Math 215 Project 2: Cryptography

Written by: Rebecca Strautman

Problem Description: Sometimes it is necessary to encode messages, for instance, when sensitive information that is sent over the internet.The main idea in cryptography is to encode a message so that the only person who can decode it is someone with a deciphering *key.* One simple way of encoding messages is through the use of matrices to transform the message. A message, in the form of numbers, can be created, then *encoded* with a matrix operation acting on these numbers. A *key* would consist of a *decoding* matrix which one can use to reproduce the original message.

In its most basic form, one would take a message consisting of letters a through z and assign each a number 1 through 26, encode the message, then send a user the key. Clearly, this can be easily broken in many different ways. One can simply guess entries for the *key ,* decode the message, and attempt to see if the resulting message makes sense for different combinations of a through z. For small matrices, a computer can easily crack these codes. In the real world, one does not simply use numbers 1 through 26. One may use modular arithmetic, and perhaps assign a through z to be multiple values from the product of two large prime numbers, say $1,747,822,896,920,092,227,343$ and $105,646,155,480,762,397$. In order to decode, one would need to know both of these large primes. It is very difficult to find large primes in general, and using the previous guess and check type method would not work for messages of this form. We will consider the basic case for this project.

Example Problem: Let us encode the message *LINEAR ALGEBRA IS FUN.* . First, we must assign a number to each character (letter and punctuation mark). For simplicity's sake, let's let $A = 1, B = 2, ...Z = 26$. We also need numbers for the space character and the period, so we can choose 27 for space, and 28 for period. Now we can convert each character to a number.

L	*I*	*N*	*E*	*A*	*R*		*A*	*L*	*G*	*E*	*B*	*R*	*A*		*I*	*S*		*F*	*U*	*N*	.
12	*9*	*14*	*5*	*1*	*18*	*27*	*1*	*12*	*7*	*5*	*2*	*18*	*1*	*27*	*9*	*19*	*27*	*6*	*21*	*14*	*28*

If we stopped here, the code could be very easily broken, so we need to scramble it. We can use a matrix to do so. We consider the following 3×3 encoding matrix:

$$B = \begin{bmatrix} -3 & -3 & -4 \\ 0 & 1 & 1 \\ 4 & 3 & 4 \end{bmatrix}$$

In order to multiply our message by this matrix, we need to put our message into a matrix with 3 rows. More specifically, we need to put our message into a $3 \times N$ matrix, so that the multiplication is defined. We can do this putting the message into 3×1 column vectors:

$$\begin{bmatrix} 12 \\ 9 \\ 14 \end{bmatrix}, \begin{bmatrix} 5 \\ 1 \\ 18 \end{bmatrix}, \begin{bmatrix} 27 \\ 1 \\ 12 \end{bmatrix}, \begin{bmatrix} 7 \\ 5 \\ 2 \end{bmatrix}, \begin{bmatrix} 18 \\ 1 \\ 27 \end{bmatrix}, \begin{bmatrix} 9 \\ 19 \\ 27 \end{bmatrix}, \begin{bmatrix} 6 \\ 21 \\ 14 \end{bmatrix}, \begin{bmatrix} 28 \\ 27 \\ 27 \end{bmatrix}$$

Note that since the number of letters in the message is not divisible by 3, we fill in the last column vector with space characters. Now we use these column vectors as the columns of a 3×8 matrix:

$$A = \begin{bmatrix} 12 & 5 & 27 & 7 & 18 & 9 & 6 & 28 \\ 9 & 1 & 1 & 5 & 1 & 19 & 21 & 27 \\ 14 & 18 & 12 & 2 & 27 & 27 & 14 & 27 \end{bmatrix}.$$

Multiplying the encoding matrix by our the message matrix gives:

$$BA = \begin{bmatrix} -119 & -90 & -132 & -44 & -165 & -192 & -137 & -273 \\ 23 & 19 & 13 & 7 & 28 & 46 & 35 & 54 \\ 131 & 95 & 159 & 51 & 183 & 201 & 143 & 301 \end{bmatrix}$$

The encoded message is the list the numbers in order of the column vectors (ie $-119, 23, 131, -90, 19, 95, -132, \dots$). To decode this message, we would write these numbers as column vectors, and then multiply by the decoding matrix (the inverse of the encoding matrix). We have a decoding matrix of

$$B^{-1} = \begin{bmatrix} 1 & 0 & 1 \\ 4 & 4 & 3 \\ -4 & -3 & -3 \end{bmatrix}$$

and clearly

$$\begin{bmatrix} 1 & 0 & 1 \\ 4 & 4 & 3 \\ -4 & -3 & -3 \end{bmatrix} \begin{bmatrix} -119 & -90 & -132 & -44 & -165 & -192 & -137 & -273 \\ 23 & 19 & 13 & 7 & 28 & 46 & 35 & 54 \\ 131 & 95 & 159 & 51 & 183 & 201 & 143 & 301 \end{bmatrix} = B^{-1}BA = A$$

which, with the numbers in order by column vector $(12, 9, 14, 5, \dots = L, I, N, E, \dots)$, is our original message.

Assigned Problem: Use the previous method to translate a code into the original message. The following was used as an ENCODING matrix:

$$\begin{bmatrix} 4 & 3 & 4 & 6 & -2 & -1 \\ 8 & 9 & 4 & -2 & -2 & -5 \\ -3 & 0 & 3 & -1 & 1 & 2 \\ 8 & 9 & 8 & 9 & 8 & -9 \\ 4 & 2 & -2 & 3 & 0 & 0 \\ 4 & 3 & -1 & -1 & 4 & 5 \end{bmatrix}.$$

The encoded code is then

272, 152, -53, 467, 167, 124, 64, -30, 69, 101, 35, 220, 249, 23, 36, 408, 77, 73, 299, 249, 6, 770, 159, 208, 157, 155, 26, 418, 113, 223, 284, 386, 27, 496, 150, 320, 86, -92, 53, 10, 44, 159, 248, 66, 20, 555, 77, 42, 29, -31, 51, 205, 66, 306, 279, 187, -67, 530, 167, 89, 248, 302, -7, 781, 173, 286, 228, 274, 69, 384, 94, 264, 202, -92, 89, 458, 78, 230.

Translate the code back into a message (using the same numbering convention *A = 1, B = 2*, etc).

Note: When Matlab does computations, it will do everything in floating point arithmetic. Sometimes there will a condition known as roundoff error in which a solution which would come out exact by hand comes out approximate on a computer. For this problem, you can safely assume that if Matlab returns numbers such as 8.000000000000014 then the answer should really be 8.