Part VI
Ciphers based on matrices

Unit 85
Matrices and vectors

A *vector* is an ordered set (a list) of numbers. The length of that list is its *dimension*. The numbers in the list are the *components* of the vector. In this section of the book, vectors will be written as *column vectors*, i.e., as a column of numbers. The name of the vector is written either in bold face or with a vector arrow over it; for example:

$$\mathbf{A} = \vec{\mathrm{A}} = \begin{pmatrix} 2 \\ 3 \\ 5 \end{pmatrix}$$

The components are written with a subscript. For the above example, $A_0 = 2$, $A_1 = 3$, and $A_2 = 5$. Notice that we strart counting with zero; this is not the usual convention, but we are programmers and we use languages that index lists beginning from zero.

The set of all vectors with the same dimension is called a *vector space*; its dimension is the same as the dimension of its vectors. The *origin* of the vector space is the vector all of whose components are zero; This vector is called the *zero vector*, **0**. In a vector space, we can add vectors by adding their components. If **C** = **A** + **B**, then

$$C_i = A_i + B_i$$

for all $i = 0, ..., d-1$, where d is the dimension of the vectors. Since it does matter in which order we add two numbers, it is also true that we can add vectors in either order; we say that addition is *commutative*. Of course, if we add the zero vector to any other vector, then that vector remains unchanged; for any vector **A**,

$$\mathbf{A} + \mathbf{0} = \mathbf{0} + \mathbf{A} = \mathbf{A}$$

A *scalar* is a number that is not part of a vector. We can multiply a vector **A** by a scalar by multiplying each component by that scalar. If **B** = c**A**, then

$$B_i = (cA)_i = c\,A_i$$

Recall from Unit 7 that we saw how to multiply vectors (it’s not the only way, but it is the only way that we will need) by defining the inner product (scalar product, dot product) as

$$\mathbf{U}\cdot\mathbf{V} = \sum_{i=0}^{d-1} U_i V_i$$

An *m*×*n matrix* is an two-dimensional array of numbers with *m* rows and *n* columns. For example:

$$\mathbf{M} = \begin{pmatrix} 3 & 4 & 2 \\ 7 & 9 & 5 \end{pmatrix}$$

The numbers in a matrix **M** are its *elements* (or *entries*) M_{ij}, where *i* labels the row, and *j* the column, in which a component exists. For the example above, $M_{01} = 4$. The *dimension* of a matrix is the number of rows by the number of columns. For our example, the dimension is 2×3. The set of all matrices with the same dimension has algebraic properties similar to a vector space. We can add two matrices in the set by adding their elements:

$$(\mathbf{M} + \mathbf{N})_{ij} = M_{ij} + N_{ij}$$

The matrix all of whose entries are zero acts as the additive identity. We can also perform scalar multiplication:

$$(c\mathbf{M})_{ij} = c\, M_{ij}$$

Multiplying matrices is a little more complicated. We can only multiply two matrices **A** and **B** if the number of columns in **A** matches the number of rows in **B**. The rule is that if **C** = **AB**, and **A** has dimension *m*×*n* and **B** has dimension *n*×*p*, then **C** has dimension *m*×*p*, and

$$C_{ik} = (\mathbf{AB})_{ik} = \sum_{j=0}^{n-1} A_{ij} B_{jk}$$

for all $i = 0, ..., m-1$ and $j = 0, ..., p-1$. Notice that if *m* and *p* are different, then we can multiply **AB**, but not **BA**.

The set of all square matrices with the same dimension is special. A matrix is *square* if it has the same number of rows as columns. In this set, it is possible to multiply in either order. In general, however, the result will depend on the order, so matrix multiplication is noncommutative. In this set of square matrices, the additive identity is the matrix all of whose entries are zero. Adding it to any other matrix leaves the matrix unchanged. The multiplicative identity is the matrix **I** such that

$$I_{ij} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases}$$

All of the entries on its diagonal are one; all other entries are zero. The multiplicative *inverse* of a square matrix **A** is another matrix called $\mathbf{A}^{-1}$ such that

$$\mathbf{A}\,\mathbf{A}^{-1} = \mathbf{A}^{-1}\,\mathbf{A} = \mathbf{I}$$

To find the inverse of a matrix we must go on a short detour.

Given a row number and a column number, the *minor matrix* of a matrix is formed by taking the matrix and removing that row and that column. For example, the (1, 2) minor matrix of this matrix

$$\mathbf{A} = \begin{pmatrix} 4 & 3 & 9 & 4 & 3 & 2 \\ 7 & 6 & 8 & 5 & 7 & 0 \\ 8 & 2 & 1 & 3 & 2 & 5 \end{pmatrix}$$

is found by deleting the second row (because the first is row 0) and the third column (because the first is column 0):

$$\mathbf{M}_{1,2} = \begin{pmatrix} 4 & 3 & 4 & 3 & 2 \\ 8 & 2 & 3 & 2 & 5 \end{pmatrix}$$

The *determinant* of a square matrix is a number derived from the matrix. We will define it in a recursive manner. The determinant of a 1×1 matrix is the value of the only entry. The determinant of a 2×2 matrix **A** is

$$\det \mathbf{A} = A_{00}\, A_{11} - A_{01}\, A_{10}$$

The determinant of a square matrix with arbitrary dimension is found from the determinants of its minor matrices:

$$\det \mathbf{A} = \sum_{i=0}^{n-1} (-1)^i A_{0\,i} \det \mathbf{M}_{0,i}$$

where $\mathbf{M}_{0,i}$ is the (0, *i*) minor matrix of **A**. With this definition, we can find the determinant of any square matrix by working our way down until we are working with determinants of 2×2 matrices. An important thing to know is that a matrix is invertible in the set of matrices if and only if its determinant is invertible in the set of numbers. In the realm of real numbers, all numbers are invertible except zero. However, in modular arithmetic we know that not all numbers have inverses.

The combination of a power of −1 and the determinant of a minor matrix that we see in the formula above is called a cofactor. In general, the *cofactor* of an element A_{ij} is

$$C_{i,j} = (-1)^{i+j} \det \mathbf{M}_{i,j}$$

The matrix **C** whose *i*, *j* entry is the cofactor of A_{ij} is called the *cofactor matrix* of **A**. The *transpose* of a matrix **A** is another matrix $\mathbf{A}^{\mathrm{T}}$ that is obtained by exchanging the rows with the columns of **A**:

$$(A^{\mathrm{T}})_{ij} = A_{ji}$$

If we take the transpose of the cofactor matrix of **A**, what get is called the *adjugate matrix* of **A**. Finally, we are able to say that the inverse of a square matrix **A** is the multiplicative inverse of the determinant of **A** multiplied by the adjugate matrix of **A**.

Another way to invert a matrix is by using elementary row operations. There are three *elementary row operations*:

- swap two rows
- multiply every element in a row by a scalar
- add one row to another (add elements that are in the same column)

To invert a square matrix, we first extend it with a copy of the identity matrix with the same dimensions. Then we apply row operations until the identity matrix appears on the other side. The matrix that is now where the identity used to be is the inverse matrix. As an example, consider this matrix:

$$\mathbf{M} = \begin{pmatrix} 2 & 1 & 2 \\ 1 & 3 & 3 \\ 5 & 6 & 7 \end{pmatrix}$$

Extend it with the 3×3 identity matrix:

$$\left(\begin{array}{ccc|ccc} 2 & 1 & 2 & 1 & 0 & 0 \\ 1 & 3 & 3 & 0 & 1 & 0 \\ 5 & 6 & 7 & 0 & 0 & 1 \end{array}\right)$$

Swap the top and middle rows (R_0 and R_1, since we count from zero):

$$\left(\begin{array}{ccc|ccc} 1 & 3 & 3 & 0 & 1 & 0 \\ 2 & 1 & 2 & 1 & 0 & 0 \\ 5 & 6 & 7 & 0 & 0 & 1 \end{array}\right)$$

Replace the middle row with $R_1 - 2R_0$, and the bottom row with $R_2 - 5R_0$:

$$\left(\begin{array}{ccc|ccc} 1 & 3 & 3 & 0 & 1 & 0 \\ 0 & -5 & -4 & 1 & -2 & 0 \\ 0 & -9 & -8 & 0 & -5 & 1 \end{array}\right)$$

Replace the bottom row with $R_2 - 2R_1$:

$$\left(\begin{array}{ccc|ccc} 1 & 3 & 3 & 0 & 1 & 0 \\ 0 & -5 & -4 & 1 & -2 & 0 \\ 0 & 1 & 0 & -2 & -1 & 1 \end{array}\right)$$

Swap the middle and bottom rows:

$$\left(\begin{array}{ccc|ccc} 1 & 3 & 3 & 0 & 1 & 0 \\ 0 & 1 & 0 & -2 & -1 & 1 \\ 0 & -5 & -4 & 1 & -2 & 0 \end{array}\right)$$

Replace the bottom row with $-(R_2 + 5R_1)/4$:

$$\left(\begin{array}{ccc|ccc} 1 & 3 & 3 & 0 & 1 & 0 \\ 0 & 1 & 0 & -2 & -1 & 1 \\ 0 & 0 & 1 & 9/4 & 7/4 & -5/4 \end{array}\right)$$

Now that the leading entries of each row are all 1, we need to put zeroes in the off-diagonal entries on the left side. Replace the top row with $R_0 - 3R_1 - 3R_2$:

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & -3/4 & -5/4 & 3/4 \\ 0 & 1 & 0 & -2 & -1 & 1 \\ 0 & 0 & 1 & 9/4 & 7/4 & -5/4 \end{array}\right)$$

Now $\mathbf{M}^{-1}$ is the square matrix on the right-hand side.

Next we need to discuss the product of a matrix and a vector. We will only be concerned with square matrices here. A n-dimensional vector is the same thing as a $n \times 1$ matrix, so we can handle multiplication in the same way as we multiply matrices. The product of an $n \times n$ matrix $\mathbf{A}$ and an n-dimensional vector $\mathbf{V}$ is another n-dimensional vector $\mathbf{U}$ whose components are

$$U_i = \sum_{j=0}^{n-1} A_{ij} V_j$$

Reading and references

Wikipedia:
en.wikipedia.org/wiki/Matrix_(mathematics)
en.wikipedia.org/wiki/Square_matrix
en.wikipedia.org/wiki/Minor_(linear_algebra)
en.wikipedia.org/wiki/Determinant
en.wikipedia.org/wiki/Adjugate_matrix
en.wikipedia.org/wiki/Elementary_matrix

Programming tasks

1. Go back and find your function that evaluated the dot product of two vectors.
2. Write a function that adds two vectors.
3. Write a function that multiplies a vector by a scalar.
4. Write a function that adds two matrices. It is OK if you only consider square matrices.
5. Write a function that multiplies a matrix by a scalar.
6. Write a function that multiplies two matrices. It is OK if you only consider square matrices.

7. Write a function to multiply a square matrix by a vector. It should return another vector with the same dimension.
8. Write a function that finds the minor matrix of a square matrix given a row number and a column number.
9. Write a function to find the determinant of a square matrix.
10. Write a function to find a cofactor of an element in a square matrix.
11. Write a function to find the cofactor matrix of a square matrix.
12. Write a function to find the transpose of a matrix. It is OK if you only consider square matrices.
13. Write a function to find the adjugate of a square matrix.
14. Write a function to invert a matrix by using elementary row operations.
15. Write a function to determine whether a square matrix is invertible.
16. Write a function to find the inverse of a square matrix. It should first check whether the matrix is invertible and have some way to alert the main program if it is not.

Exercises

1. Find these products:
   a. $\begin{pmatrix} 6 & 6 & 7 \\ 4 & 9 & 7 \\ 1 & 5 & 5 \end{pmatrix}\begin{pmatrix} 6 & 6 & 4 \\ 5 & 0 & 8 \\ 9 & 7 & 6 \end{pmatrix}$
   b. $\begin{pmatrix} 4 & 3 & 2 & 6 \\ 7 & 2 & 9 & 1 \end{pmatrix}\begin{pmatrix} 7 & 0 \\ 5 & 4 \\ 3 & 8 \\ 1 & 5 \end{pmatrix}$
   c. $\begin{pmatrix} 5 & 9 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 4 \\ 7 \end{pmatrix}$
   d. $\begin{pmatrix} 5 & 7 & 0 \\ 8 & 4 & 6 \\ 9 & 7 & 4 \end{pmatrix}\begin{pmatrix} 7 \\ 1 \\ 3 \end{pmatrix}$
2. Find the inverses of these matrices:
   a. $\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$

b. $\begin{pmatrix} 2 & 2 \\ 1 & 3 \end{pmatrix}$

c. $\begin{pmatrix} 17 & 19 & 15 \\ 10 & 12 & 13 \\ 16 & 18 & 17 \end{pmatrix}$

d. $\begin{pmatrix} 3 & 9 & 5 & 8 \\ 7 & 0 & 6 & 2 \\ 2 & 3 & 2 & 6 \\ 4 & 7 & 5 & 1 \end{pmatrix}$

Unit 86
Matrices over the set of residues

For use in ciphers, we need to work with numbers from the set of residues. Review Unit 14 if you have forgotten what we mean. For the rest of this part of the book, we will be working with matrices and vectors whose elements are taken from the set of residues, usually modulo 26. This means that every number, whether it appears in a vector, in a matrix, or alone, is a member of $\mathbb{Z}_{26}$. All multiplications and additions (and subtractions) must be done modulo 26. The inverse of any number must be found by the algorithm in Unit 21.

In the last unit, we saw how to find the inverse of a square matrix as the inverse of the determinant times the adjugate matrix. We now have to be careful that we handle our arithmetic correctly, and that when we find the inverse of the determinant we find its inverse modulo 26. If that inverse does not exist, then the matrix is not invertible.

Programming tasks

1. Find your function from Unit 21 to get the multiplicative inverse of a number in modular arithmetic.

2. Make revisions of your functions from the previous unit so that you now have a function that multiplies a square matrix by a vector. It should return a vector all of whose components are in $\mathbb{Z}_{26}$. Allow for the option to change the modulus.

3. Make revisions of your functions from the previous unit so that you now have a function that finds the determinant of a square matrix whose elements are in $\mathbb{Z}_{26}$. Allow for the option to change the modulus.

4. Write a function to determine whether a square matrix over $\mathbb{Z}_{26}$ is invertible. Allow for the option to change the modulus.

5. Make revisions of your functions from the previous unit so that you now have a function that finds the inverse of a square matrix over $\mathbb{Z}_{26}$. Allow for the option to change the modulus.

Exercises

1. Find these products modulo 26:

   a. $\begin{pmatrix} 9 & 6 & 10 \end{pmatrix} \begin{pmatrix} 10 \\ 18 \\ 13 \end{pmatrix}$

   b. $\begin{pmatrix} 15 & 9 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} 8 & 4 \\ 24 & 25 \end{pmatrix}$

   c. $\begin{pmatrix} 11 & 9 & 18 \\ 5 & 19 & 7 \\ 6 & 10 & 10 \end{pmatrix} \begin{pmatrix} 4 \\ 11 \\ 20 \end{pmatrix}$

   d. $\begin{pmatrix} 25 & 18 & 19 & 18 \\ 14 & 19 & 24 & 15 \\ 8 & 4 & 1 & 7 \\ 23 & 8 & 24 & 6 \end{pmatrix} \begin{pmatrix} 23 \\ 13 \\ 5 \\ 21 \end{pmatrix}$

   e. $\begin{pmatrix} 16 & 22 & 1 & 8 & 1 \\ 24 & 9 & 16 & 1 & 25 \\ 2 & 4 & 2 & 19 & 11 \\ 11 & 17 & 0 & 1 & 12 \\ 18 & 9 & 25 & 1 & 5 \end{pmatrix} \begin{pmatrix} 13 \\ 4 \\ 16 \\ 7 \\ 6 \end{pmatrix}$

2. Find the inverses of these matrices over $\mathbb{Z}_{26}$:

   a. $\begin{pmatrix} 8 & 19 \\ 21 & 18 \end{pmatrix}$

   b. $\begin{pmatrix} 15 & 14 & 8 \\ 0 & 15 & 11 \\ 6 & 7 & 0 \end{pmatrix}$

   c. $\begin{pmatrix} 21 & 3 & 0 \\ 8 & 8 & 19 \\ 13 & 2 & 4 \end{pmatrix}$

   d. $\begin{pmatrix} 16 & 19 & 2 & 14 \\ 20 & 8 & 0 & 13 \\ 0 & 5 & 19 & 0 \\ 13 & 8 & 8 & 11 \end{pmatrix}$

e. $\begin{pmatrix} 4 & 17 & 4 & 14 & 7 \\ 11 & 14 & 15 & 6 & 8 \\ 4 & 4 & 7 & 17 & 2 \\ 2 & 13 & 0 & 0 & 0 \\ 19 & 2 & 11 & 15 & 11 \end{pmatrix}$

Unit 87
Hill cipher

The *n*×*n Hill cipher* is a block cipher that uses matrix multiplication to encipher each block of *n* letters. (A *block cipher* is a cipher that acts on blocks of texts, rather than individual letters.) Each block is written as a vector whose components are the numerical equivalents of its letters, where ‘A’ = 0, ‘B’ = 1, ..., ‘Z’ = 25. Encipherment is done by multiplying this vector by the key, which is a (square) *n*×*n* matrix. Decipherment is done with the inverse matrix. All operations are done modulo 26 (or the length of the alphabet, if we are using a different one).

Let’s work an example. Suppose we have this short message and that we want to encipher it with the matrix that follows:

```
THIS MESSAGE WAS ENCRYPTED WITH A HILL CIPHER
```

$$\mathbf{M} = \begin{pmatrix} 7 & 8 & 11 \\ 11 & 2 & 8 \\ 15 & 7 & 4 \end{pmatrix}$$

We first divide the plaintext into blocks of three letters, and pad with nulls if necessary:

```
THI SME SSA GEW ASE NCR YPT EDW ITH AHI LLC IPH ERX
```

The first block, `THI`, is expressed as this column vector:

$$\mathbf{V} = \begin{pmatrix} 19 \\ 7 \\ 8 \end{pmatrix}$$

This block in enciphered to a new vector:

$$\mathbf{U} = \mathbf{M}\mathbf{V} = \begin{pmatrix} 7 & 8 & 11 \\ 11 & 2 & 8 \\ 15 & 7 & 4 \end{pmatrix} \begin{pmatrix} 19 \\ 7 \\ 8 \end{pmatrix} = \begin{pmatrix} 17 \\ 1 \\ 2 \end{pmatrix}$$

This vector becomes the block `RBC` in the ciphertext. The full ciphertext is

RBC GUG KAG EQY GQM IXR DEV ISN ZAV OAD FDQ TST BCL

Decipherment uses the inverse matrix:

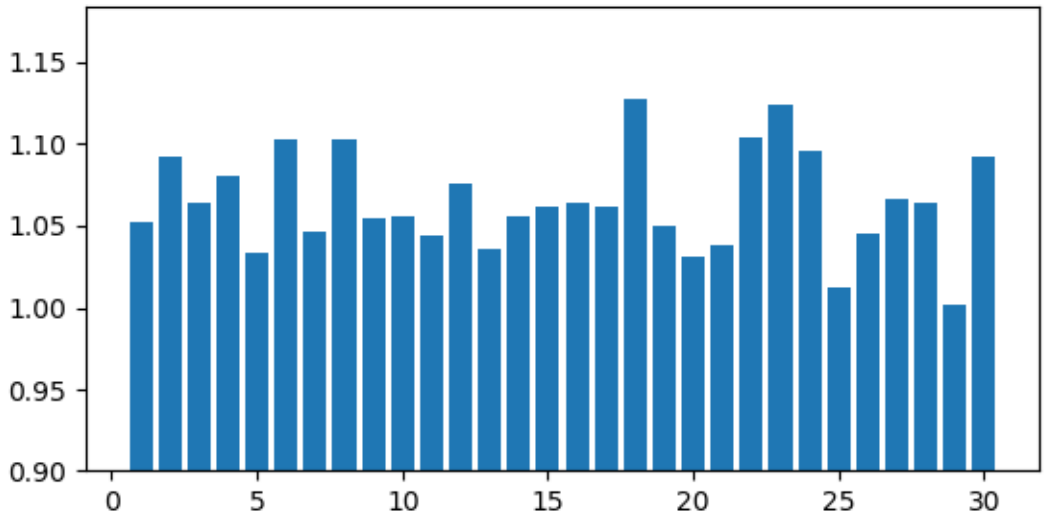
$$\mathbf{M}^{-1} = \begin{pmatrix} 12 & 5 & 22 \\ 20 & 5 & 13 \\ 11 & 5 & 12 \end{pmatrix}$$

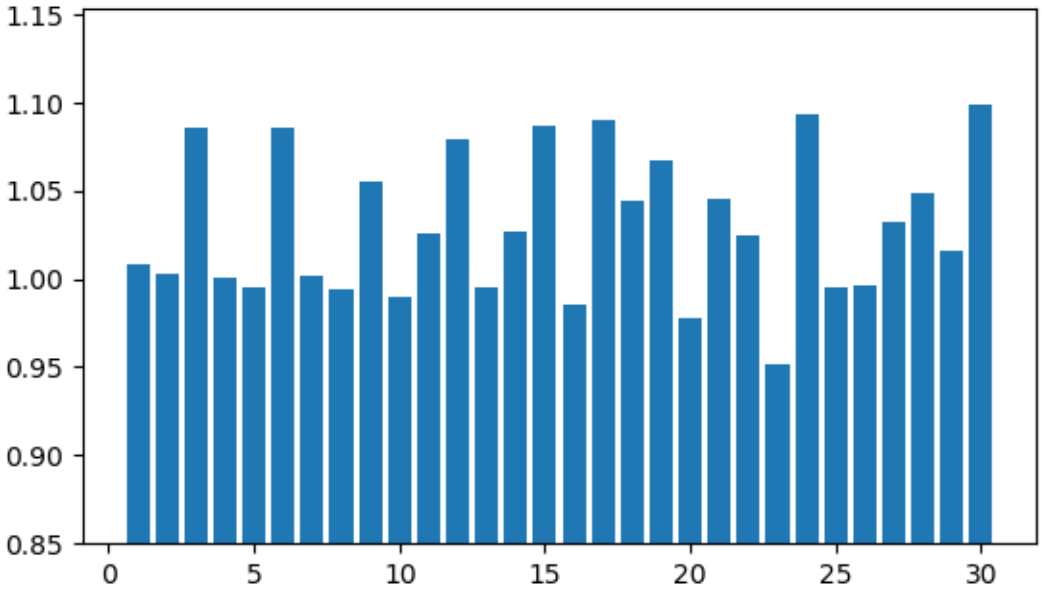
For the first block of the ciphertext:

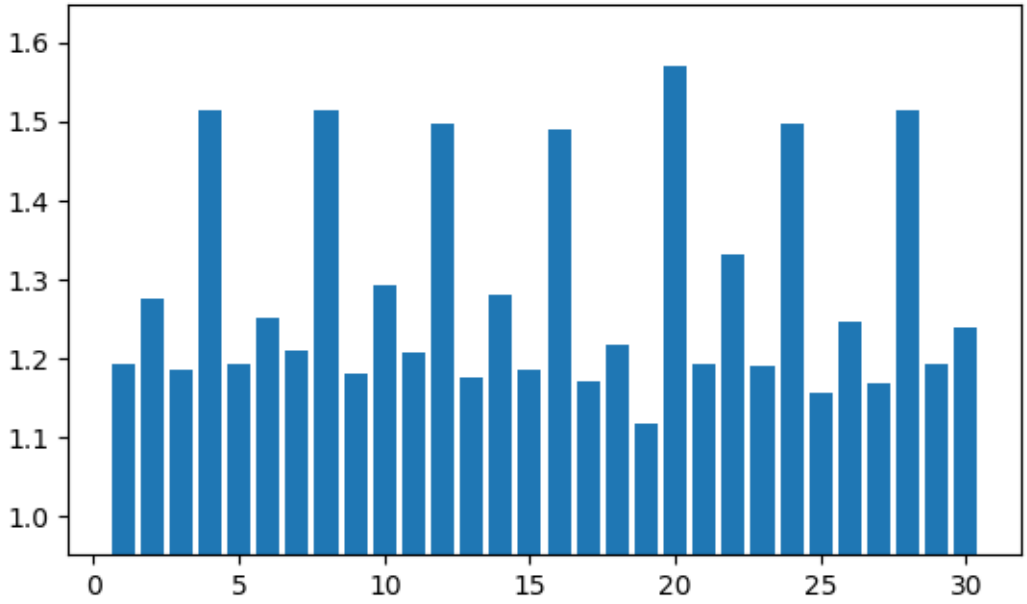
$$\mathbf{V} = \mathbf{M}^{-1}\mathbf{U} = \begin{pmatrix} 12 & 5 & 22 \\ 20 & 5 & 13 \\ 11 & 5 & 12 \end{pmatrix} \begin{pmatrix} 17 \\ 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 19 \\ 7 \\ 8 \end{pmatrix}$$

The matrix used in a Hill cipher can often be expressed as a keyword by using letters that are equivalent to the entries of the matrix. For the example above, the keyword is HILLCIPHE(r).

Given a ciphertext whose blocksize is unknown, it is often possible to find it using the index of coincidence. The same technique we used in Unit 31 to find the period of polyalphabetic substitution cipher can help here. However, the height of the peaks in the graph of IoC versus blocksize is not always above a predetermined threshold. Often it is even difficult to discern the peaks. These examples are for 2×2, 3×3, and 4×4 Hill ciphers. As you can see, it is sometimes difficult to determine the block size.







Hill had another cipher, in which the plaintext was inserted into a matrix that was multiplied by the key matrix. In this cipher, the key matrix was chosen so that it was its own inverse. An interested reader can see his second paper in the references below.

Reading and references

Lester S. Hill, “Cryptography in the Algebraic Alphabet,” *The American Mathematical Monthly* 36:6 (1929) 306-312, DOI: 10.2307/2298294, www.jstor.org/stable/2298294, web.archive.org/web/20110719235517/http://w08.middlebury.edu/INTD1065A/Lectures/Hill Cipher Folder/Hill1.pdf

Lester S. Hill, “Concerning Certain Linear Transformation Apparatus of Cryptography,” *The American Mathematical Monthly* 38:3 (1931) 135-154, DOI: 10.1080/00029890.1931.11987161, www.jstor.org/stable/2300969, www.cs.jhu.edu/~cgarman/files/Hill2.pdf

Abraham Sinkov, *Elementary Cryptanalysis: A Mathematical Approach*, 2nd edition, revised by Todd Feil, published by Mathematical Association of America, 2009; www.jstor.org/stable/10.4169/j.ctt19b9krf; chapter 4.

David Kahn, *The Codebreakers: The Story of Secret Writing*, New York: Simon & Schuster, 1967, revised and updated 1996, pages 404-409.

Wikipedia, en.wikipedia.org/wiki/Hill_cipher

Crypto Corner, crypto.interactive-maths.com/hill-cipher.html

Chris Christensen, “Lester Hill Revisited,” *Cryptologia* 38:4 (2014) 293-332, DOI: 10.1080/01611194.2014.915260

Programming tasks

1. Implement an encryptor. Allow for any block size. Verify that the matrix is invertible before enciphering.
2. Implement a decryptor. Allow for any block size. Remember that the key is the matrix used to encipher, so you must find its inverse.
3. Write a function to try to determine the block size. Perhaps it could return a value of zero if it is unable to clearly determine the size.
4. Implement a brute-force attack. For an $n \times n$ matrix, there are n^2 values to vary. You should discard matrices that are not invertible.
5. Compile a list of four-letter words that can fill a square matrix so that it is invertible. Do the same for nine- and sixteen-letter words.
6. Implement a dictionary attack. You should discard matrices that are not invertible.

Exercises

1. Encipher this text with keyword `HILL`.

> This, and enough, premised, I go souse into my personal history. My maiden name was Frances Hill. I was born at a small village near Liverpool, in Lancashire, of parents extremely poor, and, I piously believe, extremely honest.
>
> (from *Memoirs of Fanny Hill* by John Cleland)

2. Decipher this text with keyword PROFESSED.

```
SNSPTIRUZMCUZBWJZIJVGYIVSGJUUZYWDDWMDLPVWKTVYJPUGLPAGQ
LLYBVHLIPLXXIKKGKAOHSDCBDSKSKXTTTMERLNKQFBNDBPULJUSHQR
TWNSNSLWDPWGBAKKMZMOEPKGDGEABNTIARIAVQHMAZEKPGYKRXXXTT
EKNSNSYMOISTGYXSSRJAKUAYVKKKPCXGYRHKCARPVPOIAQNYGGLVRM
XJBUYLXVHSSRKIIRPAZVKVYJBKJFGSDYIGSCBDPBVRHCMTLIJXHJMK
RPOCNWRZMELXHWBRWVZWXTJFHTKDWMHVYPHHXLDEYYDJZREVWQDAZE
JOGHUXREVVGEDIZOTGELXOSJFUMVXCCPCZVJWMMKITKYZHIIXJMXRD
XSBDISSHTZOGYNOZOGBLZUSFSNSWPCQAFPGFRAIMKLHABVGGUZNTJS
HHRXSUVUUBZTYWDDCJVBTBKJEQDCQIHZNPPGSNSZDODCJPFICIAWHE
PFKHRWOTGXJMDIZFXPDQQISQZHERGXTSSZEWNGEQKHTYIXJMTOGDKN
LUN
```

3. Can you determine the block size for this ciphertext?

```
IPKGRGNLJUQQGFTQMMYKABOCFYRBGEWEOMYICYODZYLOMJGFXXZDYN
TOBIBZLTITEMDSQITHEDUUWUQJOJVAWIFKVXFUDCGFUTECOTWBSQLO
CVCSEBSSNKMBJFRGECMIERCGEVAXSBLVBZEAQNIHGHJZGBJMIJJQHN
MAKHJCBIOZRDTAZMJDSWORRBHXOOCFDBMWNYAPXZASTVSXIOFITERX
WVQMLAFSWXSTWKUVWSEFOIBKUQIHZAELVRPVUQQSGBPCZRKTVAKBLK
ZXZMAWMNUHBMOLUWNIMUMICASOLWHJVLEKONNGOFCDKKTQSGEEZELI
BCGDCBCIJLEOHEJODPHRISLZVAWZVJJZXJFHESLRYWDITGQUMERAFO
XEJUWGCSQYGGJYVAZPJVHMPGYBMKERBHCFJUUUUJGPGICASFJHWWCN
TEMAYXBUQZVDYDGHQQRMGBWBZKMRAIXDUYMTQXDRMQZVMTDZUYTRVK
WIMPVAREZPRLJIURMIDPAMSDRQZOZRIARCPJPLHWWICHDRQCWVILQK
AIMXMYWVEJRQZOZDUXOOGYXNNOMZXRMYJYHGRLFWJWPJHNADQCTXFR
EHIJDNXKZONJTWWCRYESVCGWGVUWALSSPQACCGEVAXSBLVIERGK
```

4. Use a brute-force attack to break this ciphertext. Is there a keyword?

```
TOPGRHYADTVKXXSQYQXZUBWLGFALFCMGEZUBEBHGQKWODTGFYUBIMO
WJFVKXEVFOVTRYCNXWOJYMSVZWPLANLUCCTOKEUIPVWVSOJWURXAFV
CUMPCUQJDEWJFVSNEVJQEFIWJHQTVHYASVFVSQZWFVSAMROUTOHBIG
WSFVCUQJHECURMNQEZFVIGPCMYURTOLQLBXWOYOFZSSAGBITRYAJSA
CRFIHEANLIANGMGPZWAHQWFCDIVGQWDTZWPYDTXWQAYWRMSSZWXWIK
KRZGGDJK
```

5. Use a dictionary attack to break this ciphertext.

```
RMYKUKRLHLXETRSMCYXFOUYMHHNOJAMEERFEDXQRXOWOSIEAPXFDOR
ZBCZZHLNBRMYRRGVTCNOZNYSOHYDORWPKRHPCEJEKOPGAJTDRPERMY
VZUCHCDORSWGAYJMJOVBULXRVXTGKOZNHAQDYZHKYRMNXKMCMIREMZ
ITKDDLYRYWGDJZUZNXCEJKALYRYCEEDMTBMAQYUHHNWJWCEEDMTHMK
REIZXYOCQEUHSMWJLYTTQDKSEUHZNHPXFWGDOVAVJSBROHYBYUCFOC
IKHXIKPSLUJYJWWAWUHQKSAGXXMHHNMJOOINZNHROZUMKMCYRBEMLK
VTCAJUJZUBRODKSFEBJZWPBUMJOYGNGGCNTZZBSJZUHYYVEBOWNZUL
TPRRLHJWWAWURYPKRKQSEAJUPUKITIYFYHYBQBEQBQQYUUUYUFCRLH
WFGVXCYKCBTRHRSAUEZNXKKYBTRRMYXISLBNSQJVNZPWGJLYKXMGVU
```

DPQRHPQAFXISRBXWOSIEAIHUHYBDGAZMZSMYBHXLGUSTSIKHCVOKKO
PYDDORZCVOLXYWBWGDRLUXLRYWNLWBZNHOSTZNXGWKZREYMTNDKOMA
FSIHRADORGHSOAFZWQMRMLSSWXWOFQZGCECLPAPROXSEIMLL

madness's book on classical cryptography
unit 88: attacking the hill cipher with cribs
last modified 2020-08-14

Unit 88
Attacking the Hill cipher with cribs

An $n \times n$ Hill cipher has n^2 parameters, and therefore we need n^2 constraints to completely determine the key matrix. These constraints take the form of equations that we get by matching a crib to a piece of ciphertext. If the crib is too short, or some parts of it repeat, then there may be too few constraints; in that case, we have to brute-force the remaining parameters.

When we are matching a crib to a piece of ciphertext, we must remember that we can only use parts of the crib that cover complete blocks. Let's return to the example from the previous unit. If our crib is `MESSAGE`, then we move the crib over the plaintext and try each position until we can find a key that gives a good plaintext.

```
MES SAG E
RBC|GUG|KAG|EQY|GQM|IXR|DEV|ISN|ZAV|OAD|FDQ|TST|BCL
```

In the above position, the constraints from the first block are (remember that the key is a matrix **M**, and 'A'=0, 'B'=1, ...)

$$\begin{aligned} 17 &= 12\,M_{00} + 4\,M_{01} + 18\,M_{02} \\ 1 &= 12\,M_{10} + 4\,M_{11} + 18\,M_{12} \\ 2 &= 12\,M_{20} + 4\,M_{21} + 18\,M_{22} \end{aligned}$$

From the second block we get

$$\begin{aligned} 6 &= 18\,M_{00} + 6\,M_{02} \\ 20 &= 18\,M_{10} + 6\,M_{12} \\ 6 &= 18\,M_{20} + 6\,M_{22} \end{aligned}$$

From the third block we do not get any constraints, unless we brute-force the next two plaintext characters. The equations we get, where x and y are the missing characters from the third block, are:

$$\begin{aligned} 10 &= 4\,M_{00} + x\,M_{01} + y\,M_{02} \\ 0 &= 4\,M_{10} + x\,M_{11} + y\,M_{12} \\ 6 &= 4\,M_{20} + x\,M_{21} + y\,M_{22} \end{aligned}$$

Collect the equations for the top row:

$$\begin{aligned} 17 &= 12\,M_{00} + 4\,M_{01} + 18\,M_{02} \\ 6 &= 18\,M_{00} + 6\,M_{02} \\ 10 &= 4\,M_{00} + x\,M_{01} + y\,M_{02} \end{aligned}$$

From these three equations, we can solve for M_{00}, M_{01}, and M_{02}, for each choice of x and y. We then collect the three equations for the middle row of the matrix, and the bottom row. For each of the $26^2 = 676$ choices for x and y, we get a complete key matrix (unless some of the equations are not independent and so cannot be solved). We invert each matrix and decipher the ciphertext. Unfortunately, for this position of the crib, none of the plaintexts that we get are acceptable.

Next, we move the crib over by one position:

```
 ME SSA GE
RBC|GUG|KAG|EQY|GQM|IXR|DEV|ISN|ZAV|OAD|FDQ|TST|BCL
```

Now we must brute-force the first character of the first block and the last character of the third block. The process is similar to the above, and again we do not find an acceptable plaintext. We can shift the crib one space at a time. When we do find a good plaintext, it is in this position:

```
    ME SSA GE
RBC|GUG|KAG|EQY|GQM|IXR|DEV|ISN|ZAV|OAD|FDQ|TST|BCL
```

We need to check all possibilities for the first plaintext character in the second block and the last plaintext character in the fourth block. For each of the $26^2 = 676$ choices, we set up nine equations. The equations form three sets of three. Each set allows us to solve for one row of the key matrix.

Programming tasks

1. Implement the attack. Build an attack for 2×2 and 3×3 Hill ciphers, at least. If you are very clever, perhaps you can build an attack for any block size. Be careful in handling cribs of various lengths. Use tetragram fitness to determine when you have found an acceptable plaintext.

Exercises

1. Finish the example above by hand to find the key. Check that it matches the one we used in the previous unit to encipher the message.

2. Break this ciphertext with the crib `FOLLY`. What might the keyword be?

```
GHTSYUESFCYMNNBGCGESCNGHAXDULTNVXOYRWPWTKPGUNKAWNPDVES
BBRFNTRNBGOWXLWTYNDPNSDLEYGUBBEHYQLAGOGJCNGIYUWTEULBBB
REYVZVBGCGESAEDNCQLHDLZZGRXOYKGUNKAWNNNSCNYBYCOLYRYWLI
AWNLXOGULXGBDNLGYTAEXWGJDLEXYQFBWDDLJHBFFXNNNSDLERDDLH
QGXREYWTKBWQDGGPGULNGJDNBBREEXWKMTOIZBTRMUDPYSWXYQNSUW
```

```
WLYDMOUUGSYJAMMLAYMOUUGHTCWTKVDNPDOLYUXRCILNGJESYVGVKE
FWDNEACILWLZNXTCGYWMEQNQWKZBCILULJUVDAOCWGEOGUNKAWNSGY
YNGIXWYOXLNSXRCXANFWGGYBGHTCWTKMYJLJUSGIDVUU
```

3. Break this ciphertext with the crib `PITILESS`.

```
GQTIRDBVCMCVUCJCPZFCSPEOFOPRBDTOQGQAHWBXNBWLEWJCNUBRMR
ETIEVCJSKZLCUDWHWBGESEPEVBRGXWCHDQSFAHWNRUJMHTRRPAAZGY
GEZTDGDLELCSEMNLOCGESEPMNRUAFKGQTUHGHSFCPEIYGHCRDCSIHW
RTIOJVKAXSRIGSFZMKNAZWMVIRDVUHYDBPEUPEVRBDOFSJVCHSFLCS
LWFRBDREQCRUKPMFDHAHOEMNZRJQLYYGHGFZSQHLAQBPQTQHGFSLQB
WKHOFNMACHRFGXEKUYYDBFYZVJGATEDYGGXERWCPPKGISLAALBOCST
CIYIVFMKPUSKCEGFYWUWSQMAYALQCEKMWWFSNSWMCVUCJBQDUWCKLS
YDBTUNDKPXMKEIZAFKIBRDXAGKE
```

Unit 89
Affine Hill cipher

The *affine Hill cipher* is an extension of the Hill and affine ciphers. We take the Hill cipher and add a constant term to the equation. Of course, since we are dealing with actions on vectors, the constant term is also a vector. When the key is a matrix **A** and vector **B**, a block of plaintext in a vector **P** is enciphered to a block of ciphertext in vector **C** with this equation:

$$\mathbf{C} = \mathbf{AP} + \mathbf{B}$$

Decipherment is done by solving for the plaintext vector:

$$\mathbf{P} = \mathbf{A}^{-1} (\mathbf{C} - \mathbf{B})$$

This cipher can be factored into a Hill cipher followed by a Vigenère cipher. This should be obvious from the first equation above. Furthermore, if in that equation **A** = **I** (the identity matrix), then the affine Hill degenerates to a simple Vigenère cipher.

Programming tasks

1. Implement an encryptor. Remember to check that the matrix is invertible.
2. Implement a decryptor. Remember that the key includes the matrix for encipherment and that you must invert it.
3. Modify your brute-force attack on the Hill cipher to accommodate the affine Hill cipher.
4. Modify your dictionary attack on the Hill cipher to accommodate the affine Hill. Use one keyword for the matrix and one for the added vector.
5. Modify your attack on the Hill cipher with cribs to accommodate the affine Hill cipher. Each equation will now have a constant term on the right-hand side. There are n^2+n parameters for which to solve.

Exercises

1. Encipher this text with the given matrix and vector.

$$\mathbf{A}=\begin{pmatrix} 21 & 3 & 0 \\ 8 & 8 & 19 \\ 13 & 2 & 4 \end{pmatrix}, \mathbf{B}=\begin{pmatrix} 5 \\ 7 \\ 9 \end{pmatrix}$$

   After marriage arrives a reaction, sometimes a big, sometimes a little one; but it comes sooner or later, and must be tided over by both parties if they desire the rest of their lives to go with the current.

   (from “Three and—an Extra” in *Plain Tales from the Hills* by Rudyard Kipling)

2. Decipher this ciphertext that was enciphered with the given matrix and vector.

$$\mathbf{A}=\begin{pmatrix} 8 & 19 \\ 21 & 18 \end{pmatrix}, \mathbf{B}=\begin{pmatrix} 23 \\ 11 \end{pmatrix}$$

   ```
   VSWEZMIFVTXDROVWQYJVVKQRZMVSWFWQXZJFFESTHETMQRQYXQHCID
   BCTUNNOPNPKLNBJIIFRUQDVPVAGBEYDFUPWQWJNVWXTMHCWFXQXJIF
   BLZSLZNUISVTXDROVWQYJVVKQRPINCFJHDXQGPBLDWONQNTUUCKLOO
   FBWXHDMPRVDRCAGWBXDRCABRTMNBHICAUBQRMFOLZWBCLZZMNZHUMC
   SLOLQNBVCYOLCLJWRJKLZMJGXUPONJGNPBKLVVMJDQHEGANJNAIDNA
   HEKLOOMFOLKRWVBEWXUPQRTQAINTDWWDKGBXXQST
   ```

3. Break this ciphertext with a brute-force attack. What are the keywords?

   ```
   HFPQQDGRZVKFKUEMZTGEUZSIUVNGRFBCWNHNGBGWUOZAGEGIWZGCPK
   NZBHZRBKEJTZZFXXWNSDZCUKSQUKGUTXZRGOGWWIIHGEGIPQPXWZKM
   OEJRPQWNXPGKZZHBUUBDHRZJSYMGPNTZKMZIGEAZCQBFZVTZKBZIGE
   GMSVHCHNWTHPDAFPBRNPGTZGJRLZAZZYSYYAHNSVEKHPPQPXWZKMSC
   HPCVZVZVTJJGKGZDGESIGTPZHBUZJOTHZVCNAFGVGSSMAMCBZGHNPD
   WZGOJOHNGBWAHPSMDEFBDPHNUHUCNSGQHNKMIAUFNSKGSUUZGOOMQD
   UXNGTJWG
   ```

4. Break this ciphertext with a dictionary attack.

   ```
   QJCSKYUYIUASSRXELHDERHZYKTHNQMJDIHYIAKLZBCKYQXYMVNKILM
   CFIRYOZZILKMCTRNDAONWVAUPAPJKDBYLWKQOLYHNSKYOUGDALEKJT
   JNMQJJTFGJKDOWNXOAWLOGTXTEDSLLUNTQDZNNSGWLDQGQQTENQTTU
   QCOVQULXROLPWCOUGDEJWZASZTOQYQOESIIWYXPPOAIWHQLZQHSLQR
   CVYJDGMTMOGPIBRDYJFAVGWDTOAWJRSDATSAIEIHSEQDIXBHTDMTDA
   IYKTLSVKAQGFYYWRNAYWYKTJQBSCQFJYGRATUGWLKLAUQ
   ```

5. Break this ciphertext with the crib `AFTERNOON`.

   ```
   KGMJEIVFJJGJHCKOGBRDZRHBUETXPIWNHUVFEWPTHHDXODTKUCEWUO
   CITNOLEYCEUOUNHFRFKIXNPXIGEYKOCLHUVFPLWMETRIVFXZDGUHCV
   ```

```
HYWKTGCOURTTPHEDHYDZPBVHWNNPELPUWFHYKOGJTLDXTQDLMHHYSX
XNNNSXHHTGCUWFHYMEKGCBDPICETHUVFEYWFDICUETRIVFVRDGEMPT
XZHHTGCTPXVTNDMLUVPLEFTGCKKNDIEDHYNDMWTGCNPVEFQYWXNSVT
TQUTCXTGCIUZCLRFTKCHTZVFHNTEUREPEIVSWFGGHFPAVCWWSTPM
```

6. Break this ciphertext from the 2013 British National Cipher Challenge with the crib TELEGRAM.

```
HCHKK MEYXE YMEZW RLFMS RPJSU BPWHR PMDEY XWHRM IQKEC
JNBCM CJFHY HSELD JXTSH IGEXH BMBIB GJRCN VAQJX FJYHL
LQPAF DJHAO NUCJF LRXTS XKTNQ JATUU VBGSH OHKOC VXHRZ
WWHFY EEZIU DSJFJ IBGQK XSUIN IFDPI HRXVZ BGGYT SXTPV
NETDB QMZAT SXRDW HLXKT EIWAZ WRTSX JWHSH VSIGE YRKBZ
WJQYI UVPTU VFMPR SOTWI DTEXL NMKNB YYMTV JIJWR JQFLZ
WJDHM LCVXH RZWWH FYEQH XMIQK ECUZK PJSBF KXVAY NZSUY
AJRRJ QRDWG DHENG CMRNR RBOVU FDUAE PRGAU QTBWN PLOJV
RLSVP TNBPI KHOQL PFXOT FMCWM KIRFG YNVBZ AXRS
```