomer. The newcomer obtains

$$\begin{pmatrix} M_{j_1} X \Phi_\ell^\top \\ M_{j_2} X \Phi_\ell^\top \\ \vdots \\ M_{j_d} X \Phi_\ell^\top \end{pmatrix} = \begin{pmatrix} M_{j_1} \\ M_{j_2} \\ \vdots \\ M_{j_d} \end{pmatrix} \cdot X \cdot \Phi_\ell^\top \triangleq M_D X \Phi_\ell^\top.$$

Since M_D can be seen as $\Theta(M'_D)$ for some full rank Vandermonde matrix $M'_D \in \mathbb{F}_{q^{b/k}}^{d \times d}$, it follows from Lemma 3 that M_D is invertible, and hence the newcomer may obtain

$$(X \Phi_\ell^\top)^\top = \Phi_\ell \cdot (S_1 \ S_2),$$

and restore $M_\ell X$ by (6). □

Theorem 4. *In the above code, reconstruction may be achieved by downloading $\alpha = \frac{b^2(k-1)}{k^2}$ field elements per node from any k nodes.*

Proof. Let $K = \{j_1, \dots, j_k\}$ be a subset of $[n]$ of size k , and download $M_{j_i} X$ from node j_i for each $j_i \in K$. The data collector obtains

$$M_K X = \Phi_K S_1 + \Lambda_K \Phi_K S_2,$$

where Λ_K and Φ_K are the row-submatrices of Λ and Φ which consist of the block-rows which are indexed by K . By multiplying from the right by Φ_K^\top , the data collector obtains

$$\Gamma \triangleq \Phi_K S_1 \Phi_K^\top + \Lambda_K \Phi_K S_2 \Phi_K^\top \triangleq W + \Lambda_K Q,$$

where W and Q are symmetric matrices. For $s \in [k]$ denote i_{j_s} by ℓ_s , and notice that for distinct s and t in $[k]$,

$$\llbracket \Gamma \rrbracket_{s,t} = \llbracket W \rrbracket_{s,t} + P^{\ell_s(k-1)} \llbracket Q \rrbracket_{s,t} \tag{7}$$

$$\llbracket \Gamma \rrbracket_{t,s} = \llbracket W \rrbracket_{t,s} + P^{\ell_t(k-1)} \llbracket Q \rrbracket_{t,s}$$

$$= \llbracket W \rrbracket_{s,t}^\top + P^{\ell_t(k-1)} \llbracket Q \rrbracket_{s,t}^\top$$

$$\llbracket \Gamma \rrbracket_{t,s}^\top = \llbracket W \rrbracket_{s,t} + \llbracket Q \rrbracket_{s,t} \cdot (P^{\ell_t(k-1)})^\top. \tag{8}$$

Thus, by subtracting (8) from (7) we have that

$$\begin{aligned} \llbracket \Gamma \rrbracket_{s,t} - \llbracket \Gamma \rrbracket_{t,s}^\top &= P^{\ell_s(k-1)} \llbracket Q \rrbracket_{s,t} - \llbracket Q \rrbracket_{s,t} (P^{\ell_t(k-1)})^\top \\ \left(\llbracket \Gamma \rrbracket_{s,t} - \llbracket \Gamma \rrbracket_{t,s}^\top \right) \cdot (P^{-\ell_t(k-1)})^\top &= P^{\ell_s(k-1)} \llbracket Q \rrbracket_{s,t} (P^{-\ell_t(k-1)})^\top - \llbracket Q \rrbracket_{s,t}. \end{aligned}$$

Now, it follows from Lemma 4 that vectorizing both sides of this equation results in

$$\left(P^{-\ell_t(k-1)} \otimes P^{\ell_s(k-1)} - I_{\frac{b^2}{k^2}} \right) \cdot \text{vec}(\llbracket Q \rrbracket_{s,t}) = \text{vec} \left(\left(\llbracket \Gamma \rrbracket_{s,t} - \llbracket \Gamma \rrbracket_{t,s}^\top \right) \cdot (P^{-\ell_t(k-1)})^\top \right), \tag{9}$$

which may be seen as a linear system of equations whose variables are the unknown entries of $\llbracket Q \rrbracket_{s,t}$. According to Lemma 6, this equation has a unique solution if and only if 1 is not an eigenvalue of $P^{-\ell_t(k-1)} \otimes P^{\ell_s(k-1)}$.

Since the characteristic polynomial of any companion matrix is its corresponding polynomial, and since for P this polynomial is primitive, the eigenvalues of P are $\gamma, \gamma^q, \dots, \gamma^{q^{b/k-1}}$, where γ is some primitive element in $\mathbb{F}_{q^{b/k}}$ [6, Th. 4.1.1, p. 123]. Therefore, the eigenvalues of $P^{\ell_s(k-1)}$ are $\gamma^{\ell_s(k-1)}, \gamma^{\ell_s(k-1)q}, \dots, \gamma^{\ell_s(k-1)q^{b/k-1}}$, the eigenvalues of $P^{-\ell_t(k-1)}$ are $\gamma^{-\ell_t(k-1)}, \gamma^{-\ell_t(k-1)q}, \dots, \gamma^{-\ell_t(k-1)q^{b/k-1}}$, and by Lemma 5, the eigenvalues of $P^{-\ell_t(k-1)} \otimes P^{\ell_s(k-1)}$ are

$$\Delta \triangleq \left\{ \gamma^{\ell_s(k-1)q^e - \ell_t(k-1)q^h} \mid e, h \in \{0, 1, \dots, b/k - 1\} \right\}.$$

If $1 \in \Delta$, it follows that there exist e and h in $\{0, 1, \dots, b/k - 1\}$ such that

$$\gamma^{\ell_s(k-1)q^e - \ell_t(k-1)q^h} = 1,$$

which implies that $\ell_s(k-1)q^e = \ell_t(k-1)q^h \pmod{q^{b/k} - 1}$. Therefore, there exists an integer t such that

$$\begin{aligned} \ell_s(k-1)q^e &= \ell_t(k-1)q^h + t(q^{b/k} - 1) \\ \ell_s q^e \cdot \frac{k-1}{g} &= \ell_t q^h \cdot \frac{k-1}{g} + t \cdot \frac{q^{b/k} - 1}{g}, \end{aligned}$$

and thus,

$$\ell_s q^e \cdot \frac{k-1}{g} = \ell_t q^h \cdot \frac{k-1}{g} \pmod{\frac{q^{b/k} - 1}{g}}. \quad (10)$$

Since clearly, $\gcd(\frac{k-1}{g}, \frac{q^{b/k}-1}{g}) = 1$, it follows that $\frac{k-1}{g}$ is invertible modulo $\frac{q^{b/k}-1}{g}$. Therefore, (10) implies that $\ell_s q^e = \ell_t q^h \pmod{\frac{q^{b/k}-1}{g}}$. Since $\gcd(q, \frac{q^{b/k}-1}{g}) = 1$, it follows that q is invertible modulo $\frac{q^{b/k}-1}{g}$. Hence, we have that $\ell_s = \ell_t q^{h-e} \pmod{\frac{q^{b/k}-1}{g}}$ if $h \geq e$ and $\ell_s q^{e-h} = \ell_t \pmod{\frac{q^{b/k}-1}{g}}$ if $h < e$. Either way, it follows that ℓ_t and ℓ_s , which are notations for i_{j_t} and i_{j_s} , respectively, are in the same q -cyclotomic coset modulo $\frac{q^{b/k}-1}{g}$, a contradiction to the choice of i_1, \dots, i_n . Therefore, $1 \notin \Delta$, which implies that (9) is solvable, and the data collector may obtain $[[Q]]_{s,t}$ and $[[W]]_{s,t}$ for all distinct s and t in $[k]$.

Having this information, the data collector may consider the i -th block-row of Q , excluding the diagonal element,

$$\Phi_i S_2 \left(\Phi_1^\top \quad \dots \quad \Phi_{i-1}^\top \quad \Phi_{i+1}^\top \quad \dots \quad \Phi_k^\top \right),$$

in which the matrix on the right is invertible by construction, and by Lemma 3. Hence, the data collector obtains $\Phi_1 S_2, \dots, \Phi_k S_2$, out of which any $k-1$ may once again be used to extract S_2 by the same argument. Clearly, S_1 may be obtained similarly from the submatrices $[[W]]_{s,t}$. \square

Note that in the above code $B = \frac{b^2(k-1)}{k} \left(1 - \frac{1}{k} + \frac{1}{b}\right)$, and $\alpha k = \frac{b^2(k-1)}{k}$. Thus, the construction in this section *does not* provide an MSR code. However, $\frac{B}{\alpha k} \xrightarrow{k \rightarrow \infty} 1$, and thus the cut-set bound is achieved asymptotically. A detailed comparison with PM-MSR codes and numerical examples appear in the Subsection 5.2.

This construction can be used to obtain NMSR codes for $d > 2k - 2$ in a recursive manner. By following a very similar outline to that of [20, Th. 6], we have the following.

Theorem 5. *If there exists an $(n', k', d', B', q, \alpha, \beta)$ regenerating code \mathcal{C}' such that $\frac{B'}{\alpha k'} \xrightarrow{k' \rightarrow \infty} 1$, then there exists a $(n = n' - 1, k = k' - 1, d = d' - 1, B' = B - \alpha, q, \alpha, \beta)$ regenerating code \mathcal{C} such that $\frac{B}{\alpha k} \xrightarrow{k \rightarrow \infty} 1$.*

Proof. Without loss of generality assume that \mathcal{C}' is systematic, and let \mathcal{C} be the code which results from *puncturing* the first systematic node of \mathcal{C}' . It follows from the properties of \mathcal{C}' that \mathcal{C} is a code with $n = n' - 1$ nodes, in which any $d = d' - 1$ nodes can be used for repair, and any $k = k' - 1$ nodes may be used for reconstruction. Moreover, $B = B' - \alpha$, and

$$\frac{B}{\alpha k} = \frac{B' - \alpha}{\alpha(k' - 1)} = \frac{B'}{\alpha k'} \cdot \frac{k'}{k' - 1} - \frac{1}{k' - 1} \xrightarrow{k \rightarrow \infty} 1$$

□

Notice that in Theorem 5, if $d' = ik' + j$ then $d = ik + j + (i - 1)$. Hence, given the construction for $d = 2k - 2$, one may obtain NMSR codes for larger values of d . Moreover, it is evident that $\frac{B'}{\alpha k'} \leq 1$, $\frac{k'}{k' - 1} > 1$, and that $\frac{1}{k' - 1}$ is negligible as k grows. Hence, the proof of Theorem 5 implies that $\frac{B}{\alpha k}$ tends to 1 *faster* than $\frac{B'}{\alpha k'}$ does.

5.2 The proximity of NMSR codes to MSR codes

In this subsection the construction from Subsection 5.1 is compared with PM-MSR codes for the case $d = 2k - 2$. A comparison for the case $d > 2k - 2$ will appear in future versions of this paper. Following the reasoning which is described in Subsection 4.1, the codes are compared over fields of even characteristic. That is, our codes are considered with $q = 2$, and since PM-MSR codes require $q \geq n(k - 1)$, they are considered with $q = 2^{\lceil \log(n(k-1)) \rceil}$.

Similar to Definition 6 and its subsequent discussion, EPM-MSR codes may also be defined. Note that a comparable loss of rate is apparent, not only due to the redundant representation, but also due to the symmetry which is required from the submatrices on the main diagonals of S_1 and S_2 .

The codes PM-MSR and EPM-MSR are concatenated to themselves in order to obtain the same n, k, d, α , and β , and only then the resulting file size and code rate are compared. The comparison for general parameters appears in Table 3, in which the values of α, β , and B are given in bits. Note that as in Subsection 4.2, the value of B for EPM-MSR is the number of *information* bits, rather than the number of bits in the redundant representation. Further, numerical examples are given in Table 4. Notice that it is possible to reduce the field size of PM-MSR codes in some cases [15]. Yet, we compare our NMSR codes to PM-MSR for simplicity and generality.

6 Discussion and future research

In this paper, asymptotically optimal regenerating codes were introduced. These codes attain the cut-set bound asymptotically as the reconstruction degree k increases, and may be defined over any field if the file size is reasonably large. Further, these codes enjoy several properties which are inherited from product matrix codes, such as the fact that helper nodes do not need to know the identity of each other³, and the ability to add an extra storage node without encoding the file anew.

³Although this property is apparent in most regenerating codes constructions, some constructions do require otherwise, such as [3], and some of the work of [21].

	NMSR	PM-MSR concatenated $\frac{b^2}{\lceil \log(n(k-1)) \rceil k^2}$ times	EPM-MSR concatenated $\frac{b^2}{\lceil \log(n(k-1)) \rceil^2 k^2}$ times
q	2	$2^{\lceil \log(n(k-1)) \rceil}$	2
β	$\frac{b^2}{k^2}$	$\frac{b^2}{\lceil \log(n(k-1)) \rceil k^2}$ field elements in vector form, i.e., $\frac{b^2}{k^2}$ bits.	$\frac{b^2}{\lceil \log(n(k-1)) \rceil^2 k^2}$ field elements in matrix form, i.e., $\frac{b^2}{k^2}$ bits.
α	$\frac{b^2}{k^2} \cdot (k-1)$	$\frac{b^2}{k^2} \cdot (k-1)$	$\frac{b^2}{k^2} \cdot (k-1)$
B	$\frac{b^2(k-1)}{k} \left(1 - \frac{1}{k} + \frac{1}{b}\right)$	$\frac{b^2(k-1)}{k}$	$\frac{b^2(k-1)}{k \lceil \log(n(k-1)) \rceil}$
Rate	$\frac{k}{n} \left(1 - \frac{1}{k} + \frac{1}{b}\right)$	$\frac{k}{n}$	$\frac{k}{n \lceil \log(n(k-1)) \rceil}$

Table 3: A comparison of parameters between our MSR codes (Subsection 5.1) and the PM-MSR codes [20, Sec. V] for general $n, k, d = 2k - 2$.

	n	k	d	α	β	b	q	B	Rate
NMSR	20	10	18	45KB	5KB	$200 \cdot k$	2	405.225KB	0.45025
PM-MSR							256	450KB	0.5
EPM-MSR							2	56.25KB	0.0625
NMSR	100	40	78	7.02MB	0.18MB	$1200 \cdot k$	2	≈ 0.27 GB	≈ 0.39
PM-MSR							4096	0.28GB	0.4
EPM-MSR							2	0.02GB	≈ 0.033
NMSR	100	40	78	175.5MB	4.5MB	$6000 \cdot k$	2	≈ 6.84 GB	≈ 0.39
PM-MSR							4096	7.02GB	0.4
EPM-MSR							2	0.585GB	≈ 0.033
NMSR	1000	400	798	≈ 1800 KB	≈ 4.5 KB	$190 \cdot k$	2	≈ 0.718 GB	≈ 0.39
PM-MSR							524288	0.72GB	0.4
EPM-MSR							2	37.9MB	≈ 0.02

Table 4: A comparison of parameters between our code (Subsection 5.1) and the PM-MSR code [20, Sec. V] for several common parameters $n, k, d = 2k - 2$.

It is evident from Table 2 and Table 4 that for $q = 2$, a small loss of code rate is apparent already for feasible values of k , and clearly, similar results hold for larger q as well. Since large finite field arithmetics is often infeasible, our results contribute to the feasibility of storage codes.

The research of storage codes has gained a considerable amount of attention lately. In particular, the results of [20], which inspired ours, was expanded and improved in few recent papers. For example, [3] generalized the PM-MBR construction to achieve other points of the trade-off through minor matrices, and [16] presented an MBR code which supports an arbitrary number of helper nodes in the repair process. Among the research directions we currently pursue are the application of the techniques from the current paper to the aforementioned works, as well as to high rate MSR constructions, and analyzing the encoding, decoding, repair, and reconstruction complexities of our codes in comparison with PM codes.

Acknowledgments

The work of Netanel Raviv was supported in part by the Israeli Science Foundation (ISF), Jerusalem, Israel, under Grant no. 10/12, The IBM Ph.D. fellowship, and the Mitacs organization, under the Globalink Israel-Canada Innovation Initiative. The author would like to express his sincere gratitude to Prof. Frank Kschischang, Prof. Tuvi Etzion, and Prof. Itzhak Tamo for many insightful discussions.

References

- [1] R. Bhatia and P. Rosenthal, “How and why to solve the operator equation $AX - XB = Y$,” *Bulletin of the London Mathematical Society*, vol. 29, no. 1, pp. 1–21, 1997.
- [2] S. El-Rouayheb and K. Ramchandran, “Fractional repetition codes for repair in distributed storage systems,” *48th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1510–1517, 2010.
- [3] M. Elyasi and S. Mohajer, “New exact-repair codes for distributed storage systems using matrix determinant,” *IEEE International Symposium on Information Theory (ISIT)*, pp. 1212–1216, 2016.
- [4] S. Goparaju, A. Fazeli, and A. Vardy, “Minimum storage regenerating codes for all parameters,” *IEEE International Symposium on Information Theory (ISIT)*, pp. 76–80, 2016.
- [5] H. Hou, K. W. Shum, M. Chen and H. Li, “BASIC codes: low-complexity regenerating codes for distributed storage systems,” *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3053–3069, 2016.
- [6] W. C. Huffman and V. Pless, *Fundamentals of error-correcting codes*, Cambridge University Press, 2010.
- [7] S. Jiekak, A. M. Kermarrec, N. Le Scouarnec, G. Straub, and A. Van Kempen, “Regenerating codes: a system perspective,” *ACM SIGOPS Operation Systems Review*, vol. 47, no. 2, pp. 23–32, 2013.
- [8] G. M. Kamath, N. Prakash, V. Lalitha and P. V. Kumar, “Codes with local regeneration,” *Information Theory and Applications Workshop (ITA)*, pp. 1–5, 2013.
- [9] G. M. Kamath, N. Silberstein, N. Prakash, A. S. Rawat, “Explicit MBR all-symbol locality codes,” *IEEE International Symposium on Information Theory (ISIT)*, pp. 504–508, 2013.
- [10] D. E. Knuth, *The art of computer programming, volume 1: fundamental algorithms*, Redwood City, 1997.
- [11] M. N. Krishnan and P. V. Kumar, “On MBR codes with replication,” *arXiv:1601.08190 [cs.IT]*, 2016.
- [12] A. J. Laub, *Matrix analysis for scientists and engineers*, Siam, 2005.
- [13] R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*, Cambridge University Press, 1994.

- [14] S. J. Lin and W. H. Chung, “Novel repair-by-transfer codes and systematic exact-MBR codes with lower complexities and smaller field sizes,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3232–3241, 2014.
- [15] S. J. Lin, W. H. Chung, Y. S. Han, and T. Y. Al-Naffouri, “A unified form of exact-MSR codes via product-matrix frameworks,” *IEEE Transactions on Information Theory*, vol. 61, no. 2, pp. 873–886, 2015.
- [16] K. Mahdavian, A. Khisti, and S. Mohajer, “Bandwidth adaptive and error resilient regenerating codes with minimum repair bandwidth,” *IEEE International Symposium on Information Theory (ISIT)*, pp. 235–239, 2016.
- [17] H. Neudecker, “A note on Kronecker matrix products and matrix equation systems,” *SIAM Journal on Applied Mathematics*, vol. 17, no. 3, pp. 603–606, 1969.
- [18] P. Nakkiran, K. V. Rashmi, and K. Ramchandran, “Optimal systematic distributed storage codes with fast encoding,” arXiv:1509.01858v1 [cs.IT], 2015.
- [19] J. S. Plank and C. Huang, “Tutorial: Erasure coding for storage applications,” *11th Usenix Conference on File and Storage Technologies (FAST)*, <http://web.eecs.utk.edu/~plank/plank/papers/FAST-2013-Tutorial.html>, 2013.
- [20] K. V. Rashmi, N. B. Shah, and P. V. Kumar, “Optimal exact-regenerating codes for distributed storage at the MSR and MBR point via a product-matrix construction,” *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5227–5239, 2011.
- [21] N. Raviv and T. Etzion, “Distributed storage systems based on intersecting subspace codes,” *IEEE International Symposium on Information Theory (ISIT)*, pp. 1462–1466, 2015.
- [22] N. Raviv, N. Silberstein, and T. Etzion, “Constructions of high-rate minimum storage regenerating codes over small fields,” *IEEE International Symposium on Information Theory (ISIT)*, pp. 61–65, 2016.
- [23] R. Roth, *Introduction to coding theory*, Cambridge University Press, 2006.
- [24] B. Sasidharan, M. Vajha, P. V. Kumar, “An explicit, coupled-layer construction of a high-rate MSR Code with low sub-packetization level, small field size and all-node repair,” arXiv:1607.07335 [cs.IT], 2016.
- [25] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, “Distributed storage codes with repair-by-transfer and nonachievability of interior points on the storage-bandwidth tradeoff,” *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1837–1852, 2012.
- [26] K. V. Rashmi, N. B. Shah, K. Ramchandran and P. V. Kumar, “Regenerating codes for errors and erasures in distributed storage,” *IEEE International Symposium on Information Theory (ISIT)*, pp. 1202–1206, 2012.
- [27] Y. Wu, A. G. Dimakis, and K. Ramchandran, “Deterministic regenerating codes for distributed storage,” *Allerton conference on control, computing, and communication*, pp. 1–5, 2007.
- [28] W. P. Wardlaw, “Matrix representation of finite fields,” *Mathematics Magazine*, vol. 67, no. 4, pp. 289–293, 1994.

- [29] M. Ye and A. Barg, “Explicit constructions of high-rate MDS array codes with optimal repair bandwidth,” *IEEE International Symposium on Information Theory (ISIT)*, pp. 1202–1206, 2016.
- [30] M. Ye and A. Barg, “Explicit constructions of optimal-access MDS codes with nearly optimal sub-packetization,” *arXiv:1605.08630 [cs.IT]*, 2016.