

Universally Decodable Matrices for Distributed Matrix-Vector Multiplication

Aditya Ramamoorthy, Li Tang

Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50010, U.S.A.

Pascal O. Vontobel

Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong, S. A. R.

Abstract—Coded computation is an emerging research area that leverages concepts from erasure coding to mitigate the effect of stragglers (slow nodes) in distributed computation clusters, especially for matrix computation problems. In this work, we present a class of distributed matrix-vector multiplication schemes that are based on codes in the Rosenbloom-Tsfasman metric and universally decodable matrices. Our schemes take into account the inherent computation order within a worker node. In particular, they allow us to effectively leverage partial computations performed by stragglers (a feature that many prior works lack). An additional main contribution of our work is a companion-matrix-based embedding of these codes that allows us to obtain sparse and numerically stable schemes for the problem at hand. Experimental results confirm the effectiveness of our techniques.

I. INTRODUCTION

Distributed computation clusters are routinely used in domains such as machine learning and scientific computing. In these applications, datasets are often so large that they cannot be housed in the disk of a single server. Furthermore, processing the data on a single server is either infeasible or unacceptably slow. Thus, the data and the processing is distributed and processed across a large number of nodes.

While large clusters have numerous advantages, they also present newer operational challenges. These clusters (which can be heterogeneous in nature) suffer from the problem of “stragglers”, which are defined as slow nodes (node failures are an extreme form of a straggler). It is evident that the overall speed of a computation on these clusters is typically dominated by stragglers in the absence of a sophisticated assignment of tasks to the worker nodes.

In recent years, approaches based on coding theory (referred to as “coded computation”) have been effectively used for straggler mitigation [1]–[9]. Coded computation offers significant benefits for specific classes of problems, e.g., matrix computations. We illustrate this by means of a matrix-vector multiplication example in Fig. 1, where a matrix \mathbf{A} is block-row decomposed as $\mathbf{A}^T = [\mathbf{A}_0^T \ \mathbf{A}_1^T \ \mathbf{A}_2^T]^T$. Each worker node is given the responsibility of computing two submatrix-vector products (denoted by $\hat{\mathbf{A}}_{ij}$ ’s in Fig. 1) so that the computational load on each worker is 2/3-rd of the original. It can be observed that even if one worker fails, there is enough information for a master node to compute the final result.

This work was supported in part by the National Science Foundation (NSF) under grant CCF-1718470.

However, this requires the master node to solve simple systems of equations. This approach can be generalized (and also adapted for matrix multiplication) by using Reed-Solomon (RS) code like approaches [1]–[5]. These methods allow the master node to recover $\mathbf{A}\mathbf{x}$ if any τ of the worker nodes complete their computation; τ is called the recovery threshold.

A significant amount of prior work treats stragglers as node failures (see [6], [8], [9] for exceptions), or, equivalently from the point of view of coding theory, as erasures. This matches the conventional erasure coding problem very well and allows the adaptation of well-known approaches, e.g., RS codes to the problem of distributed matrix computations. However, there are certain features of the distributed matrix-vector multiplication problem that distinguish it from classical erasure correction that we now discuss.

- *Leveraging partial computation performed by stragglers.* Each worker node operates in a sequential fashion on its assigned rows. E.g., in Fig. 1, worker W_0 , first computes $\hat{\mathbf{A}}_{00}\mathbf{x}$ and only then $\hat{\mathbf{A}}_{01}\mathbf{x}$. If node 0 is a straggler (but not a failure), ignoring the partial computation it performs will be wasteful.
- *Numerically stable decoding.* The RS-based approach requires the master node to solve a real Vandermonde system of linear equations or equivalently perform polynomial interpolation. It is well recognized that real Vandermonde matrices have a rather large condition number¹ which translates into significant numerical issues in recovering $\mathbf{A}\mathbf{x}$. This numerical issue is especially important in Krylov subspace methods for solving large linear systems of equations (which repeatedly compute matrix-vector products) and in machine learning, where gradient computations are often approximate.
- *Dealing with sparse \mathbf{A} matrices.* The case when the matrix \mathbf{A} is sparse is often an important one in practice. RS-based approaches typically generate submatrices that are sent to the worker nodes by combining a large number of rows of \mathbf{A} , thus destroying the inherent sparsity of the problem. This can significantly increase the computation time [7] at the worker nodes. Thus, techniques that only require sparse combinations of the rows of \mathbf{A} are of great interest.

Main contributions of our work: We present a class of distributed matrix-vector multiplication schemes that provably

¹While there is literature on choosing good evaluation points to reduce the condition number, in the distributed matrix vector multiplication context, we require decoding from any τ evaluation points. This makes the worst case condition number quite bad.

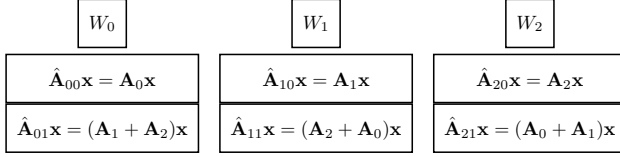


Fig. 1: Matrix \mathbf{A} is partitioned into three submatrices $\mathbf{A}_0, \mathbf{A}_1$ and \mathbf{A}_2 by rows. The assignment of submatrices $\hat{\mathbf{A}}_{ij}$, $0 \leq i \leq 2, 0 \leq j \leq 1$ to the worker nodes is shown in the figure. It can be verified that the master node can decode if any three submatrices are processed, while respecting the sequential computation order (from top to bottom).

leverage partial computations by stragglers, while possessing a numerically stable decoding algorithm. These schemes are related to codes in the Rosenbloom-Tsfasman metric [10] and universally decodable matrices (UDM) [11] that were presented in different contexts. Roughly speaking, while the RS-based approach corresponds to polynomial evaluation/interpolation, our approach can be viewed as working with polynomials with roots of higher multiplicity. An additional main contribution of our work is the usage of companion matrices [12] that allow for an embedding of finite-field matrices into the real field; this significantly improves the condition numbers of the relevant matrices. Recently, UDMs have also been used in a distributed storage scenario where nodes may have differing download speeds [13] and in a related coded computation setup [14].

II. PROBLEM FORMULATION

We consider a scenario where the master node has a matrix \mathbf{A} and vector \mathbf{x} (both real-valued) and is connected to N worker nodes. For convenience, for an arbitrary positive integer n , let $[n] \triangleq \{0, \dots, n-1\}$. The master node first partitions \mathbf{A} into Δ block-rows (or submatrices) denoted by $\mathbf{A}_0, \dots, \mathbf{A}_{\Delta-1}$, each of the same dimension. Following this, it generates submatrices denoted by $\hat{\mathbf{A}}_{i,j}$, $i \in [N], j \in [\ell]$ (of the same dimension as the \mathbf{A}_i 's) such that worker W_i is sent submatrices $\hat{\mathbf{A}}_{i,j}$, $j \in [\ell]$, and the vector \mathbf{x} . Let $\gamma \triangleq \ell/\Delta$. Then each worker is assigned the equivalent of a γ -fraction of the rows of \mathbf{A} . In this paper, we assume \mathbf{A} is large enough so that Δ can be chosen large enough. Throughout this paper the submatrices $\hat{\mathbf{A}}_{i,j}$ will be linear combinations of $\mathbf{A}_0, \dots, \mathbf{A}_{\Delta-1}$, such that the master node only calculates linear combinations of \mathbf{A}_i 's.

In what follows, we say that worker W_i has processed a submatrix $\hat{\mathbf{A}}_{i,j}$ if it has calculated $\hat{\mathbf{A}}_{i,j}\mathbf{x}$. A key feature of the distributed matrix-vector multiplication problem is that the matrices $\hat{\mathbf{A}}_{i,j}$ are processed sequentially in the order $\hat{\mathbf{A}}_{i,0}, \hat{\mathbf{A}}_{i,1}, \dots$, i.e., a worker node W_i processes $\hat{\mathbf{A}}_{i,j}$ only if it has finished processing $\hat{\mathbf{A}}_{i,0}, \dots, \hat{\mathbf{A}}_{i,j-1}$. Each time a worker node computes a product or a block of consecutive products, it sends the result to the master node. Our system requirement dictates that the master node should be able to decode $\mathbf{A}\mathbf{x}$ as long as it receives a minimum number of products from the worker nodes. Fig. 1 demonstrates a system we consider.

It is evident that the properties of a given scheme depend upon the properties of the matrices $\hat{\mathbf{A}}_{i,j}$, $i \in [N], j \in [\ell]$. To specify this encoding we discuss constructions of collections of matrices that have certain desired properties. Some of our constructions are first designed over a finite field and then embedded into \mathbb{R} using an appropriately defined procedure. Accordingly, we define certain rank conditions that depend on an underlying field of operation denoted by \mathbb{K} . We will explicitly specify \mathbb{K} when discussing the constructions. Consider N matrices G_k , $0 \leq k < N$, over \mathbb{K} with dimension $\Delta \times \ell$ and let $G_k(i, j)$ represent the (i, j) -th entry of G_k . Let $\mathbf{v} = (v_0, \dots, v_{N-1})$. We define the set

$$\Psi_{N,\ell,s}^{=Q_b} = \left\{ \mathbf{v} \mid v_i \in [\ell], i \in [N], \sum_{i \in [N]} \left\lfloor \frac{v_i}{s} \right\rfloor = Q_b \right\}.$$

Definition 1. *s-weak full-rank matrices.* Let N, ℓ, Δ, s be positive integers such that s divides ℓ and Δ . Let $Q_b = \Delta/s$. Consider matrices $G_k, k \in [N]$, of dimension $\Delta \times \ell$. Let $\mathbf{v} \in \Psi_{N,\ell,s}^{=Q_b}$ and let $\Delta_{\mathbf{v}} = \sum_{i \in [N]} v_i \geq Q_b s$. If the $\Delta \times \Delta_{\mathbf{v}}$ matrix \mathbf{G} composed of the first v_0 columns of G_0 , the first v_1 columns of G_1, \dots , and the first v_{N-1} columns of G_{N-1} , has full rank over \mathbb{K} , i.e., $\text{rank}_{\mathbb{K}}(\mathbf{G}) = \Delta$ for all $\mathbf{v} \in \Psi_{N,\ell,s}^{=Q_b}$ we say the collection $\{G_k\}_{k=0}^{N-1}$ satisfies the *s-weak full-rank condition*.

Example 1. Consider the system in Fig. 1 with $N = 3, \Delta = 3, \ell = 2$. Matrix \mathbf{A} is partitioned into three submatrices by rows, $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2$. Each worker node is assigned two submatrices and the vector \mathbf{x} . The following real-valued matrices satisfy the conditions in Definition 1 for $s = 1$ (see Fig. 1 for the corresponding $\hat{\mathbf{A}}_{i,j}$ matrices).

$$G_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, G_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \text{ and } G_2 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The collection $\{G_k\}_{k=0}^{N-1}$ is used to obtain the ℓ submatrices used by worker W_k when $\mathbb{K} = \mathbb{R}$ as follows.

$$\hat{\mathbf{A}}_{k,j} = \sum_{i \in [\Delta]} G_k(i, j) \mathbf{A}_i, \text{ for } k \in [N], j \in [\ell].$$

Consider first the case $s = 1$ and assume that worker W_i has finished processing v_i submatrices and $v_0 + \dots + v_{N-1} \geq \Delta$. Let \mathbf{G} be as specified in Definition 1. The system requirement of decoding from any $Q_b = \Delta$ submatrix-vector products is equivalent to the condition that \mathbf{G} is full-rank over \mathbb{R} for all possible patterns (v_0, \dots, v_{N-1}) . For instance, picking the first column of each of the G_k matrices in Example 1 gives a full rank 3×3 matrix as does picking the first column of G_0 and all the columns of G_2 . Thus, designing $\{G_k\}_{k=0}^{N-1}$ that satisfy Definition 1 is sufficient for the problem at hand.

Values of $s > 1$ correspond to a relaxation of this condition. Specifically, suppose that each worker node returns the results in blocks of size s . For instance, worker node W_i computes $\hat{\mathbf{A}}_{i,0}\mathbf{x}, \hat{\mathbf{A}}_{i,1}\mathbf{x}, \dots, \hat{\mathbf{A}}_{i,s-1}\mathbf{x}$ and then reports the result back to the master. Following this, it focuses on the next block $\hat{\mathbf{A}}_{i,s}\mathbf{x}, \dots, \hat{\mathbf{A}}_{i,2s-1}\mathbf{x}$, and so on. In this case, decoding by the

master node is guaranteed if it receives any Q_b blocks of size s (this explains our choice of subscript b in Q_b). If the s -weak full-rank condition holds for $s = 1$, we will refer to the system as satisfying the *strong full-rank condition*.

Remark 1. When $s = 1$, then the master node can recover \mathbf{Ax} when any Q_b submatrices have been processed across the N workers, i.e., the worst case computational load on the system, measured at the granularity of a submatrix is Δ . If $s > 1$, then the worst case computational load can be as high as

$$\Delta \cdot \left(1 + \frac{(N-1)(s-1)}{\Delta}\right). \quad (1)$$

For our constructions, the second term $\frac{(N-1)(s-1)}{\Delta}$ can be made as small as desired by choosing a large enough Δ , i.e., the multiplicative gap can be made arbitrarily close to 1.

III. CODED SCHEMES SATISFYING THE STRONG FULL RANK CONDITION

In this section, we present two schemes that satisfy the strong full-rank condition. The first scheme is essentially an embedding of an RS code in the matrix-vector multiplication framework and has appeared in [4]. The second one is inspired by the constructions in [10], [11].

Let $u(x) = \sum_{k=0}^d u_k x^k$ be a polynomial of degree d with real coefficients, i.e., $u(x) \in \mathbb{R}[x]$ where $\mathbb{R}[x]$ denotes the ring of polynomials with real coefficients. Let $u^{(j)}(x)$ denote the j -th derivative of $u(x)$. It can be verified that

$$u^{(j)}(x) = \sum_{k=0}^d u_k \binom{k}{j} j! x^{k-j}, \quad (2)$$

where $\binom{k}{i} = 0$ if $k < i$. Furthermore, note that we can also represent $u(x)$ by considering its Taylor series expansion around a point $\beta \in \mathbb{R}$, i.e.,

$$u(x) = \sum_{k=0}^d \frac{u^{(k)}(\beta)}{k!} (x - \beta)^k. \quad (3)$$

It is well known that $u(x)$ has a zero of multiplicity m at $\beta \in \mathbb{R}$ if and only if $u^{(i)}(\beta) = 0$ for $0 \leq i < m$ and $u^{(m)}(\beta) \neq 0$.

A. RS-based scheme

In the first scheme we simply choose the columns of G_k for $k \in [N]$ to correspond to a polynomial of degree $\Delta - 1$ being evaluated at distinct points in \mathbb{R} , i.e.,

$$G_k(i, j) = \beta_{k,j}^i, \text{ for } i \in [\Delta], j \in [\ell]. \quad (4)$$

where $\beta_{k,j} \in \mathbb{R}$ are distinct for $k \in [N], j \in [\ell]$.

B. UDM-based scheme

Our second construction works by choosing the columns of G_k corresponding to the evaluations of a polynomial and its derivatives of order $1, \dots, \ell - 1$. We first choose N distinct real numbers $\beta_0, \dots, \beta_{N-1}$. For worker node k , we choose the j -th column in correspondence with the evaluation of the j -th

derivative of a degree- $(\Delta - 1)$ polynomial at value β_k scaled by $j!$ (cf. Eq. (2)). Thus, for $k \in [N], i \in [\Delta]$ and $j \in [\ell]$,

$$G_k(i, j) = \begin{cases} \binom{i}{j} \beta_k^{i-j} & \text{if } i \geq j, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

C. Properties of Coded Schemes

Claim 1. The N matrices defined in (4) and (5) satisfy the strong full-rank condition in Definition 1.

Proof. Consider any vector pattern $\mathbf{v} = (v_0, \dots, v_{N-1})$ such that $v_0 + \dots + v_{N-1} = \Delta$. Let \mathbf{G} be composed of the first v_i columns of G_i , $i = 0, \dots, N - 1$. For the construction in (4), \mathbf{G} is a Vandermonde matrix as each column contains successive powers of the same real number with exponents from 0 to $\Delta - 1$. As the $\beta_{k,j}$'s are distinct, \mathbf{G} has full rank. The result for the construction in (5) follows from the determinant formula for a generalized Vandermonde matrix [15]. ■

The above constructions satisfy the strong full-rank condition. However, experimental results (see also [16]) show these constructions result in badly conditioned \mathbf{G} matrices in the worst case. In addition, both (4) and (5) result in dense linear combinations of \mathbf{A} , rendering them unsuitable in the scenario when \mathbf{A} is sparse. Nevertheless, the UDM-based construction (5), provides a systematic way to take into account the sequential processing order of the worker nodes.

Remark 2. The RS-based scheme is in one-to-one correspondence with polynomial interpolation from any Δ (out of $N\ell$) distinct evaluation points. The UDM-based scheme uses much fewer evaluation points (only N) but is equivalent to interpolating a polynomial with roots of higher multiplicity.

IV. CODED SCHEMES SATISFYING THE WEAK FULL RANK CONDITION

In this section we present schemes that satisfy the s -weak full-rank condition. However, they have excellent numerical stability and are much sparser than those discussed in Section III. These schemes are obtained by first constructing a collection of matrices over a finite field \mathbb{F}_{p^n} (where p is prime) and then embedding the finite field matrices into real field by companion matrix. Towards this end, let $\tilde{u}(x) = \sum_{k=0}^d \tilde{u}_k x^k$ be a polynomial with coefficients from \mathbb{F}_{p^n} , i.e., $\tilde{u}(x) \in \mathbb{F}_{p^n}[x]$. The i -th Hasse derivative² of $\tilde{u}(x)$ is defined as

$$\tilde{u}^{[i]}(x) = \sum_{k=0}^d \binom{k}{i} \tilde{u}_k x^{k-i}, \quad (6)$$

where we emphasize that the quantity $\binom{k}{i}$ is interpreted as a element of \mathbb{F}_p . In this scenario, it can be shown that $\tilde{u}(x)$ has a zero of multiplicity m at a point $\beta \in \mathbb{F}_{p^n}$ (or in an appropriate extension field) if $\tilde{u}^{[i]}(\beta) = 0$ for $0 \leq i < m$ and $\tilde{u}^{[m]}(\beta) \neq 0$.

²To avoid confusion with the case of real-valued polynomials, we superscript the finite field polynomials with \sim and represent the Hasse derivatives with square brackets.

The work of [10], [11] shows that the following matrices $G_k, k \in [N]$, satisfy the strong full-rank condition over $\mathbb{K} = \mathbb{F}_{p^n}$, assuming $p^n \geq N + 1$ (see Remark 3 below).

$$G_k(i, j) = \begin{cases} \binom{i}{j} \beta_k^{i-j} & \text{if } i \geq j \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

where $\beta_k, k \in [N]$, are distinct non-zero elements in \mathbb{F}_{p^n} . We remark here that while the expression above is the same as the one in (5), the elements of (7) lie in \mathbb{F}_{p^n} .

One reason for considering the matrices in (7) is as follows. Suppose that we operate over \mathbb{F}_{2^n} , i.e., $p = 2$. Note that in (7) we can equivalently calculate $\binom{i}{j}$ over the integers and then reduce it modulo 2. In particular, this implies that whenever $\binom{i}{j}$ is even, the corresponding matrix entry will be zero. Thus, over finite fields, the $\{G_k\}_{k=0}^{N-1}$ matrices obtained using (7) are likely to be much sparser than those obtained from (5).

Example 2. Let $p = 2, n = 3, \ell = 3, \Delta = 4, N = 6$. Consider the polynomial $\tilde{u}(x) = \tilde{u}_0 + \tilde{u}_1 x + \tilde{u}_2 x^2 + \tilde{u}_3 x^3$ over \mathbb{F}_8 . Its i -th Hasse derivatives, $i = 0, 1, 2$, are

$$\begin{aligned} \tilde{u}^{[0]}(x) &= \tilde{u}_0 + \tilde{u}_1 x + \tilde{u}_2 x^2 + \tilde{u}_3 x^3, \\ \tilde{u}^{[1]}(x) &= \tilde{u}_1 + \tilde{u}_2 x, \text{ and} \\ \tilde{u}^{[2]}(x) &= \tilde{u}_2 + \tilde{u}_3 x. \end{aligned}$$

Note here that $\tilde{u}^{[1]}(x)$ has only two non-zero coefficients, whereas when considering derivatives over the reals, it will have three non-zero coefficients. Then,

$$G_i = \begin{bmatrix} 1 & 0 & 0 \\ \beta_i & 1 & 0 \\ \beta_i^2 & 0 & 1 \\ \beta_i^3 & \beta_i^2 & \beta_i \end{bmatrix}, \quad (8)$$

where $\beta_i \in \mathbb{F}_8$ and β_i values are distinct for $i \in [N]$.

Next, we utilize a representation of \mathbb{F}_{p^n} by $n \times n$ matrices over \mathbb{F}_p [12]. Let $\mathbb{F}_p[x]$ denote the ring of polynomials in x with coefficients from \mathbb{F}_p . Let α be a primitive element in \mathbb{F}_{p^n} and let $\pi_\alpha(x) = x^n + \sum_{i=0}^{n-1} \pi_i x^i \in \mathbb{F}_p[x]$ denote the primitive polynomial associated with α . The $n \times n$ companion matrix (over \mathbb{F}_p) associated with $\pi_\alpha(x)$ is

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & -\pi_0 \\ 1 & 0 & \cdots & 0 & -\pi_1 \\ 0 & 1 & \cdots & 0 & -\pi_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \vdots & 1 & -\pi_{n-1} \end{bmatrix}. \quad (9)$$

The elements of \mathbb{F}_{p^n} are represented by polynomials in α of degree smaller than n (with coefficients in \mathbb{F}_p) with regular polynomial addition and multiplication being reduced to lower powers by using $\pi_\alpha(\alpha) = 0$. Let $\Gamma : \mathbb{F}_{p^n} \rightarrow \mathbb{F}_p^n$ represent the mapping of a polynomial $a(\alpha)$ to its vector representation. The mapping $\zeta : \mathbb{F}_{p^n} \rightarrow \mathbb{F}_p^{n \times n}$ defined by $\zeta(\alpha^l) = C^l, \zeta(0) = 0$, maps elements of \mathbb{F}_{p^n} to their matrix representation. It is well known (a complete argument appears in [17]) that this is isomorphic to a representation that maps $a(\alpha) \in \mathbb{F}_{p^n}$ to $\Gamma(a)$.

The product of $a_1(\alpha)$ and $a_2(\alpha)$ is mapped to $a_1(C) \cdot \Gamma(a_2)$, where $a_1(C)$ represents the polynomial $a_1(x)$ evaluated at the matrix argument C .

Lemma 1. Let \mathbf{B} be a $n \times n$ matrix with entries from \mathbb{F}_{p^m} . Let $\tilde{\mathbf{B}}$ denote the $mn \times mn$ matrix obtained by applying the map ζ to each entry of \mathbf{B} . Note that $\det(\mathbf{B}) \in \mathbb{F}_{p^m}$ and $\det(\tilde{\mathbf{B}}) \in \mathbb{F}_p$. We claim that

$$\det(\mathbf{B}) \neq 0 \iff \det(\tilde{\mathbf{B}}) \neq 0. \quad (10)$$

Furthermore, let $\hat{\mathbf{B}}$ denote the $mn \times mn$ matrix over the integers \mathbb{Z} obtained by mapping each element of $\tilde{\mathbf{B}}$ to the corresponding integer in $\{0, \dots, p-1\}$. If $\det(\mathbf{B}) \neq 0$ we have $\det(\hat{\mathbf{B}}) \neq 0$ over the reals.

Proof. The result in (10) follows from the isomorphism discussed above (for details see [17]). Note that $\det(\tilde{\mathbf{B}}) \in \mathbb{F}_p$. It can also be equivalently computed by finding $\det(\hat{\mathbf{B}})$ over the reals and reducing the result modulo p . Thus, we have that $\det(\tilde{\mathbf{B}}) \neq 0$ over reals. ■

We now present the construction of systems that satisfy the s -weak full-rank property.

Lemma 2. Let $G_k, 0 \leq k < N$, be a collection of N matrices with size $\Delta \times \ell$ over \mathbb{F}_{p^n} that satisfy the strong full-rank property. Consider the N matrices G'_k of dimension $n\Delta \times n\ell$ over \mathbb{F}_p , where G'_k is obtained by applying mapping ζ to each entry of G_k . Then, the collection $\{G'_k\}_{k=0}^{N-1}$ satisfies the n -weak full rank condition over \mathbb{R} .

Proof. This is an immediate consequence of Lemma 1. ■

Example 3. Consider collection of $N = 6$ matrices presented in Example 2 over \mathbb{F}_8 . Let the primitive polynomial over \mathbb{F}_8 be $\pi_\alpha(x) = x^3 + x + 1$. Suppose that $\beta_k = \alpha^k, k \in [N]$. Then,

$$G_k = \begin{bmatrix} 1 & 0 & 0 \\ \alpha^k & 1 & 0 \\ \alpha^{2k} & 0 & 1 \\ \alpha^{3k} & \alpha^{2k} & \alpha^k \end{bmatrix}, \quad G'_k = \begin{bmatrix} I & 0 & 0 \\ C^k & I & 0 \\ C^{2k} & 0 & I \\ C^{3k} & C^{2k} & C^k \end{bmatrix}.$$

By Lemma 2, the collection $\{G'_k\}_{k=0}^5$ satisfies the 3-weak full rank condition over \mathbb{R} .

We note here that Lemma 2 can also be applied to an RS code defined over a finite field.

Remark 3. Our proposed scheme requires us to operate over an extension field large enough so that $p^n \geq N + 1$ for the UDM based approach and $p^n \geq N\ell + 1$ for the RS-based approach. Thus, the second term in the worst case computational load (cf. Eq. (1)) can be made as small as desired by choosing Δ large enough. Increasing Δ does come at the cost of high condition numbers (cf. Section V).

V. COMPARISONS OF THE DIFFERENT SCHEMES

In this section, we compare the performance of the different schemes that have been proposed in this work. For each scheme, we construct all possible matrices \mathbf{G} based on $\{G_k\}_{k=0}^{N-1}$ and $\Psi_{N,\ell,s}^{=Q_b}$ and calculate their condition number. We

RS +
RS + C. M. of
RS + Embedding
RS + C. M. of
UDM +
UDM + Embedding
UDM + C. M. of
UDM + C. M. of

■ Density of G_i

Fig. 2: Performance comparison for system with $N = 6, \gamma = 3/4$ and $Q_b = 4$.

report the maximum condition number over all such possible G_i 's. Furthermore, we also report the average percentage of non-zero elements in the G_i matrices for each collection.

In Fig. 2 we report results for a system with $N = 6$ workers and storage capacity for each worker $\gamma = 3/4$. For the “RS-based scheme”, we set $\beta_{i,j}, i = 0, \dots, 5, j = 0, 1, 2$, in (4) to 18 equally spaced reals within the interval $[-1, 1]$. For the “UDM-based scheme”, we set β_k in (5), $k = 0, \dots, 5$ to 6 equally spaced reals within the interval $[-1, 1]$. For “RS + Embedding from $GF(19)$ ”, we construct (4) over $GF(19)$. Then we embed (4) into \mathbb{R} by using the natural mapping of $GF(19)$ into the integers. We construct “UDM + Embedding from $GF(7)$ ” in a similar manner. It can be seen that the condition number of “UDM scheme + Embedding from $GF(7)$ ” is the lowest when compared the other three schemes discussed thus far.

The other bars of Fig. 2 correspond to the companion matrix approach. In each of these cases we first design the RS-based or the UDM-based scheme over the corresponding extension field and then use the companion matrix idea introduced in Section IV. One can observe that the RS + Companion matrix schemes typically have high condition number. This is because the size of the companion matrix needs to be large enough to accommodate $N\ell$ evaluation points. The UDM + Companion matrix schemes can work with extension fields larger than N , so their companion matrices tend to be smaller. Another advantage of the companion matrix approach is that the schemes are much sparser. Indeed, the “UDM + Companion matrix $GF(3^2)$ ” in Fig. 2 not only has a very low worst case condition number of 182 but also a sparsity level of 36% which is the second lowest among all the schemes.

To better understand the performance corresponding to different choices of extension field, we consider a larger system with $N = 15, \gamma = 1/2$ in Fig. 3. We observe that the RS-based scheme is worse than the UDM-based scheme. Another observation is that the “UDM + Companion matrix $GF(3^3)$ ” has the lowest condition number (624) and the G_i matrices become sparser when the size of companion matrix increases.

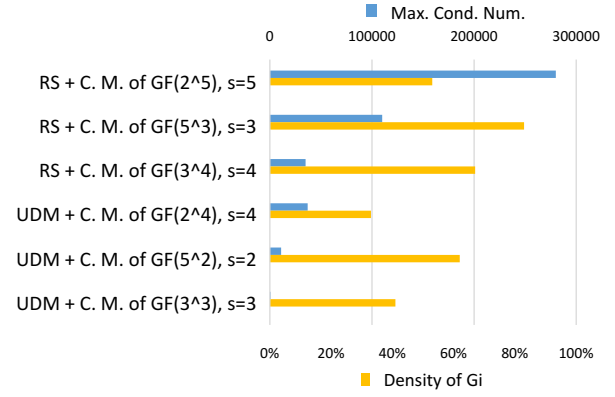


Fig. 3: Performance comparison of different extension fields for a system with $N = 15, \gamma = 1/2$ and $Q_b = 4$.

REFERENCES

- [1] Q. Yu, M. Maddah-Ali, and S. Avestimehr, “Polynomial codes: an optimal design for high-dimensional coded matrix multiplication,” in *Proc. of Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2017, pp. 4403–4413.
- [2] L. Tang, K. Konstantinidis, and A. Ramamoorthy, “Erasure coding for distributed matrix multiplication for matrices with bounded entries,” *IEEE Comm. Lett.*, vol. 23, no. 1, pp. 8–11, 2019.
- [3] K. Lee, C. Suh, and K. Ramchandran, “High-dimensional coded matrix multiplication,” in *IEEE Int. Symp. on Inf. Theory*, 2017, pp. 2418–2422.
- [4] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, “Speeding up distributed machine learning using codes,” *IEEE Trans. on Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, 2018.
- [5] S. Dutta, V. Cadambe, and P. Grover, “Short-dot: Computing large linear transforms distributedly using coded short dot products,” in *Proc. of Adv. in Neural Inf. Proc. Sys. (NIPS)*, 2016, pp. 2100–2108.
- [6] A. Mallick, M. Chaudhari, and G. Joshi, “Rateless codes for near-perfect load balancing in distributed matrix-vector multiplication,” preprint, 2018, [Online] Available: <https://arxiv.org/abs/1804.10331>.
- [7] S. Wang, J. Liu, and N. B. Shroff, “Coded sparse matrix multiplication,” in *Proc. 35th Int. Conf. on Mach. Learning, ICML*, 2018, pp. 5139–5147.
- [8] S. Kiani, N. Ferdinand, and S. C. Draper, “Exploitation of stragglers in coded computation,” in *IEEE Int. Symp. on Inf. Theory*, 2018, pp. 1988–1992.
- [9] A. B. Das, L. Tang, and A. Ramamoorthy, “C³LES: Codes for coded computation that leverage stragglers,” in *IEEE Inf. Th. Workshop*, 2018, pp. 1–5.
- [10] M. Y. Rosenbloom and M. A. Tsfasman, “Codes for the m-metric,” *Probl. Inf. Transm.*, vol. 33, no. 1, pp. 45–52, 1997.
- [11] A. Ganesan and P. O. Vontobel, “On the existence of universally decodable matrices,” *IEEE Trans. on Inf. Theory*, vol. 53, no. 7, pp. 2572–2575, 2007.
- [12] W. P. Wardlaw, “Matrix representation of finite fields,” *Mathematics Magazine*, vol. 67, no. 4, pp. 289–293, 1994.
- [13] N. Raviv, Y. Cassuto, R. Cohen, and M. Schwartz, “Erasure correction of scalar codes in the presence of stragglers,” in *IEEE Int. Symp. on Inf. Theory*, 2018, pp. 1983–1987.
- [14] N. Raviv, Q. Yu, J. Bruck, and S. Avestimehr, “Download and access tradeoffs in Lagrange coded computing,” in *IEEE Int. Symp. on Inf. Theory*, 2019.
- [15] G. Sobczyk, “Generalized Vandermonde determinants and applications,” *Aportaciones Matematicas, Serie Comunicaciones*, vol. 30, pp. 203–213, 2002.
- [16] W. Gautschi, “How (un) stable are Vandermonde systems?” *Asymptotic and Computational Analysis (Lecture Notes in Pure and Applied Mathematics)*, vol. 124, pp. 193–210, 1990.
- [17] A. Ramamoorthy, L. Tang, and P. O. Vontobel, “Universally decodable matrices for distributed matrix-vector multiplication,” preprint, 2019, [Online] Available: <https://arxiv.org/abs/1901.10674>.