# Analysis for RS-coding and fountain-code for erasure coding

**Yusheng Li**

University of Wisconsin Madison

li2339@wisc.edu

**Abstract.** This research provides an in-depth analysis of RS-coding and fountain-coding for erasure coding, focusing on the different circumstances that data storage may face during daily usage. The essay outlines the fundamental information of RS-code and fountain-code, and compares their effectiveness in addressing varying degrees of erasure in the data disk. The research includes case studies and content analysis to identify appropriate circumstances for using different coding methods. Based on these findings, the research concludes that RS-coding is more effective when fixing erasures up to a certain limit, beyond which fountain-coding is a better option. Finally, the research presents a research example to demonstrate the practical usage of different coding methods in specific circumstances, highlighting the advantages and disadvantages between the two coding systems. Overall, this research provides valuable insights into erasure coding and can aid in optimizing data storage systems for efficient and reliable performance.

**Keywords:** RS-code, fountain-code, erasures.

## 1. Introduction

Erasure coding is a critical method for securing data during transmission against packet loss and other types of data loss [1]. This process involves dividing data into smaller chunks, adding redundancy to each chunk, and distributing them across multiple storage devices or communication routes [2,3]. Erasure coding finds wide application in today's society, including cloud storage, data centers, video streaming, and wireless networks. Reed-Solomon (RS) codes and Fountain codes are two commonly used erasure code types [4]. In RS-coding, data is broken up into blocks with redundant information added to each block [5, 6].

Even if some blocks are lost, the redundant information permits reconstruction of the original data. Several applications, such as CD-ROMs, DVDs, RAID systems, and satellite communication systems, have made substantial use of RS-coding [7]. Conversely, fountain codes, a type of rateless erasure code, do not require block division. Fountain codes generate a stream of encoded packets, which are transmitted to the receiver using a random number generator. As long as enough packets of encoded data are received, the receiver can decode the original data. This characteristic makes fountain codes particularly useful in applications with erratic networks or channels [8]. This study examines both RS-coding and fountain-coding erasure coding methods. We analyze the advantages and disadvantages of these two approaches, compare and contrast them, and evaluate their performance under various network and channel setups [9]. Moreover, we explore some of the possible applications of both coding approaches in various contexts [10]. Our analysis aims to provide researchers and engineers with a better understanding of the benefits and limitations of these coding methods, helping them make informed decisions about their application in various scenarios.

## 2. Relevant theory

### *2.1. Rs-code*

In Rs coding, data will first be divided into blocks of fixed length. By adding additional redundancy symbols in to each block, Rs code is being generate. The number of added symbols depends on the desired level of error correction and the number of erasures that the code needs to be correct. The added redundancy symbols are being calculated using a polynomial multiplication algorithm based on Galois fields. Assuming that the block length is n and each block is using polynomial of degree n-1. Notice that the additional redundancy symbols are computed using the polynomial remainder theorem, in other worlds, it divides the data polynomial by using a generator matrix. The generator matrix should be carefully chosen in order to ensure the desire error-correction capabilities.

As data is received, the RS decoder computes the syndrome of the received data using a similar polynomial multiplication technique. The amount of redundancy symbols, or m, determines the degree of the polynomial that represents the syndrome. The symptom ought to equal 0 if there are no mistakes or erasures. The decoder can use the syndrome, which won't be zero if errors or erasures exist, to pinpoint the locations and kinds of errors or erasures. The error or erasure values are then calculated by the RS decoder using the error/erasure locator polynomial and the error/erasure evaluator polynomial. The decoder then fixes any mistakes or erasures and outputs the updated data.

In general, polynomial multiplication methods based on Galois fields are used to add redundancy symbols to data in Reed-Solomon coding. The code can now rectify errors and erasures that occur during data transfer or storage, preserving the integrity and dependability of the data. As shown in Figure 1.

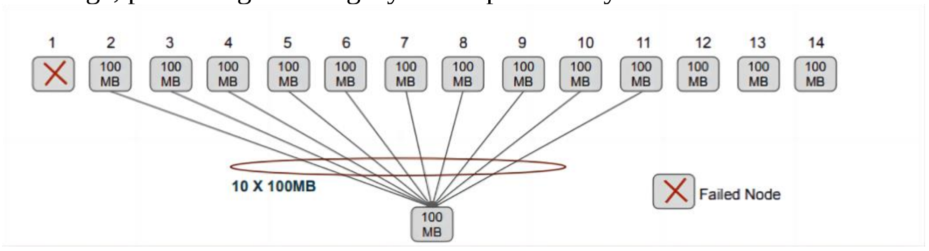

**Figure 1.** [14,10] RS code.

This is illustrated in Fig. 1 in the case of Facebook's RS code. If each node stores 100MB of data, then the total amount of data download needed for the repair of a node storing 100MB from the 10 helper nodes equals 1GB. Thus, the repair degree in the case of a [n, k] MDS code equals k, and the repair bandwidth is the total file size, which is k times the amount of data stored in a node.

## 3. Analysis

The following is the analysis of the generator matrix for a k = 3, n= 5 Reed-Solomon code over GF (7).

$$G = \begin{matrix} 1 & \alpha 1 & \alpha 1^2 & \dots & \alpha 1^{n-1} \\ 1 & \alpha 2 & \alpha 2^2 & \dots & \alpha 2^{n-1} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \alpha k & \alpha k^2 & \dots & \alpha k^{n-1} \end{matrix} \tag{1}$$

The figure shows the formation of generator matrix. The Reed-Solomon code generator matrix comprises k rows and n columns, where k is the number of message symbols and n is the length of the codewords. The generator matrix is applied to a field F of q elements. The Reed-Solomon code is represented by the generator matrix, where each row represents a distinct codeword and each column represents a different symbol. The coefficients of the polynomials used to create the codewords are the entries in the generator matrix. In this circumstance, k is equal to 3 and n is equal to 5. Then get the generator matrix by using the equation of the generator matrix.

$$G = \begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 1 & 2 \\ 1 & 3 & 2 & 6 & 4 \end{matrix} \quad (2)$$

The above figure is the generator matrix that being calculate by the generator matrix equation. Notice that for a2^3, a2^3 is equal to 8. Since the generator matrix should be over GF(7), a2^3 need to mod 7 in order to get the answer, which is 1. It is also similar for the other part of the matrix if ak^k is bigger than 7 (where k is smaller or equal to n-1).

Reed-Solomon codes are made to fix mistakes in messages sent across noisy channels. The number of parity symbols used in a Reed-Solomon code, which in turn depends on the code parameters k (number of message symbols) and n, determines the amount of errors that can be repaired by the code (total number of symbols in the code). In a Reed-Solomon code, the number of parity symbols is n-k. The code can correct up to t errors, where t is the largest integer such that: t <= ⌊(n-k)/2⌋.This means that the code can correct any set of t or fewer errors, where t is less than or equal to half the number of parity symbols. In the given circumstances, k = 3 and n = 5. Therefore, the number of parity symbols in the code is 5 - 3 = 2. The code can correct up to t errors, where t is the largest integer such that: t <= ⌊ (5-3)/2⌋ = 1. Hence, the Reed-Solomon code can correct up to ⌊(n-k)/2⌋ = ⌊ (5-3)/2⌋ = 1 symbol errors.

Also, this Reed-Solomon code can correct up to n-k = 5-3 = 2 symbol erasures

## 4. Implementation

Reed-Solomon code is able to the complete decoding up to the maximum number of correctable erasures. By using [2 1 3] as the encoded message, the generator matrix, and the basic equation of RS-code (figure below). The code after decoded should be c = [6 2 4 4 1].

$$m * G = C \quad (3)$$

Then the following matrixes listed all the possible circumstances for the one erasure and erasure circumstances. Notice that “?” represent the data being erased.

$$\begin{matrix} ? & 2 & 4 & 4 & 1 \\ 6 & ? & 4 & 4 & 1 \\ 6 & 2 & ? & 4 & 1 \\ 6 & 2 & 4 & ? & 1 \\ 6 & 2 & 4 & 4 & ? \end{matrix} \quad \begin{matrix} ? & ? & 4 & 4 & 1 \\ ? & 2 & ? & 4 & 1 \\ ? & 2 & 4 & ? & 1 \\ ? & 2 & 4 & 4 & ? \end{matrix} \quad \begin{matrix} 6 & ? & ? & 4 & 1 \\ 6 & ? & 4 & ? & 1 \\ 6 & ? & 4 & 4 & ? \end{matrix} \quad \begin{matrix} 6 & 2 & 4 & ? & ? \end{matrix} \quad (4)$$

However, when a failure that has more erasures than the maximum correctable number of erasures

## 5. Fountain code

Fountain codes are a class of error control codes that can generate an infinite number of coded symbols from a limited number of source symbols [2].

A set of forward error correction (FEC) codes called fountain codes, often known as rateless codes, offer a versatile and effective technique to send data over shaky communication links. Fountain codes don't need a predetermined number of encoded symbols to be generated beforehand, unlike conventional block codes. Instead, they produce an endless number of encoded symbols that can be sent until the recipient has enough information to decode them and recover the original data. Fountain codes use a random linear combination of the source symbols as the basis for encoding. The source symbols can be any kind of data, though they are typically binary digits. A fountain code creates a series of random coefficients and multiplies each coefficient by a source symbol to encrypt a message. The final step is to combine the products to create an encoded symbol. There is no limit to the number of encoded symbols that can be produced by repeating this process indefinitely. A decoding algorithm can be used by the recipient of the encoded symbols to reveal the original message. The random linear structure of the fountain code is intended to be utilized by the decoding algorithm. The process recovers a set of intermediate symbols by repeatedly gathering a portion of the encoded symbols. The original message can then be recreated by combining the intermediate symbols with additional encoded symbols.

The fact that fountain codes are rateless is one of their main benefits. They can adjust to the unique requirements of the communication channel because they generate an infinite number of encoded symbols. The sender can simply send out more encoded symbols until the receiver has enough information to decode the message if the channel is noisy or has a high error rate. Due to the wide variations in channel quality, wireless and satellite communications are particularly well-suited for fountain codes.

Fountain codes' simplicity is an additional benefit. On a variety of hardware platforms, the encoding and decoding algorithms can be successfully implemented because they are not particularly complex. Since they can be used in real-time applications like file transfer protocols and video streaming, fountain codes are a desirable choice.

The BitTorrent protocol for file sharing, the Digital Fountain video streaming system, and the Interplanetary Internet, a proposed network for space exploration missions are just a few examples of the many applications that have used fountain codes. With ongoing efforts to improve their performance and modify them to new communication scenarios, they remain an active area of research and development. The principle of fountain codes can be thought of as analogous to solving a set of simultaneous linear equations. If there are 3 equations in 3 unknowns, the equations can be solved for the unknowns. If there are more than 3 equations available in the same 3 unknowns, any 3 equations may be chosen to solve for the unknowns [3]. Usually, there are 5 steps for fountain codes to fix the erasure. Firstly, fountain code has the random symbol generation. A random number generator can generate a finite stream of random symbols. These symbols can be any value from the alphabet of the code. The second step is encoding. Previously, Reed-Solomon code is used to encode each block of m random symbols. In this case, Reed-Solomon code can be used to encode the original message in a similar way by using the same generator matrix as before but with m symbols. The third step is the transmissions: By transmitting the encoded symbol over the communication channel. Any symbols that are lost due to erasures are simply discarded. The fourth step is decoding: at the receiver side, The same Reed-Solomon code can be used to decode the remaining encoded symbols, as well as any additional encoded symbols that are received in the future. The decoding procedure is the same as for the original Reed-Solomon code. Last but not least, once a sufficient number of symbols is being decoded, the recovered symbols can be used to reconstruct the original message. In this case, at least three symbols are needed to be decoded in order to recover the original message. For the reason that three symbols are being lost during transmission, and it is significant to recover these symbols to reconstruct the original message. Also, there is an algorithm in fountain code, which is called LT decoding algorithm. An (n,k) fountain code can be decoded using the probabilistic LT decoding algorithm to restore the original message in an erasure channel. The algorithm is based on the straightforward notion that new equations can be created and the number of unknowns decreased by using randomly generated linear combinations of the received encoded symbols. Notice that a trustworthy decoder for fountain codes is one that can decode any lC = k(1 + e) encoded symbol to yield the k message symbols. The fraction of excess symbols over k that are needed to decode the k message symbols is represented by the factor 1 + e, which is also known as the decoding inefficiency. For effective fountain codes, lC is close to k, or nearly zero [3]. To use the LT decoding algorithm to solve the erasure problem with an(n,k) fountain code with k= 3 and n =5 , GF(7). Suppose that there are 3 erasures which exceed the maximum erasure Reed-Solomon code could handle.

First of all, generating the encoding matrix A with size n*K, where each entry is chosen uniformly at random from the field GF(7). The encoding matrix maps the original k = 3 message symbols and the n = 5 encoded symbols. Secondly, generate an encoded symbol vector y of length n, by multiplying the encoding matrix A with the message vector x of length k. Specifically, y = A*x. Thirdly, assuming that two received encoded symbols of y remain after the deletion of three y-encoded symbols during transmission. Next, Using the two received encoded symbols, a linear equation is formed in order to solve for the original message vector x. Specifically, set S be the set of indices of the received symbols. $y[i] = A[i,1]*x[1] + A[i,2]*x[2] + A[i,3]*x[3]$ for i in S Selecting a random linear combination of the encoding matrix rows that correspond to the received symbols for each I that is not in S. A random vector r[i] of length 2 is selected such that r[i][j] is uniformly selected at random from GF(7) for j = 1,2. This is done for each I that is not in S. The following equation is then created:

$$y[i] \ = \ A[i,1] * r[i][1] * yS[0] + r[i][2] * y[S[1]]) \ + \ A[i,2]* (r[i][1] * y[S[0]] + r[i][2] * y[S[1]]) \ + \ A[i,3]* (r[i][1] * y[S[0]] + r[i][2] * y[S[1]]) \tag{5}$$

A set of n linear equations in k unknowns currently exists. The Gaussian elimination approach can be used to find the solution for the unidentified message vector x. Specifically, using the basic row operations to reduce the system of equations to row echelon form, and then back-substitute to find the unknowns. Lastly, retrieve the original message after the unknown message vector x has been solved for.

## 6. Conclusion

Pros**:** High error correction capacity: RS codes are excellent for applications requiring high reliability and data integrity because they can repair a significant number of faults in the received data. Easy decoding technique: RS codes include a decoding algorithm that is both effective and simple, making them a viable option for applications that require little computing resources. Technology that is well-established: RS codes have been used for many years in a variety of applications, and it is well known how they work and behave **Cons:** High redundancy: RS codes need a substantial level of redundancy in the sent data to achieve strong error correcting capabilities, which can lower the transmission efficiency. Fixed data length: RS codes are less adaptable than other coding systems since they require a fixed data length.

Pros: High transmission efficiency: Fountain codes can produce an infinite number of encoded symbols from a small number of source symbols, leading to great transmission efficiency. Flexibility: Fountain codes are adaptable to various transmission conditions and error rates and can be used to safeguard data of any length. Reduced decoding complexity: Fountain codes are appropriate for low-power devices since their decoding method is rather straightforward and only needs a modest amount of memory.

Cons**:** Reduced error correction capability: Fountain codes are less capable of correcting errors than RS codes and may not be appropriate for applications requiring great dependability. Complicated encoding algorithm: Compared to RS codes, fountain codes have a more complicated encoding procedure that may call for more computing power and memory. Restricted knowledge: Because fountain codes are a relatively new technology, their functionality and behavior are yet not completely understood.

## References

[1] C. Ezekannagha, A. Becker, D. Heider et al., "Design considerations for advancing data storage with synthetic DNA for long-term archiving," Mater. Today Bio, vol. 12, Art. no. 100306, pp. 1-9, 2022.
[2] M. R. Z. Haddad, "Visible light communication system using 8x1 WDM," Master's thesis, Altınbaş University, Graduate School of Education Sciences, Istanbul, Turkey, 2020.
[3] J. Jeong, S. J. Park, J. W. Kim et al., "Cooperative sequence clustering and decoding for DNA storage system with fountain codes," Bioinformatics, vol. 37, no. 19, pp. 3136-3143, 2021.
[4] H. Tian, D. F. Zhao, Y. F. Yang et al., "Research of LT code based on key information feedback in deep space communication," IEEE Access, vol. 8, pp. 103956-103972, 2020.
[5] K. Pang, Z. Lin, B. F. Uchoa-Filho et al., "Distributed network coding for wireless sensor networks based on rateless LT codes," IEEE Wireless Communications Letters, vol. 1, no. 6, pp. 561-564, 2012.
[6] C. Fu, T. Lin, C. Gong et al., "Anti error and erasure coding for water-to-air visible light communication through wavy water surface with wave height up to 0.6 meters," Opt. Express, vol. 30, no. 11, pp. 18743-18761, 2022.
[7] M. Welzel, P. M. Schwarz, H. F. Löchel et al., "DNA-Aeon provides flexible arithmetic coding for constraint adherence and error correction in DNA storage," Nat. Commun., vol. 14, no. 1,

Art. no. 628, 2023.
[8] X. He and K. Cai, "On Decoding Fountain Codes with Erroneous Received Symbols," in 2020 IEEE Information Theory Workshop (ITW), Riva del Garda, Italy, 2021, pp. 1-5.
[9] K. W. U. Kesong, C. A. O. Xianbin, Z. Chen et al., "Adaptive mobile video delivery based on fountain codes and DASH: a survey," ZTE Commun., vol. 16, no. 3, pp. 9-14, 2020.
[10] A. Yatribi, M. Belkasmi, and F. Ayoub, "An Efficient and Secure Forward Error Correcting Scheme for DNA Data Storage," in Proceedings of the Tenth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2018) 10, Agadir, Morocco, 2020, pp. 226-237.