Math 2568H: Project 1

Instructions: To complete this project, you may reference any resource available on the internet. Feel free to write code in MATLAB, Octave, or Python; other languages are also acceptable. You may discuss with other students, but you must write your own computer code and writeup. You are encouraged to type your writeup in LaTeX. Show all work, cite all references used, and provide all of your code.

Submit a draft of your solution to the first 6 problems before class on Sep 17, 2025; this will be graded based on completeness. Submit a final draft of your entire solution before class on Oct 3, 2025; this will be graded based on accuracy, brevity, and clarity.

Suppose Alice and Bob wish to communicate through a channel. In this channel, one may transmit a stream of **bits** (i.e., zeros and ones), but a small fraction of the bits will be *flipped*. For example, if Alice transmits $(0, 1, 1, 0, 0, 1, 0)$, Bob might receive $(0, 0, 1, 0, 0, 1, 1)$, which can be regarded as a *noisy* version of Alice's message. Here, we denote the bit flips in red, but Bob will not be told which bits were flipped. Alice and Bob need to agree on a code that will allow them to communicate through this noisy channel. For example, suppose the intended message is either `Yes` or `No`. Then they can encode `Yes` as the sequence of bits $(1, 1, 1)$ and `No` as $(0, 0, 0)$. Provided at most one of the bits is flipped, Bob can decode the message. For example, Bob can interpret $(1, 1, 0)$ as a noisy version of $(1, 1, 1)$, meaning Alice's message was `Yes`.

Coding theory provides a general method of encoding messages in a way that is robust to such noise. A **code** is a subset C of $\{0, 1\}^n$, and the members of C are known as **codewords**. To measure robustness to noise, it is helpful to consider the **Hamming distance** on $\{0, 1\}^n$, measured by the number of bits that differ. For example, $(0, 0, 0)$ and $(1, 1, 1)$ have Hamming distance 3. Recall that Bob interpreted $(1, 1, 0)$ as a noisy version of $(1, 1, 1)$ since their distance is only 1, whereas the distance from $(1, 1, 0)$ to $(0, 0, 0)$ is 2. The **minimum distance** of C is the smallest Hamming distance between distinct codewords in C. For example, the minimum distance of the code $\{(0, 0, 0), (1, 1, 1)\}$ is 3.

(Q1) Suppose Alice and Bob agree to communicate using a code $C \subseteq \{0, 1\}^n$. Alice wishes to communicate $x \in C$, but Bob receives a noisy version $y \in \{0, 1\}^n$. Assuming bit flips are rare, which codeword in C is most likely to be the codeword that Alice transmitted?

(Q2) Given a code C of minimum distance d, what is the maximum number of bit flips that the channel can introduce while still allowing Bob to correctly identify the transmitted codeword?

(Q3) Design a code of two codewords in $\{0, 1\}^3$ so that the minimum distance is as large as possible. Do the same with three and four codewords in $\{0, 1\}^3$.

Error-correcting codes can be made computationally efficient with the help of linear algebra. For this, we think of $\{0, 1\}^n$ as a vector space—not over the real numbers $\mathbb{R}$, but rather, over the binary numbers $\mathbb{F}_2$. Unlike the real numbers, $\mathbb{F}_2$ consists of only two elements 0 and 1, and addition and multiplication are defined modulo 2:

$+$	0	1
0	0	1
1	1	0

$\times$	0	1
0	0	0
1	0	1

It's convenient to perform computations over the integers before reducing modulo 2, e.g.:

$$(1 \times 0) + (1 \times 1) + (1 \times 1) + (0 \times 0) + (1 \times 1) = 0 + 1 + 1 + 0 + 1 = 3 \equiv 1.$$

Overall, we may identify $\{0,1\}^n$ with the vector space $\mathbb{F}_2^n$ over the scalar field $\mathbb{F}_2$. With this formalism, we use the following matrices to define a code of 16 codewords in $\{0,1\}^7$:

$$G := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}, \qquad H := \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Given a message $x \in \mathbb{F}_2^4$, we consider the corresponding **linear encoding** $Gx \in \mathbb{F}_2^7$. The resulting **linear code** is the subspace $C := \operatorname{im} G$ of all such codewords.

(Q4) Verify that the minimum distance of the linear code C equals 3.

(Q5) How can you determine a message $x \in \mathbb{F}_2^4$ from its linear encoding $Gx \in C$?

(Q6) Verify that $\ker H = C$.

(Q7) How can you use H to test whether a given $y \in \mathbb{F}_2^7$ resides in C?

We can use the above linear encoder G to encode English messages as bit streams that are robust to noise. In the following example, we use the ASCII character encoding to obtain a binary representation of each character of a message:

Original message:	`Hi!`					
ASCII encoding:	72		105		33	
Convert to binary:	00010010		10010110		10000100	
Split into blocks:	0001	0010	1001	0110	1000	0100
Linear encoding:	0001111	0010011	1001001	0110110	1000110	0100101
Noisy version:	0011111	0010010	1001001	0010110	1000110	0101101

We split the 8-bit representation of each ASCII encoding into two blocks of size 4 so that we can linearly encode the resulting members of $\mathbb{F}_2^4$ by applying G. Notice that the convert-to-binary step above maps $\sum_{k=0}^{7} a_k 2^k$ to $(a_0, \ldots, a_7)$, i.e., the least significant bit is first.

(Q8) Explain how to invert each step of the above example. You may assume that each block of size 7 in the noisy version has at most one bit flip. (Hint: H will be useful in inverting this final step.)

(Q9) The above process was used to produce the noisy sequence of 980 bits available at https://tinyurl.com/y4h9dtvt. Determine the original message.